Q-Chem 6.3 User's Manual

Version 6.3

May, 2025

Q-CHEM User's Manual

Version 6.3 editor:
Dr. Shannon Houck
Prof. John Herbert

Version 6.2 editor:
Dr. Shannon Houck

Version 5.4 editors:
Dr. Eric Berquist
Dr. Andrew Gilbert
Prof. John Herbert

Versions 5.2 and 5.3 editor:
Dr. Andrew Gilbert

Versions 5.0 and 5.1 editors:
Dr. Andrew Gilbert
Prof. John Herbert

Version 4 editors:
Prof. John Herbert
Prof. Anna Krylov
Dr. Narbe Mardirossian
Prof. Martin Head-Gordon
Dr. Emil Proynov
Dr. Andrew Gilbert
Dr. Jing Kong

The contributions of individual developers to each version are highlighted in "New Features", Section 1.3

Published by:
Q-Chem, Inc.
6601 Owens Dr.
Suite 240
Pleasanton, CA 94588

Customer Support:
Telephone: (412) 687-0695
Facsimile: (412) 687-0698
email: support@q-chem.com
website: www.q-chem.com

The information in this document applies to version 6.3 of Q-CHEM.

This document version generated on May 21, 2025.

Contents

Chapter 1

Introduction

1.1 About This Manual

This manual is intended as a general-purpose user's guide for Q-CHEM, a modern electronic structure program. The manual contains background information that describes Q-CHEM methods and user-selected parameters. It is assumed that the user has some familiarity with the Unix/Linux environment, an ASCII file editor, and a basic understanding of quantum chemistry.

After installing Q-CHEM and making necessary adjustments to your user account, it is recommended that particular attention be given to Chapters 3 and 4. The latter, which describes Q-CHEM's self-consistent field capabilities, has been formatted so that advanced users can quickly find the information they require while supplying new users with a moderate level of important background information. This format has been maintained throughout the manual, and every attempt has been made to guide the user forward and backward to other relevant information so that a logical progression through this manual is not necessary.

Documentation for IQMOL, a graphical user interface designed for use with Q-CHEM, can be found on the http://www.iqmol.org websitge. IQMOL functions as a molecular structure builder, as an interface for local or remote submission of Q-CHEM jobs, and as a post-calculation visualization program for densities and molecular orbitals.

1.1.1 Chapter Summaries

Ch. 1: General overview of Q-CHEM's features, contributors, and contact information.

Ch. 2: Procedures to install, test, and run Q-CHEM on your machine.

Ch. 3: Overview of the Q-CHEM command-line input.

Ch. 4: Running ground-state self-consistent field calculations.

Ch. 5: Details specific to running density functional theory (DFT) calculations.

Ch. 6: Running post-Hartree-Fock correlated wave function calculations for ground states.

Ch. 7: Running calculations for excited states and open-shell species.

Ch. 8: Using Q-CHEM's built-in basis sets, or specifying a user-defined basis set.

Ch. 9: Using Q-CHEM's effective core potential capabilities.

Ch. 10: Options available for exploring potential energy surfaces, such as determining critical points (transition states and local minima on a single surface, or minimum-energy crossing points between surfaces) as well as *ab initio* molecular dynamics.

Ch. 11: Molecular properties and *a posteriori* wave function analysis.

Ch. 12: Methods for molecules in complex environments, including implicit solvation models, QM/MM models, the Effective Fragment Potential, and density embedding.

Ch. 13: Fragment-based approaches for efficient calculations on large systems, calculation of non-covalent interactions, and energy decomposition analysis.

App. A: Overview of the AOINTS library, which contains some of the fastest two-electron integral code currently available.

App. B: Quick-reference section containing an alphabetized list of job control variables.

App. C: Overview of third-party packages.

1.2 Q-CHEM, Inc.

1.2.1 Contact Information and Customer Support

For general information regarding Q-CHEM program, visit www.q-chem.com. Full customer support is promptly provided via telephone or email (support@q-chem.com) for those customers who have purchased Q-CHEM's "QMP" maintenance contract. In addition to free customer support, this contract provides discounts on future updates and releases of Q-CHEM. For details of the maintenance contract please see www.q-chem.com.

1.2.2 About the Company

Q-CHEM, Inc. was founded in 1993 and was based in Pittsburgh, PA until 2013, when it relocated to Pleasanton, CA. Q-CHEM's scientific contributors include leading quantum chemists around the world. The company is governed by the Board of Directors which currently consists of Peter Gill (Canberra), Anna Krylov (USC), John Herbert (Ohio State), and Hilary Pople. Fritz Schaefer (Georgia) is a Board Member Emeritus. Martin Head-Gordon is a Scientific Advisor to the Board. The close coupling between leading university research groups and Q-CHEM Inc. ensures that the methods and algorithms available in Q-CHEM are state-of-the-art.

In order to create this technology, the founders of Q-CHEM, Inc. built entirely new methodologies from the ground up, using the latest algorithms and modern programming techniques. Since 1993, well over 300 person-years have been devoted to the development of the Q-CHEM program. The author list of the program shows the full list of contributors to the current version, and the journal citations for Q-CHEM versions 2, 3, and 4 [1,3,4] illustrate the breadth of the Q-CHEM developer community. The current group of developers consist of more than 100 people in 9 countries. A brief history of Q-CHEM is given in the article *Q-Chem: An Engine for Innovation*.[2]

1.2.3 Company Mission

The mission of Q-CHEM, Inc. is to develop, distribute, and support innovative and *sustainable* quantum chemistry software for industrial, government and academic researchers in the chemical, petrochemical, biochemical, pharmaceutical and material sciences.

1.3 Q-CHEM Features

1.3.1 Overview of Q-CHEM Features

Quantum chemistry methods have proven invaluable for studying chemical and physical properties of molecules. The Q-CHEM system brings together a variety of advanced computational methods and tools in an integrated *ab initio* software package, greatly improving the speed and accuracy of calculations being performed. In addition, Q-CHEM will accommodate larger molecular structures than previously possible, with no loss in accuracy, thereby bringing the power of quantum chemistry to critical research projects for which this tool was previously unavailable. Below is a reverse-chronological listing of new features added to Q-CHEM.

1.3.2 New Features in Q-CHEM 6.3

1.3.2.1 Features in 6.3.0

- **Changes to keywords and default behavior:**
  - Use MDQ-uFERF as default for EDA2 jobs (EDA_UFERF = TRUE) (Hengyuan Shen, Abdulrahman Aldossary, Martin Head-Gordon)
  - Add REM control option for fragment CPSCF max iterations (FRAG_CPSCF_MAXITER) (Yuezhi Mao)
  - Change alpha in BW-s2 jobs to n/1000 rather than n/100 (Linus Dittmer, Nikolay V. Tkachenko, Martin Head-Gordon)
  - Make O-V projection the default for basis2 calculations (BASISPROJTYPE = OVPROJECTION) (Kuan-Yu Liu)
  - All NEO jobs will now use a tight initial guess by default (Mathew Chow, Sharon Hammes-Schiffer)
  - Use nuclear core guess by default for NEO jobs with ghost atoms (Mathew Chow, Sharon Hammes-Schiffer)
  - Use simultaneous SCF by default for NEO calculations (NEO_SIMULTANEOUS_SCF = TRUE) (Mathew Chow, Sharon Hammes-Schiffer)
- **General features and improvements:**
  - MPI parallelization for finite-difference and many-body expansion calculations (Kaushik Nanda)
  - Added torsional and flat-bottom potential restraints for geometry optimizations (Chance W. Lander, Yihan Shao)
  - Added bond stretch, r12pr34, angle, 1 to midpoint of 2 atoms, 1 to COM, and dihedral restraints to list of available restrained PES scan options (Chance W. Lander, Yihan Shao)
  - Re-enabled NTOs for RPAs, corrected to use proper particle and hole density matrices (John Herbert)
  - New mechanochemical pressure model using PV term energy correction (Felix Zeller)
  - **Resolved issues with:**
    - Hessian calculation rerun unnecessarily when GEOM_OPT_HESSIAN = READ in libopt3 (Yuezhi Mao)
    - Remove bottlenecks in electronic JK gradient evaluation in libneo and librtneo (Mathew Chow, Sharon Hammes-Schiffer)
    - Eliminate NaN values in Feshbach-Fano calculations for large, diffuse basis sets (Saikat Roy, Wojciech Skomorowski)
    - Fix orbitals not being printed to .fchk file used by IQmol (Kuan-Yu Liu)

  - Enable FAST_XAS crash with STATE_ANALYSIS = TRUE or GUI = 2 (Kaushik Nanda)
  - Fix incompatibility of SCF_FINAL_PRINT and DC-DFT (John Herbert)
  - Fix failure to conserve molecular point-group symmetry in libopt3 (Andrew Gilbert)
  - Fix file error when calculating TS-VDW forces (John Herbert)

- **Density functional theory and self-consistent field:**
  - New "Robust SCF" procedure provides more reliable SCF convergence via automated choice of algorithm and defaults (Kaushik Nanda)
  - Performance enhancements for medium to large DFT calculations (Xintian Feng)
  - Canonical ROHF/ROSCF (Adrian L. Dempwolff)
  - Charge-transfer metrics for TDDFT (John Herbert)
  - TDDFT-1D provides improved TDDFT/TDA where the crossing of S1/S0 states is smooth (Vishikh Athavale, Hung-Hsuan Teh, Joseph Subotnik)
  - **Resolved issues with:**
    - Crash of ALMO-CIS/TDA with large basis sets (Yuezhi Mao)
    - DFT numerical quadrature consistency check not always performed (Xintian Feng)

- **Correlated methods:**
  - EOM-CCSDT for EE, SF, IP, EA, DIP and DEA (Manisha, Prashant Uday Manohar)
  - Added parallel performance to EOM-CCSDT (Manisha)
  - Analytic gradients for EOM-DEA/DIP-CCSD methods (Tingting Zhao, Anna Krylov)
  - SOC and $< S^2 >$ calculations for EOM-DEA/DIP-CCSD methods (Tingting Zhao, Sai Kotaru, Sahil Gulania, Pavel Pokhilko, Anna Krylov)
  - Automated calculations of Coulomb wave for caluclations of Auger rates within the Feshbach–Fano framework (Saikat Roi, Wojciech Skomorowski)
  - Auger Channel Projection EOMIP-CCSD with frozen core and open shell references (Robin Moorby, Florian Matz, Thomas Jagau)
  - Complex-valued RI-EOM-CCSD (Simen Camps, Cansu Utku, Thomas Jagau)
  - EOM-(EE/SF)-CCSD dynamic polarizability for imaginary frequencies (Kaushik Nanda)
  - Improved EOM-CC anisotropic polarizability formula (Kaushik Nanda)
  - CC2 with size-consistent Brillouin–Wigner Partitioning (Linus Dittmer, Nikolay V. Tkachenko, Martin Head-Gordon)
  - Maximum Physical Regularization for size-consistent Brillouin-Wigner Theory (Linus Dittmer, Nikolay V. Tkachenko, Martin Head-Gordon)
  - BW-S2 performance improvements (Zhenling Wang, Martin Head-Gordon)
  - Extend 1eX2C to calculate spin-orbit coupling components (Richard Kang, Martin Head-Gordon)
  - New algorithm to accelerate 1C-NOCIS calculations using generalized non-orthogonal Slater-Condon rules (Rachel Lynn Patterson, Martin Head-Gordon)
  - Implementation of coupling between Dyson orbitals and the continuum (plane wave) (Madhubani Mukherjee, Anna I. Krylov)
  - Add a switch to control averaging in DO/PW coupling calculation (Madhubani Mukherjee, Anna I. Krylov)
  - **Resolved issues with:**

    - Coupled cluster PES scans erroneously reporting SCF energy (Kaushik Nanda)
    - Convergence difficulties in DIP-EOM-CCSDT (Manisha)
    - Crash due to memory allocation issues in Auger decay width calculations for certain basis sets (Saikat Roy, Wojciech Skomorowski)

- **Molecular dynamics, non-adiabatic dynamics, embedding, and solvation:**
  - SAPT+PCM, dielectric boundaries for (X)SAPT calculations (John Herbert)
  - Heterogeneous PCM allows users to modify the dielectric for each atom in PCM (John Herbert)
  - Seminumerical frequency support for the SMD solvation model (John Herbert)
  - Energy-based generalized many-body expansion (GMBE) implementation (Jake A. Tan, Francisco Ballesteros, Ka Un Lao)
  - Density matrix-based generalized many-body expansion (GMBE) implementation (Jake A. Tan, Francisco Ballesteros, Ka Un Lao)
  - Allow user-defined vdW radii for Poisson equation solver, PEqS (John Herbert)
  - Enable user to explicitly set polarization screening factor in EFP (Kaushik Nanda)
  - **Resolved issues with:**
    - Many-body calculations with gradients fail with FILE_MO_COEFS error (Kaushik Nanda, Ryan Steele)
    - Missing EFP polarization energy at every SCF step when USE_LIBQINTS = TRUE (Xintian Feng)
    - Disabled NAC calculations with SMD (John Herbert)
    - Printing spatial overlaps of occ-virt orbital pairs, SPATIAL_OVERLAP_ANALYSIS (John Herbert)

- **Fragment and energy decomposition analysis:**
  - Broken bond orbitals (BBOs) (Alistair J. Sterling, Daniel S. Levine, Abdul Aldossary, Martin Head-Gordon)
  - Enable ALMO(MSDFT) diabatic coupling calculation involving Delta-SCF excited states (Yuezhi Mao)
  - Uncoupled FERF (Hengyuan Shen, Abdul Aldossary, Martin Head-Gordon)

- **Incorporation of quantum nuclear effects (NEO suite):**
  - NEO-PCM analytic Hessian for solution phase frequency and normal mode analysis (Mathew Chow, Sharon Hammes-Schiffer)
  - External point charges are now supported for NEO electrostatic embedding calculations (Mathew Chow, Sharon Hammes-Schiffer)
  - Enabled exchange-correlation functionals that have additional dispersion correction baked-in (e.g., wB97X-D, wB97X-D3, B3LYP-D3(BJ), etc.) to be used in NEO-DFT (Mathew Chow, Sharon Hammes-Schiffer)
  - Added full DFT-D support for NEO (energies, gradients, Hessians) and RT-NEO (real-time and real-time-Ehrenfest) calculations (Mathew Chow, Sharon Hammes-Schiffer)
  - Pseudopotentials are now fully supported with NEO (including energies, gradients, and Hessians) (Mathew Chow, Sharon Hammes-Schiffer)
  - DIIS_GDM is now supported for the NEO simultaneous solver (Mathew Chow, Sharon Hammes-Schiffer)
  - **Improved performance by:**
    - Simultaneous GDM optimization for NEO-HF and NEO-DFT methods (Mathew Chow, Sharon Hammes-Schiffer)
    - Nuclear SCF initial guess (Mathew Chow, Sharon Hammes-Schiffer)

    - Generalized simultaneous DIIS and GDM optimization to treat 'N' number of quantum protons to achieve faster NEO Hartree Product calculations (Mathew Chow, Sharon Hammes-Schiffer)
    - Various improvements to simultaneous NEO-PCM calculations, including support with Hartree product and multiple proton CNEO Hessian calculations (Mathew Chow, Sharon Hammes-Schiffer)
    - Updates and improvements to general orbital optimizer library (Christopher Malbon, Sharon Hammes-Schiffer)
    - Performance improvements to CNEO Hessian calculations via addition of preconditioner (Eno Paenurk, Sharon Hammes-Schiffer)
    - Improved stability and performance for NEO Hessian and CNEO Hessian codes (Mathew Chow, Sharon Hammes-Schiffer)
  - **Resolved issues with:**
    - Fixed issue where nuclear AUTOSAD would not read NEO_PURECART for child instructions (Mathew Chow, Sharon Hammes-Schiffer)
    - Fixed issue where MAX_SCF_CYCLES did not have an effect on the simultaneous NEO-SCF solvers (Mathew Chow, Sharon Hammes-Schiffer)
    - Fixed memory issue for calculations using NEO simultaneous SCF combined with NEO Hartree product (Mathew Chow, Sharon Hammes-Schiffer)

1.3.3 New Features in Q-CHEM 6.2

1.3.3.1 Features in 6.2.0

- **Changes to default behavior:**
  - Set POINT_GROUP_SYMMETRY to FALSE for freq jobs reading guess from previous job to avoid reorientation
  - Make NO_REORIENT = 1 and POINT_GROUP_SYMMETRY = 0 the defaults for harmonic confiner jobs
  - Make improved initial guess Hessian the default for optimizations (INITIAL_HESSIAN = MODEL)
  - Enable RI-K virtual orbital correction by default for occ-RI-K frequency jobs
- **General features and improvements:**
  - NEO Multistate DFT (NEO-MSDFT) (Joseph Dickinson, Qi Yu, Sharon Hammes-Schiffer)
  - Added simulation of nuclear-electronic quantum dynamics based on real-time nuclear-electronic orbital (RT-NEO) (Tao E. Li, Mathew Chow, Sharon Hammes-Schiffer)
  - Added support for D3 empirical dispersion correction (energies and gradients) and Effective Core Potential (ECP) (energies and gradients) for NEO approach and RT-NEO methods (Mathew Chow, Sharon Hammes-Schiffer)
  - Constrained NEO (CNEO) for a single NEO center (Eno Paenurk, Sharon Hammes-Schiffer)
  - Addition of SCS-RIMP2 and SOS-OOMP2 to NEO methods (Jonathan Fetherolf, Sharon Hammes-Schiffer)
  - Faster NEO-SCF via simultaneous DIIS optimization (Mathew Chow, Sharon Hammes-Schiffer)
  - Add atomic multipole moment calculation using IAOs (Alexandra McIsaac, Abdulrahman Aldossary, Martin Head-Gordon)
  - Update the list of occasions not allowed for the AUTOSAD -> CORE fallback when atomic jobs fail (Yuezhi Mao)

- New syntax allowing external electric field to follow a bond (Tarek Scheele, Tim Neudecker)
- Real-time extension of TAO-DFT (RT-TAO) (Hung-Yi Tsai, Jeng-Da Chai)
- SCF energy prints now break one-electron terms into kinetic, nuclear attraction and remainder when SCF_FINAL_PRINT is set to TRUE
- Resolved issues with:
  - Incorrect hydrogen isotope mass when using EPC functionals with NEO (Mathew Chow, Zhen (Coraline) Tao, Sharon Hammes-Schiffer)
  - NEO Hessian implementation for range-separated functionals (Eno Paenurk and Sharon Hammes-Schiffer)
  - Failure of the READ_REAL guess of complex RHF (Yuezhi Mao)
  - Libopt3 misclassification of structure type for near-zero eigenvalues
  - "SVD Failed" invalid step error in rare geometry optimization cases

- **Density functional theory and self-consistent field:**
  - Implemented DFT/CIS, including a new parameterization for X-ray spectroscopy (Aniket Mandal, John Herbert)
  - Addition of new TZVP basis set for DFT/CIS to QCAUX (Aniket Mandal, John Herbert)
  - Implemented suppression of open-shell mixing for ROKS calculations (Juanes Arias-Martinez, Martin Head-Gordon)
  - Enable negative s8 values for Grimme's D4 dispersion correction
  - Dipole filtering for restricted and unrestricted TDKS (John Herbert, Avik Kumar Ojha)
  - Print spatial overlaps of orbitals (intermediate overlaps for occ/virt pairs) (John Herbert)
  - Make virtual orbital correction after SCF convergence optional for occ-RI-K
  - Enable MakeMeSAD for GEN_SCFMAN
  - Add SAD guesses for cc-pV6Z and aug-cc-pV6Z
  - Disable AUTOSAD guess for CDFT-CI (John Herbert)
  - Disable NTOs for RPA (John Herbert)
  - Resolved issues with:
    - Job control through FDIFF_DER for anharmonic frequency calculations (Yuezhi Mao)
    - Anharmonic frequency calculation failure with DFT (Yuezhi Mao)
    - Restart initialization for unrestricted TDKS (Hung-Yi Tsai, Jeng-Da Chai)
    - Bug in RI related to basis sets that use multishells (note that this does not affect most RI calculations, unless an auxiliary multishell basis is used)

- **Correlated methods:**
  - Added hyperfine coupling code to calculate couplings between singlet/triplet excited CIS states (Samuel May, Joseph Subotnik)
  - Generalization of 1C-NOCIS to two-electron open-shell singlets (Juanes Arias-Martinez, Hamlin Wu, Martin Head-Gordon)
  - Implemented the ACP-EOMIP-CCSD method to compute partial Auger decay widths (Florian Matz, Thomas Jagau)
  - Built-in implementation of calculations of cross-sections and angular distributions using Dyson orbitals (Madhubani Mukherjee, Sarai Folkestad, Anna Krylov)

  - Natural Auger Orbitals for Auger decay, ICD, and related processes for CVS-EOM methods (Nayanthara K. Jayadev, Anna I. Krylov)
  - Added BW-s2 method (Kevin Carter-Fenk, Martin Head-Gordon)
  - Stochastic resolution of identity to CC2 (sRI-CC2) for excited state energy calculations (Chongxiao Zhao, Wenjie Dou, Joonho Lee)
  - EOM-CCSDT for EE and SF states (Manisha, Prashant Uday Manohar)
  - Resolved issues with:
    * Legacy RI-MP2 code crashes when MEM_STATIC value is too large (Yuezhi Mao)
    * Error in calculation of the separable part of the one-particle density matrix with CCSD (Yongbin Kim, Madhubani Mukherjee, Anna Krylov)
    * Overwriting of Molden files for natural Auger orbitals in resonant Auger decay calculations (Nayanthara Karippara Jayadev, Anna Krylov)

- **Molecular dynamics, non-adiabatic dynamics, embedding, and solvation improvements:**
  - External/embedded potential read (Huseyin Aksu, Barry Dunietz)
  - Turn on PCM at the first SCF iteration when SCF_GUESS = READ, allowing faster convergence (Yuezhi Mao)
  - Turn on PCM at the first SCF iteration in the 2nd geom opt cycle for FRAGMO and AUTOSAD (Yuezhi Mao)
  - Clarified ambiguous output for COSMO (John Herbert)
  - Added analytic X-HCFF Hessian (Rahel Weiss, Felix Zeller, Tim Neudecker)
  - Resolved issues with:
    * Out-of-bound access error for SMX with ghost atoms
    * Undesirable behavior of SOLVENT_METHOD = TRUE invoking Kirkwood model (John Herbert)
    * Add missing qarchive files for QM/MM calculations (Xiaoliang Pan, Yihan Shao)
    * Incorrectly increasing van der Waals radii for I and At atoms in jobs with multiple SMD calculations (Zheng Pei)

- **Fragment and energy decomposition analysis improvements:**
  - Add additional printout for libwfa regarding ionic/covalent states (Felix Plasser)
  - Resolved issues with:
    * Occasional crash in non-perturbative CTA (Yuezhi Mao)

- **Post-processing features:**
  - Calculations of g-tensors using ezMagnet (Sven Kähler, Antonio Cebreiro-Gallardo, Pavel Pokhilko, David Casanova, and Anna I. Krylov)

1.3.4 New Features in Q-CHEM 6.1

1.3.4.1 Features in 6.1.1

- **Changes to default behavior:**
  - Set TDKS = FALSE in fragment REM variable (*$rem_frgm*) section (John Herbert)

  - Remove redundant CIS_ and SET_ variables (John Herbert)
  - Update the XDM damping parameters and added variables for user control (John Herbert)
  - Formatted printing of SCF energy (John Herbert)

- **General features and improvements:**
  - Add ability to print CHELPG grid points and weights (John Herbert)
  - Add Fock matrix correction for RI-JK kernels (Xintian Feng)
  - Increase precision for orbital energies printout, allowing 1s energies to be obtained for heavy elements (John Herbert)

- **Density functional theory and self-consistent field:**
  - Add XYG3, XYGJ-OS to the list of available methods (Yuezhi Mao)
  - Disable double-hybrids and post-HF methods with CDFT (Yuezhi Mao)
  - Improved error catching in TRNSS LR-TDDFT code for core-level excitations (John Herbert)
  - Maximum Overlap Method for ROKS Calculations (Kevin Carter-Fenk, Martin Head-Gordon)
  - Fixed issues with:
    * unclear value of NATOMS_MOL1 in XDM user interface (John Herbert)
    * κ value printing in XDM10 (John Herbert)

- **Correlated methods:**
  - Added hyperfine coupling code to calculate couplings between singlet/triplet excited CIS states (Samuel May, Joseph Subotnik)
  - Improved MOM functionality in the ccman2 Davidson solver (Florian Matz, Quyen Nguyen, Thomas-C. Jagau)
  - Support for TDDFT/TDA-SOC overlaps and external magnetic fields (Yanze Wu, Joseph Subotnik)
  - Acceleration for TDDFT/TDA-SOC derivative coupling calculations, especially for time-reversal cases (Yanze Wu, Joseph Subotnik)
  - Added triplet support for CIS/TDDFT overlaps (Yanze Wu, Joseph Subotnik)
  - Added correct header label for NOONs in CAS-SCF calculation output (John Herbert)
  - Implemented stochastic resolution of indentity approximation to CC2 model (sRI-CC2) for ground-state and excited state energy calculations (Chongxiao Zhao, Wenjie Dou, and Joonho Lee)
  - Fixed issues with:
    * typo in SOC output labels (“ms=1" changed to “ms=-1") (Yanze Wu, Joseph Subotnik)
    * acceptor-donor order dependence in RAS-FCD and RAS-FED (Aaditya Manjanath, Chao-Ping Hsu)

- **Molecular dynamics, non-adiabatic dynamics, embedding, and solvation:**
  - Enable SMx debug printing (John Herbert)
  - Improve error catching for incorrect PCM inputs (John Herbert)
  - Fixed issues with:
    * error handling for **Td_CEq_NonEq** in PCM with SF-TDDFT calculations (John Herbert)
    * printing bug in PCM (Kuan-Yu Liu)

- **Fragment and energy decomposition analysis:**

  - Change the fragment index convention in non-perturbative PolA/CTA to start from 1 (Yuezhi Mao)

- **Miscellaneous:**

  - Disable analytic nuclear Hessian for ROKS jobs (Yuezhi Mao)
  - Formatted printing for TDDFT with truncated subspace (John Herbert)
  - Replace term “Energy Transfer" with "Polarization Energy" in non-perturbative PolA/CTA (Yuezhi Mao)
  - Replace term “Charge Transfer" with "On-Fragment Charge Shift" in non-perturbative PolA/CTA (Yuezhi Mao)
  - TDDFT formatted print for core-level excitations (John Herbert)

1.3.4.2 Features in 6.1.0

- **Changes to default behavior:**

  - Renamed SASF_RPA to SASF_CIS
  - Renamed TDDFT_PCM to TDDFT_LR_PCM (John Herbert)
  - Renamed SYMMETRY to INTEGRAL_SYMMETRY and made INTEGRAL_SYMMETRY default to FALSE
  - Added POINT_GROUP_SYMMETRY
  - Add ENABLE_ARCHIVE keyword to control writing of archive files (Eric Berquist, Peter McLaughlin)

- **General features and improvements:**

  - Implemented gauge-independent atomic orbital (GIAO) basis for SCF calculations (Jonathan Wong, Brad Ganoe, Tim Neudecker, Adam Rettig, Xiao Liu, Joonho Lee)
  - Implemented the GIAO complex calculations for DFT jobs (Jiashu Liang)
  - Implemented on-the-fly SQC and Ehrenfest nonadiabatic dynamics algorithms using restricted CIS and TDDFT/TDA (Justin Talbot, Stephen J. Cotton, Martin Head-Gordon)
  - Implemented slater transition method and transition potential method for calculating core-level binding energies and X-ray absorption spectra (Subrata Jana, John Herbert)
  - Generated data for vibronic decomposition analysis (Kuan-Yu Liu, Peter McLaughlin, Andrew Gilbert)
  - Enabled state following with libopt3
  - Changed transition state geometry optimization default to new optimizer (Peter McLaughlin)
  - Implemented new initial model Hessian for geometry optimization (Peter McLaughlin)
  - Enabled new plots section format for ESP cube plots (Yuezhi Mao)
  - Added vibrational circular dichroism (VCD) spectroscopy (Yu Zhang, Kuan-Yu Liu, Eric Berquist, Evgeny Epifanovsky)
  - Added RRHO print out (Abdul Aldossary, Alistair Sterling, Tim Schramm)
  - Added computation of MO overlaps at displaced geometries (John Herbert)
  - Resolved issues with:
    * performance of B97M-V gradients
    * NEO-epc19 deuterium mass being passed incorrectly (Zhen (Coraline) Tao, Mathew Chow, Sharon Hammes-Schiffer)
    * NEO-epc19 gradients (Zhen (Coraline) Tao, Mathew Chow, Sharon Hammes-Schiffer)

    - NEO-epc172 Hessian and multiple proton NEO Hessian (Zhen (Coraline) Tao, Mathew Chow, Sharon Hammes-Schiffer)
    - missing STEP constraint in RHF jobs (Kevin Carter-Fenk)
    - Gaussian blur failing to work with ECP (John Herbert)
    - many-body expansion job failure above fourth order

- **Density functional theory and self-consistent field:**
  - Enabled spin-orbit NTO analysis for TDDFT and SF-TDDFT (Saikiran Kotaru)
  - Implemented electron-affinity time-dependent density functional theory (Kevin Carter-Fenk, Juan Arias-Martinez, Leonardo Cunha)
  - Enabled projection-based embedding with ECPs (Valentina Parravicini, Thomas Jagau)
  - Added a memory check before SCF for DFT jobs
  - Improved EA-TDDFT/STEX job control (Kevin Carter-Fenk)
  - Added SPADE localization capabilities and initial guess functionality in EA-TDDFT/STEX jobs (Kevin Carter-Fenk)
  - Resolved issues with:
    - TDA excited state frequencies crash
    - double counting in VV10 calculations
    - meta-GGA TDDFT analytical frequency

- **Correlated methods:**
  - Added ability to obtain partial widths for CCSD calculations on core-vacant states (Florian Matz, Thomas Jagau)
  - Implemented RI-CC2-EOM-SF, EA, and IP (Garrette Paran, Thomas Jagau, Cansu Utku)
  - Implemented CCSDT (Prashant Uday Manohar, Manisha)
  - Enable the use of new plots section format for the excited-state analysis of ADC jobs (Yuezhi Mao)
  - Correct coefficient of EOM-CC 2PA cross section with circular polarization (Kaushik Nanda)
  - Add DALTON solver for damped CC equations to speed up convergence (Kaushik Nanda)
  - Allow user-defined active and inactive frozen-core orbitals for fc-CVS-EOM-CC calculations (Kaushik Nanda)
  - Resolved issues with:
    - Failure of iFCI when using GEN_SCFMAN
    - Failure when calculating CC transition properties from the initial CCSD state when both CVS-EOM-EE-CCSD and CVS-EOM-IP-CCSD states are requested together (Kaushik Nanda)

- **Molecular dynamics, non-adiabatic dynamics, embedding, and solvation:**
  - Implemented PCM contribution for nuclear-electronic orbital (NEO) method (NEO) (Mathew Chow, Sharon Hammes-Schiffer)
  - Disabled $N = 14$ PCM discretization grid (John Herbert)
  - New and improved state-specific (SS)-PCM-TDDFT (Thomas Froitzheim, John Herbert, Jan-Michael Mewes, Stefan Grimme)
  - Print clearer errors for incorrect PEqS input formatting (John Herbert)
  - Resolved issues with:

    - SMD gradient with IEF-PCM electrostatics (John Herbert)
    - SS(V)PE and IEF-PCM gradients in NEO calculations (Mathew Chow, Sharon Hammes-Schiffer)
    - AIMD crash with DACF calculations (J. A. Gyamfi, Thomas C. Jagau)
    - Failure of frozen energy gradient calculations with implicit solvent when IDERIV = 0 (Yuezhi Mao)

- **Fragment and energy decomposition analysis:**
  - Implemented decomposition of molecular polarizability using the iterative Hirshfeld scheme (Yihan Shao, Yuezhi Mao)
  - Implemented force decomposition analysis (Abdulrahman Aldossary, Yuezhi Mao, Marti Gimferrer)
  - Added POL and NOCV analysis for EDA2 (Hengyuan Shen)
  - Enabled calculation of non-perturbative PolA/CTA with ALMO-EDA2 (Hengyuan Shen, Yuezhi Mao)
  - Added pre- and post-processing scripts for XSAPT (Montgomery Gray, John Herbert)
  - Resolved issues with:
    - schema-related crash for FSSH jobs (Yuezhi Mao, Justin Talbot)
    - error in MO SAPT calculation of $E_{\text{elst}}^{(1)}$ and $E_{\text{exch}}^{(1)}$ (Montgomery Gary, John Herbert)

- **Miscellaneous:**
  - Disabled SMD for MECP (John Herbert)
  - Added ability to define solvent dielectric constants in PCM using solvent name (John Herbert)
  - Added warning for BrianQC jobs if the feature is not supported GEN_SCFMAN = FALSE
  - Remove an outdated warning about use of ECPs with Gaussian blurring (John Herbert)
  - Enabled comments in all input sections (John Herbert)
  - Fixed memory leak for X2C with ROHF orbitals (Kevin Carter-Fenk)
  - Allow input parser to interpret 1 as TRUE (Eric Berquist)

1.3.5 New Features in Q-CHEM 6.0

1.3.5.1 Features in 6.0.2

- **General features and improvements:**
  - Enabled Molden support for the new default geometry optimizer
  - Improved output for rxn path (IRC) calculations (John Herbert)
  - Improved formatting of output for orbital radii of gyration (John Herbert)
  - Improved error-catching for input section (John Herbert)
  - Inlcuded one-electron matrices in fchk and archive
  - Resolved pure and Cartesian assignment of nuclear-electronic orbital (NEO) basis (Mathew Chow, Sharon Hammes-Schiffer)
  - Resolved issues with:
    - AUTOSAD/I-Hirshfeld multiplicities for highly-charged transition metal atoms (Kevin Carter-Fenk)
    - molden output for penalty function optimization
    - calculations of the exact gradients and derivative couplings of CIS Boys diabatic states (Vishikh Athavale, Joseph Subotnik)

    - immutable SOS_UFACTOR in SOS-CIS(D) calculations
    - inconsistent charges and multipole moments in DUAL_BASIS_ENERGY calculations
    - SCF_GUESS_MIX calculations with SAD guess (Yuezhi Mao)

- **Density functional theory and self-consistent field:**
  - Improved algorithm for RI-JK derivatives
- **Correlated methods:**
  - Enabled plotting of RAS-CI properties in cube, MOLDEN, and FChk files (Abel Carreras, David Casanova)
- **Molecular dynamics, non-adiabatic dynamics, embedding, and solvation:**
  - Added printing of single-atom thermodynamic information for PCM jobs (John Herbert)
- **Fragment and energy decomposition analysis improvements:**
  - Resolved issues with erroneous interaction energies with mixed basis sets for fragments (Yihan Shao, Yuezhi Mao)
- **Miscellaneous:**
  - Added sample jobs for SCF_ALGORITHM options NEWTON_CG and NEWTON_MINRES (Yuezhi Mao)
  - Added sample job for EDA2 calculation with external charges (Yuezhi Mao, Yihan Shao)
  - Renamed sample jobs for variational PA/CTA based on GEN_SCFMAN_EDA2 (Yuezhi Mao, Hengyuan Shen)
  - Added SOC-NTO analysis for TDDDFT and SF-TDDFT (Saikiran Kotaru, Anna Krylov)

1.3.5.2 Features in 6.0.1

- **General features and improvements:**
  - Restored support for IGDESP used by CHARMM (John Herbert)
  - Implemented a memory-efficient GOSTSHYP algorithm (Felix Zeller)
  - Enabled finite-field chemical shielding and magnetizability calculations using gauge-independent atomic orbitals (GIAOs) for HF and MP2 (Jonathan Wong, Brad Ganoe, Tim Neudecker, Adam Rettig, Xiao Liu, Joonho Lee)
  - Resolved issues with:
    - molden output for optimization in libopt3
    - libopt3 Hessian calculations using PCM
    - writing MO coefficients and energies to qarchive file
    - cleaned up topology checking printing
    - fixed printing of RMS of step size during geometry optimization
- **Molecular dynamics, non-adiabatic dynamics, embedding, and solvation improvements:**
  - Implemented state-specific PCM with RAS-SF (Bushra Alam, Hanjie Jiang, John Herbert, Paul Zimmerman)
- **Fragment and energy decomposition analysis improvements:**

  - Resolved issues with projection-based embedding calculations where frozen (environment) occupied orbitals are not ordered based on energies (Yuezhi Mao)

- **Miscellaneous:**

  - Added comments to DFT SOC calculations for identifying the progress of the job (Saikiran Kotaru)
  - Added warning about not having analytic Hessians for some optimization jobs
  - Added printing to distinguish CPCM1 from CPCM2

1.3.5.3 Features in 6.0.0

- **Changes to default behavior:**

  - Tightened default integral threshold (THRESH) to SCF_CONVERGENCE + 4 and used same threshold for DIIS and GDM
  - Set default of FD_MAT_VEC_PROD to FALSE for VV10 functional (Yuezhi Mao)
  - Turned off automatic evaluation of electrostatic potentils on a grid (Felix Plasser)
  - Set finite difference as default for energy derivatives in electric field (Yuezhi Mao)

- **General features and improvements:**

  - Next-generation interface of Q-Chem with external tools (generation of archive files in the HDF5 format)
  - Implemented the nuclear-electronic orbital CCSD (NEO-CCSD) method (Fabijan Pavosevic, Sharon Hammes-Schiffer)
  - Implemented NEO-TDDFT analytical gradient and Hessian (Zhen (Coraline) Tao, Patrick E. Schneider, Sharon Hammes-Schiffer)
  - Enabled subset selection of atoms in NMR J-coupling calculations (JOBTYPE = ISSC) via *$spin-spin* input section
  - Disabled steepest descent in geometry optimization with fixed atoms
  - Added delocalized natural internal coordinate optimization in new optimizer
  - Updated geometry in the MOLDEN file for each step in finite difference optimizations (John Herbert)
  - Stabilized density fitting for JK and MP2
  - Set new optimizer as default for unconstrained optimization (GEOM_OPT_DRIVER = LIBOPT3)
  - Added the minimal-augmented and heavy-augmented versions of the Karlsruhe basis sets (John Herbert)
  - Removed MPI support
  - Resolved issues with:
    * incorrect Hirshfeld charges based on molecule input orders (Abdulrahman Aldossary)
    * not-a-number (NAN) errors in SOC calculations
    * missing nuclear repulsion energies in Fock projection (BASIS2) calculations
    * removed restriction on number of atoms (MAX_ATOM) that can be included in random search and basin hopping
    * ordering of localized MOs in formatted checkpoint files (Abdulrahman Aldossary)
    * missing ECP for the def2-SVPD basis set
    * failure to compute NMR properties with linearly dependent basis sets
    * parsing input files with 100k+ lines

    - character table of C3 point group

- **Density functional theory and self-consistent field:**
  - Accelerated convergence of the SCF algorithm ADIIS and add a new combined algorithm option ADIIS_DIIS. (Yuezhi Mao)
  - Enabled gauge-independent atomic orbitals (GIAOs) in SCF calculations using GEN_SCFMAN (Brad Ganoe, Tim Neudecker, Joonho Lee, Adam Rettig, Jonathan Wong)
  - Disabled user setting of coefficients (via HFK_LR_COEF and HFK_SR_COEF), if using built-in range-separated functionals
  - Implemented frequency calculation and analytic Hessian for the VV10 functional (Jiashu Liang)
  - Enabled generation of formatted checkpoint files in CIS/TDDFT calculations with frozen occupied/virtual orbitals via GUI = 2 (Yuezhi Mao)
  - Enabled STATE_ANALYSIS for the new plot section (PLOT = 1) (Yuezhi Mao)
  - Performed consistency check on TDKS Fock matrices based on the SCF convergence threshold instead of the field amplitude (John Herbert)
  - Added new energy density functionals: revSCAN, regSCAN, r++SCAN, r2SCAN, r4SCAN, TASK, mTASK, regTM, rregTM, revTM
  - Enabled computing spin-orbit couplings (SOC) (1-electron and 2-electron mean-field) with TDDFT (both restricted and unrestricted) and spin-flip TDDFT (SF-TDDFT) (Saikiran Kotaru, Ana Krylov)
  - Implemented analytic gradient for density-corrected DFT (DC-DFT) for self-interaction correction (Marc Coons, Bhaskar Rana, John Herbert)
  - Resolved issues with:
    - incorrect results of fractional electron SCF calculations using GEN_SCFMAN (Yuezhi Mao)
    - hanging qints (USE_LIBQINTS = TRUE) jobs with large number of OpenMP threads
    - non-variational initial SCF guess for ADIIS (Yuezhi Mao)
    - incorrect memory estimation in TDDFT/TDA calculations
    - crash of TDA excited state frequency jobs
    - crash of geometry optimization with fixed atoms
    - frequency calculations using basis functions with g or higher angular momenta
    - sign error with TDDFT spin-orbit coupling calculations (Nicole Bellonzi)
    - crash of projection-based embedding calculations (Yuezhi Mao)
    - incorrect result of RPA TDDFT frequency using non-Pople basis set
    - insufficient memory allocation for NMR calculations with meta-GGA functionals
    - erroneous results in DC-DFT calculations using hybrid functionals with larger basis sets (Marc Coons, Bhaskar Rana, John Herbert)
    - crash of excited state potential energy surface scans with CIS/TDDFT (John Herbert)

- **Correlated methods:**
  - Implemented EOM oscillator strengths in velocity and mixed gauges (Josefine Andersen, Sonia Coriani)
  - Implemented CCSD optical rotation evaluation (Josefine Andersen, Kaushik Nanda)
  - Implemented the fragment charge difference (FCD) scheme in RASMAN2 (Chou-Hsun (Jeff) Yang, Aaditya Manjanath, Chao-Ping (Cherri) Hsu)
  - Implemented complex-valued CC2, RI-CC2, and RI-CCSD (Cansu Utku, Garrette Paran, Thomas Jagau)

  - Implemented the complex absorption potential (CAP) method in AIMD calculations (Jerryman A. Gyamfi, Thomas Jagau)
  - Implemented the v2RDM-CASSCF-PDFT method using density fitted basis sets (Mohammad Mostafanejad, Run Li, A. Eugene DePrince III)
  - Resolved formatting error in output of SOC calculation with RAS-CI method (Abel Carreras, David Casanova)

- **Molecular dynamics, non-adiabatic dynamics, embedding, and solvation:**
  - Implemented projection-based embedding with complex basis functions (Valentina Parravicini, Thomas Jagau)
  - Enabled user-defined permittivity grid for Poisson equation solver (PEqS) (Suranjan Paul)
  - Improved PCM printing (John Herbert)
  - Implemented CIS and TDDFT wavefunction overlaps including their spin-flip variants for (A)FSSH (Theta Chen, Junhan Chen, Zuxin Jin, Vishikh Athavale, Vale Cofer-Shabica, Joe Subotnik)
  - Resolved issues with QM/MM optimzation not reading previous MOs as a guess for the next cycle

- **Fragment and energy decomposition analysis:**
  - Implemented pairwise fragment excitation energy decomposition analysis (EDA) in QM/EFP calculations (Lyudmila Slipchenko)
  - Increased the maximum angular momentum of basis functions to 5 for XSAPT calculations
  - Implemented SPADE- and ALMO-based partitioning schemes for electric field calculations (Yuezhi Mao)
  - Implemented a new MP2 EDA scheme and added a non-perturbative polarization analysis for DFT EDA (Kevin Ikeda, Hengyuan Shen)
  - Enabled ALMO-CIS/TDA calculations with excitation amplitudes localized on one fragment (Yuezhi Mao)
  - Enabled ALMO-CIS/TDA calculations with excitation from one fragment's occupied orbitals to all virtuals in the system (Yuezhi Mao)
  - Enabled ALMO-CIS/TDA calculations with excitation from one fragment's occupied orbitals to another's virtual orbitals (Yuezhi Mao)
  - Enabled user-defined occupied-virtual pairs in ALMO-CIS/TDA calculations (Yuezhi Mao)
  - Resolved miscellaneous issues with ALMO-CIS and excited-state ALMO-EDA calculations (Yuezhi Mao)

- **Miscellaneous:**
  - Printed orbital kinetic energies using SCF_PRINT = 3
  - Enabled EXTERNAL_CHARGES specification in an external file (Vale Cofer-Shabica, Joseph Subotnik)
  - Added parameter check for many-body dispersion calculations (John Herbert)
  - Restored finite difference for ωB97M-2 and the XYG series of energy functionals
  - Restored finite difference banner for SA-SF-RPA

1.3.6 New Features in Q-CHEM 5.4

1.3.6.1 Features in 5.4.2

- **Changes to default behavior:**

  - Made default SCF convergence criterion for supersystem and fragment jobs in EDA and BSSE calculations consistent (Yuezhi Mao)

- **General features and improvements:**
  - Enabled mixed basis for AUTOSAD guess (Kevin Carter-Fenk, Yuezhi Mao, John Herbert)
  - Enabled compatibility with the NBO7 program (John Herbert)
  - Implementation of intrinsic bond orbital (IBO) analysis (Alexander Zech, Christopher Stein, Abdulrahman Aldossary, Martin Head-Gordon)
  - Resolved issues with:
    * frequency job failure when number of threads is thrice larger than number of atoms
    * frequency job failure when CPSCF segments are equal to number of atoms
    * incorrect alpha density generated when using new plots section format

- **Density functional theory and self-consistent field:**
  - Enabled analytic Hessian for TPPS/TM/SCAN TDDFT calculations
  - Added printing of information about memory requirements for TDDFT (John Herbert)
  - Added an experimental implementation of the X2C method for relativistic quantum chemistry (Diptarka Hait, Leonardo Cunha, Richard Kang, Martin Head-Gordon)
  - Improved CIS/TDA/RPA guess to avoid missing roots
  - Implementation of projection-based embedding with complex basis functions (Valentina Parravicini, Thomas Jagau)
  - Improved performance of the GOSTSHYP method through integral screening (Felix Zeller, Tim Neudecker, Eric Berquist)
  - Resolved issues with:
    * AIFDEM crash when a larger fragment is listed first
    * NAN in SCF energies using VV10 functionals
    * unrestricted RPA TDDFT analytic Hessian for singlet excited state
    * failure to compute non-adiabatic couplings (NACs) using pure TDDFT
    * incorrect TDDFT energies with FAST_XAS using multiple threads
    * incorrect results from projection-based embedding using LRC-DFT as the low-level theory (Yuezhi Mao)
    * incompletely converged energies in RPA calculations
    * failure to evaluate spin-orbit integrals in TDDFT SOC calculations
    * GPU acceleration of unrestricted pure DFT gradient when using BrianQC
    * incorrect ROHF gradient when using BrianQC

- **Correlated methods:**
  - Implementation of CCSD damped polarizability and first hyperpolarizability (Kaushik Nanda)
  - Resolved issues with:
    * wrongfully activated ECD properties with EOM-IP-CCSD (Josefine Andersen, Sonia Coriani)
    * convergence issues in EOM-DIP and EOM-DEA methods
    * symmetry check for v2RDM (Rain Li, Eugene DePrince)
    * failure to write ASCI energy to checkpoint files

- **Molecular dynamics:**
  - Enabled the use of new SCF drivers (GEN_SCFMAN = TRUE) in path integral MD
  - Resolved issues with:
    - missing energy-component file for AIMD when GEN_SCFMAN=TRUE
- **Fragment and energy decomposition analysis:**
  - When EDA2_MOM is used with EDA_BSSE, apply IMOM to BSSE calculations with ghost atoms to prevent collapsing to the lower-energy states (Yuezhi Mao)
  - Allowed SCFMI_MOM and EDA2_MOM to preserve the electronic configuration of the frozen state (Yuezhi Mao)
  - Multiple stability improvements in ALMO-EDA (Yuezhi Mao)
  - Implemented non-perturbative CT analysis for ALMO-EDA (Hengyuan Shen, Srimukh Prasad, Martin Head-Gordon)
  - Resolved issues with:
    - final print of the one-side CT energy in VFB CT analysis incorrectly contained the contribution from SMD's CDS (non-electrostatic) term, when using the SMD solvent model (Yuezhi Mao)
    - double-counting of environment frozen core orbitals with default N_FROZEN_CORE setting for projection-based embedding (Yuezhi Mao)
    - display of preparation energy for ALMO-EDA (Yuezhi Mao)
    - many-body expansion (MBE) geometry optimization (John Herbert)
    - convergence of linear solvers for orthogonal frozen decomposition (Yuezhi Mao)
    - the dispersion term in classic frozen decomposition in non-*Aufbau* ALMO-EDA (Yuezhi Mao)
- **Miscellaneous:**
  - Disabled analytic force calculation with projection-based embedding (Yuezhi Mao)
  - Disabled complex SCF for fragment jobs (Yuezhi Mao)
  - Resolved issue with NAN printing `efield` file for in QM/MM calculations when external charges are set to zero
  - Added warning that CDFT does not support algorithms other than DIIS and RCA (Yuezhi Mao)
  - Added warning when 3c methods are used without recommended basis sets (John Herbert)
  - Added NBO version number in output (John Herbert)
  - Fixed minor spelling errors in the printing of TDDFT (Bushra Alam, John Herbert)

1.3.6.2 Features in 5.4.1

- **Changes to default behavior:**
  - Renamed Onsanger SOLVENT_METHOD to Kirkwood (John Herbert)
  - Updated the SM8 solvation model to use Cartesian Gaussians (PURCAR = 2222) (John Herbert)
  - Renamed spin-specific keywords to EA_ALPHA, EA_BETA, IP_ALPHA and IP_BETA (Wojtek Skomorowski)
- **General features and improvements:**

  - Added Intrinsic Atomic Orbitals (IAO) and Intrinsic Bond Orbitals (IBO) (Abdulrahman Aldossary, Alexander Zech, Christopher Stein)
  - Added a new localization method, Oxidation State Localized Orbitals (OSLO) (Abdulrahman Aldossary, Alexander Zech, Christopher Stein)
  - Included installation of Romberg utilities

- **Density functional theory and self-consistent field:**
  - Implemented hybrid functionals for TAO-DFT (Shaozhi Li, Jeng-Da Chai)
  - Improved efficiency of range-separated DFT frequency calculations when run in parallel with shared memory and multithreading
  - Added option to turn off ground-state PCM calculations for TDDFT (John Herbert)
  - Added option to enforce level shifting in every SCF cycle for state-targeted energy projection (STEP) (Kevin Carter-Fenk)
  - Implemented projection-based embedding for unrestricted calculations (Yuezhi Mao)
  - Added printing of more digits for the TDDFT transition strength
  - Improved SCF guess for optimization jobs using BASIS2
  - Resolved issues with:
    * segmentation fault in CIS frequency calculations when using libqints
    * index out of bounds error with TDKS sample in manual (Hung-Yi Tsai, Jeng-Da Chai)
    * errors in unrestricted TDDFT Hessian calculations
    * small error in RPA excitation energies
    * incorrect SCF energy with libqints-based SRC-DFT
    * crash when computing numerical derivatives with BASIS = GEN
    * crash while running large frequency jobs due to insufficient memory in CPSCF
    * incorrect evaluation of iterative Hirshfeld charges (Abdulrahman Aldossary)

- **Correlated methods:**
  - Added options for custom scaling in complex basis function calculations (Florian Matz, Thomas Jagau)
  - Improved projected CAP-EOM-CC (James Gayvert)
  - Implemented EOM-DEA-CCSD two-photon absorption (Kaushik Nanda, Sahil Gulania, Anna Krylov)
  - Implemented complex-valued CC2 and RI-CCSD (Cansu Utku, Garrette Pauley Paran, Thomas Jagau)
  - Implemented effective nuclear charge approximation for SOCs using EOM (Saikiran Kotaru, Anna Krylov)
  - Resolved issues with:
    * freezing string method (FSM) reading SCF energy instead of correlated energy value
    * missing triples corrections for EOM calculations in ccman2 (Pavel Pokhilko)
    * using frozen core and virtual orbitals in projector-based embedding calculations (Yuezhi Mao)

- **Large systems, QM/MM, and solvation:**
  - Implemented user-defined permittivity grid for Poisson equation solver (PEqS) (Suranjan Kumar Paul)
  - Enabled SCRF for GEN_SCFMAN-based ROHF/ROKS calculations (Yuezhi Mao)
  - Implemented state-specific PCM/TDDFT (SS-PCM/TDDFT) method based on the constrained equilibrium theory (Haisheng Ren, Fan Wang, Xiangyuan Li, Yingli Su)

  - Improved GROMACS QM/MM interface (Vale Cofer-Shabica)
  - Improved gradient performance of the SM8 solvation model (John Herbert)
  - Improved memory usage of the SM8 solvation model (John Herbert)
  - Added TDDFT_LR_PCM to control linear-response solvent correction (John Herbert)

- **Fragment and energy decomposition analysis:**

  - Enabled the linearized approximation in projection-based embedding (Yuezhi Mao)
  - Implemented POD2L and POD2GS for projection operator diabatization (POD) (Yuezhi Mao)
  - Enabled calculation of couplings between multiple pairs of diabatic orbitals for POD (Yuezhi Mao)
  - Added printing of separate energy components in the SAPT output (John Herbert)

1.3.6.3 Features in 5.4.0

- **Changes to default behavior:**

  - Use of automatically generated superposition of atomic densities SCF guess for custom basis sets (Yuezhi Mao, Kevin Carter-Fenk)
  - Use atomic size-corrected Becke weights for CDFT (Kevin Carter-Fenk)

- **General features and improvements:**

  - New methods to distort molecules using force and pressure: HCFF, X-HCFF, GOSTSHYP (Tim Stauch, Maximilian Scheurer)
  - Overhauled library of standard basis sets for consistency with Basis Set Exchange and extended support through element 118
  - Improved stability of ECP fitting and updated definitions of fitted ECPs (CRENBS, CRENBL, HWMB, LACVP, LANL2DZ, SBKJC)
  - Evaluation of electric field at nuclei (Yuezhi Mao)
  - Frequency calculations for rigid fixed-atom constraints (Saswata Dasgupta)
  - Save additional calculation output files to unique folder
  - Resolved issues with:
    * inconsistent application of quadrupole field to resolve orbital degeneracies
    * definition of jun-cc-pVDZ basis set (John Herbert)
    * some jobs crashing with the FILE_SET_SYM_REP read error
    * cleaning up in PES scan jobs on Windows
    * unnecessary gradient evaluation at every point of frozen PES scan

- **Features and improvements in density functional theory and self-consistent field:**

  - TAO-DFT for global hybrid GGAs (Jeng-Da Chai)
  - Vibronic and resonance Raman spectroscopy (Xunkun Huang, Huili Ma, WanZhen Liang)
  - Integrated DFT-D4 empirical dispersion model (Kuan-Yu Liu, Romit Chakraborty)
  - New implementation of direct propagation of the time-dependent Kohn-Sham equation (real-time TDKS) with support for unrestricted SCF and implicit solvation (Ying Zhu, John Herbert)
  - State-targeted energy projection method (Kevin Carter-Fenk, John Herbert)

  - Multiple improvements to frozen-density embedding methods (Cristina Gonzalez-Espinoza, Alexander Zech, Tomasz A. Wesolowski)
  - Faster algorithm for ωGDD tuning (John Herbert)
  - Improvements in the IP/EA omega tuning scripts for long range corrected functionals (John Herbert)
  - Support for high angular momentum in DFT frequency calculations
  - Superposition of atomic potentials (SAP) guess for SCF (Yu Zhang, Susi Lehtola)
  - Expand density functionals available for NMR chemical shift calculations (Jiashu Liang, Khadiza Begam, Barry Dunietz, Yihan Shao)
  - Nuclear gradient and analytical 2nd functional derivative of the VV10 functional (Jiashu Liang)
  - Performance improvements in the evaluation of DFT-D3 nuclear hessian contribution
  - Consistent constrained DFT and SCF convergence criteria (Kevin Carter-Fenk)
  - NVIDIA GPU computing improvements via interface with BrianQC:
    * Accelerated force and vibrational frequency computations with range-separated functionals
    * Accelerated Fock derivative computation in DFT vibrational frequency jobs
  - Resolved issues with:
    * buffer overflow in a special case of very large DFT jobs
    * a special case of crashing unrestricted CIS derivative coupling calculations
    * evaluation of finite-difference nonlocal correlation orbital Hessian (Yuezhi Mao)
    * use of AO integrals in general response module
    * differences in DFT quadrature between Linux and macOS
    * using ghost atoms in MBD-vdW calculations (Kevin Carter-Fenk, Evgeny Epifanovsky)
    * using arbitrary density functionals for MBD-vdW and TS-vdW
    * crashing large CIS state following calculations
    * SOC constants with unrestricted TDDFT
    * RI-J/RI-K gradient
    * DFT hyperpolarizabilities

- **Features and improvements in correlated methods:**
  - Calculation of electronic g-tensors with CCSD (Sven Kähler, Anna Krylov);
  - Calculation of electronic circular dichroism (ECD) using EOM-CC (Josefine Andersen, Sonia Coriani)
  - Evaluation of spin-orbit couplings using CVS-EOM methods, L-edge XAS/XES spectroscopy calculations (Marta Vidal, Pavel Pokhilko, Sonia Coriani)
  - Feshbach method with EOM-CC states and Coulomb wave expanded in terms of plane wave Gaussian type orbitals (Wojciech Skomorowski)
  - Improved performance in small to medium CC/EOM jobs via in-core computations
  - Improvements in projected CAP EOM-CC (James Gayvert)
  - IP/EA-ADC methods and intermediate state representation (ISR) properties (Adrian Dempwolff, Matthias Schneider, Alexander Paul)
  - Dramatic speedup of ADC(3) (Adrian Dempwolff)
  - Improved fourth-order static self-energy for all ADC variants (PP (EE), IP, EA) (Adrian Dempwolff)
  - Subspace-projected CAP-ADC for all ADC variants (PP (EE), IP, EA) (Adrian Dempwolff)

  - Evaluation of spin-orbit couplings using RAS-CI and RAS2-SF methods (Abel Carreras, Anna Krylov, David Casanova, Hanjie Jiang, Pavel Pokhilko, Paul M. Zimmerman)
  - Use of resolution-of-the-identity integrals in LibRASSF-based implementation of RAS-SF (Shannon Houck)
  - Implementation of the Bloch effective Hamiltonian approach within LibRASSF-based RAS-SF (Shannon Houck)
  - Experimental implementation of the CC2 and RI-CC2 methods (Garrette Paran, Thomas Jagau)
  - Implementation of the Brueckner CC2 method (Adam Rettig)
  - Implementation of direct RPA for the ground state correlation energy (Joonho Lee)
  - Cubic storage RI-MP3 and Laplace-transformed RI-MP2 and RI-MP3 (Joonho Lee)
  - Added access to κ-regularized orbital optimized MP2 via METHOD = koomp2
  - New implementation of v2RDM and v2RDM-CASSCF solvers (Rain Li, Wayne Mullinax, Eugene DePrince, Marcus Liebenthal)
  - Improved defaults in incremental FCI (Alan Rask)
  - Experimental implementation of tensor hypercontraction methods (Joonho Lee)
  - Resolved issues with:
    * 2 GB limit on temporary files in CC/EOM/ADC calculations on Windows
    * evaluation of analytic gradients of κ-regularized OO-MP2
    * crashing in fragment excitation difference (FED) calculations due to insufficient memory (Aaditya Manjanath)
    * crashing in large RI-MP2 calculations
    * initial guess in EOM-DIP-CCSD calculations
    * crashing in large RI-CCSD calculations

- **Features and improvements in molecular dynamics:**
  - New AIMD variable (AIMD_INIT_VELOC_NANO_RANDOM) for better random seeds (Tarek Scheele)
  - Resolved issues with:
    * activating vibrational spectra computation in special cases

- **Features and improvements for large systems, QM/MM, and solvation:**
  - AIRBED: A simplified density functional theory model for physisorption on surfaces (Nick Besley, Stephen Mason)
  - Resolved issues with:
    * SM12 crashes with general basis set (Yuezhi Mao)
    * MM finite difference force calculations
    * printing of EFG principal components
    * SM12 gradient
    * implicit solvation in SCF and DFT response property calculations
    * requiring explicit derivative level to be set for IEF-PCM frequencies (John Herbert)
    * RI-MP2 + PCM jobs
    * out-of-memory error in large SMD jobs

- **Features and improvements for fragment and energy decomposition analysis methods:**
  - Enable geometry optimization on POL and VFB-CT surfaces in the presence of solvent (Yuezhi Mao)

- Enable ALMO-EDA for systems with non-*Aufbau* electronic configurations (Yuezhi Mao)
- Enable the separation of electrostatic and non-electrostatic terms in SMD solvation energy (Yuezhi Mao)
- Improve error message when attempting ROHF-based SCFMI and EDA
- Improve error message when attempting to use unsupported solvent models with SCFMI and EDA
- Control number of subspace vectors and convergence threshold in SAPT CPSCF (Kevin Carter-Fenk)
- Improved SAPT+aiDx and SAPT+MBD keywords (Kevin Carter-Fenk)
- Resolved issues with:
  - crashing during large projection-based embedding calculations (Yuezhi Mao)
  - requiring explicit derivative level to be set for adiabatic EDA geometries and frequencies (Yuezhi Mao)
  - interoperability between SAPT features and various SAPT basis sets (Kevin Carter-Fenk)
  - crashing when using SAPT(KS) + cDFT with fragment-based Hirshfeld populations (Kevin Carter-Fenk)
  - memory usage in XSAPT (Kevin Carter-Fenk, John Herbert)

1.3.7 New Features in Q-CHEM 5.3

- **Changes in default settings:**
  - Renamed rem variable ADIABATIC_CTA to VFB_CTA
  - Changed ROHF_DIAG_SPEC default from 0 to 2 for ROHF and set GEN_SCFMAN as default ROSCF engine
- **General improvements:**
  - Added support for the jun-cc-pVDZ basis set (Kevin Carter-Fenk)
- **New features and improvements in the DFT suite:**
  - TD-DFT analytic force and frequencies for meta-GGA density functionals
  - Level shifting in DIIS for better SCF convergence in difficult cases (Section 4.5)
  - M06-SX density functional (Pierpaolo Morgante, Roberto Peverati)
  - HF-3c method (Bhaskar Rana, John Herbert)
- **New features and improvements in the CC/EOM-CC package:**
  - Calculation of RIXS and orbital analysis of RIXS transition moments (Kaushik Nanda, Anna Krylov; Section 7.10.8.1)
  - New features in the CVS-EOM-CC suite (Marta Vidal, Sonia Coriani)
  - Energies and properties for EOM-DEA-CCSD (Sahil Gulania, Maxim Ivanov, Anna Krylov; Section 7.10.7)
  - Transition properties and $\langle \hat{S}^2 \rangle$ for EOM-DIP-CCSD (Sahil Gulania, Wojciech Skomorowski, Anna Krylov)
  - New NLO properties (hyperpolarizabilities) in EOM-CC (Kaushik Nanda, Anna Krylov)
  - New tools for strongly correlated and magnetic systems: Extension of FNO to open-shell references (Pavel Pokhilko, Anna Krylov; Section 7.10.13)
  - Construction of effective Hamiltonians from EOM-CC wavefunctions (Pavel Pokhilko, Anna Krylov; Section 13.6)
  - NTO analysis of spin-forbidden transitions (Pavel Pokhilko, Anna Krylov; Section 7.10.20.4)

  - Search for special points of complex PES (minima, MECP, and exceptional points) within CAP-EOM-CCSD (Zsuzsanna Koczor-Benda, Thomas Jagau)
  - Voronoi CAP and projected CAP methods (James Gayvert, Ksenia Bravaya; Section 7.10.9)
  - Two-body Dyson orbitals for computing Auger decay rates and resonance lifetimes (Wojciech Skomorowski, Anna Krylov)
  - Stability improvements in EOM-CC (Pavel Pokhilko, Anna Krylov)

- **New features and improvements in MP2 methods:**
  - Geometry optimization with regularized orbital-optimized second-order Møller-Plesset perturbation theory (κ-OOMP2) (Joonho Lee, Martin Head-Gordon; Section 6.6.6)

- **New capabilities for intermolecular interactions:**
  - Implementation of the XSAPT+MBD method (Kevin Carter-Fenk, John Herbert)

- **QM/MM improvements:**
  - L-BFGS algorithm for geometry optimization (Bhaskar Rana, John Herbert)
  - Harmonic confining potentials (Saswata Dasgupta, John Herbert)

- **New methods and capabilities:**
  - Nuclear-electronic orbital DFT and TD-DFT methods (Fabian Pavosevic, Zhen Tao, Sharon Hammes-Schiffer)
  - New module for RAS-SF methods (Shannon Houck, Nick Mayhall)
  - A family of configuration-interaction methods: non-orthogonal configuration interaction singles (NOCIS), static exchange (STEX), and one-center NOCIS (Katherine Oosterbaan, Martin Head-Gordon)
  - Integral screening and resolution-of-the-identity capabilities for complex basis functions (Thomas Jagau)
  - RI-MP2 method for complex basis functions (Mario Hernández Vera, Thomas Jagau; Section 6.6.10)
  - New method (concentric localization) for truncating the virtual space in projector-based embedding theory (Yuezhi Mao)
  - Square gradient minimization for excited-state orbital optimization (Diptarka Hait, Martin Head-Gordon)
  - Resonance Raman spectroscopy simulation (Saswata Dasgupta, John Herbert)
  - Population analysis of antibonding orbitals (Abdulrahman Aldossary)
  - Fragment-based diabatization schemes (Yuezhi Mao)
  - Enabled ghost atoms without basis functions (Bushra Alam, John Herbert)
  - Electron localization function (Bushra Alam, John Herbert
  - New input options for wavefunction analysis (Felix Plasser)

- **New features in the BrianQC GPU module:**
  - Extended support for GPU accelerated DFT exchange-correlation with support for LDA, GGA, and meta-GGA functionals
  - Partially GPU accelerated DFT frequency calculations

1.3.8 New Features in Q-CHEM 5.2

- **Changes in default settings:**
  - Single-node shared-memory parallelism becomes default and recommended for most jobs. New command line key *-mpi* is required to use distributed-memory MPI-parallel features (Section 2.2.1.1).
  - Pure basis functions are used by default with BASIS = GEN.
  - Default number of grid points in Lebedev grids in solvent models changed from 302 to 194 points (non-Hydrogen) and 110 points (Hydrogen) atoms.
  - Use of SWIG charges for SMx models.
  - Input format for XPol, SAPT and XSAPT, and MBE jobs has changed.
  - Use EDA2 as the default driver for ALMO-EDA.
  - Frozen core approximation no longer applied by default in RAS-CI calculations.
- **General improvements:**
  - Increased availability of basis sets: High angular momentum basis functions (up to k-functions) supported for most SCF, RI-MP2, CC, EOM-CC, ADC calculations.
  - Streamlined input format for RI-SCF calculations.
  - Added the def2- family of density fitted (RI) basis sets for SCF and post-SCF calculations (courtesy of Dr. Florian Weigend).
  - On-the-fly generation for the superposition of atomic densities guess for SCF (Kevin Carter-Fenk, John Herbert).
  - Reintroduction of legacy ECPs without fitting.
  - Easy specification of basis sets on fragments, reading of basis sets from an external file (Zheng Pei and Yihan Shao).
- **Improvements to the DFT capabilities:**
  - Support for analytic frequency calculations using meta-GGA density functionals (available only with shared-memory parallelism).
  - Support for analytic frequency calculations using resolution-of-the-identity (density-fitted) Coulomb (available only with shared-memory parallelism).
  - Improved performance of analytic partial Hessian calculations using DFT.
  - New density functionals: revM06, revM11 (Pierpaolo Morgante and Roberto Peverati).
- **Improvements in implicit solvation models:**
  - Revised PCM tessellation grids for improved performance (John Herbert).
  - Improved performance of the general SCF program with SMx solvation models (Yuezhi Mao).
- **New MP2 features:**
  - Addition of regularized orbital-optimized second-order Møller-Plesset perturbation theory (κ-OOMP2) (Joonho Lee, Martin Head-Gordon; Section 6.6.6).
- **Enhancements to the coupled-cluster package:**
  - Mixed-precision CCSD and EOM-CCSD (Pavel Pokhilko, Evvgeny Epifanovsky, Anna Krylov, with additional contributions from Ilya Kaliman, Kaushik Nanda, Marta Vidal, and Sonia Coriani; Sections 6.18 and 7.10.14).

  - Damped response, dynamic polarizabilities for two-electron absorption using EOM-CC (Kaushik Nanda and Anna Krylov).
  - Better handling of linear point groups in ADC and CC methods.
  - Improved performance of disk-based ADC/CC algorithms.
  - Projected and Voronoi CAP for CAP-EOM-CC/CC calculations (Ksenia Bravaya, Alexander Kunitsa; Section 7.10.9).
  - Dynamic polarizabilities for CCSD and EOM-CCSD (Kaushik Nanda, Anna Krylov; Section 7.10.20.8).
  - Improved evaluation of spin-orbit coupling constants using EOM-CC wavefunctions (Pavel Pokhilko and Anna Krylov).
  - New features for SOC calculation and analysis (Pavel Pokhilko, Anna Krylov; Section 7.10.20.4).
  - Dyson orbitals for CVS-EOM-CCSD (Marta Vidal, Sonia Coriani, Anna Krylov; Section 7.10.8).

- **Improvements in energy decomposition analysis methods:**
  - Added electron density difference (EDD) plots and the ETS-NOCV analysis (Yuezhi Mao).
  - Added support for PCM and SMD solvation models in ALMO-EDA (Yuezhi Mao).
  - Resolved several issues that caused instabilities in MP2-EDA calculations (Yuezhi Mao).
- **New capabilities for explicit solvation modeling:**
  - Polarizable Embedding (PE) Model for ground-state and ADC calculations (Maximilian Scheurer; Section 11.8).
- **Other new methods and capabilities:**
  - Incremental FCI method (Paul Zimmerman).
  - Transition potential DFT for core-valence excitations.
  - Analytic evaluation of Raman intensities (Zheng Pei and Yihan Shao).

1.3.9 New Features in Q-CHEM 5.1

- Improved OpenMP parallelization for:
  - SCF vibrational frequency calculations (Zhengting Gan)
  - RIMP2 gradient (Fazle Rob, Joonho Lee, Xintian Feng, Evgeny Epifanovsky)
- Complete active space self-consistent field (CASSCF) and adaptive sampling CI (Daniel Levine, Martin Head-Gordon)
- Tkatchenko-Scheffler van der Waals method (Section 5.7.5) and many-body dispersion method (Section 5.7.6) (Denis Barton, Ka Un Lao, & Rob DiStasio)
- Enhancements to the coupled-cluster package:
  - Core/valence separation for EOM-CCSD core-level excited and ionized states (Marta Vidal, Anna Krylov, Xintian Feng, Evgeny Epifanovsky, Sonia Coriani), Section 7.10.8.
  - NTO analysis of two-photon transitions (Kaushik Nanda, Anna Krylov), Section 7.10.20.6.
  - NTO analysis of the complex-valued EOM wave functions (Anna Krylov, Wojciech Skomorowski), Section 7.10.20.

  - Analytic gradients for Cholesky-decomposed and resolution-of-identity CCSD and EOM-CCSD (Xintian Feng, Anna Krylov).
  - Improved performance, reduced disk usage by coupled-cluster methods (Evgeny Epifanovsky, Ilya Kaliman, Xintian Feng).

- New features in NTO analysis: Energies of NTOs (Anna Krylov), Section 10.2.12.
- Finite-difference evaluation of non-linear properties (Marc de Wergifosse, Anna Krylov), Section 10.12.3.
- Poisson boundary conditions for SCF calculations (Marc Coons, John Herbert), Section 11.2.11.
  - Enables quantum chemistry calculations in an arbitrary (anisotropic and inhomogeneous) dielectric environment.
  - Nonequilibrium solvent corrections for vertical ionization energies.
- Energy decomposition analysis (EDA):
  - EDA based on symmetry-adapted perturbation theory and constrained DFT (SAPT/cDFT-EDA), Section 12.11 (Ka Un Lao, Kevin Carter-Fenk, John Herbert)
  - ALMO-EDA for CIS and TDDFT/TDA excited states, Section 12.7.5 (Qinghui Ge, Yuezhi Mao, Martin Head-Gordon)
  - Perturbative ALMO-CTA and COVP analysis in EDA2 (Yuezhi Mao, Martin Head-Gordon)
- Analytic derivative couplings for computing excitation/vibration energy couplings within the *ab initio* Frenkel-Davydov exciton model (Adrian Morrison, John Herbert), Section 12.14.
- Hyperfine and nuclear quadrupole couplings, Section 10.10.4 (Eric Berquist, Daniel Lambrecht)
- Variational two-electron reduced-density-matrix (v2RDM) and v2RDM-driven complete active space self-consistent field (v2RDM-CASSCF) method (Gergely Gidofalvi, Lauren Koulias, Wayne Mullinax, Eugene DePrince)
- Frozen and restrained potential energy scans, Section 9.6 (Yihan Shao)
- Extended ESP charge fitting procedure to the computation of RESP charges (Yihan Shao)

1.3.10 New Features in Q-CHEM 5.0

- Enhancements to the coupled-cluster package:
  - Analytic gradients for Cholesky-decomposed CCSD and EOM-CCSD; efficiency improvement for canonical CCSD and EOM-CCSD gradients (Xintian Feng, Evgeny Epifanovsky).
  - CAP-EOM-CCSD analytic gradients (Zsuzsanna Koczor-Benda, Thomas Jagau) and Dyson orbitals for metastable states (Thomas Jagau, Anna Krylov), Section 7.10.9).
  - CAP-EOM-MP2 method (Alexander Kunitsa, Ksenia Bravaya).
  - Evaluation of polarizabilities using CCSD and EOM-CCSD (EE and SF) wave functions using full derivative formulation (Kaushik Nanda and Anna Krylov), Section 7.10.20.8).
  - Evaluation of $\langle \hat{S}^2 \rangle$ for EOM-CCSD wave functions (Xintian Feng).
  - Evaluation of NACs for EOM-CCSD wave functions (Shirin Faraji, Anna Krylov, Evgeny Epifanovsky, Xintian Feng), Section 7.10.20.5).
  - Efficiency improvement and new multicore-parallel code for (T) correction (Ilya Kaliman).

  - New coupled-cluster based methods for core states (Anna Krylov).

- New capabilities for implicit solvation modeling:

  - Equilibrium PCM capabilities for computing vertical excitation, ionization, and electron attachment energies at EOM-CC and MP2 levels (Section 11.2).
  - State-specific equilibrium and nonequilibrium solvation for all orders and variants of ADC (Jan Mewes, Andreas Dreuw) , Section 7.11.10.
  - Poisson equation boundary conditions allowing use of an arbitrary, anisotropic dielectric function $\varepsilon(\mathbf{r})$, with full treatment of volume polarization (Marc Coons, John Herbert), Section 11.2.11.
  - Composite Model for Implicit Representation of Solvent (CMIRS), an accurate model for free energies of solvation (Zhi-Qiang You, John Herbert), Section 11.2.7.

- New density functionals (Narbe Mardirossian and Martin Head-Gordon), Section 5.3):

  - GGA functionals: BEEF-vdW, HLE16, KT1, KT2, KT3, rVV10
  - Meta-GGA functionals: B97M-rV, BLOC, mBEEF, oTPSS, TM
  - Hybrids: CAM-QTP(00), CAM-QTP(01), HSE-HJS, LC-ωPBE08, MN15, rCAM-B3LYP, WC04, WP04
  - Double hybrids: B2GP-PLYP, DSD-PBEB95-D3, DSD-PBEP86-D3, DSD-PBEPBE-D3, LS1DH-PBE, PBE-QIDH, PTPSS-D3, PWPB95-D3
  - Grimme's PBEh-3c "low-cost" composite method
  - rVV10 non-local correlation functional

- Additional DFT developments:

  - New forms of DFT-D3 (J. Witte; Section 5.7.3).
  - New standard integration grids, SG-2 and SG-3 (Saswata Dasgupta, John Herbert), Section 5.5.3.
  - More efficient propagator algorithms for time-dependent Kohn-Sham calculations, also known as "real-time" TDDFT (Ying Zhu, John Herbert), Section 7.4).

- New integral package for for computing effective core potential (ECP) integrals (Simon McKenzie, Evgeny Epifanovsky), Section 8.10.2).

  - More efficient analytic algorithms for energies and first derivatives.
  - Support for arbitrary projector angular momentum.
  - Support up to h angular momentum in the basis set.

- Analytic derivative couplings for the *ab initio* Frenkel-Davydov exciton model (Adrian Morrison, John Herbert); Section 12.14).

- New ALMO-based energy decomposition analysis (EDA) methods:

  - The second-generation ALMO-EDA methods for DFT (Paul Horn, Yuezhi Mao, Martin Head-Gordon); Section 12.6.
  - The extension of ALMO-EDA to RIMP2 theory (Jonathan Thirman, Martin Head-Gordon); Section 12.7.1.
  - The "adiabatic" EDA method for decomposing changes in molecular properties (Yuezhi Mao, Paul Horn, MartinHead-Gordon); Section 12.7.3.

- Wave function correlation capabilities:

- Coupled cluster valence bond (CCVB) method for describing open-shell molecules with strong spin correlations (David Small, Martin Head-Gordon); Section 6.19.3.
- Implementation of coupled-cluster valence bond with singles and doubles (CCVB-SD) for closed-shell species (Joonho Lee, David Small, Martin Head-Gordon); Section 6.14.4.

Note: Several important changes in Q-CHEM's default settings have occurred since version 4.4.

- Core electrons are now frozen by default in most post-Hartree-Fock calculations; see Section 6.2.
- The keywords for calculation of SOCs and NACs were renamed for consistency between different methods.
- Some newer density functionals now use either the SG-2 or SG-3 quadrature grid by default, whereas all functionals used SG-1 by default in v. 4.4. Table 5.3 lists the default grid for various classes of functionals.

1.3.11 New Features in Q-CHEM 4.4

- occ-RI-K algorithm for the evaluation of exact exchange in energy and force calculations (S. Manzer, F. Rob, M. Head-Gordon); Section 4.6.8.
- Combinatorially-optimized exchange-correlation functionals (N. Mardirossian, M. Head-Gordon); Section 5.3):
  - ωB97M-V (range-separated hybrid, meta-GGA functional with VV10 non-local correlation)
  - B97M-V (meta-GGA functional with VV10 non-local correlation)
  - ωB97X-V (range-separated hybrid functional with VV10 non-local correlation)
- Implementation of new exchange-correlation functionals from the literature (N. Mardirossian and M. Head-Gordon; Section 5.3). These include:
  - MGGA_MS0, MGGA_MS1, MGGA_MS2, MGGA_MS2h, MGGA_MVS, MGGA_MVSh, PKZB, revTPSS, revTPSSh, SCAN, SCAN0, PBEsol, revPBE, revPBE0
  - N12, N12-SX, GAM, MN12-L, MN12-SX, MN15-L, dlDF
  - VV10, LC-VV10
  - B97-K, B97-D3(0), B97-3, τ-HCTH, τ-HCTHh
  - SRC1-R1, SRC1-R2, SRC2-R1, SRC2-R2
  - B1LYP, B1PW91, MPW1K, LRC-BOP, BHH, BB1K, PW6B95, PWB6K, B2PLYP
- Hessian-free minimum point verification (S. Sharada, M. Head-Gordon); Section 9.3.3)
- Exciton-based excited-state models:
  - *Ab initio* Frenkel-Davydov model for coupled excitations in multi-chromophore systems (A. F. Morrison, J. M. Herbert); Section 12.14.
  - TDDFT for molecular interactions [TDDFT(MI)], a set of local excitation approximations for efficient TDDFT calculations in multi-chromophore systems and for single chromophores in the presence of explicit solvent molecules (J. Liu, J. M. Herbert); Section 12.15.
- Improvements to many-body and XSAPT methods (K. U. Lao, J. M. Herbert):
  - MPI-parallelized many-body expansion with analytic gradient (Section 12.13).

  - Efficient atomic orbital implementation of XSAPT for both closed- and open-shell systems (Section 12.10.3).

- Thermostats for *ab initio* molecular dynamics (R. P. Steele, J. M. Herbert).
- Analytic energy gradient for the Ewald summation in QM/MM calculations (Z. C. Holden, J. M. Herbert).
- Zeolite QM/MM methods (J. Gomes, M. Head-Gordon).
- EOM-MP2 methods for excitation, ionization and electron attachment energies (A. Kunitsa, K. Bravaya); Section 7.10.15.1.
- Evaluation of polarizabilities using CCSD and EOM-CCSD wave functions (K. Nanda, A. I. Krylov); Section 7.10.20.8.
- Distributed-memory parallel implementation of CC and EOM-CC methods and performance improvements in disk-based algorithms (E. Epifanovsky, I. Kaliman, A. I. Krylov).
- Improvements to the maximum overlap method (MOM) for SCF calculations (A. T. B. Gilbert); Section 7.6.
- Non-equilibrium PCM method to describe solvent effects in ADC excited-state calculations (J.-M. Mewes, A. Dreuw); Section 7.11.10.
- Spin-flip ADC method (D. Lefrancois, A. Dreuw); Section 7.11.7.

1.3.12 New Features in Q-CHEM 4.3

- Analytic derivative couplings (*i.e.*, nonadiabatic couplings) between electronic states computed at the CIS, spin-flip CIS, TDDFT, and spin-flip TDDFT levels (S. Fatehi, Q. Ou, J. E. Subotnik, X. Zhang, J. M. Herbert); Section 9.8.
- A third-generation ("+D3") dispersion potential for XSAPT (K. U. Lao, J. M. Herbert); Section 12.10.
- Non-equilibrium PCM for computing vertical excitation energies (at the TDDFT level) and ionization energies in solution (Z.-Q. You, J. M. Herbert); Section 11.2.3.3.
- Spin-orbit couplings between electronic states for CC and EOM-CC wave functions (E. Epifanovsky, J. Gauss, A. I. Krylov); Section 7.10.20.4.
- PARI-K method for evaluation of exact exchange, which affords dramatic speed-ups for triple-ζ and larger basis sets in hybrid DFT calculations (S. Manzer, M. Head-Gordon).
- Transition moments and cross sections for two-photon absorption using EOM-CC wave functions (K. Nanda, A. I. Krylov); Section 7.10.20.6.
- New excited-state analysis for ADC and CC/EOM-CC methods (M. Wormit); Section 10.2.12).
- New Dyson orbital code for EOM-IP-CCSD and EOM-EA-CCSD (A. Gunina and A. I. Krylov; Section 7.10.28).
- Transition moments, state dipole moments, and Dyson orbitals for CAP-EOM-CCSD (T.-C. Jagau and A. I. Krylov; Sections 7.10.9 and 7.10.28).
- TAO-DFT: Thermally-assisted-occupation density functional theory (J.-D. Chai; Section 5.12.3).
- MP2[V], a dual basis method that approximates the MP2 energy (J. Deng and A. Gilbert).
- Iterative Hirshfeld population analysis for charged systems, and CM5 semi-empirical charge scheme (K. U. Lao and J. M. Herbert; Section 10.2.2).

- New DFT functionals: (Section 5.3):
  - Long-range corrected functionals with empirical dispersion-: ωM05-D, ωB97X-D3 and ωM06-D3 (Y.-S. Lin, K. Hui, and J.-D. Chai.
  - PBE0_DH and PBE0_2 double-hybrid functionals (K. Hui and J.-D. Chai; Section 5.9).
  - AK13 (K. Hui and J.-D. Chai).
  - LFAs asymptotic correction scheme (P.-T. Fang and J.-D. Chai).
- LDA/GGA fundamental gap using a frozen-orbital approximation (K. Hui and J.-D. Chai; Section 5.12.2).

1.3.13 New Features in Q-CHEM 4.2

- Input file changes:
  - New keyword METHOD simplifies input in most cases by replacing the pair of keywords EXCHANGE and CORRELATION (see Chapter 4).
  - Keywords for requesting excited-state calculations have been modified and simplified (see Chapter 7 for details).
  - Keywords for solvation models have been modified and simplified (see Section 11.2 for details).
- New features for NMR calculations including spin-spin couplings (J. Kussmann, A. Luenser, and C. Ochsenfeld; Section 10.10.2).
- New built-in basis sets (see Chapter 8).
- New features and performance improvements in EOM-CC:
  - EOM-CC methods extended to treat meta-stable electronic states (resonances) via complex scaling and complex absorbing potentials (D. Zuev, T.-C. Jagau, Y. Shao, and A. I. Krylov; Section 7.10.9).
  - New features added to EOM-CC iterative solvers, such as methods for interior eigenvalues and user-specified guesses (D. Zuev; Section 7.10.16)
  - Multi-threaded parallel code for (EOM-)CC gradients and improved CCSD(T) performance.
- New features and performance improvements in ADC methods (M. Wormit, A. Dreuw):
  - RI-ADC can tackle much larger systems at reduced cost (Section 7.11.4).
  - SOS-ADC methods (Section 7.11.5).
  - State-to-state properties for ADC (Section 7.11.9).
- SM12 implicit solvation model (A. V. Marenich, D. G. Truhlar, and Y. Shao; Section 11.2.9.1).
- Interface to NBO v. 6 (Section 10.3.1).
- Optimization of MECPs between electronic states at the SOS-CIS(D) and TDDFT levels (X. Zhang and J. M. Herbert; Section 9.8.3).
- ROKS method for ΔSCF calculations of excited states (T. Kowalczyk and T. Van Voorhis; Section 7.8.3).
- Fragment-based initial guess for SCF methods (Section 12.3).
- Pseudo-fractional occupation number method for improved SCF convergence in small-gap systems (D. S. Lambrecht; Section 4.5.6).

- Density embedding scheme (B. J. Albrecht, E. Berquist, and D. S. Lambrecht; Section 11.6).
- New features and enhancements in fragment-based many-body expansion methods (K. U. Lao and J. M. Herbert):
  - XSAPT(KS)+D: A dispersion corrected version of symmetry-adapted perturbation theory for fast and accurate calculation of interaction energies in non-covalent clusters (Section 12.10).
  - Many-body expansion and fragment molecular orbital (FMO) methods for clusters (Section 12.13).
- Periodic boundary conditions with proper Ewald summation, for energies only (Z. C. Holden and J. M. Herbert; Section 11.3).

1.3.14 New Features in Q-CHEM 4.1

- Fundamental algorithms:
  - Improved parallel performance at all levels including new OpenMP capabilities for Hartree-Fock, DFT, MP2, and coupled cluster theory (Z. Gan, E. Epifanovsky, M. Goldey, and Y. Shao; Section 2.2.1.1).
  - Significantly enhanced ECP capabilities, including gradients and frequencies in all basis sets for which the energy can be evaluated (Y. Shao and M. Head-Gordon; Section 8.10.7).
- SCF and DFT capabilities:
  - TDDFT energy with the M06, M08, and M11 series of functionals.
  - XYGJ-OS analytical energy gradient.
  - TDDFT/C-PCM excitation energies, gradient, and Hessian (J. Liu and W. Liang; Section 7.3.6).
  - Additional features in the maximum overlap method (MOM) approach for converging difficult SCF calculations (N. A. Besley; Section 4.5.14).
- Wave function correlation capabilities:
  - RI and Cholesky decomposition implementation of all CC and EOM-CC methods enabling applications to larger systems with reduced disk and memory requirements and improved performance (E. Epifanovsky, X. Feng, D. Zuev, Y. Shao, and A. I. Krylov; Sections 6.12.9 and 6.12.10).
  - Attenuated MP2 theory in the aug-cc-pVDZ and aug-cc-pVTZ basis sets, which truncates two-electron integrals to cancel basis set superposition error, yielding results for intermolecular interactions that are much more accurate than standard MP2 in the same basis set (M. Goldey and M. Head-Gordon; Section 6.7).
  - Extended RAS-nSF methodology for ground and excited states involving strong non-dynamical correlation (P. M. Zimmerman, D. Casanova, and M. Head-Gordon; Section 7.12).
  - Coupled cluster valence bond (CCVB) method for describing molecules with strong spin correlations (D. W. Small and M. Head-Gordon; Section 6.19.3).
- Searching and scanning potential energy surfaces:
  - Potential energy surface scans (Y. Shao; Section 9.6).
  - Improvements in automatic transition structure searching via the "freezing string" method, including the ability to perform such calculations without a Hessian calculation (S. M. Sharada and M. Head-Gordon; Section 9.3.3).
  - Enhancements to partial Hessian vibrational analysis (N. A. Besley; Section 10.7.4).

- Calculating and characterizing inter- and intramolecular interactions
  - Extension of EFP to macromolecules: fEFP approach (A. Laurent, D. Ghosh, A. I. Krylov, and L. V. Slipchenko; Section 11.5.5).
  - Symmetry-adapted perturbation theory level at the "SAPT0" level, for intermolecular interaction energy decomposition analysis into physically-meaningful components such as electrostatics, induction, dispersion, and exchange. An RI version is also available (L. D. Jacobson, J. M. Herbert; Section 12.9).
  - The "explicit polarization" (XPol) monomer-based SCF calculations to compute many-body polarization effects in linear-scaling time via charge embedding (Section 12.8), which can be combined either with empirical potentials (*e.g.*, Lennard-Jones) for the non-polarization parts of the intermolecular interactions, or better yet, with SAPT for an *ab initio* approach called XSAPT that extends SAPT to systems containing more that two monomers (L. D. Jacobson and J. M. Herbert; Section 12.10).
  - Extension of the absolutely-localized molecular orbital (ALMO)-based energy decomposition analysis to unrestricted cases (P. R. Horn and M. Head-Gordon; Section 12.5.3).
  - Calculation of the populations of "effectively unpaired electrons" in low-spin state using DFT, a new method of evaluating localized atomic magnetic moments within Kohn-Sham without symmetry breaking, and Mayer-type bond order analysis with inclusion of static correlation effects (E. I. Proynov; Section 10.2.4).
- Quantum transport calculations including electron transmission functions and electron tunneling currents under applied bias voltage (B. D. Dunietz and N. Sergueev; Section 13.4).
- Searchable online version of the Q-CHEM PDF manual (J. M. Herbert and E. Epifanovsky).

1.3.15 New Features in Q-CHEM 4.0.1

- Remote submission capability in IQMOL (A. T. B. Gilbert).
- Scaled nuclear charge and charge-cage stabilization capabilities (T. Kús and A. I. Krylov; Section 7.10.12).
- Calculations of excited state properties including transition dipole moments between different excited states in CIS and TDDFT as well as couplings for electron and energy transfer (Z.-Q. You and C.-P. Hsu; Section 10.14).

1.3.16 New Features in Q-CHEM 4.0

- New exchange-correlation functionals (Section 5.3):
  - Density-functional dispersion using Becke and Johnson's XDM model in an efficient, analytic form (Z. Gan, E. I. Proynov, and J. Kong; Section 5.7.4).
  - Van der Waals density functionals vdW-DF-04 and vdW-DF-10 of Langreth and coworkers (O. Vydrov; Section 5.7.2).
  - VV09 and VV10, new analytic dispersion functionals (O. Vydrov, T. Van Voorhis; Section 5.7.2)
  - DFT-D3 empirical dispersion methods for non-covalent interactions (S.-P. Mao and J.-D. Chai; Section 5.7.3).
  - ωB97X-2, a double-hybrid functional based on the long-range corrected B97 functional, with improved accounting for medium- and long-range interactions (J.-D. Chai and M. Head-Gordon; Section 5.9).
  - XYGJ-OS, a double-hybrid functional for predictions of non-bonded interactions and thermochemistry at nearly chemical accuracy (X. Xu, W. A. Goddard, and Y. Jung; Section 5.9).

  - Short-range corrected functional for calculation of near-edge X-ray absorption spectra (N. A. Besley; Section 7.13.2).
  - LB94 asymptotically-corrected exchange-correlation functional for TDDFT (Y.-C. Su and J.-D. Chai; Section 5.10.2).
  - Non-dynamical correlation in DFT with an efficient RI implementation of the Becke05 model in a fully analytic formulation (E. I. Proynov, Y. Shao, F. Liu, and J. Kong; Section 5.3).
  - TPSS and its hybrid version TPSSh, and rPW86 (F. Liu and O. Vydrov).
  - Double-hybrid functional B2PLYP-D (J.-D. Chai).
  - Hyper-GGA functional MCY2 from Mori-Sánchez, Cohen, and Yang (F. Liu).
  - SOGGA, SOGGA11 and SOGGA11-X family of GGA functionals (R. Peverati, Y. Zhao, and D. G. Truhlar).
  - M08-HX and M08-SO suites of high HF exchange meta-GGA functionals (Y. Zhao and D. G. Truhlar).
  - M11-L and M11 suites of meta-GGA functionals (R. Peverati, Y. Zhao, D. G. Truhlar).

- Improved DFT algorithms:

  - Multi-resolution exchange-correlation (mrXC) for fast calculation of grid-based XC quadrature (S. T. Brown, C.-M. Chang, and J. Kong; Section 5.5.5).
  - Efficient computation of the XC part of the dual basis DFT (Z. Gan and J. Kong; Section 4.4.5).
  - Fast DFT calculation with "triple jumps" between different sizes of basis set and grid, and different levels of functional (J. Deng, A. T. B. Gilbert, and P. M. W. Gill; Section 4.8).
  - Faster DFT and HF calculation with an atomic resolution-of-identity algorithm (A. Sodt and M. Head-Gordon; Section 4.6.6).

- Post-Hartree–Fock methods:

  - Significantly enhanced coupled-cluster code rewritten for better performance on multi-core architectures, including energy and gradient calculations with CCSD and energy calculations with EOM-EE/SF/IP/EA-CCSD, and CCSD(T) energy calculations (E. Epifanovsky, M. Wormit, T. Kús, A. Landau, D. Zuev, K. Khistyaev, I. Kaliman, A. I. Krylov, and A. Dreuw; Chaps. 6 and 7).
  - Fast and accurate coupled-cluster calculations with frozen natural orbitals (A. Landau, D. Zuev, and A. I. Krylov; Section 6.15.1).
  - Correlated excited states with the perturbation-theory based, size-consistent ADC scheme (M. Wormit and A. Dreuw; Section 7.11).
  - Restricted active space, spin-flip method for multi-configurational ground states and multi-electron excited states (P. M. Zimmerman, F. Bell, D. Casanova, and M. Head-Gordon; Section 7.2.3.2).

- Post-Hartree–Fock methods for describing strong correlation:

  - "Perfect quadruples" and "perfect hextuples" methods for strong correlation problems (J. A. Parkhill and M. Head-Gordon; Section 6.14.5).
  - Coupled-cluster valence bond (CCVB) methods for multiple-bond breaking (D. W. Small, K. V. Lawler, and M. Head-Gordon; Section 6.19).

- TDDFT for excited states:

  - Nuclear gradients for TDDFT (Z. Gan, C.-P. Hsu, A. Dreuw, M. Head-Gordon, and J. Kong; Section 7.3.1).

  - Direct coupling of charged states for study of charge transfer reactions (Z.-Q. You and C.-P. Hsu; Section 10.14.2).
  - Analytical excited-state Hessian for TDDFT within the Tamm-Dancoff approximation (J. Liu and W. Liang; Section 7.3.7).
  - Self-consistent excited-states with the maximum overlap method (A. T. B. Gilbert, N. A. Besley, and P. M. W. Gill; Section 7.6).
  - Calculation of reactions via configuration interactions of charge-constrained states computed with constrained DFT (Q. Wu, B. Kaduk and T. Van Voorhis; Section 5.11).
  - Overlap analysis of the charge transfer in a TDDFT excited state (N. A. Besley; Section 7.3.3).
  - Localizing diabatic states with Boys or Edmiston-Ruedenberg localization, for charge or energy transfer (J. E Subotnik, R. P. Steele, N. Shenvi, and A. Sodt; Section 10.14.1.3).
  - Non-collinear formalism for spin-flip TDDFT (Y. Shao, Y. A. Bernard, and A. I. Krylov; Section 7.3)

- Solvation and condensed-phase modeling

  - Smooth free energy surface for solvated molecules via SWIG-PCMs, for QM and QM/MM calculations, including a linear-scaling QM/MM/PCM algorithm (A. W. Lange and J. M. Herbert; Sections 11.2.3 and 11.2.5).
  - Klamt's COSMO solvation model with DFT energy and gradient (Y. Shao; Section 11.2.8).
  - Polarizable explicit solvent via EFP, for ground- and excited-state calculations at the DFT/TDDFT and CCSD/EOM-CCSD levels, as well as CIS and CIS(D). A library of effective fragments for common solvents is also available, along with energy and gradient for EFP–EFP calculations (V. Vanovschi, D. Ghosh, I. Kaliman, D. Kosenkov, C. F. Williams, J. M. Herbert, M. S. Gordon, M. W. Schmidt, Y. Shao, L. V. Slipchenko, and A. I. Krylov; Section 11.5).

- Optimizations, vibrations, and dynamics:

  - "Freezing" and "growing" string methods for efficient automated reaction-path finding (A. Behn, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon; Section 9.3.2).
  - Improved robustness of the intrinsic reaction coordinate (IRC)-following code (M. Head-Gordon).
  - Quantum-mechanical treatment of nuclear motion at equilibrium via path integrals (R. P. Steele; Section 9.10).
  - Calculation of local vibrational modes of interest with partial Hessian vibrational analysis (N. A. Besley; Section 10.7.4).
  - Accelerated *ab initio* molecular dynamics MP2 and/or dual-basis methods, based on Z-vector extrapolation (R. P. Steele; Section 4.7.3).
  - Quasi-classical *ab initio* molecular dynamics (D. S. Lambrecht and M. Head-Gordon; Section 9.9.6).

- Fragment-based methods:

  - Symmetry-adapted perturbation theory (SAPT) for computing and analyzing dimer interaction energies (L. D. Jacobson, M. A. Rohrdanz, and J. M. Herbert; Section 12.9).
  - Many-body generalization of SAPT ("XSAPT"), with empirical dispersion corrections for high accuracy and low cost in large clusters (L. D. Jacobson, K. U. Lao, and J. M. Herbert; Section 12.10).
  - Methods based on a truncated many-body expansion, including the fragment molecular orbital (FMO) method (K. U. Lao and J. M. Herbert; Section 12.13).

- Properties and wave function analysis:
  - Analysis of metal oxidation states via localized orbital bonding analysis (A. J. W. Thom, E. J. Sundstrom, and M. Head-Gordon; Section 10.2.6).
  - Hirshfeld population analysis (S. Yeganeh; Section 10.2.2).
  - Visualization of non-covalent bonding using Johnson and Yang's NCI algorithm (Y. Shao; Section 10.5.6).
  - Electrostatic potential on a grid for transition densities (Y. Shao; Section 10.5.8).
- Support for modern computing platforms
  - Efficient multi-threaded parallel performance for CC, EOM, and ADC methods.
  - Better performance for multi-core systems with shared-memory parallel DFT and Hartree-Fock (Z. Gan, Y. Shao, and J. Kong) and RI-MP2 (M. Goldey and M. Head-Gordon; Section 6.17).
  - Accelerated RI-MP2 calculation on GPUs (R. Olivares-Amaya, M. Watson, R. Edgar, L. Vogt, Y. Shao, and A. Aspuru-Guzik; Section 6.6.4).
- Graphical user interfaces (GUIs):
  - Input file generation, Q-CHEM job submission, and visualization is supported by IQMOL, a fully integrated GUI developed by Andrew Gilbert. IQMOL is a free software and does not require purchasing a Q-CHEM license. See http://www.iqmol.org for details and installation instructions.
  - Other graphical interfaces are also available, including MOLDEN, MACMOLPLT, and AVOGADRO (Chapter 10 and elsewhere).

1.3.17 Summary of Features in Q-CHEM Versions 3.*x*

- DFT functionals and algorithms:
  - Long-ranged corrected (LRC) functionals, also known as range-separated hybrid functionals (M. A. Rohrdanz and J. M. Herbert)
  - Constrained DFT (Q. Wu and T. Van Voorhis)
  - Grimme's "DFT-D" empirical dispersion corrections (C.-D. Sherrill)
  - "Incremental" DFT method that significantly accelerates exchange-correlation quadrature in later SCF cycles (S. T. Brown)
  - Efficient SG-0 quadrature grid with approximately half the number of grid points relative to SG-1 (S.-H. Chien)
- Solvation models:
  - SM8 model (A. V. Marenich, R. M. Olson, C. P. Kelly, C. J. Cramer, and D. G. Truhlar)
  - Kirkwood-Onsager reaction-field model (C.-L. Cheng, T. Van Voorhis, K. Thanthiriwatte, and S. R. Gwaltney)
  - Chipman's SS(V)PE model (S. T. Brown)
- Second-order perturbation theory algorithms for ground and excited states:
  - Dual-basis RIMP2 energy and analytical gradient (R. P. Steele, R. A. DiStasio Jr., and M. Head-Gordon)
  - O2 energy and gradient (R. C. Lochan and M. Head-Gordon)

  - SOS-CIS(D), SOS-CIS(D_0), and RI-CIS(D) for excited states (D. Casanova, Y. M. Rhee, and M. Head-Gordon)
  - Efficient resolution-of-identity (RI) implementations of MP2 and SOS-MP2 (including both energies and gradients), and of RI-TRIM and RI-CIS(D) energies (Y. Jung, R. A. DiStasio, Jr., R. C. Lochan, and Y. M. Rhee)

- Coupled-cluster methods (P. A. Pieniazek, E. Epifanovsky, A. I. Krylov):
  - IP-CISD and EOM-IP-CCSD energy and gradient
  - Multi-threaded (OpenMP) parallel coupled-cluster calculations
  - Potential energy surface crossing minimization with CCSD and EOM-CCSD methods (E. Epifanovsky)
  - Dyson orbitals for ionization from the ground and excited states within CCSD and EOM-CCSD methods (M. Oana)

- QM/MM methods (H. L. Woodcock, A. Ghysels, Y. Shao, J. Kong, and H. B. Brooks)
  - Q-CHEM/CHARMM interface (H. L. Woodcock)
  - Full QM/MM Hessian evaluation and approximate mobile-block-Hessian evaluation
  - Two-layer ONIOM model (Y. Shao).
  - Integration with the MOLARIS simulation package (E. Rosta).

- Improved two-electron integrals package
  - Rewrite of the Head-Gordon–Pople algorithm for modern computer architectures (Y. Shao)
  - Fourier Transform Coulomb method for linear-scaling construction of the Coulomb matrix, even for basis sets with high angular moment and diffuse functions (L. Fusti-Molnar)

- Dual basis self-consistent field calculations, offering an order-of-magnitude reduction in the cost of large-basis DFT calculations (J. Kong and R. P. Steele)

- Enhancements to the correlation package including:
  - Most extensive range of EOM-CCSD methods available including EOM-SF-CCSD, EOM-EE-CCSD, EOM-DIP-CCSD, EOM-IP/EA-CCSD (A. I. Krylov).
  - Available for RHF, UHF, and ROHF references.
  - Analytic gradients and properties calculations (permanent and transition dipoles *etc.*.).
  - Full use of Abelian point-group symmetry.

- Coupled-cluster perfect-paring methods applicable to systems with > 100 active electrons (M. Head-Gordon)

- Transition structure search using the "growing string" algorithm (A. Heyden and B. Peters):

- *Ab initio* molecular dynamics (J. M. Herbert)

- Linear scaling properties for large systems (J. Kussmann, C. Ochsenfeld):
  - NMR chemical shifts
  - Static and dynamic polarizabilities
  - Static hyper-polarizabilities, optical rectification, and electro-optical Pockels effect

- Anharmonic frequencies (C. Y. Lin)

- Wave function analysis tools:
  - Analysis of intermolecular interactions with ALMO-EDA (R. Z. Khaliullin and M. Head-Gordon)
  - Electron transfer analysis (Z.-Q. You and C.-P. Hsu)
  - Spin densities at the nuclei (V. A. Rassolov)
  - Position, momentum, and Wigner intracules (N. A. Besley and D. P. O'Neill)
- Graphical user interface (GUI) options:
  - IQMOL, a fully integrated GUI. IQMOL includes input file generator and contextual help, molecular builder, job submission tool, and visualization kit (molecular orbital and density viewer, frequencies, etc). For the latest version and download/installation instructions, please see the IQMOL homepage (http://www.iqmol.org).
  - Seamless integration with the SPARTAN package (see www.wavefun.com).
  - Support for several other public-domain visualization programs:
    * WEBMO
      https://www.webmo.net
    * AVOGADRO
      https://avogadro.cc
    * MOLDEN
      http://www.cmbi.ru.nl/molden
    * MACMOLPLT (via a MOLDEN-formatted input file)
      https://brettbode.github.io/wxmacmolplt
    * JMOL
      http://jmol.sourceforge.net/

1.3.18 Summary of Features Prior to Q-CHEM 3.0

- Efficient algorithms for large-molecule density functional calculations:
  - CFMM for linear scaling Coulomb interactions (energies and gradients) (C. A. White).
  - Second-generation J-engine and J-force engine (Y. Shao).
  - LinK for exchange energies and forces (C. Ochsenfeld and C. A. White).
  - Linear scaling DFT exchange-correlation quadrature.
- Local, gradient-corrected, and hybrid DFT functionals:
  - Slater, Becke, GGA91 and Gill '96 exchange functionals.
  - VWN, PZ81, Wigner, Perdew86, LYP and GGA91 correlation functionals.
  - EDF1 exchange-correlation functional (R. Adamson).
  - B3LYP, B3P and user-definable hybrid functionals.
  - Analytical gradients and analytical frequencies.
  - SG-0 standard quadrature grid (S.-H. Chien).
  - Lebedev grids up to 5294 points (S. T. Brown).
- High level wave function-based electron correlation methods

  - Efficient semi-direct MP2 energies and gradients.
  - MP3, MP4, QCISD, CCSD energies.
  - OD and QCCD energies and analytical gradients.
  - Triples corrections (QCISD(T), CCSD(T) and OD(T) energies).
  - CCSD(2) and OD(2) energies.
  - Active space coupled cluster methods: VOD, VQCCD, VOD(2).
  - Local second order Møller-Plesset (MP2) methods (DIM and TRIM).
  - Improved definitions of core electrons for post-HF correlation (V. A. Rassolov).

- Extensive excited state capabilities:
  - CIS energies, analytical gradients and analytical frequencies.
  - CIS(D) energies.
  - Time-dependent density functional theory energies (TDDFT).
  - Coupled cluster excited state energies, OD and VOD (A. I. Krylov).
  - Coupled-cluster excited-state geometry optimizations.
  - Coupled-cluster property calculations (dipoles, transition dipoles).
  - Spin-flip calculations for CCSD and TDDFT excited states (A. I. Krylov and Y. Shao).

- High performance geometry and transition structure optimization (J. Baker):
  - Optimizes in Cartesian, *Z*-matrix or delocalized internal coordinates.
  - Impose bond angle, dihedral angle (torsion) or out-of-plane bend constraints.
  - Freezes atoms in Cartesian coordinates.
  - Constraints do not need to be satisfied in the starting structure.
  - Geometry optimization in the presence of fixed point charges.
  - Intrinsic reaction coordinate (IRC) following code.

- Evaluation and visualization of molecular properties
  - Kirkwood-Onsager, SS(V)PE, and Langevin dipoles solvation models.
  - Evaluate densities, electrostatic potentials, orbitals over cubes for plotting.
  - Natural Bond Orbital (NBO) analysis.
  - Attachment/detachment densities for excited states via CIS, TDDFT.
  - Vibrational analysis after evaluation of the nuclear coordinate Hessian.
  - Isotopic substitution for frequency calculations (R. Doerksen).
  - NMR chemical shifts (J. Kussmann).
  - Atoms in Molecules (AIMPAC) support (J. Ritchie).
  - Stability analysis of SCF wave functions (Y. Shao).
  - Calculation of position and momentum molecular intracules A. Lee, N. A. Besley, and D. P. O'Neill).

- Flexible basis set and effective core potential (ECP) functionality: (Ross Adamson and Peter Gill)
  - Wide range of built-in basis sets and ECPs.
  - Basis set superposition error correction.

- Support for mixed and user-defined basis sets.
- Effective core potentials for energies and gradients.
- Highly efficient PRISM-based algorithms to evaluate ECP matrix elements.
- Faster and more accurate ECP second derivatives for frequencies.

1.4 Citing Q-CHEM

Users who publish papers based on Q-CHEM calculations are asked to cite the official peer-reviewed literature citation for the software. For versions corresponding to 5.0 and later, this is:

> Evgeny Epifanovsky, Andrew T. B. Gilbert, Xintian Feng, Joonho Lee, Yuezhi Mao, Narbe Mardirossian, Pavel Pokhilko, Alec F. White, Marc P. Coons, Adrian L. Dempwolff, Zhengting Gan, Diptarka Hait, Paul R. Horn, Leif D. Jacobson, Ilya Kaliman, Jörg Kussmann, Adrian W. Lange, Ka Un Lao, Daniel S. Levine, Jie Liu, Simon C. McKenzie, Adrian F. Morrison, Kaushik D. Nanda, Felix Plasser, Dirk R. Rehn, Marta L. Vidal, Zhi-Qiang You, Ying Zhu, Bushra Alam, Benjamin J. Albrecht, Abdulrahman Aldossary, Ethan Alguire, Josefine H. Andersen, Vishikh Athavale, Dennis Barton, Khadiza Begam, Andrew Behn, Nicole Bellonzi, Yves A. Bernard, Eric J. Berquist, Hugh G. A. Burton, Abel Carreras, Kevin Carter-Fenk, Romit Chakraborty, Alan D. Chien, Kristina D. Closser, Vale Cofer-Shabica, Saswata Dasgupta, Marc de Wergifosse, Jia Deng, Michael Diedenhofen, Hainam Do, Sebastian Ehlert, Po-Tung Fang, Shervin Fatehi, Qingguo Feng, Triet Friedhoff, James Gayvert, Qinghui Ge, Gergely Gidofalvi, Matthew Goldey, Joe Gomes, Cristina E. González-Espinoza, Sahil Gulania, Anastasia O. Gunina, Magnus W. D. Hanson-Heine, Phillip H. P. Harbach, Andreas Hauser, Michael F. Herbst, Mario Hernández Vera, Manuel Hodecker, Zachary C. Holden, Shannon Houck, Xunkun Huang, Kerwin Hui, Bang C. Huynh, Maxim Ivanov, Ádám Jász, Hyunjun Ji, Hanjie Jiang, Benjamin Kaduk, Sven Kähler, Kirill Khistyaev, Jaehoon Kim, Gergely Kis, Phil Klunzinger, Zsuzsanna Koczor-Benda, Joong Hoon Koh, Dimitri Kosenkov, Laura Koulias, Tim Kowalczyk, Caroline M. Krauter, Karl Kue, Alexander Kunitsa, Thomas Kus, István Ladjánszki, Arie Landau, Keith V. Lawler, Daniel Lefrancois, Susi Lehtola, Run R. Li, Yi-Pei Li, Jiashu Liang, Marcus Liebenthal, Hung-Hsuan Lin, You-Sheng Lin, Fenglai Liu, Kuan-Yu Liu, Matthias Loipersberger, Arne Luenser, Aaditya Manjanath, Prashant Manohar, Erum Mansoor, Sam F. Manzer, Shan-Ping Mao, Aleksandr V. Marenich, Thomas Markovich, Stephen Mason, Simon A. Maurer, Peter F. McLaughlin, Maximilian F. S. J. Menger, Jan-Michael Mewes, Stefanie A. Mewes, Pierpaolo Morgante, J. Wayne Mullinax, Katherine J. Oosterbaan, Garrette Paran, Alexander C. Paul, Suranjan K. Paul, Fabijan Pavošević, Zheng Pei, Stefan Prager, Emil I. Proynov, Ádám Rák, Eloy Ramos-Cordoba, Bhaskar Rana, Alan E. Rask, Adam Rettig, Ryan M. Richard, Fazle Rob, Elliot Rossomme, Tarek Scheele, Maximilian Scheurer, Matthias Schneider, Nickolai Sergueev, Shaama M. Sharada, Wojciech Skomorowski, David W. Small, Christopher J. Stein, Yu-Chuan Su, Eric J. Sundstrom, Zhen Tao, Jonathan Thirman, Gábor J. Tornai, Takashi Tsuchimochi, Norm M. Tubman, Srimukh Prasad Veccham, Oleg Vydrov, Jan Wenzel, Jon Witte, Atsushi Yamada, Kun Yao, Sina Yeganeh, Shane R. Yost, Alexander Zech, Igor Ying Zhang, Xing Zhang, Yu Zhang, Dmitry Zuev, Alán Aspuru-Guzik, Alexis T. Bell, Nicholas A. Besley, Ksenia B. Bravaya, Bernard R. Brooks, David Casanova, Jeng-Da Chai, Sonia Coriani, Christopher J. Cramer, György Cserey, A. Eugene DePrince III, Robert A. DiStasio Jr., Andreas Dreuw, Barry D. Dunietz, Thomas R. Furlani, William A. Goddard III, Sharon Hammes-Schiffer, Teresa Head-Gordon, Warren J. Hehre, Chao-Ping Hsu, Thomas-C. Jagau, Yousung Jung, Andreas Klamt, Jing Kong, Daniel S. Lambrecht, WanZhen Liang, Nicholas J. Mayhall, C. William McCurdy, Jeffrey B. Neaton, Christian Ochsenfeld, John A. Parkhill, Roberto Peverati, Vitaly A. Rassolov, Yihan Shao, Lyudmila V. Slipchenko, Tim Stauch, Ryan P. Steele, Joseph E. Subotnik, Alex J. W. Thom, Alexandre Tkatchenko, Donald G. Truhlar, Troy Van Voorhis, Tomasz A. Wesolowski, K. Birgitta Whaley, H. Lee Woodcock III, Paul M. Zimmerman, Shirin Faraji, Peter M. W. Gill, Martin Head-Gordon, John M. Herbert, and Anna I. Krylov. *Software for the frontiers of quantum chemistry: An overview of developments in the Q-Chem 5 package.* [*J. Chem. Phys.*. **155**, 084801 (2021)]

Literature citations for Q-CHEM v. 2.0[1], v. 3.0[3], and v. 4.0[4] are also available, and the most current list of Q-CHEM authors can always be found on the website, www.q-chem.com. The primary literature is extensively referenced throughout this manual, and users are urged to cite the original literature for particular theoretical methods. This is how our large community of academic developers gets credit for its effort.

References and Further Reading

[1] J. Kong, C. A. White, A. I. Krylov, D. Sherrill, R. D. Adamson, T. R. Furlani, M. S. Lee, A. M. Lee, S. R. Gwaltney, T. R. Adams, C. Ochsenfeld, A. T. B. Gilbert, G. S. Kedziora, V. A. Rassolov, D. R. Maurice, N. Nair, Y. Shao, N. A. Besley, P. E. Maslen, J. P. Dombroski, H. Daschel, W. Zhang, P. P. Korambath, J. Baker, E. F. C. Byrd, T. Van Voorhis, M. Oumi, S. Hirata, C.-P. Hsu, N. Ishikawa, J. Florian, A. Warshel, B. G. Johnson, P. M. W. Gill, M. Head-Gordon, and J. A. Pople. *J. Comput. Chem.*, 21:1532, 2000. DOI: 10.1002/1096-987X(200012)21:16<1532::AID-JCC10>3.0.CO;2-W.

[2] A. I. Krylov and P. M. W. Gill. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 3:317, 2013. DOI: 10.1002/wcms.1122.

[3] Y. Shao, L. Fusti-Molnar, Y. Jung, J. Kussmann, C. Ochsenfeld, S. T. Brown, A. T. B. Gilbert, L. V. Slipchenko, S. V. Levchenko, D. P. O'Neill, R. A. DiStasio Jr., R. C. Lochan, T. Wang, G. J. O. Beran, N. A. Besley, J. M. Herbert, C. Y. Lin, T. Van Voorhis, S. H. Chien, A. Sodt, R. P. Steele, V. A. Rassolov, P. E. Maslen, P. P. Korambath, R. D. Adamson, B. Austin, J. Baker, E. F. C. Byrd, H. Dachsel, R. J. Doerksen, A. Dreuw, B. D. Dunietz, A. D. Dutoi, T. R. Furlani, S. R. Gwaltney, A. Heyden, S. Hirata, C.-P. Hsu, G. Kedziora, R. Z. Khalliulin, P. Klunzinger, A. M. Lee, M. S. Lee, W. Liang, I. Lotan, N. Nair, B. Peters, E. I. Proynov, P. A. Pieniazek, Y. M. Rhee, J. Ritchie, E. Rosta, C. D. Sherrill, A. C. Simmonett, J. E. Subotnik, H. L. Woodcock III, W. Zhang, A. T. Bell, A. K. Chakraborty, D. M. Chipman, F. J. Keil, A. Warshel, W. J. Hehre, H. F. Schaefer III, J. Kong, A. I. Krylov, P. M. W. Gill, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 8:3172, 2006. DOI: 10.1039/B517914A.

[4] Y. Shao, Z. Gan, E. Epifanovsky, A. T. B. Gilbert, M. Wormit, J. Kussmann, A. W. Lange, A. Behn, J. Deng, X. Feng, D. Ghosh, M. Goldey, P. R. Horn, L. D. Jacobson, I. Kaliman, R. Z. Khaliullin, T. Kús, A. Landau, J. Liu, E. I. Proynov, Y. M. Rhee, R. M. Richard, M. A. Rohrdanz, R. P. Steele, E. J. Sundstrom, H. L. Woodcock III, P. M. Zimmerman, D. Zuev, B. Albrecht, E. Alguire, B. Austin, G. J. O. Beran, Y. A. Bernard, E. Berquist, K. Brandhorst, K. B. Bravaya, S. T. Brown, D. Casanova, C.-M. Chang, Y. Chen, S. H. Chien, K. D. Closser, D. L. Crittenden, M. Diedenhofen, R. A. DiStasio Jr., H. Do, A. D. Dutoi, R. G. Edgar, S. Fatehi, L. Fusti-Molnar, A. Ghysels, A. Golubeva-Zadorozhnaya, J. Gomes, M. W. D. Hanson-Heine, P. H. P. Harbach, A. W. Hauser, E. G. Hohenstein, Z. C. Holden, T.-C. Jagau, H. Ji, B. Kaduk, K. Khistyaev, J. Kim, J. Kim, R. A. King, P. Klunzinger, D. Kosenkov, T. Kowalczyk, C. M. Krauter, K. U. Lao, A. D. Laurent, K. V. Lawler, S. V. Levchenko, C. Y. Lin, F. Liu, E. Livshits, R. C. Lochan, A. Luenser, P. Manohar, S. F. Manzer, S.-P. Mao, N. Mardirossian, A. V. Marenich, S. A. Maurer, N. J. Mayhall, E. Neuscamman, C. M. Oana, R. Olivares-Amaya, D. P. O'Neill, J. A. Parkhill, T. M. Perrine, R. Peverati, A. Prociuk, D. R. Rehn, E. Rosta, N. J. Russ, S. M. Sharada, S. Sharma, D. W. Small, A. Sodt, T. Stein, D. Stück, Y.-C. Su, A. J. W. Thom, T. Tsuchimochi, V. Vanovschi, L. Vogt, O. Vydrov, T. Wang, M. A. Watson, J. Wenzel, A. White, C. F. Williams, J. Yang, S. Yeganeh, S. R. Yost, Z.-Q. You, I. Y. Zhang, X. Zhang, Y. Zhao, B. R. Brooks, G. K. L. Chan, D. M. Chipman, C. J. Cramer, W. A. Goddard III, M. S. Gordon, W. J. Hehre, A. Klamt, H. F. Schaefer III, M. W. Schmidt, C. D. Sherrill, D. G. Truhlar, A. Warshel, X. Xua, A. Aspuru-Guzik, R. Baer, A. T. Bell, N. A. Besley, J.-D. Chai, A. Dreuw, B. D. Dunietz, T. R. Furlani, S. R. Gwaltney, C.-P. Hsu, Y. Jung, J. Kong, D. S. Lambrecht, W. Liang, C. Ochsenfeld, V. A. Rassolov, L. V. Slipchenko, J. E. Subotnik, T. Van Voorhis, J. M. Herbert, A. I. Krylov, P. M. W. Gill, and M. Head-Gordon. *Mol. Phys.*, 113:184, 2015. DOI: 10.1080/00268976.2014.952696.

Chapter 2

Installation, Customization, and Execution

2.1 Installing Q-CHEM

2.1.1 Downloading and Licensing

Users are referred to the detailed installation instructions available at https://www.q-chem.com/install/.

An encrypted license file must be obtained from your vendor before you will be able to use Q-CHEM. Q-Chem licenses can be issued in one of two ways. In the first method, node-locked licensing, Q-CHEM will only operate correctly on the machine for which the license file(s) have been generated. In the second method, FlexNet licensing, the license is issued for one node, and the other nodes check out licenses in order to run Q-Chem. These licensing types are described in further detail below, and information about obtaining these files can be found in the installation section.

Do not alter the license file unless directed by Q-CHEM, Inc.

2.1.1.1 Node-locked Licensing

Node-locked licensing requires obtaining a license for each machine that will be running Q-CHEM. On a supercomputing cluster, for example, host IDs need to be generated for each individual node, and the license file must be regenerated whenever nodes are added or removed from the cluster. This licensing option works best for running Q-Chem on workstations, and in some special cases where FlexNet licensing is untenable.

2.1.1.2 FlexNet Licensing

FlexNet (formerly known as flexlm) is a convenient option for managing Q-Chem licenses in a computer cluster or supercomputer setting. One node (for example, the head node in a cluster or another dedicated node) runs the licensing server software and provides access to the Q-Chem license to all client compute nodes. This method requires coordination with the administrator of the cluster to set up, but information only needs to be collected for one node, and you can easily add or remove nodes from the cluster without needing to reissue the license. The FlexNet server licensing option is available free of charge to eligible users running Q-Chem for Linux.

2.1.2 Installation Requirements

2.1.2.1 Execution Environment

Q-CHEM is shipped as a single executable along with several scripts. No compilation is required. Once the package is installed it is ready to run. Please refer to the installation notes for your particular platform, which are distributed with the software. The system software required to run Q-CHEM on your platform is minimal, and includes:

- A suitable operating system.
- Run-time libraries (usually provided with your operating system).

Please check the Q-CHEM web site (www.q-chem.com) or contact Q-CHEM support (support@q-chem.com) if further details are required.

2.1.2.2 Hardware Platforms and Operating Systems

Q-CHEM runs on a wide variety of computer systems, ranging from Intel and AMD microprocessor-based PCs and workstations, to high-performance server nodes used in clusters and supercomputers. Q-CHEM supports the Linux, Mac, and Windows operating systems. To determine the availability of a specific platform or operating system, please contact support@q-chem.com.

2.1.2.3 Memory and Disk Requirements

Memory

Q-CHEM, Inc. has endeavored to minimize memory requirements and maximize the efficiency of memory usage. Still, the larger the structure or the higher the level of theory, the more memory is needed. Although Q-CHEM can be run successfully in very small-memory environments, this is seldom an issue nowadays and we recommend 2 GB per CPU core as a minimum. Q-CHEM also offers the ability for user control of important, memory-intensive aspects of the program. In general, the more memory your system has, the larger the calculation you will be able to perform.

Q-CHEM uses two types of memory: a chunk of static memory that is used by multiple data sets and managed by the code, and dynamic memory which is allocated using system calls. The size of the static memory is specified by the user through the *$rem* variable MEM_STATIC and has a default value of 192 MB.

The *$rem* variable MEM_TOTAL specifies the limit of the total memory the user's job can use. The default value is sufficiently large that on most machines it will allow Q-CHEM to use all the available memory. This value should be reduced on machines where this is undesirable (for example if the machine is used by multiple users). The limit for the dynamic memory allocation is given by (MEM_TOTAL − MEM_STATIC). The amount of MEM_STATIC needed depends on the size of the user's particular job. Please note that one should not specify an excessively large value for MEM_STATIC, otherwise it will reduce the available memory for dynamic allocation. Memory settings in CC, EOM, and ADC calculations are described in Section 6.17. The use of *$rem* variables will be discussed in the next Chapter.

Disk

The Q-CHEM executables, shell scripts, auxiliary files, samples and documentation require about 1.4GB of disk space, depending on the platform. The default Q-CHEM output, which is printed to the designated output file, is usually only a few kilobytes. This will be exceeded, of course, in difficult geometry optimizations, QM/MM and QM/EFP jobs, as well as in cases where users invoke non-default print options. In order to maximize the capabilities of your copy of Q-CHEM, additional disk space is required for scratch files created during execution, and these are automatically

deleted upon normal termination of a job. The amount of disk space required for scratch files depends critically on the type of job, the size of the molecule and the basis set chosen.

Q-CHEM uses direct methods for Hartree-Fock and density functional theory calculations, which do not require a large amount of scratch disk space. Wave function-based correlation methods, such as MP2 and coupled-cluster theory, require substantial amounts of temporary (scratch) disk storage, and the faster the access speeds, the better these jobs will perform. With the low cost of disk drives, it is feasible to have between 100 and 1000 GB of scratch space available as a dedicated file system for these large temporary job files. The more you have available, the larger the jobs you will be able to run. In the case of some jobs, like MP2, the jobs will also run faster as two-electron integrals are computed less often.

2.1.3 Q-CHEM Auxiliary files (*$QCAUX*)

The `$QCAUX` environment variable determines the directory where Q-CHEM searches for auxiliary files and the machine license. If not set explicitly, it defaults to `$QC/qcaux`.

The *$QCAUX* directory contains files required to run Q-CHEM calculations, including basis set and ECP specifications, SAD guesses (see Chapter 4), library of standard effective fragments (see Section 11.5), and instructions for the AOINTS package for generating two-electron integrals efficiently.

2.1.4 Q-CHEM Run-time Environment Variables

Q-CHEM requires the following shell environment variables setup prior to running any calculations:

QC	Defines the location of the Q-CHEM directory structure. The `qchem.install` shell script determines this automatically.
QCAUX	Defines the location of the auxiliary information required by Q-CHEM, which includes the license required to run Q-CHEM. If not explicitly set by the user, this defaults to `$QC/qcaux`.
QCSCRATCH	Defines the directory in which Q-CHEM will store temporary files. Q-CHEM will usually remove these files on successful completion of the job, but they can be saved, if so wished. Therefore, `$QCSCRATCH` should not reside in a directory that will be automatically removed at the end of a job, if the files are to be kept for further calculations. Note that many of these files can be very large, and it should be ensured that the volume that contains this directory has sufficient disk space available. The `$QCSCRATCH` directory should be periodically checked for scratch files remaining from abnormally terminated jobs. `$QCSCRATCH` defaults to the working directory if not explicitly set. Please see section 2.2 for details on saving temporary files and consult your systems administrator.
QCLOCALSCR	On certain platforms, such as Linux clusters, it is sometimes preferable to write the temporary files to a disk local to the node. `$QCLOCALSCR` specifies this directory. The temporary files will be copied to `$QCSCRATCH` at the end of the job, unless the job is terminated abnormally. In such cases Q-CHEM will attempt to remove the files in `$QCLOCALSCR`, but may not be able to due to access restrictions. Please specify this variable only if required.

2.1.5 User Account Adjustments

In order for individual users to run Q-CHEM, User file access permissions must be set correctly so that the user can read, write and execute the necessary Q-CHEM files. It may be advantageous to create a *qchem* user group on your machine and recursively change the group ownership of the Q-CHEM directory to *qchem* group.

The Q-CHEM run-time environment need to be initiated prior to running any Q-CHEM calculations, which is done by sourcing the environment setup script qcenv.sh (for bash) or qcenv.csh (for csh and tcsh) placed in your Q-CHEM top directory after a successful installation. It might be more convenient for user to include the Q-CHEM environment setup in their shell startup script, *e.g.*, `.cshrc` or `.tcshrc` for csh or tcsh, respectively, or `.bashrc` for bash.

If using the csh or tcsh shell, add the following lines to the `.cshrc` file in the user's home directory:

```
#
setenv  QC          qchem_root_directory_name
setenv  QCSCRATCH   scratch_directory_name
source  $QC/qcenv.csh
#
```

If using the Bourne-again shell (bash), add the following lines to the `.bashrc` file in the user's home directory:

```
#
export QC=qchem_root_directory_name
export QCSCRATCH=scratch_directory_name
. $QC/qcenv.sh
#
```

2.1.6 Further Customization

Q-CHEM has developed a simple mechanism for users to set user-defined long-term defaults to override the built-in program defaults. Such defaults may be most suited to machine specific features such as memory allocation, as the total available memory will vary from machine to machine depending on specific hardware and accounting configurations. However, users may identify other important uses for this customization feature. Q-CHEM obtains input initialization variables from four sources:

1. User input file
2. `$HOME/.qchemrc` file
3. `$QC/config/preferences` file
4. "Factory installed" program defaults

Input mechanisms higher in this list override those that are lower. Mechanisms #2 and #3 allow the user to specify alternative default settings for certain variables that will override the Q-CHEM "factory-installed" defaults. This can be done by a system administrator via a `preferences` file added to the `$QC/config` directory, or by an individual user by means of a `.qchemrc` file in her home directory.

Note: The `.qchemrc` and `preferences` files are not requisites for running Q-CHEM and currently only support keywords in the *$rem* input section.

The format of the `.qchemrc` and `preferences` files consists of a *$rem* keyword section, as in the Q-CHEM input file, terminated with the usual *$end* keyword. Any other `$whatever` section will be ignored. To aid in reproducibility, a copy of the `.qchemrc` file (if present) is included near the top of the job's output file. (The `.qchemrc` and

`preferences` files must have file permissions such that they are readable by the user invoking Q-CHEM.) The format of both of these files is as follows:

```
$rem
   rem_variable   option   comment
   rem_variable   option   comment
   ...
$end
```

Example 2.2.1 An example of a `.qchemrc` file to override default *$rem* settings for all of the user's Q-CHEM jobs.

```
$rem
   DIIS_SUBSPACE_SIZE   5         Modify max DIIS subspace size
   THRESH               10        10**(-10) threshold
   MAX_SCF_CYCLES       100       More than the default of 50
$end
```

The following *$rem* variables are specifically recommended as those that a user might want to customize:

- MEM_STATIC
- SCF_CONVERGENCE
- THRESH
- MAX_SCF_CYCLES
- GEOM_OPT_MAX_CYCLES

2.2 Running Q-CHEM

2.2.1 General Usage

Once installation is complete, and any necessary adjustments are made to the user account, the user is now able to run Q-CHEM. There are several ways to invoke Q-CHEM:

1. IQMOL offers a fully integrated graphical interface for the Q-CHEM package and includes a sophisticated input generator with contextual help which is able to guide you through the many Q-CHEM options available. It also provides a molecular builder, job submission and monitoring tools, and is able to visualize molecular orbitals, densities and vibrational frequencies. For the latest version and download/installation instructions, please see the IQMOL homepage (http://www.iqmol.org).

2. *qchem* command line shell script. The simple format for command line execution is given below. The remainder of this manual covers the creation of input files in detail.

3. Via a third-party graphical user interface (GUI). The two most popular ones are:

   - A general web-based interface for electronic structure software, WEBMO (www.webmo.net).
   - Wavefunction's SPARTAN user interface on some platforms. Contact Wavefunction, Inc. (www.wavefun.com) or Q-CHEM for full details of current availability.

Using the Q-CHEM command line shell script (*qchem*) is straightforward provided Q-CHEM has been correctly installed on your machine and the necessary environment variables have been set in your `.cshrc`, `.profile`, or equivalent login file. If done correctly, the necessary changes will have been made to the `$PATH` variable automatically on login so that Q-CHEM can be invoked from your working directory.

The *qchem* shell script can be used in either of the following ways:

```
qchem infile outfile
qchem infile outfile savename
qchem -save infile outfile savename
qchem -archive infile outfile
```

where `infile` is the name of a suitably formatted Q-CHEM input file (detailed in Chapter 3, and the remainder of this manual), and the `outfile` is the name of the file to which Q-CHEM will place the job output information.

Note: If the `outfile` already exists in the working directory, it will be overwritten.

The use of the `savename` command line variable allows the saving of a few key scratch files between runs, and is necessary when instructing Q-CHEM to read information from previous jobs. If the `savename` argument is not given, Q-CHEM deletes all temporary scratch files at the end of a run. The saved files are in `$QCSCRATCH/savename/`, and include files with the current molecular geometry, the current molecular orbitals and density matrix and the current force constants (if available). The *–save* option in conjunction with `savename` means that all temporary files are saved, rather than just the few essential files described above. Normally this is not required. When `$QCLOCALSCR` has been specified, the temporary files will be stored there and copied to `$QCSCRATCH/savename/` at the end of normal termination.

The name of the input parameters `infile`, `outfile` and *save* can be chosen at the discretion of the user (usual UNIX file and directory name restrictions apply). It maybe helpful to use the same job name for `infile` and `outfile`, but with varying suffixes. For example:

```
localhost-1> qchem water.in water.out &
```

invokes Q-CHEM where the input is taken from `water.in` and the output is placed into `water.out`. The `&` places the job into the background so that you may continue to work in the current shell.

```
localhost-2> qchem water.com water.log water &
```

invokes Q-CHEM where the input is assumed to reside in `water.com`, the output is placed into `water.log` and the key scratch files are saved in a directory `$QCSCRATCH/water/`.

2.2.1.1 OpenMP Parallelization

Parallel execution of Q-CHEM can be threaded across multiple processors on a single node using the OpenMP protocol. To run a Q-CHEM calculation with OpenMP threads, specify the number of threads (*nthreads*) using the *qchem* command option *-nt*. Since each thread uses one CPU core, you should not specify more threads than the total number of available CPU cores for performance reason. When unspecified, the number of threads defaults to 1 (serial calculation).

```
qchem -nt nthreads infile outfile
qchem -nt nthreads infile outfile save
qchem -save -nt nthreads infile outfile save
```

To run parallel Q-CHEM via the Slurm job scheduling system, add the *-slurm* command line argument when starting Q-CHEM. For example:

```
qchem -slurm -nt nthreads infile outfile
```

2.2.1.2 Hybrid MPI+OMP Parallelization

Starting v6.3.0, parallel execution of Q-CHEM using the message-passing interface (MPI) protocol across multiple processor cores and/or multiple compute nodes is possible on top of the OpenMP parallelization across multiple threads for certain job types; the older MPI-parallelized features predating v5.4.0 have been deprecated. This hybrid scheme involves multiple MPI processes, each spawning several OpenMP threads. In this hybrid scheme, cross-node communication is handled by the MPI protocol and intra-node communication is done implicitly using OpenMP threading for efficient utilization of shared-memory parallel (SMP) systems. This parallelization strategy reflects current trends towards multi-core architectures in cluster computing. As of v6.3.0, the new hyrbid MPI+OMP scheme is *only* available for calculations involving finite-difference derivatives or the many-body interaction method that don't depend on a prior calculation. For example, these calculations cannot be started using the SCF_GUESS = READ. Further, geometry optimization using the LIBOPT3 library is possible with MPI+OMP-enabled finite-difference gradients.

To run parallel calculations with MPI, use the option *-np* to specify the number of MPI processes to be spawned as follows:

```
qchem -mpi -np nprocs infile outfile
qchem -mpi -np nprocs infile outfile savename
qchem -save -mpi -np nprocs infile outfile savename
```

where *nprocs* is the number of processors to use. If the *-np* switch is not given, Q-CHEM will default to running locally on a single node.

To run hybrid MPI+OpenMP calculations, use options *-mpi -np* and *-nt* together, where *-np* followed by the number of MPI processes to be spawned and *-nt* followed by the number of OpenMP threads used in each MPI process as follows:

```
qchem -mpi -np nprocs -nt nthreads infile outfile
qchem -mpi -np nprocs -nt nthreads infile outfile savename
qchem -save -mpi -np nprocs -nt nthreads infile outfile savename
```

When the additional argument *savename* is specified, the temporary files on MPI rank `n` are stored in a folder named `$QCSCRATCH/savename.n`. Upon successful completion of the job, the relevant scratch information is copied back to the `$QCSCRATCH/savename` folder on the head node. If the job terminates abnormally, the files will not be copied to the head node.

On computer systems with batch schedulers such as PBS and SLURM, users may need to set `$QCMPIRUN` environment variable to point to the *mpirun* or *srun* command used in the system. In addition, `$QCMPI` may need to be set to "SLURM" for SLURM scheduler. For further details users should read the `$QC/README.Parallel` file. Please contact Q-CHEM if any problems are encountered (`support@q-chem.com`).

2.2.1.3 GPU-accelerated Q-CHEM with BRIANQC

Starting with version 5.0, the core parts of Q-CHEM calculations can be accelerated using the BRIANQC GPU module. It does so by providing routines for computing all components of the Fock matrix (Eq. (4.18)): the core Hamiltonian,

Coulomb, exchange, and exchange-correlation (Eq. (5.9)) integrals, along with their first derivatives and the most time-consuming parts of their second derivatives. This can lead to significant speedups when computing Hartree-Fock and density functional theory energies, gradients, vibrational frequencies, and other calculations requiring these quantities. Range-separated hybrid density functionals, where the exchange contribution is split into two terms (Eq. (5.12)), are also supported.

In order to invoke BRIANQC, pass the *-gpu* flag when starting Q-CHEM. Because BRIANQC does not accelerate all parts of Q-CHEM calculations, and GPU acceleration works transparently with OpenMP threading, it is still important to parallelize the remaining parts of a calculation using OpenMP threading.

```
qchem -gpu -nt nthreads infile outfile
```

Requirements for using BRIANQC are:

- A separate BRIANQC license
- A 64-bit Linux or Windows-based operating system
- An Nvidia GPU based on the Pascal, Volta, Turing, or Ampere architecture
- A basis set with g angular momentum or lower functions
- Only one Q-CHEM calculation running per GPU

To learn more, visit https://www.brianqc.com/.

2.2.2 Integration with IQMOL

2.2.2.1 Installation and Server Setup

IQMOL provides a fully integrated molecular builder and viewer for the Q-CHEM package. It is available for the Windows, Linux, and Mac OS X platforms and instructions for downloading and installing the latest version can be found at http://www.iqmol.org/downloads.html.

IQMOL can be run as a stand-alone package which is able to open existing Q-CHEM input/output files, but it can also be used as a fully functional front end which is able to submit and monitor Q-CHEM jobs, and to analyze the resulting output. By default, IQMOL submits Q-CHEM jobs to a server that is owned by Q-CHEM, Inc., which provides prospective users with the opportunity to run short Q-CHEM demonstration jobs for free simply by downloading IQMOL, without the need to install Q-CHEM.

For customers who own Q-CHEM, it is necessary to configure IQMOL to submit jobs to an appropriate server. To do this, first ensure Q-CHEM has been correctly installed on the target machine and can be run from the command line. Second, open IQMOL and carry out the following steps:

1. Select the Calculation→Edit Servers menu option. A dialog will appear with a list of configured servers (which will initially be empty).
2. Click the Add New Server button with the '+' icon. This opens a dialog which allows the new server to be configured. The server is the machine which has your Q-CHEM installation.
3. Give the server a name (this is simply used to identify the current server configuration and does not have to match the actual machine name) and select if the machine is local (*i.e.* the same machine as IQMOL is running on) or remote.

4. If there is PBS software running on the server, select the PBS 'Type' option, otherwise in most cases the Basic option should be sufficient. Please note that the server must be Linux based and cannot be a Windows server.

5. If required, the server can be further configured using the Configure button. Details on this can be found in the embedded IQMOL help which can be accessed via the Help→Show Help menu option.

6. For non-PBS servers the number of concurrent Q-CHEM jobs can be limited using a simple inbuilt queuing system. The maximum number of jobs is set by the Job Limit control. If the Job Limit is set to zero the queue is disabled and any number of jobs can be run concurrently. Please note that this limit applies to the current IQMOL session and does not account for jobs submitted by other users or by other IQMOL sessions.

7. The `$QC` environment variable should be entered in the given box.

8. For remote servers the address of the machine and your user name are also required. IQMOL uses SSH2 to connect to remote machines and the most convenient way to set this up is by using authorized keys () for details on how these can be set up). IQMOL can then connect via the SSH Agent and will not have to prompt you for your password. If you are not able to use an SSH Agent, several other authentication methods are offered:

   - **Public Key** This requires you to enter your SSH passphrase (if any) to unlock your private key file. The passphrase is stored in memory, not disk, so you will need to re-enter this each time IQMOL is run.
   - **Password Prompt** This requires each server password to be entered each time IQMOL is run. Once the connection has been established the memory used to hold the password is overwritten to reduce the risk of recovery from a core dump.

   Further configuration of SSH options should not be required unless your public/private keys are stored in a non-standard location.

It is recommended that you test the server configuration to ensure everything is working before attempting to submit a job. Multiple servers can be configured if you have access to more than one copy of Q-CHEM or have different account configurations. In this case the default server is the first on the list and if you want to change this you should use the arrow buttons in the Server List dialog. The list of configured servers will be displayed when submitting Q-CHEM jobs and you will be able to select the desired server for each job.

Please note that while Q-CHEM is file-based, as of version 2.1 IQMOL uses a directory to keep the various files from a calculation. More details can be found in the IQMOL `user manual`.

2.2.2.2 Reading Q-CHEM Outputs

In addition to having IQmol communicate with Q-CHEM, it can work standalone by reading output files directly. A number of file formats are supported:

- Q-CHEM outputs produced by running the *qchem* shell script (Section 2.2.1)
- Formatted checkpoint (`.fchk`) files: A formatted checkpoint file can be requested by setting GUI = 2 in the *$rem* section of the input, or equivalently by setting IQMOL_FCHK = TRUE. The checkpoint file name is determined by the `$GUIFILE` environment variable which by default is set to `${input}.fchk`.
- *qarchive* files: We have created a new file format designed to overcome limitations of using text-based formats with post-processing and visualization tools. The *qarchive* format is HDF5-based, supports more of Q-CHEM's novel feature set than fchk files, and is tightly integrated with IQMOL. In order to create this file in your working directory, pass the *-archive* flag to the *qchem* shell script and add the following to the input:

```
$archive
    enable_archive = True !Turns on generation of Archive
$end
```

The archive file is also always present in each job's scratch directory when the *-save* flag is used. It appears under the file name `$QCSCRATCH/savename/qarchive.h5`

2.2.3 Testing and Exploring Q-CHEM

Q-CHEM is shipped with a small number of test jobs which are located in the `$QC/samples` directory. If you wish to test your version of Q-CHEM, run the test jobs in the samples directory and compare the output files with the reference files of the same name, having suffixes `.out`.

These test jobs are not an exhaustive quality control test (a small subset of the test suite used at Q-CHEM Inc.), but they should all run correctly on your platform. If any fault is identified in these, or any output files created by your version, do not hesitate to contact customer service immediately.

These jobs are also an excellent way to begin learning about Q-CHEM's text-based input and output formats in detail. In many cases you can use these inputs as starting points for building your own input files, if you wish to avoid reading the rest of this manual.

Please check the Q-CHEM web page (`www.q-chem.com`) and the README files in the `$QC/doc` directory for updated information.

Chapter 3

Q-CHEM Inputs

3.1 IQMOL

The easiest way to run Q-CHEM is by using the IQMOL interface which can be downloaded for free from http://www.iqmol.org. Before submitting a Q-CHEM job from you will need to configure a Q-CHEM server and details on how to do this are given in Section 2.2.2 of this manual.

IQMOL provides a free-form molecular builder and a comprehensive interface for setting up the input for Q-CHEM jobs. Additionally calculations can be submitted to either the local or a remote machine and monitored using the built in job monitor. The output can also be analyzed allowing visualization of molecular orbitals and densities, and animation of vibrational modes and reaction pathways. A more complete list of features can be found at http://www.iqmol.org/features.html.

The IQMOL program comes with a built-in help system that details how to set up and submit Q-CHEM calculations. This help can be accessed via the Help→Show Help menu option.

3.2 General Form

IQMOL (or another graphical interface) is the simplest way to control Q-CHEM. However, the low level command line interface is available to enable maximum customization and allow the user to exploit all Q-CHEM's features. The command line interface requires a Q-CHEM input file which is simply an ASCII text file. This input file can be created using your favorite editor (*e.g.*, vi, emacs, jot, *etc.*) following the basic steps outlined in the next few chapters.

Q-CHEM's input mechanism uses a series of **keywords** to signal user input sections of the input file. As required, the Q-CHEM program searches the input file for supported keywords. When Q-CHEM finds a keyword, it then reads the section of the input file beginning at the keyword until that keyword section is terminated the *$end* keyword. A short description of all Q-CHEM keywords is provided in Table 3.1 and the following sections. The user **must** understand the function and format of the *$molecule* (Section 3.3) and *$rem* (Section 3.4) keywords, as these keyword sections are where the user places the molecular geometry information and job specification details.

Table 3.1: Partial list of Q-CHEM input sections; the first two (*$molecule* and *$rem*) are required for all jobs, whereas the rest are required only for certain job types, or else are optional places to specify additional job-control variables. Each input section ("*$section*") should be terminated with *$end*. See the `$QC/samples` directory that is included with your release for specific examples of Q-CHEM input files using these keywords.

Section Name	Description
$molecule	Contains the molecular coordinate input (input file requisite).
$rem	Job specification and customization parameters (input file requisite).
$active_orbitals	Specify active orbitals for nonorthogonal configuration interaction (Section 7.5.2).
$alist	Specify active orbitals for TDDFT with a truncated subspace (Section 7.13.2).
$aux_basis	User-defined auxiliary basis set for resolution-of-identity calculations (Chapter 8.4).
$aux_basis_j	User-defined auxiliary basis set for resolution-of-identity Coulomb calculations (Chapter 8.4).
$aux_basis_k	User-defined auxiliary basis set for resolution-of-identity exact exchange calculations (Chapter 8.4).
$aux_basis_corr	User-defined auxiliary basis set for resolution-of-identity correlation methods (Chapter 8.4).
$basis	User-defined basis set information (Chapter 8).
$cdft	Options for the constrained DFT method (Section 5.11).
$chem_sol	Job control for the Q-CHEM/CHEMSOL interface (Langevin dipoles model; Section 11.2.10).
$comment	User comments for inclusion into output file (Section B.1.1.4).
$complex_ccman	Contains parameters for complex-scaled and CAP-augmented EOM-CC calculations (Chapter 7.10).
$ecp	User-defined effective core potentials (Section 8.10.5).
$efei	Application of external forces in a geometry optimization (Section 9.5.1).
$efp_fragments	Specifies labels and positions of EFP fragments (Section 11.5).
$efp_params	Contains user-defined parameters for effective fragments (Section 11.5).
$empirical_dispersion	User-defined van der Waals parameters for DFT dispersion correction (Section 5.7.3).
$eom_user_guess	User-defined guess for EOM-CC calculations (Chapter 7.10).
$external_charges	Specifies external point charges and their positions (Section B.1.1.7).
$fde	Specifies frozen density embedding options (Section 11.7).
$force_field_params	Force-field parameters for QM/MM calculations (Section 11.3).
$geom_opt	Geometry optimization controls for LIBOPT3 calculations (Section 9.2).
$harmonic_opt	Information for optimization with soft harmonic constraints (Section 9.4.7).

`Continued on next page`

Table 3.1 – Continued from previous page

Section Name	Description
$intracule	Intracule parameters (Section 13.2).
$isotopes	Isotopic substitutions for vibrational calculations (Section 10.7.2).
$localized_diabatization	Information for mixing together multiple adiabatic states into diabatic states (Chapter 10).
$magnet	Job control for magnetic field-related response properties (Section 10.10.4).
$mass	User-defined atomic mass (Chapter 9.9).
$multipole_field	Details of an external multipole field (Section B.1.1.10).
$nbo	Options for the Natural Bond Orbital package (Section 10.3.1).
$occupied	Guess orbitals to be occupied (Section 4.4.4).
$opt	Constraint definitions for geometry optimizations (Section 9.4).
$pcm	Job control for polarizable continuum models (Section 11.2.4).
$plots	Generate plotting information over a grid of points (Section 10.5).
$qct_active_modes *$qct_vib_distribution* *$qct_vib_phase*	Information for quasi-classical trajectory calculations (Section 9.9.6).
$qm_atoms	Specify the QM region for QM/MM calculations (Section 11.3).
$response	Job control for the generalized response solver (Section 10.13).
$solvent	Additional parameters and variables for implicit solvent models (Section 11.2).
$smx	Job control for SMx implicit solvent models (Section 11.2.9).
$spin-spin	Indices for atoms to include in spin-spin coupling calculations (Section 10.10.2.1).
$swap_occupied_virtual	Guess orbitals to be swapped (Section 4.4.4).
$svp	Special parameters for the iso-density SS(V)PE module (Section 11.2.6).
$svpirf	Initial guess for the iso-density SS(V)PE module (Section 11.2.6).
$van_der_waals	User-defined atomic radii for Langevin dipoles solvation (Section 11.2.10) and PCMs (Section 11.2.3).
$velocity	User-defined nuclear velocity for AIMD calculations (Chapter 9.9).
$xc_functional	User-defined DFT exchange-correlation functional (Section 5.3.7).
$xdm	Job control for the XDM dispersion method (Section 5.7.4).
$zbasis	User-defined complex basis set information (Section 8.7).
$2pa	Additional parameters for two-photon absorption calculations (Section 7.10.20.6).

The keywords *$rem* and *$molecule* are required in any Q-CHEM input file

As each keyword has a different function, the format required for specific keywords varies somewhat, to account for these differences (format requirements are summarized in Appendix B). However, because each keyword in the input file is sought out independently by the program, the overall format requirements of Q-CHEM input files are much less stringent. For example, the *$molecule* section does not have to occur at the very beginning of the input file.

Note: (1) Users are able to enter keyword sections in any order.
(2) Each keyword section must be terminated with the *$end* keyword.
(3) The *$rem* and *$molecule* sections must be included.
(4) It is not necessary to have all keywords in an input file.
(5) Each keyword section is described in Appendix B.
(6) The entire Q-CHEM input is case-insensitive.

The second general aspect of Q-CHEM input is that there are effectively four input sources:

- User input file (required)
- `.qchemrc` file in `$HOME` (optional)
- `preferences` file in `$QC/config` (optional)
- Internal program defaults and calculation results (built-in)

The order of preference is as shown, *i.e.*, the input mechanism offers a program default override for *all* users, default override for *individual* users and, of course, the input file provided by the user overrides all defaults. Refer to Section 2.1.6 for details of `.qchemrc` and `preferences`. Currently, Q-CHEM only supports the *$rem* keyword in `.qchemrc` and `preferences` files.

In general, users will need to enter variables for the *$molecule* and *$rem* keyword section and are encouraged to add a *$comment* for future reference. The necessity of other keyword input will become apparent throughout the manual.

3.3 Molecular Coordinate Input (*$molecule*)

3.3.1 Introduction

The *$molecule* section communicates to the program the charge, spin multiplicity, and geometry of the molecule being considered. The molecular coordinates input begins with two integers: the net charge and the spin multiplicity of the molecule. The net charge can be any integer, including 0 for neutral molecules, positive for cations, negative for anions. The multiplicity can be any integer as well (1 for a singlet, 2 for a doublet, 3 for a triplet, *etc.*). Each subsequent line of the molecular coordinate input corresponds to a single atom in the molecule (or dummy atom), regardless of whether using *Z*-matrix internal coordinates or Cartesian coordinates.

Note: The coordinate system used for declaring an initial molecular geometry by default does not affect that used in a geometry optimization procedure. See Chapter 9.1 which discusses the geometry optimization packages in further detail.

Q-CHEM begins all calculations by rotating and translating the user-defined molecular geometry into a Standard Nuclear Orientation whereby the center of nuclear charge is placed at the origin. This is a standard feature of most quantum chemistry programs. This action can be turned off by using POINT_GROUP_SYMMETRY FALSE.

Note: POINT_GROUP_SYMMETRY = FALSE will also turn off determining and using of the point group symmetry.

Note: Q-CHEM ignores commas and equal signs, and requires all distances, positions and angles to be entered as Ångstroms and degrees unless the INPUT_BOHR *$rem* variable is set to TRUE, in which case all lengths are assumed to be in bohr.

3.3.2 Specifying the Molecular Coordinates Manually

Coordinates in the *$molecule* section can be given either in Cartesian form or in *Z*-matrix form, as discussed below. In either case, Q-CHEM rotates the user-specified coordinates into the so-called *standard nuclear orientation*,[2] defined as the principle axes of nuclear charge. This helps to make outputs more reproducible, because different structure-building tools might provide coordinates in different orientations, and quadrature grids used in DFT and some other types of calculations are not strictly rotationally invariant.[1] Thus, two structures with identical internal coordinates but differing by a rigid rotation can give slightly different energies. By rotating the user's input into a standard orientation, identical structures prepared with different graphical user interfaces should afford the same energy. Occassionally, however, one might want to use the absolute (laboratory-fixed) coordinate frame that is defined by the coordinates provided in the *$molecule* section. This can be obtained by setting NO_REORIENT = TRUE in the *$rem* section, which disables rotation into the standard orientation. The coordinate system that Q-CHEM uses, whether it is the standard one or the user's, is printed in the Q-CHEM output file immediately below where the input file is reproduced.

NO_REORIENT

Disable rotation into the standard nuclear orientation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use the standard nuclear orientation.

TRUE Use the coordinate system defined by the *$molecule* section.

RECOMMENDATION:

Use the default unless you have a reason to want a different coordinate system.

3.3.2.1 Cartesian Coordinates

Q-CHEM can accept a list of N atoms and their $3N$ Cartesian coordinates. The atoms can be entered either as atomic numbers or atomic symbols where each line corresponds to a single atom. The Q-CHEM format for declaring a molecular geometry using Cartesian coordinates (in Ångstroms) is:

```
atom  x-coordinate  y-coordinate  z-coordinate
```

Note: The geometry can by specified in bohr by setting the *$rem* variable INPUT_BOHR equal to TRUE.

Example 3.3.1 Atomic number Cartesian coordinate input for H_2O. The first line species the molecular charge and multiplicity, respectively.

```
$molecule
   0 1
   8   0.000000   0.000000  -0.212195
   1   1.370265   0.000000   0.848778
   1  -1.370265   0.000000   0.848778
$end
```

Example 3.3.2 Atomic symbol Cartesian coordinate input for H_2O.

```
$molecule
   0 1
   O   0.000000   0.000000  -0.212195
   H   1.370265   0.000000   0.848778
   H  -1.370265   0.000000   0.848778
$end
```

Note:

1. Atoms can be declared by either atomic number or symbol.
2. Coordinates can be entered either as variables/parameters or real numbers.
3. Variables/parameters can be declared in any order.
4. A single blank line separates parameters from the atom declaration.

Once all the molecular Cartesian coordinates have been entered, terminate the molecular coordinate input with the *$end* keyword.

3.3.2.2 *Z*-matrix Coordinates

For small molecules, *Z*-matrix notation is a common input format. The *Z*-matrix defines the positions of atoms relative to previously defined atoms using a length, an angle and a dihedral angle. Again, note that all bond lengths and angles must be in Ångstroms and degrees, unless INPUT_BOHR is set to TRUE, in which case bond lengths are specified in bohr.

The first three atom entries of a *Z*-matrix are different from the subsequent entries. The first *Z*-matrix line declares a single atom. The second line of the *Z*-matrix input declares a second atom, refers to the first atom and gives the distance between them. The third line declares the third atom, refers to either the first or second atom, gives the distance between them, refers to the remaining atom and gives the angle between them. All subsequent entries begin with an atom declaration, a reference atom and a distance, a second reference atom and an angle, a third reference atom and a dihedral angle. This can be summarized as:

1. First atom.
2. Second atom, reference atom, distance.
3. Third atom, reference atom A, distance between A and the third atom, reference atom B, angle defined by atoms A, B and the third atom.

4. Fourth atom, reference atom A, distance, reference atom B, angle, reference atom C, dihedral angle (A, B, C and the fourth atom).

5. All subsequent atoms follow the same basic form as (4)

Example 3.3.3 *Z*-matrix input for hydrogen peroxide

```
O1
O2   O1   oo
H1   O1   ho   O2   hoo
H2   O2   ho   O1   hoo   H1   hooh
```

Line 1 declares an oxygen atom (O1). Line 2 declares the second oxygen atom (O2), followed by a reference to the first atom (O1) and a distance between them denoted `oo`. Line 3 declares the first hydrogen atom (H1), indicates it is separated from the first oxygen atom (O1) by a distance HO and makes an angle with the second oxygen atom (O2) of `hoo`. Line 4 declares the fourth atom and the second hydrogen atom (H2), indicates it is separated from the second oxygen atom (O2) by a distance HO and makes an angle with the first oxygen atom (O1) of `hoo` and makes a dihedral angle with the first hydrogen atom (H1) of `hooh`.

Some further points to note are:

- Atoms can be declared by either atomic number or symbol.
  - If declared by atomic number, connectivity needs to be indicated by *Z*-matrix line number.
  - If declared by atomic symbol either number similar atoms (*e.g.*, H1, H2, O1, O2 *etc.*) and refer connectivity using this symbol, or indicate connectivity by the line number of the referred atom.
- Bond lengths and angles can be entered either as variables/parameters or real numbers.
  - Variables/parameters can be declared in any order.
  - A single blank line separates parameters from the *Z*-matrix.

All the following examples are equivalent in the information forwarded to the Q-CHEM program.

Example 3.3.4 Using parameters to define bond lengths and angles, and using numbered symbols to define atoms and indicate connectivity.

```
$molecule
   0 1
   O1
   O2  O1  oo
   H1  O1  ho  O2  hoo
   H2  O2  ho  O1  hoo  H1  hooh

   oo   =   1.5
   oh   =   1.0
   hoo  = 120.0
   hooh = 180.0
$end
```

Example 3.3.5 Not using parameters to define bond lengths and angles, and using numbered symbols to define atoms and indicate connectivity.

```
$molecule
   0 1
   O1
   O2  O1  1.5
   H1  O1  1.0  O2  120.0
   H2  O2  1.0  O1  120.0  H1  180.0
$end
```

Example 3.3.6 Using parameters to define bond lengths and angles, and referring to atom connectivities by line number.

```
$molecule
   0 1
   8
   8  1  oo
   1  1  ho  2  hoo
   1  2  ho  1  hoo  3  hooh

   oo   =   1.5
   oh   =   1.0
   hoo  = 120.0
   hooh = 180.0
$end
```

Example 3.3.7 Referring to atom connectivities by line number, and entering bond length and angles directly.

```
$molecule
  0 1
  8
  8  1  1.5
  1  1  1.0  2  120.0
  1  2  1.0  1  120.0  3  180.0
$end
```

Obviously, a number of the formats outlined above are less appealing to the eye and more difficult for us to interpret than the others, but each communicates *exactly* the same *Z*-matrix to the Q-CHEM program.

3.3.2.3 Dummy Atoms

Dummy atoms are indicated by the identifier X and followed, if necessary, by an integer. (*e.g.*, $X1$, $X2$. Dummy atoms are often useful for molecules where symmetry axes and planes are not centered on a real atom, and have also been useful in the past for choosing variables for structure optimization and introducing symmetry constraints.

Note: Dummy atoms play no role in the quantum mechanical calculation, and are used merely for convenience in specifying other atomic positions or geometric variables.

3.3.3 Reading Molecular Coordinates from a Previous Job or File

Often users wish to perform several calculations in sequence, where the later calculations rely on results obtained from the previous ones. For example, a geometry optimization at a low level of theory, followed by a vibrational analysis and then, perhaps, single-point energy at a higher level. Rather than having the user manually transfer the coordinates from the output of the optimization to the input file of a vibrational analysis or single point energy calculation, Q-CHEM can transfer them directly from job to job.

To achieve this requires that:

- The READ variable is entered into the molecular coordinate input
- Scratch files from a previous calculation have been saved. These may be obtained *explicitly* by using the save option across multiple job runs as described below and in Chapter 2, or *implicitly* when running multiple calculations in one input file, as described in Section 3.5.

Example 3.3.8 Reading a geometry from a prior calculation.

```
$molecule
   READ
$end
```

In this example, the *job1* scratch files are saved in a directory *$QCSCRATCH/job1* and are then made available to the *job2* calculation. This is achieved with the following commands:

```
localhost-1> qchem job1.in job1.out job1
localhost-2> qchem job2.in job2.out job1
```

In this example, the `job1` scratch files are saved in a directory `$QCSCRATCH/job1` and are then made available to the `job2` calculation.

Note: The program must be instructed to read specific scratch files by the input of `job2`.

The READ function can also be used to read molecular coordinates from a second input file. The format for the coordinates in the second file follows that for standard Q-CHEM input, and must be delimited with the *$molecule* and *$end* keywords.

Example 3.3.9 Reading molecular coordinates from another file. `filename` may be given either as the full file path, or path relative to the working directory.

```
$molecule
   READ filename
$end
```

3.4 Job Specification: The *$rem* Input Section

The *$rem* section in the input file is the means by which users specify the type of calculation that they wish to perform (*i.e.*, level of theory, basis set, convergence criteria, additional special features, *etc.*). The keyword *$rem* signals the beginning of the overall job specification. Within the *$rem* section the user inserts ***$rem* variables** (one per line) which define the essential details of the calculation. The allowed format is either

```
REM_VARIABLE     VALUE     [ comment ]
```

or alternatively

```
REM_VARIABLE   =  VALUE    [ comment ]
```

The "=" sign is automatically discarded and only the first two remaining arguments are read, so that all remaining text is ignored and can be used to place comments in the input file. Thus the *$rem* section that provides Q-CHEM job control takes the form shown in the following example.

The general format of the *$rem* section of the text input file is

```
$rem
  REM_VARIABLE  value  [ comment ]
  REM_VARIABLE  value  [ comment ]
  ...
$end
```

Note:

1. Tab stops can be used to format input.
2. A line prefixed with an exclamation mark '!' is treated as a comment and will be ignored by the program.
3. *$rem* variables are case-insensitive (as is the whole Q-CHEM input file).
4. Depending on the particular *$rem* variable, "`value`" may be a keyword (string), an integer, or a logical value (true or false).
5. A complete list of *$rem* variables can be found in Appendix B.

In this manual, *$rem* variables will be described using the following format:

REM_VARIABLE_NAME
: A short description of what the variable controls.

TYPE:
: The type of variable (INTEGER, LOGICAL or STRING)

DEFAULT:
: The default value, if any.

OPTIONS:
: A list of the options available to the user.

RECOMMENDATION:
: A brief recommendation, where appropriate.

For example:

IQMOL_FCHK

Controls printing of a formatted checkpoint file that can be read by the IQMOL program.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not generate the checkpoint file.

OPTIONS:

TRUE Generate a checkpoint file named `inputfilename.fchk`.

RECOMMENDATION:

For many Q-CHEM jobs there is no reason *not* to generate the checkpoint file. Note that GUI = 2 (used by IQMOL) is synonymous with IQMOL_FCHK = TRUE.

If a default setting is indicated for a particular *$rem* variable, then it is not necessary to declare that variable in order for the default setting to be used. For example, the default value for the variable JOBTYPE is SP, indicating a single-point energy calculation, so to perform such a calculation the user does **not** need to set the JOBTYPE variable. To perform a geometry optimization, however, it is necessary to override this default by setting JOBTYPE = OPT. System administrator preferences for default *$rem* settings can be specified in the `$QC/config/preferences` file, and user preferences in a `$HOME/.qchemrc` file, both of which are described in Section 2.1.6.

Q-CHEM provides defaults for most *$rem* variables, but the user will always have to stipulate a few others. In a single point energy calculation, for example, the minimum requirements will be BASIS (defining the basis set) and METHOD (defining the level of theory for correlation and exchange). For example, METHOD = HF invokes a Hartree-Fock calculation, whereas METHOD = CIS specifies a CIS excited-state calculation.

Example 3.3.10 Example of minimal *$rem* requirements to run an MP2/6-31G* single-point energy calculation.

```
$rem
  BASIS          6-31G*    Just a small basis set
  METHOD         mp2       MP2
$end
```

The level of theory can alternatively be specified by setting values for two other *$rem* variables, EXCHANGE (defining the level of theory to treat exchange) and CORRELATION (defining the level of theory to treat electron correlation, if required). For excited states computed using equation-of-motion (EOM) methods (Chapter 7), there is a third *$rem* variable, EOM_CORR, which specifies the level of correlation for the target states.

For DFT calculations, METHOD specifies an exchange-correlation functional; see Section 5.4 for a list of supported functionals. For wave function approaches, supported values of METHOD can be found in Section 6.1 for ground-state methods and in Section 7.1 for excited-state methods. If a wave function-based correlation treatment such as MP2 or CC is requested using the CORRELATION keyword, then HF is taken as the default for EXCHANGE.

3.5 Batch Jobs: Multiple Inputs in a Single File

It is sometimes useful to place a sequence of jobs into a single Q-CHEM input file, where the individual inputs should be separated from one another by a line consisting of the string `@@@`. The output from these jobs is then appended sequentially to a single output file. This is useful to (a) use information obtained in a prior job (*i.e.*, an optimized geometry) in a subsequent job; or (b) keep related calculations together in a single output file.

Some limitations should be kept in mind:

- The first job will overwrite any existing output file of the same name in the working directory. Restarting the job will also overwrite any existing file.
- Q-CHEM reads all the jobs from the input file immediately and stores them. Therefore no changes can be made to the details of subsequent jobs following command-line initiation of Q-CHEM, even if these subsequent jobs have not yet run.
- If any single job fails, Q-CHEM proceeds to the next job in the batch file, for good or ill.
- No check is made to ensure that dependencies are satisfied, or that information is consistent. For example, in a geometry optimization followed by a frequency calculation, no attempt is made by the latter to check that the optimization was successful. When reading MO coefficients from a previous job, it is the user's responsibility to ensure that the basis set is the same in both calculations, as this is assumed by the program.
- Scratch files are saved from one job to the next in a batch job, so that information from previous jobs can be shared with subsequent ones, but are deleted upon completion of the entire batch job unless the *–save* command-line argument is supplied, as discussed in Chapter 2.

The following example requests a batch job consisting of (i) a HF/6-31G* geometry optimization; followed by (ii) a frequency calculation at the same level of theory that uses the previously-optimized geometry (and also reads in the final MOs from the optimization job); and finally (iii) a single-point calculation at the same geometry but at a higher

level of theory, MP2/6-311G(d,p).

Example 3.11 Example of using information from previous jobs in a single input file.

```
$comment
   Optimize H-H at HF/6-31G*
$end

$molecule
   0 1
   H
   H  1  r

   r = 1.1
$end

$rem
   JOBTYPE       opt    Optimize the bond length
   METHOD        hf
   BASIS         6-31G*
$end

@@@

$comment
   Now calculate the frequency of H-H at the same level of theory.
$end

$molecule
   read
$end

$rem
   JOBTYPE       freq    Calculate vibrational frequency
   METHOD        hf
   BASIS         6-31G*
   SCF_GUESS     read    Read the MOs from disk
$end

@@@

$comment
  Now a single point calculation at at MP2/6-311G(d,p)//HF/6-31G*
$end

$molecule
   read
$end

$rem
   METHOD        mp2
   BASIS         6-311G(d,p)
$end
```

3.6 Q-CHEM Output File

When Q-CHEM is invoked using

```
# qchem infile outfile
```

the output file *outfile* contains a variety of information, depending on the type of job(s), but in general consists of the following.

- Q-CHEM citation
- User input (for record-keeping purposes)
- Molecular geometry in Cartesian coordinates
- Molecular point group, nuclear repulsion energy, number of α- and β-spin electrons
- Basis set information (number of functions, shells and function pairs)
- SCF details (method, guess, and convergence procedure)
- Energy and DIIS error for each SCF iteration
- Results of any post-SCF calculation that is requested
- Results of any excited-state calculation that is requested
- Molecular orbital symmetries and energies
- Wave function analysis
- Message signaling successful job completion

Note: If *outfile* above already exists when the job is started, then the existing file is overwritten with the results of the new calculation.

References and Further Reading

[1] S. Dasgupta and J. M. Herbert. *J. Comput. Chem.*, 38:869, 2017. DOI: 10.1002/jcc.24761.

[2] P. M. W. Gill, B. G. Johnson, and J. A. Pople. *Chem. Phys. Lett.*, 209:506, 1993. DOI: 10.1016/0009-2614(93)80125-9.

Chapter 4

Self-Consistent Field Ground-State Methods

4.1 Introduction

Theoretical "model chemistries"[42] involve two principle approximations. One must specify, first of all, the type of atomic orbital (AO) basis set that will be used to construct molecular orbitals (MOs), via the "linear combination of atomic orbitals" (LCAO) *ansatz*, available options for which are discussed in Chapter 8. Second, one must specify the manner in which the instantaneous interactions between electrons ("electron correlation") are to be treated. Self-consistent field (SCF) methods, in which electron correlation is described in a mean-field way, represent the simplest, most affordable, and most widely-used electronic structure methods. The SCF category of methods includes both Hartree-Fock (HF) theory as well as Kohn-Sham (KS) density functional theory (DFT). This Chapter summarizes Q-CHEM's SCF capabilities, while Chapter 5 provides further details specific to DFT calculations. Chapter 6 describes the more sophisticated (but also more computationally expensive!) post-HF, wave function-based methods for describing electron correlation. If you are new to quantum chemistry, we recommend an introductory textbook such as Refs. 42, 90, or 48.

Section 4.2 provides the theoretical background behind SCF methods, including both HF and KS-DFT. In some sense, the former may be considered as a special case of the latter, and job-control *$rem* variables are much the same in both cases. Basic SCF job control is described in Section 4.3. Later sections introduce more specialized options that can be consulted as needed. Of particular note are the following:

- Initial guesses for SCF calculations (Section 4.4). Modification of the guess is recommended in cases where the SCF calculation fails to converge.
- Changing the SCF convergence algorithm (Section 4.5) is also a good strategy when the SCF calculation fails to converge.
- Linear-scaling, "$\mathcal{O}(N)$", and other reduced-cost methods are available for large systems (see Section 4.6).
- Unconventional SCF calculations. Some non-standard SCF methods with novel physical and mathematical features are available. These include:
  - Dual-basis SCF calculations (Section 4.7) and DFT perturbation theory (Section 4.8), which facilitate large-basis quality results but require self-consistent iterations only in a smaller basis set.
  - SCF meta-dynamics (Section 4.9.2), which can be used to locate multiple solutions to the SCF equations and to help check that the solution obtained is actually the lowest minimum.

  Some of these unconventional SCF methods are available exclusively in Q-CHEM.

4.2 Theoretical Background

4.2.1 SCF and LCAO Approximations

The fundamental equation of non-relativistic quantum chemistry is the time-independent Schrödinger equation,

$$\hat{H}(\mathbf{R},\mathbf{r})\ \Psi(\mathbf{R},\mathbf{r}) = E(\mathbf{R})\ \Psi(\mathbf{R},\mathbf{r}). \tag{4.1}$$

In quantum chemistry, this equation is solved as a function of the electronic variables ($\mathbf{r}$), for fixed values of the nuclear coordinates ($\mathbf{R}$). The Hamiltonian operator in Eq. (4.1) is

$$\hat{H} = -\frac{1}{2}\sum_{i=1}^{N}\hat{\nabla}_i^2 - \frac{1}{2}\sum_{A=1}^{M}\frac{1}{M_A}\hat{\nabla}_A^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{r_{ij}} + \sum_{A=1}^{M}\sum_{B>A}^{M}\frac{Z_A Z_B}{R_{AB}} \tag{4.2}$$

in atomic units, where

$$\hat{\nabla}^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\ . \tag{4.3}$$

In Eq. (4.2), Z is the nuclear charge, M_A is the ratio of the mass of nucleus A to the mass of an electron, $R_{AB} = |\mathbf{R}_A - \mathbf{R}_B|$ is the distance between nuclei A and B, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance between the ith and jth electrons, $r_{iA} = |\mathbf{r}_i - \mathbf{R}_A|$ is the distance between the ith electron and the Ath nucleus, M is the number of nuclei and N is the number of electrons. The total energy E is an eigenvalue of $\hat{H}$, with a corresponding eigenfunction (wave function), Ψ.

Separating the motions of the electrons from that of the nuclei, an idea originally due to Born and Oppenheimer,[9] yields the electronic Hamiltonian operator

$$\hat{H}_{\text{elec}} = -\frac{1}{2}\sum_{i=1}^{N}\hat{\nabla}_i^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{r_{ij}} \tag{4.4}$$

The solution of the corresponding electronic Schrödinger equation,

$$\hat{H}_{\text{elec}}\Psi_{\text{elec}} = E_{\text{elec}}\Psi_{\text{elec}}\ , \tag{4.5}$$

affords the total electronic energy, E_{elec}, and electronic wave function, Ψ_{elec}, which describes the distribution of the electrons for fixed nuclear positions. The total energy is obtained by simply adding the nuclear–nuclear repulsion energy [the fifth term in Eq. (4.2)] to the total electronic energy:

$$E_{\text{tot}} = E_{\text{elec}} + E_{\text{nuc}}\ . \tag{4.6}$$

Solving the eigenvalue problem in Eq. (4.5) yields a set of eigenfunctions (Ψ_0, Ψ_1, $\Psi_2\ldots$) with corresponding eigenvalues $E_0 \le E_1 \le E_2 \le \ldots$.

Our interest lies in determining the lowest eigenvalue and associated eigenfunction which correspond to the ground state energy and wave function of the molecule. However, solving Eq. (4.5) for other than the most trivial systems is extremely difficult and the best we can do in practice is to find approximate solutions.

The first approximation used to solve Eq. (4.5) is the independent-electron (mean-field) approximation, in which the wave function is approximated as an antisymmetrized product of one-electron functions, namely, the MOs. Each MO is determined by considering the electron as moving within an average field of all the other electrons. This affords the well-known Slater determinant wave function[80,81]

$$\Psi = \frac{1}{\sqrt{n!}}\begin{vmatrix} \chi_1(1) & \chi_2(1) & \cdots & \chi_n(1) \\ \chi_1(2) & \chi_2(2) & \cdots & \chi_n(2) \\ \vdots & \vdots & & \vdots \\ \chi_1(n) & \chi_2(n) & \cdots & \chi_n(n) \end{vmatrix} \tag{4.7}$$

where χ_i, a spin orbital, is the product of a molecular orbital ψ_i and a spin function (α or β).

One obtains the optimum set of MOs by variationally minimizing the energy in what is called a "self-consistent field" or SCF approximation to the many-electron problem. The archetypal SCF method is the Hartree-Fock (HF) approximation, but these SCF methods also include KS-DFT (Chapter 5). All SCF methods lead to equations of the form

$$\hat{f}(i)\,\chi(\mathbf{x}_i) = \varepsilon\,\chi(\mathbf{x}_i)\ , \tag{4.8}$$

where the Fock operator $\hat{f}(i)$ for the ith electron is

$$\hat{f}(i) = -\frac{1}{2}\hat{\nabla}_i^2 + \upsilon_{\text{eff}}(i)\ . \tag{4.9}$$

Here $\mathbf{x}_i$ are spin and spatial coordinates of the ith electron, the functions χ are spin orbitals and υ_{eff} is the effective potential "seen" by the ith electron, which depends on the spin orbitals of the other electrons. The nature of the effective potential υ_{eff} depends on the SCF methodology, *i.e.*, on the choice of density-functional approximation.

The second approximation usually introduced when solving Eq. (4.5) is the introduction of an AO basis $\{\phi_\mu\}$ linear combinations of which will then determine the MOs. There are many standardized, atom-centered Gaussian basis sets and details of these are discussed in Chapter 8.

After eliminating the spin components in Eq. (4.8) and introducing a finite basis,

$$\psi_i = \sum_\mu c_{\mu i}\phi_\mu\ , \tag{4.10}$$

Eq. (4.8) reduces to the Roothaan-Hall matrix equation

$$\mathbf{FC} = \boldsymbol{\varepsilon}\mathbf{SC}\ . \tag{4.11}$$

Here, $\mathbf{F}$ is the Fock matrix, $\mathbf{C}$ is a square matrix of molecular orbital coefficients, $\mathbf{S}$ is the AO overlap matrix with elements

$$S_{\mu\nu} = \int \phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r})d\mathbf{r} \tag{4.12}$$

and $\boldsymbol{\varepsilon}$ is a diagonal matrix containing the orbital energies. Generalizing to an unrestricted formalism by introducing separate spatial orbitals for α and β spin in Eq. (4.7) yields the Pople-Nesbet equations[69]

$$\mathbf{F}^\alpha\mathbf{C}^\alpha = \boldsymbol{\varepsilon}^\alpha\mathbf{S}\mathbf{C}^\alpha \tag{4.13}$$

$$\mathbf{F}^\beta\mathbf{C}^\beta = \boldsymbol{\varepsilon}^\beta\mathbf{S}\mathbf{C}^\beta \tag{4.14}$$

In SCF methods, an initial guess is for the MOs is first determined, and from this, an average field seen by each electron can be calculated. A new set of MOs can be obtained by solving the Roothaan-Hall or Pople-Nesbet eigenvalue equations, resulting in the restricted or unrestricted finite-basis SCF approximation. This procedure is repeated until the new MOs differ negligibly from those of the previous iteration. The Hartree-Fock approximation for the effective potential in Eq. (4.9) inherently neglects the instantaneous electron-electron correlations that are averaged out by the SCF procedure, and while the chemistry resulting from HF calculations often offers valuable qualitative insight, quantitative energetics are often poor. In principle, the DFT methodologies are able to capture all the correlation energy, *i.e.*, the difference in energy between the HF energy and the true energy. In practice, the best-available density functionals perform well but not perfectly, and conventional post-HF approaches to calculating the correlation energy (see Chapter 6) are often required.

That said, because SCF methods often yield acceptably accurate chemical predictions at low- to moderate computational cost, self-consistent field methods are the cornerstone of most quantum-chemical programs and calculations. The formal costs of many SCF algorithms is $\mathcal{O}(N^4)$, that is, they grow with the fourth power of system size, N. This is slower than the growth of the cheapest conventional correlated methods, which scale as $\mathcal{O}(N^5)$ or worse, algorithmic advances available in Q-CHEM can reduce the SCF cost to $\mathcal{O}(N)$ in favorable cases, an improvement that allows SCF methods to be applied to molecules previously considered beyond the scope of *ab initio* quantum chemistry.

Types of ground-state energy calculations currently available in Q-CHEM are summarized in Table 4.1.

Calculation	***$rem* Variable JOBTYPE**	
Single point energy (default)	SINGLE_POINT or SP	
Force (energy + gradient)	FORCE	
Equilibrium structure search	OPTIMIZATION or OPT	(Ch. 9)
Transition structure search	TS	(Ch. 9)
Intrinsic reaction pathway	RPATH	(Section 9.7)
Potential energy scan	PES_SCAN	(Section 9.6)
Vibrational frequency calculation	FREQUENCY or FREQ	(Section 10.7)
Polarizability and relaxed dipole	POLARIZABILITY, DIPOLE	(Section 10.12.2)
NMR chemical shift	NMR	(Section 10.10.2)
Indirect nuclear spin-spin coupling	ISSC	(Section 10.10.2)
Ab initio molecular dynamics	AIMD	(Section 9.9)
Ab initio path integrals	PIMD, PIMC	(Section 9.10)
BSSE (counterpoise) correction	BSSE	(Section 12.4)
Energy decomposition analysis	EDA	(Sections 12.5.3 and 12.6)
Symmetry-adapted perturbation theory	XSAPT	(Sections 12.9 and 12.10)

Table 4.1: The type of calculation to be run by Q-CHEM is controlled by the *$rem* variable JOBTYPE.

4.2.2 Hartree-Fock Theory

As with much of the theory underlying modern quantum chemistry, the HF approximation was developed shortly after publication of the Schrödinger equation, but remained a qualitative theory until the advent of the computer. Although the HF approximation tends to yield qualitative chemical accuracy, rather than quantitative information, and is generally inferior to many of the DFT approaches available, it remains as a useful tool in the quantum chemist's toolkit. In particular, for organic chemistry, HF predictions of molecular structure are very useful.

Consider once more the Roothaan-Hall equations, Eq. (4.11), or the Pople-Nesbet equations, Eq. (4.13), which can be traced back to Eq. (4.8), in which the effective potential v_{eff} depends on the SCF methodology. In a restricted HF (RHF) formalism, the effective potential can be written as

$$v_{\text{eff}} = \sum_{a}^{N/2} \left[2\hat{J}_a(1) - \hat{K}_a(1) \right] - \sum_{A=1}^{M} \frac{Z_A}{r_{1A}} \tag{4.15}$$

where the Coulomb and exchange operators are defined as

$$\hat{J}_a(1) = \int \psi_a^*(2) \frac{1}{r_{12}} \psi_a(2)\, d\mathbf{r}_2 \tag{4.16}$$

and

$$\hat{K}_a(1)\psi_i(1) = \left[\int \psi_a^*(2) \frac{1}{r_{12}} \psi_i(2)\, d\mathbf{r}_2 \right] \psi_a(1) \tag{4.17}$$

respectively. By introducing an atomic orbital basis, we obtain Fock matrix elements

$$F_{\mu\nu} = H_{\mu\nu}^{\text{core}} + J_{\mu\nu} - K_{\mu\nu} \tag{4.18}$$

where the core Hamiltonian matrix elements

$$H_{\mu\nu}^{\text{core}} = T_{\mu\nu} + V_{\mu\nu} \tag{4.19}$$

consist of kinetic energy elements

$$T_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left(-\frac{1}{2} \hat{\nabla}^2 \right) \phi_\nu(\mathbf{r})\, d\mathbf{r} \tag{4.20}$$

and nuclear attraction elements

$$V_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left(-\sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} \right) \phi_\nu(\mathbf{r}) \, d\mathbf{r} \tag{4.21}$$

The Coulomb and exchange elements are given by

$$J_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma} \, (\mu\nu|\lambda\sigma) \tag{4.22}$$

and

$$K_{\mu\nu} = \frac{1}{2} \sum_{\lambda\sigma} P_{\lambda\sigma} \, (\mu\lambda|\nu\sigma) \tag{4.23}$$

respectively, where the density matrix elements are

$$P_{\mu\nu} = 2 \sum_{a=1}^{N/2} C_{\mu a} C_{\nu a} \tag{4.24}$$

and the two electron integrals are

$$(\mu\nu|\lambda\sigma) = \int \int \phi_\mu(\mathbf{r}_1) \phi_\nu(\mathbf{r}_1) \left(\frac{1}{r_{12}} \right) \phi_\lambda(\mathbf{r}_2) \phi_\sigma(\mathbf{r_2}) \, d\mathbf{r}_1 \, d\mathbf{r}_2 \, . \tag{4.25}$$

Note: The formation and utilization of two-electron integrals is a topic central to the overall performance of SCF methodologies. The performance of the SCF methods in new quantum chemistry software programs can be quickly estimated simply by considering the quality of their atomic orbital integrals packages. See Appendix A for details of Q-CHEM's AOINTS package.

Substituting the matrix element in Eq. (4.18) back into the Roothaan-Hall equations, Eq. (4.11), and iterating until self-consistency is achieved will yield the RHF energy and wave function. Alternatively, one could have adopted the unrestricted form of the wave function by defining separate α and β density matrices:

$$\begin{aligned} P^\alpha_{\mu\nu} &= \sum_{a=1}^{n_\alpha} C^\alpha_{\mu a} C^\alpha_{\nu a} \\ P^\beta_{\mu\nu} &= \sum_{a=1}^{n_\beta} C^\beta_{\mu a} C^\beta_{\nu a} \end{aligned} \tag{4.26}$$

The total electron density matrix $\mathbf{P} = \mathbf{P}^\alpha + \mathbf{P}^\beta$. The unrestricted α Fock matrix,

$$F^\alpha_{\mu\nu} = H^{\text{core}}_{\mu\nu} + J_{\mu\nu} - K^\alpha_{\mu\nu} \, , \tag{4.27}$$

differs from the restricted one only in the exchange contributions, where the α exchange matrix elements are given by

$$K^\alpha_{\mu\nu} = \sum_\lambda^N \sum_\sigma^N P^\alpha_{\lambda\sigma} \, (\mu\lambda|\nu\sigma) \tag{4.28}$$

4.3 Basic SCF Job Control

4.3.1 Introduction

As of version 5.1, Q-CHEM uses a new SCF package, GEN_SCFMAN, developed by E. J. Sundstrom, P. R. Horn and many other coworkers. In addition to supporting the basic features of the previous SCF package (*e.g.* restricted, unrestricted and restricted open-shell HF/KS-DFT calculations), many new features are now available in Q-CHEM, including:

- Addition of several useful SCF convergence algorithms and support for user-specified hybrid algorithm (Section 4.5.12).
- More general and user-friendly internal stability analysis and automatic correction for the energy minimum (Section 4.5.17).

GEN_SCFMAN also supports a wider range of orbital types, including complex orbitals. A full list of supported orbitals is:

- Restricted (R): typically appropriate for closed shell molecules at their equilibrium geometry, where electrons occupy orbitals in pairs.
- Unrestricted (U): appropriate for radicals with an odd number of electrons, and also for molecules with even numbers of electrons where not all electrons are paired, *e.g.*, stretched bonds and diradicals.
- Restricted open-shell (RO): for open-shell molecules, where the α and β orbitals are constrained to be identical.
- Canonical restricted open-shell (can. RO or "ROSCF-DODS"): canonical form of ROSCF, see Section 4.5.18 for details.
- Open-shell singlet ROSCF (OS_RO): see the "ROKS" method documented in Section 7.8.3.
- Generalized (G): *i.e.*, each MO is associated with both α and β spin components.
- The use of complex orbitals (with Hartree-Fock only): restricted (CR), unrestricted (CU), and generalized (CG).

Aspects of an SCF calculation such as the SCF guess, the use of efficient algorithms to construct the Fock matrix like occ-RI-K (see Section 4.6.8), are unaffected by the use of GEN_SCFMAN. Likewise, using GEN_SCFMAN does not make any difference to the post-SCF procedures such as correlated methods, excited state calculations and evaluation of molecular properties.

It should be noted that many special features (*e.g.* dual-basis SCF, CDFT, *etc.*) based on Q-CHEM's old SCF code are not yet supported in GEN_SCFMAN. They will become available in the future.

4.3.2 Job Control

The following two *$rem* variables must be specified in order to run HF calculations:

METHOD

Specifies the exchange-correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use METHOD = *NAME*, where *NAME* is one of the following: HF for Hartree-Fock theory; one of the DFT methods listed in Section 5.3.5.; one of the correlated methods listed in Sections 7.10, 7.11, and 7.9;

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations for DFT are indicated in bold in Section 5.3.5.

BASIS

Specifies the basis sets to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen	User defined (*$basis* keyword required).
Symbol	Use standard basis sets as per Chapter 8.
Mixed	Use a mixture of basis sets (see Chapter 8).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

In addition, the following *$rem* variables can be used to customize the SCF calculation:

GEN_SCFMAN

Use GEN_SCFMAN for the present SCF calculation.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

FALSE	Use the previous SCF code.
TRUE	Use GEN_SCFMAN.

RECOMMENDATION:

Set to FALSE in cases where features not yet supported by GEN_SCFMAN are needed.

ORBITAL_ENERGY_PREC

Obtain additional digits of precision in the orbital energies.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Standard printout with 4 decimal digits (in Hartree).
1	5 decimal digits.
2	Scientific notation with 10 digits of precision.

RECOMMENDATION:

Set as desired. Has no effect in cases where the orbital symmetry labels are printed; use SYM_IGNORE = TRUE if additional precision is needed in such cases.

PRINT_ORBITALS

Prints orbital coefficients with atom labels in analysis part of output.

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print any orbitals.

TRUE Prints occupied orbitals plus 5 virtual orbitals.

NVIRT Number of virtual orbitals to print.

RECOMMENDATION:

Use true unless more virtual orbitals are desired.

SCF_CONVERGENCE

SCF is considered converged when the wave function error is less that $10^{-\text{SCF\_CONVERGENCE}}$. Adjust the value of THRESH at the same time.

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations (including BSSE and XSAPT jobs).

7 For job types NMR, STATPOLAR, DYNPOLAR, HYPERPOLAR, and ISSC.

8 For most other job types, including geometry optimization, transition-state search, vibrational analysis, CIS/TDDFT calculations, correlated wavefunction methods, energy decomposition analysis (EDA2), etc.

OPTIONS:

User-defined

RECOMMENDATION:

Tighter criteria are recommended for geometry optimization and vibration analysis. Larger values provide more significant figures, at greater computational cost. Beginning with Q-CHEM 3.0, the DIIS error is measured by the maximum error rather than the RMS error that was used in earlier versions.

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE Closed-shell systems.

TRUE Open-shell systems.

OPTIONS:

FALSE Constrain the spatial part of the alpha and beta orbitals to be the same.

TRUE Do not Constrain the spatial part of the alpha and beta orbitals.

RECOMMENDATION:

Use the default unless ROHF is desired. Note that for unrestricted calculations on systems with an even number of electrons it is usually necessary to break α/β symmetry in the initial guess, by using SCF_GUESS_MIX or providing *$occupied* information (see Section 4.4 on initial guesses).

The calculations using other more special orbital types are controlled by the following *$rem* variables (they are not effective if GEN_SCFMAN = FALSE):

OS_ROSCF

Run an open-shell singlet ROSCF calculation with GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE OS_ROSCF calculation is performed.

FALSE Do not run OS_ROSCF (it will run a closed-shell RSCF calculation instead).

RECOMMENDATION:

Set to TRUE if desired.

ROSCF_DODS

Run a canonical restricted open-shell calculation (also called "ROSCF-DODS", where DODS stands for different orbitals for different spins) with GEN_SCFMAN. Available for high-spin half-filled open-shell systems.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a canonical ROSCF calculation.

FALSE Do not perform a canonical ROSCF calculation (run an open-shell USCF calculation instead).

RECOMMENDATION:

Set to TRUE if desired.

GHF

Run a generalized Hartree-Fock calculation with GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Run a GHF calculation.

FALSE Do not use GHF.

RECOMMENDATION:

Set to TRUE if desired.

COMPLEX

Run an SCF calculation with complex MOs using GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use complex orbitals.

FALSE Use real orbitals.

RECOMMENDATION:

Set to TRUE if desired.

COMPLEX_MIX

Mix a certain percentage of the real part of the HOMO to the imaginary part of the LUMO.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0–100 The mix angle = π·COMPLEX_MIX/100.

RECOMMENDATION:

It may help find the stable complex solution (similar idea as SCF_GUESS_MIX).

Example 4.1 Restricted open-shell singlet ROSCF calculation for the first excited state of formaldehyde using GEN_SCFMAN. The first job provides the guess orbitals through a restricted SCF calculation.

```
$molecule
   0 1
   H -0.940372  0.000000  1.268098
   H  0.940372  0.000000  1.268098
   C  0.000000  0.000000  0.682557
   O  0.000000  0.000000 -0.518752
$end

$rem
   GEN_SCFMAN        true
   METHOD            wb97x-d
   BASIS             def2-svpd
   THRESH            14
   SCF_CONVERGENCE   9
point_group_symmetry False
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE           sp
   METHOD            wb97x-d
   BASIS             def2-svpd
   GEN_SCFMAN        true
   OS_ROSCF          true
   THRESH            14
   SCF_CONVERGENCE   9
   SCF_ALGORITHM     diis
point_group_symmetry False
   SCF_GUESS         read
$end
```

4.3.3 Additional Options

Listed below are a number of useful options to customize an SCF calculation. This is only a short summary of the function of these *$rem* variables. A full list of all SCF-related variables is provided in Appendix B. Several important

sub-topics are discussed separately, including $\mathcal{O}(N)$ methods for large molecules (Section 4.6), customizing the initial guess (Section 4.4), and converging the SCF calculation (Section 4.5).

INTEGRALS_BUFFER

Controls the size of in-core integral storage buffer.

TYPE:

INTEGER

DEFAULT:

15 15 Megabytes.

OPTIONS:

User defined size.

RECOMMENDATION:

Use the default, or consult your systems administrator for hardware limits.

DIRECT_SCF

Controls direct SCF.

TYPE:

LOGICAL

DEFAULT:

Determined by program.

OPTIONS:

TRUE Forces direct SCF.

FALSE Do not use direct SCF.

RECOMMENDATION:

Use the default; direct SCF switches off in-core integrals.

METECO

Sets the threshold criteria for discarding shell-pairs.

TYPE:

INTEGER

DEFAULT:

2 Discard shell-pairs below $10^{-\mathrm{THRESH}}$.

OPTIONS:

1 Discard shell-pairs four orders of magnitude below machine precision.

2 Discard shell-pairs below $10^{-\mathrm{THRESH}}$.

RECOMMENDATION:

Use the default.

S2THRESH

Cutoff for neglect of overlap integrals, defined via a two-electron shell-pair threshold of $10^{-\mathrm{S2THRESH}}$ (S2THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

Same as THRESH.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

Increase the value of S2THRESH if the program finds negative eigenvalues for the overlap matrix.

THRESH

Cutoff for neglect of two electron integrals. $10^{-\mathrm{THRESH}}$ (THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

8 For single point energies.

10 For optimizations and frequency calculations.

14 For coupled-cluster calculations.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

The value should satisfy THRESH $\geq$ 3 + SCF_CONVERGENCE, although THRESH = 4 + SCF_CONVERGENCE is the default (in most cases) since Q-CHEM v. 6.0. See Ref. 36 for recommended values of THRESH in the presence of diffuse basis functions, where tighter thresholds are often required and too-loose thresholds may lead to slower convergence or convergence failure.

STABILITY_ANALYSIS

Performs stability analysis for a HF or DFT solution.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform stability analysis.

FALSE Do not perform stability analysis.

RECOMMENDATION:

Set to TRUE when a HF or DFT solution is suspected to be unstable.

SCF_PRINT

Controls level of output from SCF procedure to Q-CHEM output file.

TYPE:

INTEGER

DEFAULT:

0 Minimal, concise, useful and necessary output.

OPTIONS:

0 Minimal, concise, useful and necessary output.

1 Level 0 plus component breakdown of SCF electronic energy.

2 Level 1 plus density, Fock and MO matrices on each cycle.

3 Level 2 plus two-electron Fock matrix components (Coulomb, HF exchange , orbital kinetic energies, and DFT exchange-correlation matrices) on each cycle.

RECOMMENDATION:

Proceed with care; can result in *extremely* large output files at level 2 or higher. Output of all information is only available in scfman (GEN_SCFMAN = FALSE). If GEN_SCFMAN is set to TRUE and SCF_PRINT > 1, only level 1 plus MO matrices are available in the output. These levels are primarily for program debugging.

SCF_FINAL_PRINT

Controls level of output from SCF procedure to Q-CHEM output file at the end of the SCF.

TYPE:

INTEGER

DEFAULT:

0 No extra print out.

OPTIONS:

0 No extra print out.

1 Orbital energies and break-down of SCF energy.

2 Level 1 plus MOs and density matrices.

3 Level 2 plus Fock matrix.

RECOMMENDATION:

The break-down of energies is often useful (level 1).

4.3.4 Examples

Provided below are examples of Q-CHEM input files to run ground state, HF single point energy calculations.

Example 4.2 Example Q-CHEM input for a single point energy calculation on water. Note that the declaration of the single point *$rem* variable is redundant because it is the same as the Q-CHEM default.

```
$molecule
   0  1
   O
   H1  O  oh
   H2  O  oh  H1  hoh

   oh  =   1.2
   hoh = 120.0
$end

$rem
   JOBTYPE       sp       Single Point energy
   METHOD        hf       Hartree-Fock
   BASIS         sto-3g   Basis set
$end
```

Example 4.3 UHF/6-311G calculation on the Li atom. Note that correlation and the job type were not indicated because Q-CHEM defaults automatically to no correlation and single point energies. Note also that, since the number of α and β electron differ, MOs default to an unrestricted formalism.

```
$molecule
   0,2
   Li
$end

$rem
   METHOD     HF       Hartree-Fock
   BASIS      6-311G   Basis set
$end
```

Example 4.4 ROHF/6-311G calculation on the Lithium atom.

```
$molecule
   0,2
   3
$end

$rem
   METHOD         hf       Hartree-Fock
   UNRESTRICTED   false    Restricted MOs
   BASIS          6-311G   Basis set
$end
```

4.3.5 Symmetry

Symmetry is a powerful branch of mathematics and is often exploited in quantum chemistry, both to reduce the computational workload and to classify the final results obtained.[27,28,91] Q-CHEM is able to determine the point group symmetry of the molecular nuclei and, on completion of the SCF procedure, classify the symmetry of molecular orbitals, and provide symmetry decomposition of kinetic and nuclear attraction energy (see Chapter 10).

Molecular systems possessing point group symmetry offer the possibility of large savings of computational time, by avoiding calculations of integrals which are equivalent *i.e.*, those integrals which can be mapped on to one another under one of the symmetry operations of the molecular point group. The Q-CHEM default is to use symmetry to reduce computational time, when possible.

There are several keywords that are related to symmetry, which causes frequent confusion. POINT_GROUP_SYMMETRY controls symmetry throughout all modules. The default is TRUE. In some cases it may be desirable to turn off symmetry altogether, for example if you do not want Q-CHEM to reorient the molecule into the standard nuclear orientation, or if you want to turn it off for finite difference calculations. If the POINT_GROUP_SYMMETRY keyword is set to FALSE then the coordinates will not be altered from the input, and the point group will be set to C_1.

The INTEGRAL_SYMMETRY keyword controls symmetry in some integral routines. It is set to TRUE by default. Note that setting it to FALSE does not turn point group symmetry off, and does not disable symmetry in the coupled-cluster suite (CCMAN and CCMAN2), which is controlled by CC_SYMMETRY (see Chapters 6 and 7), although we noticed that sometimes it may interfere with the determination of orbital symmetries, possibly due to numerical noise. In some cases, INTEGRAL_SYMMETRY = TRUE can cause problems (poor convergence and wildly incorrect SCF energies) and turning it off can avoid these problems.

The symmetry is turned off when ghost atoms are used because their placement may break the symmetry of the molecular structure. However, the user can force calculation to use symmetry by FORCE_SYMMETRY_ON keyword; this is illustrated in example 7.65.

Note: The user should be aware about different conventions for defining symmetry elements. The arbitrariness affects, for example, C_{2v} point group. The specific choice affects how the irreducible representations in the affected groups are labeled. For example, b_1 and b_2 irreducible representations in C_{2v} are flipped when using different conventions. Q-CHEM uses non-Mulliken symmetry convention. See `https://iopenshell.usc.edu/resources/howto/symmetry` for detailed explanations.

INTEGRAL_SYMMETRY

Controls the efficiency through the use of point group symmetry for calculating integrals.

TYPE:

LOGICAL

DEFAULT:

False Do not use symmetry for computing integrals.

OPTIONS:

TRUE Use symmetry when available.

FALSE Do not use symmetry. This is always the case for RIMP2 jobs

RECOMMENDATION:

Use the default unless benchmarking. Note that symmetry usage is disabled for RIMP2, FFT, and QM/MM jobs.

POINT_GROUP_SYMMETRY

Controls whether or not Q-CHEM determines the point group of the molecule and reorients the molecule to the standard orientation.

TYPE:

LOGICAL

DEFAULT:

TRUE Do determine the point group (disabled for RIMP2 jobs).

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Use the default unless you do not want the molecule to be reoriented. Note that symmetry usage is disabled for RIMP2 jobs.

SYM_TOL

Controls the tolerance for determining point group symmetry. Differences in atom locations less than $10^{-\mathrm{SYM\_TOL}}$ are treated as zero.

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}.

OPTIONS:

User defined.

RECOMMENDATION:

Use the default unless the molecule has high symmetry which is not being correctly identified. Note that relaxing this tolerance too much may introduce errors into the calculation. Misidentified point group symmetry can affect rotational symmetry numbers in thermochemical calculations.

FORCE_SYMMETRY_ON

Overrides turning off symmetry in calculations using ghost atoms.

TYPE:

LOGICAL

DEFAULT:

FALSE Turn symmetry off when using ghost atoms.

OPTIONS:

TRUE Force symmetry.

FALSE Do not use symmetry.

RECOMMENDATION:

Use the default unless you know what you are doing.

4.4 SCF Initial Guess

4.4.1 Introduction

The Roothaan-Hall and Pople-Nesbet equations of SCF theory are non-linear in the molecular orbital coefficients. Like many mathematical problems involving non-linear equations, prior to the application of a technique to search for a numerical solution, an initial guess for the solution must be generated. If the guess is poor, the iterative procedure

applied to determine the numerical solutions may converge very slowly, requiring a large number of iterations, or at worst, the procedure may diverge.

Thus, in an *ab initio* SCF procedure, the quality of the initial guess is of utmost importance for (at least) two main reasons:

- To ensure that the SCF converges to an appropriate ground state. Often SCF calculations can converge to different local minima in wave function space, depending upon which part of "LCAO space" in which the initial guess lands.
- When considering jobs with many basis functions requiring the recalculation of ERIs at each iteration, using a good initial guess that is close to the final solution can reduce the total job time significantly by decreasing the number of SCF iterations.

For these reasons, sooner or later most users will find it helpful to have some understanding of the different options available for customizing the initial guess. Q-CHEM currently offers six options for the initial guess:

- Superposition of Atomic Densities (SAD)
- Superposition of Atomic Potentials (SAP, only available with GEN_SCFMAN = TRUE)
- On-the-fly (automated) Superposition of Atomic Densities (AUTOSAD)
- Purified SAD guess (provides molecular orbitals; SADMO)
- Core Hamiltonian (CORE)
- Generalized Wolfsberg-Helmholtz (GWH)
- Reading previously obtained MOs from disk. (READ)
- Basis set projection (BASIS2)

The first six of these guesses are built-in, and are briefly described in Section 4.4.2. The option of reading MOs from disk is described in Section 4.4.3. The initial guess MOs can be modified, either by mixing, or altering the order of occupation. These options are discussed in Section 4.4.4. Finally, Q-CHEM's novel basis set projection method is discussed in Section 4.4.5.

4.4.2 Initial Guess Types

Core Hamiltonian The core Hamiltonian guess simply obtains the guess MO coefficients by diagonalizing the core Hamiltonian matrix in Eq. (4.19). It is also commonly known as the one-electron guess, as it completely ignores interelectronic interactions. Although the guess is exact for one-electron systems, the lack of repulsion effects leads to incorrect shell structure of atoms as well as all electrons crowding onto the heaviest atom in the system; see Ref. 53 for a discussion. Due to these effects, the core guess is typically extremely inaccurate and should only be used as a last resort; much better alternatives are provided by the various SAD and SAP guesses.

Superposition of Atomic Densities (SAD) The SAD guess[56] is constructed by summing together pretabulated, spherically averaged atomic density matrices. The SAD guess generally yields robust convergence, and its use is particularly important when large basis sets and/or large molecules are employed. There are three issues associated with the SAD guess to be aware of:

1. No molecular orbitals are obtained, which means that SCF algorithms requiring orbitals (the direct minimization methods discussed in Section 4.5) cannot directly use the SAD guess. It can, however, be generated on-the-fly for general basis sets (BASIS = GEN), as described below, though not for mixed basis sets (BASIS = MIXED).

2. The SAD guess is not available for general (read-in) basis sets (pretabulated guesses exist for all internal basis sets); and
3. The SAD guess is not idempotent and thus requires *at least* two SCF iterations to ensure proper SCF convergence (idempotency of the density).

Purified Superposition of Atomic Densities (SADMO) The purified SAD guess (called "SADMO" in Ref. 53), is otherwise the same as the SAD guess except that it removes the issues 1 and 3 above. The SADMO guess obtains guess orbitals and corresponding occupation numbers by diagonalizing the non-idempotent SAD density matrix, after which an idempotent density matrix is recreated by aufbau occupation of the SAD natural orbitals. Since the initial density matrix is created with the SAD guess, the SADMO guess is not available for a general (read-in) basis set, either.

Superposition of Atomic Potentials (SAP) The SAP guess[53] is a major improvement on the core guess as it correctly describes atomic shell structure while retaining a simple form. The SAP guess introduces the interelectronic interactions missing from the core guess with a superposition of pretabulated atomic potentials, which have been derived with fully numerical calculations;[54,55] the atomic potentials used in Q-CHEM are derived from non-relativistic exchange-only LDA calculations employing spherically averaged densities.[55] As suggested in Ref. 53, the atomic potential matrix is evaluated through quadrature on a molecular grid analogous to the one used in DFT calculations; the grid is controlled by the *$rem* variable GUESS_GRID. Importantly, the SAP guess is noniterative, available for all elements in the periodic table from H to Og, and can be used with both internal and general (read-in) basis sets, thereby offering reasonably accurate initial guesses also in the case when the other options fail to work. Note SAP guess is not available in the old SCF code but only in GEN_SCFMAN.

On-the-fly (Automated) Superposition of Atomic Densities (AUTOSAD) In contrast to the SAD option that relies on pretabulated density matrices, the AUTOSAD guess provides a means of obtaining a method-specific SAD guess on-the-fly by running separate atomic calculations on all nonequivalent atoms in the system. As a SAD guess, the AUTOSAD density matrix is not idempotent and the guess will not produce molecular orbitals, so direct minimization methods cannot be directly used. Unlike the SAD option, AUTOSAD can be used for both internally defined and user-defined (general) basis sets, but is currently unavailable for mixed basis. Note that use of AUTOSAD is not necessary when using a single internal basis set with wave function methods, as in this case the AUTOSAD density is simply equivalent to the pretabulated SAD density. The pretabulated SAD guess is based on Hartree-Fock atomic densities whereas AUTOSAD uses the SCF method specified in *$rem*.

Generalized Wolfsberg-Helmholtz (GWH) The GWH guess procedure[106] uses a combination of the overlap matrix elements in Eq. (4.12), and the diagonal elements of the core Hamiltonian matrix in Eq. (4.19). This initial guess is usually even worse than the core Hamiltonian.[53] It is constructed according to

$$H_{\mu\upsilon} = c_x S_{\mu\upsilon}(H_{\mu\mu} + \tfrac{1}{2}H_{\upsilon\upsilon}) \; . \tag{4.29}$$

where c_x is a constant, typically $c_x = 1.75$.

The selection of these choices (or whether to read in the orbitals) is controlled by the following *$rem* variables:

SCF_GUESS

Specifies the initial guess procedure to use for the SCF.

TYPE:

STRING

DEFAULT:

SAD	Superposition of atomic densities[56] (default for internal basis sets)
AUTOSAD	For internally defined or user-customized general basis sets or mixed basis
GWH	For ROHF jobs with GEN_SCFMAN = FALSE which require a set of orbitals
FRAGMO	For fragment jobs such as ALMO-based calculations
CORE	For special cases that currently can't be handled by the ones above (e.g. mixed basis with ghost atoms)

OPTIONS:

CORE	Diagonalize core Hamiltonian
SAD	Superposition of atomic density[56]
SAP	Superposition of atomic potentials[53] (only available with GEN_SCFMAN = TRUE)
AUTOSAD	On-the-fly superposition of atomic densities
SADMO	Purified superposition of atomic densities (available only with standard basis sets)
GWH	Apply generalized Wolfsberg-Helmholtz approximation
READ	Read previous MOs from disk
FRAGMO	Superimposing converged fragment MOs (see Section 12.3)

RECOMMENDATION:

SAD, AUTOSAD, or SADMO guess for standard basis sets. For either standard or user-customized general basis sets, AUTOSAD is recommended and used as default. If these options fail, use the SAP guess; try the GWH or core Hamiltonian guess only as a last resort. For mixed basis sets, only the AUTOSAD, SAP, GWH, and core Hamiltonian guesses are currently available. For ROHF it can be useful to READ guesses from an SCF calculation on the corresponding cation or anion. Note that because the density is made spherical, this may favor an undesired state for atomic systems, especially transition metals. Use FRAGMO in a fragment MO calculation.

SCF_GUESS_ALWAYS

Switch to force the regeneration of a new initial guess for each series of SCF iterations (for use in geometry optimization).

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False	Do not generate a new guess for each series of SCF iterations in an optimization; use MOs from the previous SCF calculation for the guess, if available.
True	Generate a new guess for each series of SCF iterations in a geometry optimization.

RECOMMENDATION:

Use the default unless SCF convergence issues arise

GUESS_GRID

Specifies the type of grid to use for SAP guess generation. The options are the same as those of the *$rem* variable XC_GRID.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0	Use SG-0 for H, C, N, and O; SG-1 for all other atoms.
n	Use SG-n for all atoms, $n = 1, 2$, or 3
XY	A string of two six-digit integers X and Y, where X is the number of radial points and Y is the number of angular points where possible numbers of Lebedev angular points, which must be an allowed value from Table 5.2 in Section 5.5.
$-XY$	Similar format for Gauss-Legendre grids, with the six-digit integer X corresponding to the number of radial points and the six-digit integer Y providing the number of Gauss-Legendre angular points, $Y = 2N^2$.

RECOMMENDATION:

Larger grids may be required if the SAP guess is poor.

4.4.3 Reading MOs from Disk

There are two methods by which MO coefficients can be used from a previous job by reading them from disk:

1. Running two independent jobs sequentially invoking Q-CHEM with three command line variables:.

```
localhost-1> qchem job1.in job1.out save
localhost-2> qchem job2.in job2.out save
```

Note: (1) The *$rem* variable SCF_GUESS must be set to READ in `job2.in`.
(2) Scratch files remain in `$QCSCRATCH/save` on exit.

2. Running a batch job where two jobs are placed into a single input file separated by the string `@@@` on a single line.
Note: (1) SCF_GUESS must be set to READ in the second job of the batch file.
(2) A third Q-CHEM command line variable is not necessary.
(3) As for the SAD guess, Q-CHEM requires at least two SCF cycles to ensure proper SCF convergence (idempotency of the density).

Note: It is up to the user to make sure that the basis sets match between the two jobs. There is no internal checking for this, although the occupied orbitals are re-orthogonalized in the current basis after being read in. If you want to project from a smaller basis into a larger basis, consult Section 4.4.5.

Example 4.5 Input for an ROHF calculation on the OH radical. One SCF cycle is initially performed on the cation, to get reasonably good initial guess orbitals, which are then read in as the guess for the radical. This avoids the use of Q-CHEM's default GWH guess for ROHF, which is often poor.

```
$comment
   OH radical, part 1. Do initial iterations of cation orbitals.
$end

$molecule
   1  1
   O  0.000   0.000   0.000
   H  0.000   0.000   1.000
$end

$rem
   BASIS           =  6-311++G(2df)
   METHOD          =  hf
   THRESH          =  10
   MAX_SCF_CYCLES  =  2
   SCF_CONVERGENCE =  1
$end

@@@

$comment
   OH radical, part 2. Read cation orbitals, do the radical
$end

$molecule
   0  2
   O  0.000   0.000   0.000
   H  0.000   0.000   1.000
$end

$rem
   BASIS            =  6-311++G(2df)
   METHOD           =  hf
   UNRESTRICTED     =  false
   SCF_ALGORITHM    =  dm
   SCF_CONVERGENCE  =  7
   SCF_GUESS        =  read
   THRESH           =  10
$end
```

4.4.4 Modifying the Occupied Molecular Orbitals

It is sometimes useful for the occupied guess orbitals to be other than the lowest N_α (or N_β) orbitals. Reasons why one may need to do this include:

- To converge to a state of different symmetry or orbital occupation.
- To break spatial symmetry.
- To break spin symmetry, as in unrestricted calculations on molecules with an even number of electrons.

There are two mechanisms for modifying a set of guess orbitals: either by SCF_GUESS_MIX, or by specifying the orbitals to occupy. Q-CHEM users may define the occupied guess orbitals using the *$occupied* or *$swap_occupied_virtual* keywords. In the former, occupied guess orbitals are defined by listing the α orbitals to be occupied on the first line and β on the second. In the former, only pair of orbitals that needs to be swapped is specified.

Note: (1) To prevent Q-CHEM to change orbital occupation during SCF procedure, MOM_START option is often used in combination with *$occupied* or *$swap_occupied_virtual* keywords.
(2) The need for orbitals renders these options incompatible with the SAD guess. Most often, they are used with SCF_GUESS = READ.

The format for modifying occupied guess orbitals is as follows:

```
$occupied
   1  2  3  4 ...  NAlpha
   1  2  3  4 ...  NBeta
$end
```

or alternatively

```
$swap_occupied_virtual
   <spin> <io1> <iv1>
   <spin> <io2> <iv2>
$end
```

Occupied and virtual orbitals can also be swapped

```
$swap_occupied_virtual
   alpha  5  6
   beta   6  7
$end
```

This is identical to

```
$occupied
   1 2 3 4 6 5 7
   1 2 3 4 5 7 6
$end
```

or

```
$occupied
   1:4 6 5 7
   1:5 7 6
$end
```

The other *$rem* variables related to altering the orbital occupancies are:

SCF_GUESS_PRINT

Controls printing of guess MOs, Fock and density matrices.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not print guesses.
SAD	
1	Atomic density matrices and molecular matrix.
2	Level 1 plus density matrices.
CORE and GWH	
1	No extra output.
2	Level 1 plus Fock and density matrices and, MO coefficients and eigenvalues.
READ	
1	No extra output
2	Level 1 plus density matrices, MO coefficients and eigenvalues.

RECOMMENDATION:

None

SCF_GUESS_MIX

Controls mixing of LUMO and HOMO to break symmetry in the initial guess. For unrestricted jobs, the mixing is performed only for the alpha orbitals.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)	Do not mix HOMO and LUMO in SCF guess.

OPTIONS:

0 (FALSE)	Do not mix HOMO and LUMO in SCF guess.
1 (TRUE)	Add 10% of LUMO to HOMO to break symmetry.
n	Add $n \times 10\%$ of LUMO to HOMO ($0 < n < 10$).

RECOMMENDATION:

When performing unrestricted calculations on molecules with an even number of electrons, it is often necessary to break alpha/beta symmetry in the initial guess with this option, or by specifying input for *$occupied*.

Example 4.6 Input for an unrestricted HF calculation on H_2 in the dissociation limit, showing the use of SCF_GUESS_MIX = 2 (corresponding to 20% of the alpha LUMO mixed with the alpha HOMO). Geometric direct minimization with DIIS is used to converge the SCF, together with MAX_DIIS_CYCLES = 1 (using the default value for MAX_DIIS_CYCLES, the DIIS procedure just oscillates).

```
$molecule
   0  1
   H  0.000   0.000   0.0
   H  0.000   0.000 -10.0
$end

$rem
   METHOD           =  hf
   BASIS            =  6-31g**
   UNRESTRICTED     =  true
   SCF_ALGORITHM    =  diis_gdm
   MAX_DIIS_CYCLES  =  1
   SCF_GUESS        =  gwh
   SCF_GUESS_MIX    =  2
$end
```

4.4.5 Basis Set Projection

Q-CHEM also includes a novel basis set projection method developed by Dr Jing Kong of Q-CHEM Inc. It permits a calculation in a large basis set to bootstrap itself up *via* a calculation in a small basis set that is automatically spawned when the user requests this option. When basis set projection is requested (by providing a valid small basis for BASIS2), the program executes the following steps:

- A simple DFT calculation is performed in the small basis, BASIS2, yielding a converged density matrix in this basis.
- The large basis set SCF calculation (with different values of EXCHANGE and CORRELATION set by the input) begins by constructing the DFT Fock operator in the large basis but with the density matrix obtained from the small basis set.
- By diagonalizing this matrix, an accurate initial guess for the density matrix in the large basis is obtained, and the target SCF calculation commences.

Two different methods of projection are available and can be set using the BASISPROJTYPE *$rem*. The OVPROJECTION option expands the MOs from the BASIS2 calculation in the larger basis, while the FOPPROJECTION option constructs the Fock matrix in the larger basis using the density matrix from the initial, smaller basis set calculation. Basis set projection is a very effective option for general basis sets, where the SAD guess is not available. In detail, this initial guess is controlled by the following *$rem* variables:

BASIS2

Sets the small basis set to use in basis set projection.

TYPE:

STRING

DEFAULT:

No second basis set default.

OPTIONS:

Symbol. Use standard basis sets as per Chapter 8.

BASIS2_GEN	General BASIS2
BASIS2_MIXED	Mixed BASIS2

RECOMMENDATION:

BASIS2 should be smaller than BASIS. There is little advantage to using a basis larger than a minimal basis when BASIS2 is used for initial guess purposes. Larger, standardized BASIS2 options are available for dual-basis calculations (see Section 4.7).

BASISPROJTYPE

Determines which method to use when projecting the density matrix of BASIS2

TYPE:

STRING

DEFAULT:

OVPROJECTION

OPTIONS:

FOPPROJECTION	Construct the Fock matrix in the second basis
OVPROJECTION	Projects MOs from BASIS2 to BASIS.

RECOMMENDATION:

None

Note: BASIS2 sometimes affects post-Hartree-Fock calculations. It is recommended to split such jobs into two subsequent one, such that in the first job a desired Hartree-Fock solution is found using BASIS2, and in the second job, which performs a post-HF calculation, SCF_GUESS = READ is invoked.

Example 4.7 Input where basis set projection is used to generate a good initial guess for a calculation employing a general basis set, for which the default initial guess is not available.

```
$molecule
   0  1
   O
   H  1  r
   H  1  r  2  a

   r    0.9
   a  104.0
$end

$rem
   METHOD        mp2
   BASIS         general
   BASIS2        sto-3g
$end

$basis
   O   0
   S   3   1.000000
           3.22037000E+02   5.92394000E-02
           4.84308000E+01   3.51500000E-01
           1.04206000E+01   7.07658000E-01
   SP  2   1.000000
           7.40294000E+00  -4.04453000E-01   2.44586000E-01
           1.57620000E+00   1.22156000E+00   8.53955000E-01
   SP  1   1.000000
           3.73684000E-01   1.00000000E+00   1.00000000E+00
   SP  1   1.000000
           8.45000000E-02   1.00000000E+00   1.00000000E+00
****
   H   0
   S   2   1.000000
           5.44717800E+00   1.56285000E-01
           8.24547000E-01   9.04691000E-01
   S   1   1.000000
           1.83192000E-01   1.00000000E+00
****
$end
```

4.5 Converging SCF Calculations

4.5.1 Introduction

As for any numerical optimization procedure, the rate of convergence of the SCF procedure is dependent on the initial guess and on the algorithm used to step towards the stationary point. Q-CHEM features a number of SCF optimization algorithms which can be selected via the *$rem* variable SCF_ALGORITHM, including:

Methods that are based on extrapolation or interpolation:

- The highly successful DIIS procedures. These are the default (except for restricted open-shell SCF calculations)

and are available for all orbital types (see Section 4.5.3). The damping[40] and level-shifting technique[38,44,74] can also be invoked together with DIIS (R, U only).

- ADIIS: the combination of augmented Roothaan-Hall (ARH) energy function and DIIS developed by Hu and Yang,[46] which isavailable for R and U only.

Methods that make use of orbital gradient:

- Direct Minimization (DM), which has been re-implemented as simple steepest descent with line search, and is available for all orbital types. DM can be invoked after a few DIIS iterations.
- Limited-memory Broyden-Fletcher-Goldfarb-Shanno algorithm (L_BFGS), which is a quasi-Newton method using the gradients and steps taken in the previous iterations to construct the approximate Hessian.
- Geometric Direct Minimization (GDM) which is an improved and highly robust version of DM and is the recommended fall-back when DIIS fails. Like DM, It can also be invoked after a few iterations with DIIS to improve the initial guess. GDM is the default algorithm for restricted open-shell SCF calculations and is available for all orbital types (see Section 4.5.7). GDM comes in three variants:
  - GDM: the default implementation.
  - GDM_LS: Uses a line search at every iteration.
  - GDM_QLS: Employs a quadratic energy model to perform the line search at every iteration.
- Square Gradient Minimization (SGM) family: SGM is a GDM-inspired method for excited state orbital optimization. Currently, three variants of this approach are supported: SGM (for RO and and OS_RO), SGM_LS (for R, U, RO and OS_RO) and SGM_QLS (for R and U), which are based upon the corresponding GDM approaches. For further details, see Section 4.5.15.

Methods that require orbital Hessian:

- NEWTON_CG/NEWTON_MINRES (solve $\mathbf{Hd} = -\mathbf{g}$ for the update direction with CG/MINRES solvers).
- SF_NEWTON_CG (the “saddle-free" version of NEWTON_CG).

The analytical orbital Hessian is available for R/U/RO/G/CR unless special density functionals (*e.g.*, those nonlocal functionals except for VV10[95]) are used, while the use of finite-difference Hessian is available for all orbital types by setting FD_MAT_VEC_PROD = TRUE.

In addition to these algorithms, there is also the maximum overlap method (MOM) which ensures that DIIS always occupies a continuous set of orbitals and does not oscillate between different occupancies. MOM can also be used to obtain higher-energy solutions of the SCF equations (see Section 7.6). The relaxed constraint algorithm (RCA), which guarantees that the energy goes down at every step, is also available via the old SCF code (set GEN_SCFMAN = FALSE). Nevertheless, the performance of the ADIIS[46] algorithm should be similar to it.

Since the code in GEN_SCFMAN is highly modular, the availability of different SCF algorithms to different SCF (orbital) types is largely extended in general. For example, the old ROSCF implementation requires the use of the GWH guess and the GDM algorithm exclusively. Such a limitation has been eliminated in GEN_SCFMAN based RO calculations.

4.5.2 Basic Convergence Control Options

See also more detailed options in the following sections, and note that the SCF convergence criterion and the integral threshold must be set in a compatible manner, (this usually means THRESH should be set to at least 3 higher than SCF_CONVERGENCE).

MAX_SCF_CYCLES

Controls the maximum number of SCF iterations permitted.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

Increase for slowly converging systems such as those containing transition metals.

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

DIIS Pulay DIIS.

OPTIONS:

DIIS	Pulay DIIS.
ROOTHAAN	Roothaan repeated diagonalization
DM	Direct minimizer.
L_BFGS	Limited memory BFGS algorithm with line search
GDM	Geometric Direct Minimization
GDM_LS	Like GDM, but employs a line search in L-BFGS
GDM_QLS	Like GDM_LS, but uses a quadratic line search in L-BFGS
RCA	Relaxed constraint algorithm
ADIIS	The combination of the ARH energy function and DIIS by Hu and Yang
NEWTON_CG	Solve $\mathbf{Hd} = -\mathbf{g}$ using conjugated gradients to determine Newton steps (see also Newton_MINRES, which uses the MINRES algorithm to solve the linear equation)
DIIS_DM	Uses DIIS initially, switching to direct minimizer for later iterations (See THRESH_DIIS_SWITCH, MAX_DIIS_CYCLES)
DIIS_GDM	Use DIIS and then later switch to geometric direct minimization (See THRESH_DIIS_SWITCH, MAX_DIIS_CYCLES)
LS_DIIS	Uses level-shifting initially, switching to DIIS for later iterations (See THRESH_LS_SWITCH, MAX_LS_CYCLES)
RCA_DIIS	Use RCA initially, switching to DIIS for later iterations (See THRESH_RCA_SWITCH and MAX_RCA_CYCLES)
ADIIS_DIIS	Use ADIIS initially, switching to DIIS for later iterations (See THRESH_ADIIS_SWITCH and MAX_ADIIS_CYCLES)
ROBUST	New black-box workflow for robust convergence featuring tighter thresholds and a combination of algorithms (DIIS, ADIIS, and GDM).
ROBUST_STABLE	Does stability analysis on top of the ROBUST procedure.

RECOMMENDATION:

Use DIIS unless performing a restricted open-shell calculation, in which case GDM is recommended. If DIIS fails to find a reasonable approximate solution in the initial iterations, RCA_DIIS and ADIIS_DIIS are the recommended fallback options. If DIIS approaches the correct solution but fails to finally converge, DIIS_GDM is the recommended fallback. For systems with small HOMO-LUMO gaps and DIIS fails to converge, LS_DIIS could help. Use ROBUST or ROBUST_STABLE (see 4.5.13) if nothing works; please report failed SCF convergence to the Q-Chem Office to help refine these workflows.

Note: for the usage of Square Gradient Minimization (SGM) and related algorithm options, see the documentation in Section 4.5.15.

SCF_CONVERGENCE

SCF is considered converged when the wave function error is less that $10^{-\text{SCF\_CONVERGENCE}}$. Adjust the value of THRESH at the same time. Note as of Q-CHEM 3.0 the DIIS error is measured by the maximum error rather than the RMS error.

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations (including BSSE and XSAPT jobs)

7 For job types NMR, STATPOLAR, DYNPOLAR, HYPERPOLAR, and ISSC

8 For most other job types, including geometry optimization, transition-state search, vibrational analysis, CIS/TDDFT calculations, correlated wavefunction methods, energy decomposition analysis (EDA2), etc.

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

Tighter criteria for geometry optimization and vibration analysis. Larger values provide more significant figures, at greater computational cost.

In some cases besides the total SCF energy, one needs its separate energy components, like kinetic energy, exchange energy, correlation energy, *etc.* The values of these components are printed at each SCF cycle if one specifies SCF_PRINT = 1 in the input.

4.5.3 Direct Inversion in the Iterative Subspace (DIIS)

The SCF implementation of the Direct Inversion in the Iterative Subspace (DIIS) method[70,71] uses the property of an SCF solution that requires the density matrix to commute with the Fock matrix:

$$\mathbf{SPF} - \mathbf{FPS} = \mathbf{0} \,. \tag{4.30}$$

During the SCF cycles, prior to achieving self-consistency, it is therefore possible to define an error vector $\mathbf{e}_i$, which is non-zero except at convergence:

$$\mathbf{SP}_i\mathbf{F}_i - \mathbf{F}_i\mathbf{P}_i\mathbf{S} = \mathbf{e}_i \tag{4.31}$$

Here $\mathbf{P}_i$ is obtained by diagonalizing $\mathbf{F}_i$, and

$$\mathbf{F}_k = \sum_{j=1}^{k-1} c_j \mathbf{F}_j \tag{4.32}$$

The DIIS coefficients c_k, are obtained by a least-squares constrained minimization of the error vectors, *viz*

$$Z = \left(\sum_k c_k \mathbf{e}_k \right) \cdot \left(\sum_k c_k \mathbf{e}_k \right) \tag{4.33}$$

where the constraint $\sum_k c_k = 1$ is imposed to yield a set of linear equations, of dimension $N + 1$:

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_1 \cdot \mathbf{e}_N & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{e}_N \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_N \cdot \mathbf{e}_N & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_N \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \,. \tag{4.34}$$

Convergence criteria require the largest element of the Nth error vector to be below a cutoff threshold, usually 10^{-5} a.u. for single point energies, but often increased to 10^{-8} a.u. for optimizations and frequency calculations.

The rate of convergence may be improved by restricting the number of previous Fock matrices used for determining the DIIS coefficients,

$$\mathbf{F}_k = \sum_{j=k-(L+1)}^{k-1} c_j \mathbf{F}_j \ . \tag{4.35}$$

Here L is the size of the DIIS subspace, which is set using the *$rem* variable DIIS_SUBSPACE_SIZE. As the Fock matrix nears self-consistency, the linear matrix equations in Eq. (4.34) tend to become severely ill-conditioned and it is often necessary to reset the DIIS subspace (this is automatically carried out by the program).

Finally, on a practical note, we observe that DIIS has a tendency to converge to global minima rather than local minima when employed for SCF calculations. This seems to be because only at convergence is the density matrix in the DIIS iterations idempotent. On the way to convergence, one is not on the true energy surface, and this seems to permit DIIS to "tunnel" through barriers in wave function space. This is usually a desirable property, and is the motivation for the options that permit initial DIIS iterations before switching to direct minimization to converge to the minimum in difficult cases.

The following *$rem* variables permit some customization of the DIIS iterations:

DIIS_SUBSPACE_SIZE

Controls the size of the DIIS and/or RCA subspace during the SCF.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

User-defined

RECOMMENDATION:

None

DIIS_PRINT

Controls the output from DIIS SCF optimization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Minimal print out.

1 Chosen method and DIIS coefficients and solutions.

2 Level 1 plus changes in multipole moments.

3 Level 2 plus Multipole moments.

4 Level 3 plus extrapolated Fock matrices.

RECOMMENDATION:

Use the default

Note: In Q-CHEM 3.0 the DIIS error is determined by the maximum error rather than the RMS error. For backward compatibility the RMS error can be forced by using the following *$rem*:

DIIS_ERR_RMS

Changes the DIIS convergence metric from the maximum to the RMS error.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE, FALSE

RECOMMENDATION:

Use the default, the maximum error provides a more reliable criterion.

DIIS_SEPARATE_ERRVEC

Control optimization of DIIS error vector in unrestricted calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE Use a combined α and β error vector.

OPTIONS:

FALSE Use a combined α and β error vector.

TRUE Use separate error vectors for the α and β spaces.

RECOMMENDATION:

When using DIIS in Q-CHEM a convenient optimization for unrestricted calculations is to sum the α and β error vectors into a single vector which is used for extrapolation. This is often extremely effective, but in some pathological systems with symmetry breaking, can lead to false solutions being detected, where the α and β components of the error vector cancel exactly giving a zero DIIS error. While an extremely uncommon occurrence, if it is suspected, set DIIS_SEPARATE_ERRVEC = TRUE to check.

4.5.4 Damping

Damping may be the oldest SCF acceleration scheme which was proposed by Hartree in the early days of applying quantum mechanics to study atomic structure.[40] In this simple scheme, the density matrix (or Fock matrix) of the current SCF iteration is linearly mixed with the density matrix of the previous iteration to generate a damped density matrix as the input for the next SCF iteration:

$$P_n^{\text{damped}} = (1-\alpha)P_n + \alpha P_{n-1}, \tag{4.36}$$

where α is the mixing factor with $0 \leqslant \alpha \leqslant 1$. During the SCF process, if density matrix changes drastically between consecutive iterations (usually this happens in the early stage of the SCF process), the total energy and occupied molecular orbitals are also strongly fluctuating, which may lead to slow SCF convergence or even divergence. In this scenario mixing the density (or Fock) matrix with its counterpart in the last iteration could reduce (damp) the energy and molecular orbital fluctuation and stabilize the SCF process. On the other hand, if the SCF process converges smoothly and quickly, apply damping would only slow down its convergence. Therefore damping is seldom applied solely in the full SCF process and often invoked only in the early stage of the SCF process and turned off later.

Density matrix damping is offered as an option (the DAMP algorithm) to handle difficult SCF cases in Q-CHEM (RHF and UHF only). Damping is often combined with DIIS and GDM (the DP_DIIS and DP_GDM algorithms) in practical calculations. In order to invoke damping, the *$rem* variable SCF_ALGORITHM should be set to DAMP, DP_DIIS or DP_GDM. The other relevant *$rem* variables are as the following:

NDAMP

Determine the mixing coefficient. α = NDAMP/100.

TYPE:

INTEGER

DEFAULT:

75

OPTIONS:

User-defined. Integers between 0 and 100.

RECOMMENDATION:

Increase NDAMP if strong fluctuations happen during the SCF process.

MAX_DP_CYCLES

The maximum number of SCF iterations with damping when SCF_ALGORITHM = DP_DIIS and DP_GDM. See also THRESH_DP_SWITCH.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

1 Only a single SCF step with damping, and no damping for the remaining SCF steps.

n n SCF iterations with damping before turning damping off.

RECOMMENDATION:

Increase this number if strong fluctuation continues after damping is turned off.

THRESH_DP_SWITCH

The threshold for turning off damping in SCF iterations is $10^{-\text{THRESH\_DP\_SWITCH}}$ when SCF_ALGORITHM is set to DP_DIIS or DP_GDM. See also MAX_DP_CYCLES.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined.

RECOMMENDATION:

None

Example 4.8 Input for a UKS calculation of water cation using DIIS after damping in the early stage of the SCF process.

```
$molecule
  1 2
   O    1.9158048   -5.3106212    3.9451654
   H    2.8858048   -5.3106212    3.9451654
   H    1.5924750   -5.6945720    3.1151415
$end

$rem
   METHOD              B3LYP
   BASIS               3-21G
   SCF_ALGORITHM       DP_DIIS
   THRESH_DP_SWITCH    3
   MAX_DP_CYCLES       20
   NDAMP               50
$end
```

4.5.5 Level-Shifting

Level-shifting is an old technique that may facilitate SCF convergence in systems having small HOMO/LUMO gaps.[38,44,74] If the gap is small, a simple Fock matrix diagonalization (Roothaan step) may alter the energetic ordering of the molecular orbitals, so that after re-populating the electrons according to the *aufbau* principle, the overall effect is a discontinuous switch in the electron configuration, and the SCF process fails to converge. To remedy this fluctuating SCF behavior, one can shift the diagonal elements of the virtual block of the Fock matrix ("level-shifting") to increase the calculated HOMO/LUMO gap before diagonalization. With proper level-shifting, the energetic ordering of the molecular orbitals is preserved during diagonalization and thus the shapes of the orbitals are changed in a continuous way at each SCF cycle, leading to a stable iterative process. Using perturbation theory, one can show that a proper level shift guarantees that the total energy is lowered after Fock matrix diagonalization.[44,74] It is important to note, however, that SCF solutions obtained via level-shifting are not necessarily stable ground states. To check the stability of the converged electronic state, we recommend invoking the keywords STABILITY_ANALYSIS or INTERNAL_STABILITY (see Section 4.3.3 and Section 4.5.17).

In cases where DIIS or some other SCF algorithm converges quickly, application of a level shift usually slows down convergence. In addition, experiences show that level-shifting can converge difficult SCF cases to moderate thresholds such as 10^{-5}, but becomes less efficient as the convergence threshold is tightened, say, to 10^{-8}. To obtain tightly-converged solutions, level-shifting should be combined with DIIS or another convergence algorithm. For difficult SCF cases, invoking level-shifting in the early SCF iterations and then turning it off later (in favor of DIIS, say) is often the best strategy. As such, Q-CHEM offers a hybrid SCF algorithm that combines level-shifting with DIIS. This is invoked by setting SCF_ALGORITHM = LS_DIIS. Level-shifting can also be used on its own (without DIIS), by means of the following *$rem* variables.

LEVEL_SHIFT

Determine whether to invoke level-shifting or not together with DIIS.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Apply level shifting.

FALSE Do not apply level shifting.

RECOMMENDATION:

Use TRUE if level-shifting is necessary to accelerate SCF convergence.

GAP_TOL

HOMO/LUMO gap threshold to control whether to shift the diagonal elements of the virtual block of the Fock matrix or not. If the HOMO/LUMO gap is less than this threshold, at a given SCF iteration, then the diagonal elements of the virtual block of the Fock matrix are shifted. Otherwise no level-shift is applied.

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

User-defined

RECOMMENDATION:

The input number must be an integer between 0 and 9999. The threshold applied is set to GAP_TOL/1000 Eh. The default value is provided to make the level-shifting calculation run and should not be taken as optimal for any specific problem. Trial and error may be required to find the optimal threshold. Larger values of GAP_TOL generally lead to level-shifting being used more frequently during the SCF convergence process.

LSHIFT

Constant shift applied to all diagonal elements of the virtual block of the Fock matrix.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

User-defined

RECOMMENDATION:

The input number must be an integer between 0 and 9999. The actual shift is equal to LSHIFT/1000 Eh. The default value is provided to make the level-shifting calculation run and should not be taken as optimal for any specific problem. Trial and error may be required to find the optimal threshold. Larger level shifts make the SCF process more stable but also slow down convergence, thus requiring more SCF cycles.

Note: If either GAP_TOL or LSHIFT or both are explicitly specified, then LEVEL_SHIFT is automatically set to TRUE. Setting LEVEL_SHIFT = FALSE disables any values of GAP_TOL and LSHIFT. Invoking the LS_DIIS algorithm also disables any setting of LEVEL_SHIFT.

MAX_LS_CYCLES

The maximum number of DIIS iterations with level-shifting when SCF_ALGORITHM = LS_DIIS. See also THRESH_LS_SWITCH.

TYPE:

INTEGER

DEFAULT:

MAX_SCF_CYCLES

OPTIONS:

1 Only a single DIIS step with level-shifting, and no level-shifting for the remaining DIIS steps.

n n DIIS iterations with level-shifting before turning level-shifting off.

RECOMMENDATION:

None

THRESH_LS_SWITCH

The threshold for turning off level-shifting in DIIS is $10^{-\text{THRESH\_LS\_SWITCH}}$ when SCF_ALGORITHM is set to LS_DIIS. See also MAX_LS_CYCLES.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

User-defined.

RECOMMENDATION:

None

Example 4.9 Input for a RKS calculation using DIIS with level-shifting on a uranium compound.

```
$molecule
0 1
  U   -0.7734808   -0.8815596   -0.8853446
  O   -1.3090665   -2.1863261   -2.7399692
  O   -1.6134743    1.0032462   -1.9673881
  O   -0.2537507    0.4215612    0.9749395
  O    0.0643962   -2.7662217    0.1985884
  O   -2.4384926   -1.2003830   -0.1700214
  O    0.8915310   -0.5627363   -1.6006679
  H   -0.5266214   -2.1731728   -3.3131242
  H   -2.5519330    1.0551966   -1.7276040
  H    0.6520521    0.7395638    0.8360306
  H   -0.6807322   -3.1486006    0.6879451
$end

$rem
   METHOD          = B3LYP
   BASIS           = LANL2DZ
   ECP             = fit-LANL2DZ
   MAX_SCF_CYCLES  = 200
   THRESH          = 10
   LEVEL_SHIFT     = TRUE
   GAP_TOL         = 200
   LSHIFT          = 200
   SCF_CONVERGENCE = 5
$end
```

Example 4.10 Input for a UKS calculation using LS_DIIS on a cobalt compound.

```
$molecule
   6 2
   O    0.7515076    1.2954050    1.0605230
   O    0.7506760   -1.2982554    1.0594277
   Co   0.0016554   -0.0007951    0.0017848
   O   -1.4949030   -0.0008880    1.0616115
   O    1.4981395   -0.0006773   -1.0578583
   O   -0.7482665   -1.2970503   -1.0566523
   O   -0.7473745    1.2963844   -1.0559284
$end

$rem
   METHOD             = B3LYP
   BASIS              = 6-31G
   SCF_ALGORITHM      = LS_DIIS
   MAX_SCF_CYCLES     = 200
   GUI                = 2
   GAP_TOL            = 100
   LSHIFT             = 200
   SCF_CONVERGENCE    = 8
   STABILITY_ANALYSIS = TRUE
$end
```

4.5.6 Pseudo-Fractional Occupation Number Method (pFON)

An alternative to level-shifting for cases exhibiting small (or zero) HOMO/LUMO gaps is the pseudo-fraction occupation number (pFON) approach,[72] which corresponds to a "smearing out" of the occupation numbers at the HOMO

level. Often, this improves the stability and accelerates the convergence by eliminating the discontinuous occupancy changes (from one SCF iteration to the next) that can arise in small-gap systems. Essentially, more than one electron configuration is allowed during the same orbital optimization, with fractional occupancies. This is formally equivalent to a finite-temperature formalism.

The pFON method introduces a density matrix

$$P_{\mu\nu} = \sum_{p=1}^{N} n_p C_{\mu p} C_{\nu p} \tag{4.37}$$

with occupancies $0 \leq n_p \leq 1$ that can be fractional, whereas for a conventional SCF calculation either $n_p = 1$ (occupied) or $n_p = 0$ (virtual). In pFON, the occupation numbers follow a Fermi-Dirac distribution,

$$n_p = \left(1 + e^{(\epsilon_p - \epsilon_\mathrm{F})/kT}\right)^{-1} , \tag{4.38}$$

where ϵ_p is an SCF eigenvalue (orbital energy) and T is a temperature. In Q-CHEM's implementation, the Fermi energy is set to $\epsilon_\mathrm{F} = (\epsilon_\mathrm{HOMO} + \epsilon_\mathrm{LUMO})/2$. To ensure conservation of the total number of electrons, the pFON approach re-scales the occupation numbers so that $\sum_p n_p = N_\mathrm{el}$.

There are several parameters to control the electronic temperature T throughout a pFON SCF run. The temperature can either be held constant at finite temperature ($T_\mathrm{init} = T_\mathrm{final}$), or the system can be cooled from a higher temperature down to the final temperature. So far, no zero-temperature extrapolation has been implemented.

OCCUPATIONS

Activates pFON calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Integer occupation numbers

1 Not yet implemented

2 Pseudo-fractional occupation numbers (pFON)

RECOMMENDATION:

Use pFON to improve convergence for small-gap systems.

FON_T_START

Initial electronic temperature (in K) for FON calculation.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

Any desired initial temperature.

RECOMMENDATION:

Pick the temperature to either reproduce experimental conditions (*e.g.* room temperature) or as low as possible to approach zero-temperature.

FON_T_END

Final electronic temperature for FON calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

Any desired final temperature.

RECOMMENDATION:

Pick the temperature to either reproduce experimental conditions (*e.g.* room temperature) or as low as possible to approach zero-temperature.

FON_NORB

Number of orbitals above and below the Fermi level that are allowed to have fractional occupancies.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n number of active orbitals

RECOMMENDATION:

The number of valence orbitals is a reasonable choice.

FON_T_SCALE

Determines the step size for the cooling.

TYPE:

INTEGER

DEFAULT:

90

OPTIONS:

n temperature is scaled by $0.01 \cdot n$ in each cycle (cooling method 1)

n temperature is decreased by n K in each cycle (cooling method 2)

RECOMMENDATION:

The cooling rate should be neither too slow nor too fast. Too slow may lead to final energies that are at undesirably high temperatures. Too fast may lead to convergence issues. Reasonable choices for methods 1 and 2 are 98 and 50, respectively. When in doubt, use constant temperature.

FON_E_THRESH

DIIS error below which occupations will be kept constant.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n freeze occupations below DIIS error of 10^{-n}

RECOMMENDATION:

This should be one or two numbers bigger than the desired SCF convergence threshold.

FON_T_METHOD

Selects cooling algorithm.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 temperature is scaled by a factor in each cycle

2 temperature is decreased by a constant number in each cycle

RECOMMENDATION:

We have made slightly better experience with a constant cooling rate. However, choose constant temperature when in doubt.

Example 4.11 pFON calculation of a metal cluster.

```
$molecule
   0 1
   Pt   -0.20408    1.19210    0.54029
   Pt    2.61132    1.04687    0.66196
   Pt    0.83227    0.03296   -1.49084
   Pt    0.95832   -1.05360    0.92253
   Pt   -1.66760   -1.07875   -1.02416
$end

$rem
   METHOD          pbe
   MAX_SCF_CYCLES  200
   ECP             fit-lanl2dz
   BASIS           lanl2dz
   OCCUPATIONS     2    ! pseudo-fractional occupation numbers
   FON_NORB        10   ! 10 fractionally occupied orbitals above and below the Fermi level
   FON_T_START     1000 ! starting electronic temperature: 1000 K
   FON_T_END       0    ! final electronic temperature: 0 K
   FON_T_METHOD    2    ! constant cooling scheme
   FON_T_SCALE     25   ! reduce the temperature by 25 K per cooling step
   FON_E_THRESH    5    ! freeze occupation numbers once DIIS error is 10-5
   GEN_SCFMAN        false
   INTEGRAL_SYMMETRY false
$end
```

4.5.7 Geometric Direct Minimization (GDM)

Geometric Direct Minimization (GDM) is an extremely robust SCF convergence algorithm that is only slightly less efficient than DIIS. The GDM algorithm takes steps in an orbital rotation space that properly respects the hyperspherical geometry of the manifold of allowed SCF solutions. In other words, orbital rotations are variables that describe a space that is curved like a many-dimensional sphere. Just like the optimum flight paths for airplanes are not straight lines but great circles, so too are the optimum steps in orbital rotation space. GDM takes this correctly into account, which is the origin of its efficiency and its robustness. For full details see Ref. 94. GDM is a good alternative to DIIS for SCF jobs that exhibit convergence difficulties with DIIS. The GDM algorithm has been extended to restricted open-shell SCF calculations, and results indicate that it is much more efficient as compared to older direct-minimization methods.

Section 4.5.3 discussed the fact that DIIS can efficiently head towards the global SCF minimum in the early iterations. This can be true even if DIIS fails to converge in later iterations. For this reason, a hybrid scheme has been implemented

which uses the DIIS minimization procedure to achieve convergence to an intermediate cutoff threshold. Thereafter, the geometric direct minimization algorithm is used. This scheme combines the strengths of the two methods quite nicely: the ability of DIIS to recover from initial guesses that may not be close to the global minimum, and the ability of GDM to robustly converge to a local minimum, even when the local surface topology is challenging for DIIS. This is the recommended procedure with which to invoke GDM (*i.e.*, setting SCF_ALGORITHM = DIIS_GDM). This hybrid procedure is also compatible with the SAD guess, while GDM itself is not, because it requires an initial guess set of orbitals. If one wishes to disturb the initial guess as little as possible before switching on GDM, one should additionally specify MAX_DIIS_CYCLES = 1 to obtain only a single Roothaan step (which also serves up a properly orthogonalized set of orbitals).

$rem options relevant to GDM are SCF_ALGORITHM which should be set to either GDM or DIIS_GDM and the following:

MAX_DIIS_CYCLES

The maximum number of DIIS iterations before switching to (geometric) direct minimization when SCF_ALGORITHM is DIIS_GDM or DIIS_DM. See also THRESH_DIIS_SWITCH.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

1 Only a single Roothaan step before switching to (G)DM

n n DIIS iterations before switching to (G)DM.

RECOMMENDATION:

None

THRESH_DIIS_SWITCH

The threshold for switching between DIIS extrapolation and direct minimization of the SCF energy is $10^{-\text{THRESH\_DIIS\_SWITCH}}$ when SCF_ALGORITHM is DIIS_GDM or DIIS_DM. See also MAX_DIIS_CYCLES.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined.

RECOMMENDATION:

None

Example 4.12 Input for a UHF calculation using geometric direct minimization (GDM) on the phenyl radical, after initial iterations with DIIS.

```
$molecule
   0   2
   c1
   x1  c1  1.0
   c2  c1  rc2  x1  90.0
   x2  c2  1.0  c1  90.0  x1    0.0
   c3  c1  rc3  x1  90.0  c2    tc3
   c4  c1  rc3  x1  90.0  c2   -tc3
   c5  c3  rc5  c1   ac5  x1  -90.0
   c6  c4  rc5  c1   ac5  x1   90.0
   h1  c2  rh1  x2  90.0  c1  180.0
   h2  c3  rh2  c1   ah2  x1   90.0
   h3  c4  rh2  c1   ah2  x1  -90.0
   h4  c5  rh4  c3   ah4  c1  180.0
   h5  c6  rh4  c4   ah4  c1  180.0

   rc2 =   2.672986
   rc3 =   1.354498
   tc3 =  62.851505
   rc5 =   1.372904
   ac5 = 116.454370
   rh1 =   1.085735
   rh2 =   1.085342
   ah2 = 122.157328
   rh4 =   1.087216
   ah4 = 119.523496
$end

$rem
   BASIS           = 6-31G*
   METHOD          = hf
   SCF_ALGORITHM   = diis_gdm
   SCF_CONVERGENCE = 7
   THRESH          = 10
$end
```

4.5.8 Direct Minimization (DM)

Direct minimization (DM) is a less sophisticated forerunner of the geometric direct minimization (GDM) method discussed in the previous section. DM does not properly step along great circles in the hyper-spherical space of orbital rotations, and therefore converges less rapidly and less robustly than GDM, in general. DM is retained in Q-CHEM only for legacy purposes. In general, the input options are the same as for GDM, with the exception of the specification of SCF_ALGORITHM, which can be either DIIS_DM (recommended) or DM.

PSEUDO_CANONICAL

When SCF_ALGORITHM = DM, this controls the way the initial step, and steps after subspace resets are taken.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use Roothaan steps when (re)initializing

TRUE Use a steepest descent step when (re)initializing

RECOMMENDATION:

The default is usually more efficient, but choosing TRUE sometimes avoids problems with orbital reordering.

4.5.9 Relaxed Constraint Algorithm (RCA)

The relaxed constraint algorithm (RCA) is an ingenious and simple means of minimizing the SCF energy that is particularly effective in cases where the initial guess is poor. The latter is true, for example, when employing a user-specified basis (when the "core" or GWH guess must be employed) or when near-degeneracy effects imply that the initial guess will likely occupy the wrong orbitals relative to the desired converged solution.

Briefly, RCA begins with the SCF problem as a constrained minimization of the energy as a function of the density matrix, $E(\mathbf{P})$.[13,14] The constraint is that the density matrix be idempotent, $\mathbf{P} \cdot \mathbf{P} = \mathbf{P}$, which basically forces the occupation numbers to be either zero or one. The fundamental realization of RCA is that this constraint can be relaxed to allow sub-idempotent density matrices, $\mathbf{P} \cdot \mathbf{P} \leq \mathbf{P}$. This condition forces the occupation numbers to be between zero and one. Physically, we expect that any state with fractional occupations can lower its energy by moving electrons from higher energy orbitals to lower ones. Thus, if we solve for the minimum of $E(\mathbf{P})$ subject to the relaxed sub-idempotent constraint, we expect that the ultimate solution will nonetheless be idempotent.

In fact, for Hartree-Fock this can be rigorously proven. For density functional theory, it is possible that the minimum will have fractional occupation numbers but these occupations have a physical interpretation in terms of ensemble DFT. The reason the relaxed constraint is easier to deal with is that it is easy to prove that a linear combination of sub-idempotent matrices is also sub-idempotent as long as the linear coefficients are between zero and one. By exploiting this property, convergence can be accelerated in a way that guarantees the energy will go down at every step.

The implementation of RCA in Q-CHEM closely follows the "Energy DIIS" implementation of the RCA algorithm.[49] Here, the current density matrix is written as a linear combination of the previous density matrices:

$$\mathbf{P}(x) = \sum_i x_i \mathbf{P}_i \tag{4.39}$$

To a very good approximation (exact for Hartree-Fock) the energy for $\mathbf{P}(x)$ can be written as a quadratic function of x:

$$\mathbf{E}(x) = \sum_i E_i x_i + \frac{1}{2} \sum_i x_i (\mathbf{P}_i - \mathbf{P}_j) \cdot (\mathbf{F}_i - \mathbf{F}_j) x_j \tag{4.40}$$

At each iteration, x is chosen to minimize $\mathbf{E}(x)$ subject to the constraint that all of the x_i are between zero and one. The Fock matrix for $\mathbf{P}(x)$ is further written as a linear combination of the previous Fock matrices,

$$\mathbf{F}(x) = \sum_i x_i \mathbf{F}_i + \delta \mathbf{F}_{xc}(x) \tag{4.41}$$

where $\delta \mathbf{F}_{xc}(x)$ denotes a (usually quite small) change in the exchange-correlation part that is computed once x has been determined. We note that this extrapolation is very similar to that used by DIIS. However, this procedure is guaranteed to reduce the energy $\mathbf{E}(x)$ at every iteration, unlike DIIS.

In practice, the RCA approach is ideally suited to difficult convergence situations because it is immune to the erratic orbital swapping that can occur in DIIS. On the other hand, RCA appears to perform relatively poorly near convergence, requiring a relatively large number of steps to improve the precision of a good approximate solution. It is thus advantageous in many cases to run RCA for the initial steps and then switch to DIIS either after some specified number of iterations or after some target convergence threshold has been reached. Finally, note that by its nature RCA considers the energy as a function of the density matrix. As a result, it cannot be applied to restricted open shell calculations which are explicitly orbital-based. Note: RCA interacts poorly with INCDFT, so INCDFT is disabled by default when an RCA or RCA_DIIS calculation is requested. To enable INCDFT with such a calculation, set INCDFT = 2 in the *$rem* section. RCA may also have poor interactions with incremental Fock builds; if RCA fails to converge, setting INCFOCK = FALSE may improve convergence in some cases.

Job-control variables for RCA are listed below, along with an example of its use.

RCA_PRINT

Controls the output from RCA SCF optimizations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No print out

1 RCA summary information

2 Level 1 plus RCA coefficients

3 Level 2 plus RCA iteration details

RECOMMENDATION:

None

MAX_RCA_CYCLES

The maximum number of RCA iterations before switching to DIIS when SCF_ALGORITHM is RCA_DIIS.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

N N RCA iterations before switching to DIIS

RECOMMENDATION:

None

THRESH_RCA_SWITCH

The threshold for switching between RCA and DIIS when SCF_ALGORITHM is RCA_DIIS.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

N Algorithm changes from RCA to DIIS when Error is less than 10^{-N}.

RECOMMENDATION:

None

Example 4.13 RCA_DIIS algorithm applied a radical

```
$molecule
0 2
   H     1.004123   -0.180454    0.000000
   O    -0.246002    0.596152    0.000000
   O    -1.312366   -0.230256    0.000000
$end

$rem
   UNRESTRICTED          true
   METHOD                hf
   BASIS                 cc-pVDZ
   SCF_GUESS             gwh
   SCF_ALGORITHM         RCA_DIIS
   THRESH                9
$end
```

4.5.10 Augmented Roothaan Hall Energy DIIS (ADIIS)

Similar to RCA/EDIIS,[13,14,49] the ADIIS algorithm proposed by Hu and Yang[46] is also supposed to accelerate SCF convergence in cases where DIIS performs poorly in the initial iterations. This algorithm also involves a Fock matrix extrapolation scheme:

$$\tilde{\mathbf{F}}_{n+1} = \sum_{i=1}^{n} c_i \mathbf{F}_i \tag{4.42}$$

where $\tilde{\mathbf{F}}_{n+1}$ is the extrapolated Fock matrix to be diagonalized to generate the updated MOs and electron density, $\mathbf{F}_i = \mathbf{F}[\mathbf{P}_i]$ is the Fock matrix constructed from the density matrix of the i-th iteration, and $\{c_i\}$ are the extrapolation coefficients, which are obtained by minimizing the augmented Roothaan-Hall (ARH) energy function of an extrapolated density $\tilde{\mathbf{P}}_{i+1} = \sum_{i=1}^{n} \mathbf{P}_i$

$$\begin{aligned} f^{\mathrm{ADIIS}}(c_1, \ldots, c_n) = E[\mathbf{P}_n] + \sum_{i=1}^{n} c_i (\mathbf{P}_i - \mathbf{P}_n) \cdot \mathbf{F}_n \\ + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j (\mathbf{P}_i - \mathbf{P}_n) \cdot (\mathbf{F}_j - \mathbf{F}_n) \end{aligned} \tag{4.43}$$

while subjected to the constraint $\sum_{i=1}^{n} c_i = 1, c_i \geq 0$ for all i. As suggested in the original literature,[46] variable substitutions are conducted ($c_i = t_i^2 / \sum_i t_i^2$) to convert the constrained optimization to a standard, unconstrained optimization problem (optimizers such as L-BFGS can be used to solve the latter). Note that while the sums in Eqs. 4.42 and 4.43 runs from 1 to n, in practice it is unnecessary to extrapolate using all $\mathbf{P}_i$ and $\mathbf{F}_i$ obtained in the previous SCF cycles. In the Q-CHEM implementation of ADIIS, the number of $\mathbf{P}_i$'s and $\mathbf{F}_i$'s used in the extrapolation has a maximum of 6.

The ADIIS algorithm is known to become less efficient in the region close to SCF convergence. Therefore, it is desirable to use the "ADIIS+DIIS" algorithm, which carries out ADIIS when the SCF error is below a threshold or the number of ADIIS iterations reaches a certain value. This hybrid algorithm, which can be invoked by setting "SCF_ALGORITHM = ADIIS_DIIS" in Q-CHEM, was shown to afford accelerated convergence for cases where DIIS alone was unable or took much longer to converge the SCF problem.[46]

ADIIS_INNER_CONV

Convergence criterion for the ADIIS inner loops (L-BFGS optimization of Eq. 4.43)

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

n Using 10^{-n} as the convergence criterion for the ADIIS inner loops

RECOMMENDATION:

Use the default

THRESH_ADIIS_SWITCH

The threshold for switching from ADIIS to DIIS in a ADIIS-DIIS calculations

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Switching from ADIIS to DIIS when the SCF error is below 10^{-n}

RECOMMENDATION:

3 or 4 is suitable

MAX_ADIIS_CYCLES

The maximum number of ADIIS cycles before switching to DIIS in a ADIIS-DIIS calculations

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

N Doing at most N ADIIS iterations before switching to DIIS

RECOMMENDATION:

Use the default; typically there is no benefit of doing ADIIS for too many iterations

Example 4.14 B3LYP/3-21g calculation for the Cd(II)-imidazole complex using the ADIIS_DIIS algorithm (switching to DIIS when the error is below 10^{-3}). The SADMO guess is used.

```
$molecule
2 1
Cd      0.000000      0.000000      0.000000
 N      0.000000      0.000000     -2.260001
 N     -0.685444      0.000000     -4.348035
 C      0.676053      0.000000     -4.385069
 C      1.085240      0.000000     -3.091231
 C     -1.044752      0.000000     -3.060220
 H      1.231530      0.000000     -5.300759
 H      2.088641      0.000000     -2.711077
 H     -2.068750      0.000000     -2.726515
 H     -1.313170      0.000000     -5.174718
$end

$rem
METHOD           B3LYP
BASIS            3-21G
THRESH           14
SCF_GUESS        SADMO
SCF_ALGORITHM    ADIIS_DIIS
SCF_CONVERGENCE  8
INTEGRAL_SYMMETRY    FALSE
POINT_GROUP_SYMMETRY FALSE
$end
```

4.5.11 Newton Methods

Q-CHEM also offers various Newton methods for converging SCF problems. In these methods, the step to take in the orbital rotation space is calculated as $\mathbf{\Delta} = -\mathbf{H}^{-1}\mathbf{g}$, where $\mathbf{g}$ and $\mathbf{H}$ are the gradient and *exact* Hessian of the SCF energy with respect to the orbital rotation variables, respectively. In the quadratic regime in the vicinity of an energy minimum, the Hessian matrix is positive definite and by taking Newton steps the minimum can be reached very efficiently, usually within very few steps. However, when far from the minimum the Hessian matrix is often indefinite and a Newton step may not even lead to descending. Since to obtain SCF orbital Hessian (Hessian-vector products in the actual implementation) is computationally demanding, it is typically recommended to perform SCF calculations using another algorithm (*e.g.* GDM) and then start using Newton methods after the error is below a certain threshold. See Section 4.5.12 for job control details for general hybrid-algorithm calculations and the examples therein. In addition, a line search algorithm is employed in the Q-CHEM implementation of Newton methods to prevent them from overstepping.

In the actual implementation of these methods in Q-CHEM, the Newton steps are calculated by iteratively solving the linear equation $\mathbf{H\Delta} = -\mathbf{g}$. This avoids the explicit evaluation and inversion of the SCF orbital Hessian and thus is computationally more efficient. The conjugate-gradient (CG) and minimal residual (MINRES) solvers are currently supported, which correspond to the NEWTON_CG and NEWTON_MINRES algorithms in Q-CHEM, respectively. The CG method requires the Hessian matrix to be positive definite, which may not be satisfied when far from convergence. In those cases, the CG iterations will be terminated when negative curvature is detected (based on $(\mathbf{p}^{(i)})^T\mathbf{H}\mathbf{p}^{(i)} < 0$, where $\mathbf{p}^{(i)}$ is the update to the solution vector $\mathbf{\Delta}^{(i)}$ at the i-th iteration. Then the unconverged $\mathbf{\Delta}^{(i)}$ will be adopted as the update direction (see Ref. 65 for details). The MINRES algorithm, on the other hand, doesn't require the Hessian matrix to be positive definite, so this method will be able to solve for $\mathbf{\Delta}$ in most cases except when singularity exists. However, as mentioned above, the obtained update (orbital rotation) may not lead to descending in energy when the Hessian is not positive definite.

A "saddle-free" Newton-CG algorithm (SF_NEWTON_CG) is also implemented in Q-CHEM. This methods aims to restore the curvature of the Hessian matrix when negative eigenvalues are present by using the following modified Hessian:

$$\widetilde{\mathbf{H}} = \mathbf{H} - \mathbf{v}_{-}\lambda_{-}\mathbf{v}_{-}^{T} \tag{4.44}$$

where λ_{-} is a diagonal matrix formed by negative eigenvalues and $\mathbf{v}_{-}$ represents the corresponding eigenvectors. At each iteration, the algorithm will start with solving for a few lowest eigenvalues and eigenvectors using the Davidson algorithm, and then CG will be used to solve $\widetilde{\mathbf{H}}\boldsymbol{\Delta} = -\mathbf{g}$, where the modified Hessian is guaranteed to be positive definite.

Example 4.15 SCF calculation for a water molecule using the NEWTON_CG algorithm

```
$molecule
0 1
   O  -1.551007  -0.114520   0.000000
   H  -1.934259   0.762503   0.000000
   H  -0.599677   0.040712   0.000000
$end

$rem
   jobtype  sp
   method   b3lyp
   basis    6-31g(d)
   scf_algorithm   newton_cg
   scf_convergence 9
$end
```

4.5.12 User-Customized Hybrid SCF Algorithm

It is often the case that a single algorithm is not able to guarantee SCF convergence. Meanwhile, some SCF algorithms (*e.g.*, ADIIS) can accelerate convergence at the beginning of an SCF calculation but becomes less efficient near the convergence. While a few hybrid algorithms (DIIS_GDM, RCA_DIIS) have been enabled in Q-CHEM's original SCF implementation, in GEN_SCFMAN, we seek for a more flexible setup for the use of multiple SCF algorithms so that users can have a more precise control on the SCF procedure. With the current implementation, at most four distinct algorithms (usually more than enough) can be employed in one single SCF calculation based on GEN_SCFMAN, and the basic job control is as follows:

GEN_SCFMAN_HYBRID_ALGO

Use multiple algorithms in an SCF calculation based on GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Use a single SCF algorithm (given by SCF_ALGORITHM).

TRUE Use multiple SCF algorithms (to be specified).

RECOMMENDATION:

Set it to TRUE when the use of more than one algorithm is desired.

GEN_SCFMAN_ALGO_1

The first algorithm to be used in a hybrid-algorithm calculation.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

All the available SCF_ALGORITHM options, including the GEN_SCFMAN additions (Section 4.3).

RECOMMENDATION:

None

GEN_SCFMAN_ITER_1

Maximum number of iterations given to the first algorithm. If used up, switch to the next algorithm.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

RECOMMENDATION:

None

GEN_SCFMAN_CONV_1

The convergence criterion given to the first algorithm. If reached, switch to the next algorithm.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n 10^{-n}

RECOMMENDATION:

None

Note: *$rem* variables GEN_SCFMAN_ALGO_X, GEN_SCFMAN_ITER_X, GEN_SCFMAN_CONV_X (X = 2, 3, 4) are defined and used in a similar way.

Example 4.16 B3LYP/3-21G calculation for a cadmium-imidazole complex using the ADIIS + DIIS algorithm (an example from Ref. 46). Due to the poor quality of the CORE guess, using a single algorithm such as DIIS or GDM fails to converge.

```
$molecule
   2 1
   Cd      0.000000      0.000000      0.000000
   N       0.000000      0.000000     -2.260001
   N      -0.685444      0.000000     -4.348035
   C       0.676053      0.000000     -4.385069
   C       1.085240      0.000000     -3.091231
   C      -1.044752      0.000000     -3.060220
   H       1.231530      0.000000     -5.300759
   H       2.088641      0.000000     -2.711077
   H      -2.068750      0.000000     -2.726515
   H      -1.313170      0.000000     -5.174718
$end

$rem
   EXCHANGE                 B3LYP
   BASIS                    3-21g
   UNRESTRICTED             FALSE
   THRESH                   14
   SCF_GUESS                CORE
   GEN_SCFMAN_HYBRID_ALGO   TRUE
   GEN_SCFMAN_ALGO_1        ADIIS
   GEN_SCFMAN_CONV_1        3  ! switch to DIIS when error < 1E-3
   GEN_SCFMAN_ITER_1        50
   GEN_SCFMAN_ALGO_2        DIIS
   GEN_SCFMAN_CONV_2        8
   GEN_SCFMAN_ITER_2        50
   INTEGRAL_SYMMETRY        FALSE
   POINT_GROUP_SYMMETRY     FALSE
$end
```

4.5.13 Black-box Robust SCF Pipelines

In the spirit of the User-Customized Hybrid SCF Algorithm in Section 4.5.12, starting v6.3.0, Q-Chem provides two "black-box" pipelines for improving the robustness of the SCF convergence for difficult use cases. These pipelines can be used by setting SCF_ALGORITHM to ROBUST or ROBUST_STABLE in conjunction to GEN_SCFMAN set to true. The outline for the ROBUST pipeline is below:

- SCF converges when the DIIS error or RMS gradient converges below the requested convergence threshold.
- The cutoff for the neglect of 2-electron integrals is set to 10^{-12} or a tighter user-specified value set using THRESH. Note that, by default, THRESH = SCF_CONVERGENCE+4.
- The cutoff for the neglect of overlap integrals is set to 10^{-14} or a tighter user-specified value set using S2THRESH to offset any basis-set linear dependency problems.
- We define oscillatory behavior during the SCF procedure as SCF_ROBUST_MIN_SIGN_CHANGES number of sign changes within a window of logarithms of DIIS errors or RMS gradients.
- We define a plateau detection when the slopes of SCF_ROBUST_PLATEAU_WINDOWS windows of DIIS errors or RMS gradients are less than SCF_ROBUST_SLOPE_THRESH.

- Throughout the pipeline, MAX_SCF_CYCLES corresponds to the maximum cycles allowed for an SCF algorithm to run without any parameter changes.
- The pipeline starts with the DIIS algorithm. If a plateau or oscillatory behavior in the DIIS error is detected after iteration #(2*SCF_ROBUST_WINDOW_SIZE+1), the incremental Fock algorithm, if on, is switched off for good.
- At this point, if the DIIS error is less than 0.001, the algorithm is switched to ADIIS; otherwise, DIIS is allowed to run for a further MAX_SCF_CYCLES iterations unless a plateau or oscillatory trend is detected again at which point the algorithm switches to GDM.
- ADIIS is allowed to run for MAX_SCF_CYCLES iterations unless the DIIS error becomes smaller than 0.001.
- DIIS is run again after ADIIS. The pipeline switches to GDM if a plateau or oscillatory trend is detected again after iteration #(2*SCF_ROBUST_WINDOW_SIZE+1).
- GDM is allowed to run until convergence or maximum allowed number of iterations.

The ROBUST_STABLE pipeline performs a stability analysis on top of the ROBUST pipeline until a minimum is found. Following the initial macroiteration run with ROBUST, only GDM is used in subsequent macroiterations for stability analaysis. The number of negative eigenvalues needs to decrease with each macroiteration as a necessary condition for this pipeline to continue.

Below, we describe the internal *$rem* parameters of these robust pipelines. The default values of these parameters are currently set to experimental values and may not be optimal for all use cases; so, the users are encouraged to share their settings with the Q-Chem Office, if the defaults don't yield convergence for their use cases.

SCF_ROBUST_WINDOW_SIZE
Controls the size of the window of logarithms of DIIS error or RMS gradients used for detecting oscillations or a plateau in robust SCF procedures.

TYPE:
INTEGER

DEFAULT:
7

OPTIONS:
n Customized.

RECOMMENDATION:
None

SCF_ROBUST_MIN_SIGN_CHANGES
Controls the number of sign changes in a window of logarithms of DIIS error or RMS gradients used for detecting oscillations in robust SCF procedures.

TYPE:
INTEGER

DEFAULT:
3

OPTIONS:
n Customized.

RECOMMENDATION:
None

SCF_ROBUST_PLATEAU_WINDOWS

Controls the number of windows of logarithms of DIIS error or RMS gradients used for detecting a plateau in robust SCF procedures.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n Customized.

RECOMMENDATION:

None

SCF_ROBUST_SLOPE_THRESH

Controls the slope threshold for a series of windows of logarithms of DIIS error or RMS gradients used for detecting a plateau in robust SCF procedures.

TYPE:

INTEGER

DEFAULT:

3 Corresponds to 10^{-3}

OPTIONS:

n Customized to 10^{-n}.

RECOMMENDATION:

None

4.5.14 Maximum Overlap Method (MOM)

In general, the DIIS procedure (Section 4.5.3) is remarkably successful. One difficulty that is occasionally encountered is the problem of an SCF that occupies two different sets of orbitals on alternating iterations, and therefore oscillates and fails to converge. This can be overcome by choosing orbital occupancies that maximize the overlap of the new occupied orbitals with the set previously occupied. Q-CHEM contains the maximum overlap method (MOM),[35] developed by Andrew Gilbert and Peter Gill. With GEN_SCFMAN, the MOM algorithm can be applied to R, U, and RO SCF calculations when paired with the DIIS algorithm.

MOM is therefore is a useful adjunct to DIIS in convergence problems involving flipping of orbital occupancies. It is controlled by the *$rem* variable MOM_START, which specifies the SCF iteration on which the MOM procedure is first enabled. There are two strategies that are useful in setting a value for MOM_START. To help maintain an initial configuration it should be set to start on the first cycle. On the other hand, to assist convergence it should come on later to avoid holding on to an initial configuration that may be far from the converged one.

The MOM-related *$rem* variables in full are the following:

MOM_PRINT

Switches printing on within the MOM procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Printing is turned off

TRUE Printing is turned on.

RECOMMENDATION:

None

MOM_START

Determines when MOM is switched on to stabilize DIIS iterations.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 (FALSE) MOM is not used

n MOM begins on cycle n.

RECOMMENDATION:

Set to 1 if preservation of initial orbitals is desired. If MOM is to be used to aid convergence, an SCF without MOM should be run to determine when the SCF starts oscillating. MOM should be set to start just before the oscillations.

MOM_METHOD

Determines the target orbitals with which to maximize the overlap on each SCF cycle.

TYPE:

INTEGER

DEFAULT:

MOM

OPTIONS:

MOM Maximize overlap with the orbitals from the previous SCF cycle.

IMOM Maximize overlap with the initial guess orbitals.

RECOMMENDATION:

If appropriate guess orbitals can be obtained, then IMOM can provide more reliable convergence to the desired solution.[7]

Example 4.4.17 An example showing how to converge a ROHF calculation on the $^3A_2$ state of DMX. Note the use of reading in orbitals from a previous closed-shell calculation and the use of MOM to maintain the orbital occupancies. The $^3B_1$ is obtained if MOM is not used.

View input online

4.5.15 Square Gradient Minimization (SGM)

The GDM method (Section 4.5.7) is an extremely effective energy minimizer but it cannot reliably be applied to optimize excited-state orbitals, as such states are typically unstable stationary points in orbital-rotation space. Energy

minimization based approaches therefore tend to 'slip' from these saddle points to some local minima (often the ground state, a phenomenon often described as 'variational collapse').

Diptarka Hait and Martin Head-Gordon have proposed an alternative way to optimize excited state orbitals, by minimizing the square of the energy gradient against orbital degrees of freedom.[39] This energy gradient should be zero for all stationary points in energy, and thus all such stationary points are global minima of the squared energy gradient Δ. Quasi-Newton methods therefore can reliably converge to the stationary point closest to the initial guess orbitals by minimizing Δ, without the risk of variational collapse. The resulting SGM approach is thus essentially an extension of GDM that converges to the closest state (*i.e.*, stationary point in orbital space) to the initial guess, as opposed to the closest energy minimum. SGM consequently can be used for reliable excited state optimization within a direct minimization framework, similar to how the MOM algorithm of Section 4.5.14 can be used in conjunction with iterative diagonalization methods like DIIS. Further details about SGM applying for excited-state orbital optimization can be found in Section 7.8.4. Full details of the SGM algorithm are provided in Ref. 39.

The use of SGM is controlled by the SCF_ALGORITHM variable in the *$rem* section:

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

SGM

SGM_LS

SGM_QLS for R and U orbitals only

RECOMMENDATION:

SGM should be used for RO and or OS_RO orbitals only. SGM_LS is recommended for R or U orbitals, though it can also be used for RO and OS_RO orbitals. SGM_QLS is a slower, but more robust option for R and U calculations.

DELTA_GRADIENT_SCALE

Scales the gradient of Δ by $N/100$, which can be useful for cases with troublesome convergence by reducing step size.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

N

RECOMMENDATION:

Use default. For problematic cases 50, 25, 10 or even 1 could be useful.

4.5.16 State-Targeted Energy Projection (STEP)

The maximum overlap method (Sec. 4.5.14) is successful in many cases, but when optimizing excited state orbitals it can be prone to variational collapse (falling from the target configuration to the ground state). This behavior was improved with the "initial" MOM (IMOM) procedure, which uses the initial guess molecular orbital coefficients as the reference for the overlap criterion (see Sec. 7.6 for details). While IMOM lends a significant improvement to the MOM, it occasionally also falls victim to variational collapse, incentivizing the pursuit of alternatives to these methods.

One such alternative, proposed by Kevin Carter-Fenk and John Herbert, is the state-targeted energy projection (STEP) procedure, which applies a simple level-shift formalism to converge to the target state.[15] The STEP algorithm retains the cost-effectiveness of the MOM procedures (about the same cost per cycle as a normal SCF), while simultaneously being far more robust in converging to the target state. Further details on the STEP procedure can be found in Section 7.8.5, and for a complete account of the STEP algorithm the reader is referred to Ref. 15.

The STEP-related *$rem* variables are the following:

STEP

Activates the STEP procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not apply the STEP level-shift algorithm.

TRUE Apply the STEP level-shift algorithm.

RECOMMENDATION:

None

STEP_EPSILON

Scales the size of the occupied/virtual gap imposed by the level-shift by N/100 Hartree.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

N

RECOMMENDATION:

Use the default unless convergence issues arise, in which case a larger value can be used until the desired state is found. Be aware that increasing the occupied/virtual gap in level-shift algorithms slows convergence so it may be advisable to increase SCF_MAX_CYCLES if large shifts are required.

STEP_PRINT

Controls the print level for STEP algorithm information.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Do not print any information about STEP between SCF cycles.

1 Print the level-shift applied at each SCF cycle (R- and U-STEP).

2 Print the level-shift for both mixed and triplet states at each SCF cycle (RO-STEP).

RECOMMENDATION:

Use the default. Level shifts of 0 indicate that an *aufbau* criterion is sufficient to determine orbital occupation, and shifts > 0 imply non-*aufbau* selection of the occupied space.

4.5.17 Internal Stability Analysis and Automated Correction for Energy Minima

At convergence, the SCF energy will be at a stationary point with respect to changes in the MO coefficients. However, this stationary point is not guaranteed to be an energy minimum, and in cases where it is not, the wave function is said to be unstable. Even if the wave function is at a minimum, this minimum may be an artifact of the constraints placed on the form of the wave function. For example, an unrestricted calculation will usually give a lower energy than the corresponding restricted calculation, and this can give rise to an RHF $\rightarrow$ UHF instability.

Based on our experience, even for very simple data set such as the G2 atomization energies,[19] using the default algorithm (DIIS) produces unstable solutions for several species (even for single atoms with some density functionals). In such cases, failure to check the internal stability of SCF solutions can result in flawed benchmark results. Although in general the use of gradient-based algorithms such as GDM is more likely to locate the true minimum, it still cannot entirely eliminate the possibility of finding an unstable solution.

To understand what instabilities can occur, it is useful to consider the most general form possible for the spin orbitals:

$$\chi_i(\mathbf{r},\zeta) = \psi_i^{\alpha}(\mathbf{r})\alpha(\zeta) + \psi_i^{\beta}(\mathbf{r})\beta(\zeta)\ . \tag{4.45}$$

Here, ψ_i^{α} and ψ_i^{β} are complex-valued functions of the Cartesian coordinates $\mathbf{r}$, and α and β are spin eigenfunctions of the spin-variable ζ. The first constraint that is almost universally applied is to assume the spin orbitals depend only on one or other of the spin-functions α or β. Thus, the spin-functions take the form

$$\chi_i(\mathbf{r},\zeta) = \psi_i^{\alpha}(\mathbf{r})\alpha(\zeta) \quad \text{or} \quad \chi_i(\mathbf{r},\zeta) = \psi_i^{\beta}(\mathbf{r})\beta(\zeta)\ . \tag{4.46}$$

In addition, most SCF calculations use real functions, and this places an additional constraint on the form of the wave function. If there exists a complex solution to the SCF equations that has a lower energy, the wave function exhibits a real $\rightarrow$ complex instability. The final constraint that is commonly placed on the spin-functions is that $\psi_i^{\alpha} = \psi_i^{\beta}$, *i.e.*, that the spatial parts of the spin-up and spin-down orbitals are the same. This gives the familiar restricted formalism and can lead to an RHF $\rightarrow$ UHF instability as mentioned above. Further details about the possible instabilities can be found in Ref. 76.

Wave function instabilities can arise for several reasons, but frequently occur if

- There exists a singlet diradical at a lower energy then the closed-shell singlet state.
- There exists a triplet state at a lower energy than the lowest singlet state.
- There are multiple solutions to the SCF equations, and the calculation has not found the lowest energy solution.

Q-CHEM's previous stability analysis package suffered from the following limitations:

- It is only available for restricted (close-shell) and unrestricted SCF calculations.
- It requires the analytical orbital Hessian of the wave function energy.
- The calculation terminates after the corrected MOs are generated, and a second job is needed to read in these orbitals and run another SCF calculation.

The implementation of internal stability analysis in GEN_SCFMAN overcomes almost all these shortcomings. Its availability has been extended to all the implemented orbital types. As in the old code, when the analytical Hessian of the given orbital type and theory (*e.g.* RO/B3LYP) is available, it computes matrix-vector products analytically for the Davidson algorithm.[22] If the analytical Hessian is not available, users can still run stability analysis by using the

finite-difference matrix-vector product technique developed by Sharada *et al.*,[79] which requires the gradient (related to the Fock matrix) only:

$$\mathbf{H}\mathbf{b}_1 = \frac{\hat{\boldsymbol{\nabla}}E(\mathbf{X}_0 + \xi\mathbf{b}_1) - \hat{\boldsymbol{\nabla}}E(\mathbf{X}_0 - \xi\mathbf{b}_1)}{2\xi} \tag{4.47}$$

where $\mathbf{H}$ is the Hessian matrix, $\mathbf{b}_1$ is a trial vector, $\mathbf{X}_0$ stands for the current stationary point, and ξ is the finite step size. With this method, internal stability analysis is available for all the implemented orbital types in GEN_SCFMAN. It should be noted that since the second derivative of NLC functionals (except for VV10) is not available in Q-CHEM, this finite-difference method will be used by default for the evaluation of Hessian-vector products.

GEN_SCFMAN allows multiple SCF calculations and stability analyses to be performed in a single job so that it can make use of the corrected MOs and locate the true minimum automatically. The MOs are displaced along the direction of the lowest-energy eigenvector (with line search) if an SCF solution is found to be unstable. A new SCF calculation that reads in these corrected MOs as initial guess will be launched automatically if INTERNAL_STABILITY_ITER > 0. Such macro-loops will keep going until a stable solution is reached.

Note: The stability analysis package can be used to analyze both HF and DFT wave functions.

4.5.17.1 Job Control

INTERNAL_STABILITY

Perform internal stability analysis in GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform internal stability analysis after convergence.

TRUE Perform internal stability analysis and generate the corrected MOs.

RECOMMENDATION:

Turn it on when the SCF solution is prone to unstable solutions, especially for open-shell species.

FD_MAT_VEC_PROD

Compute Hessian-vector product using the finite difference technique.

TYPE:

BOOLEAN

DEFAULT:

FALSE (TRUE when the employed functional contains non-local correlation (except VV10))

OPTIONS:

FALSE Compute Hessian-vector product analytically.

TRUE Use finite difference to compute Hessian-vector product.

RECOMMENDATION:

Set it to TRUE when analytical Hessian is not available.

Note: For simple R and USCF calculations, it can always be set to FALSE, which indicates that only the NLC part will be computed with finite difference (if its analytic orbital hessian is unavailable).

INTERNAL_STABILITY_ITER

Maximum number of new SCF calculations permitted after the first stability analysis is performed.

TYPE:

INTEGER

DEFAULT:

0 (automatically set to 1 if INTERNAL_STABILITY = TRUE)

OPTIONS:

n n new SCF calculations permitted.

RECOMMENDATION:

Give a larger number if 1 is not enough (still unstable).

INTERNAL_STABILITY_DAVIDSON_ITER

Maximum number of Davidson iterations allowed in one stability analysis.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n Perform up to n Davidson iterations.

RECOMMENDATION:

Use the default.

INTERNAL_STABILITY_CONV

Convergence criterion for the Davidson solver (for the lowest eigenvalues).

TYPE:

INTEGER

DEFAULT:

4 (3 when FD_MAT_VEC_PROD = TRUE)

OPTIONS:

n Terminate Davidson iterations when the norm of the residual vector is below 10^{-n}.

RECOMMENDATION:

Use the default.

INTERNAL_STABILITY_ROOTS

Number of lowest Hessian eigenvalues to solve for.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n Solve for n lowest eigenvalues.

RECOMMENDATION:

Use the default.

Example 4.18 Unrestricted SCF calculation of triplet B_2 using B97M-V/6-31g with the GDM algorithm. A displacement is performed when the first solution is characterized as a saddle point, and the second SCF gives a stable solution.

```
$molecule
   0 3
   b
   b 1 R

   R = 1.587553
$end

$rem
   METHOD               b97m-v
   BASIS                6-31g
   UNRESTRICTED         true
   THRESH               14
   SCF_FINAL_PRINT      1
   SCF_ALGORITHM        gdm
   SCF_CONVERGENCE      8
   INTERNAL_STABILITY   true  ! turn on internal stability analysis
   FD_MAT_VEC_PROD      false ! use finite-diff for the vv10 part only
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

4.5.18 Orbital Energy Scaling in Canonical ROSCF Calculations

For high-spin half-filled open-shell systems, canonical ROSCF calculations according to Plakhutin and Davidson[68] can be performed by setting ROSCF_DODS = TRUE. The canonical ROSCF method provides the same total wavefunction and, thus, total energy and correct $\langle \hat{S}^2 \rangle$ behavior as "conventional" ROSCF. Other than "conventional" ROSCF, it features two different sets of orbitals for α and β electrons, as unrestrcited SCF does, which fulfill Koopmans' theorem. Because of this, canonical ROSCF is also called ROSCF-DODS, where DODS stands for "different orbitals for different spins".

In general, ROSCF calculations can be hard to converge, often because of violations of the aufbau principle. In ROSCF calculations, these situations can be avoided when applying an intermediary scaling to the open-shell diagonal blocks of the α and β RO Fock matrices, chosen such that the "correct" orbital ordering is restored during the SCF iterations. The procedure is similar to level-shifting (Section 4.5.5) but much simpler. It exploits the fact that in ROSCF calculations, the diagonal blocks of the RO Fock matrices can be arbitrarily scaled without affecting the total wavefunction.

As an example, consider a system with a single unpaired α electron, where the highest α MO belonging to the doubly-occupied orbital space and the lowest α MO belonging to the (doubly-) unoccupied orbital space feature orbital energies of -0.5 and 0.5, respectively. If the unpaired electron has an orbital energy of -0.6, it might be arbitrarily shifted to, e.g., 0.3 by scaling its energy with a factor of 0.5, corresponding to a setting of ROSCF_DIAG_SCALE_A = 50, which makes life for the SCF algorithm easier.

The obvious caveat of this procedure is that the converged orbital energies need to be known in order to choose the correct scaling factors. In practice, one can try different combinations of ROHF_DIAG_SCALE_A and ROHF_DIAG_SCALE_B if poor convergence behavior is encountered.

4.5.18.1 Job Control

ROSCF_DIAG_SCALE_A

When performing canonical restricted open-shell SCF calculations, scale the open-shell diagonal α Fock matrix block (and, thus, singly occupied α orbital energies) during the SCF iterations in order to mitigate aufbau principle violations in case of poorly converging systems. After convergence, proper canonical orbital energies are restored.

TYPE:

INTEGER

DEFAULT:

50 Scale α orbital energies by a factor of 0.5 which often leads to improved convergence behavior.

OPTIONS:

n Scale α orbital energies by a factor of $n \cdot 10^-2$.

RECOMMENDATION:

In case of convergence issues, try series of different values, e.g., ± 800, ± 400, ± 200, ± 100, ± 50, ± 20, ± 10.

ROSCF_DIAG_SCALE_B

The same as ROSCF_DIAG_SCALE_A, except that singly unpccupied β orbital energies are manipulated during the SCF iterations.

TYPE:

INTEGER

DEFAULT:

50 Scale β orbital energies by a factor of 0.5.

OPTIONS:

n Scale β orbital energies by a factor of $n \cdot 10^-2$.

RECOMMENDATION:

See ROSCF_DIAG_SCALE_A for recommendations.

4.6 Large Molecules and Linear Scaling Methods

4.6.1 Introduction

Construction of the effective Hamiltonian, or Fock matrix, has traditionally been the rate-determining step in self-consistent field calculations, due primarily to the cost of two-electron integral evaluation, even with the efficient methods available in Q-CHEM (see Appendix A). However, for large enough molecules, significant speedups are possible by employing linear-scaling methods for each of the nonlinear terms that can arise. Linear scaling means that if the molecule size is doubled, then the computational effort likewise only doubles. There are three computationally significant terms:

- Electron-electron Coulomb interactions, for which Q-CHEM incorporates the Continuous Fast Multipole Method (CFMM) discussed in section 4.6.2
- Exact exchange interactions, which arise in hybrid DFT calculations and Hartree-Fock calculations, for which Q-CHEM incorporates the LinK method discussed in section 4.6.3 below.
- Numerical integration of the exchange and correlation functionals in DFT calculations, which we have already discussed in section 5.5.

Q-CHEM supports energies and efficient analytical gradients for all three of these high performance methods to permit structure optimization of large molecules, as well as relative energy evaluation. Note that analytical second derivatives of SCF energies do not exploit these methods at present.

For the most part, these methods are switched on automatically by the program based on whether they offer a significant speedup for the job at hand. Nevertheless it is useful to have a general idea of the key concepts behind each of these algorithms, and what input options are necessary to control them. That is the primary purpose of this section, in addition to briefly describing two more conventional methods for reducing computer time in large calculations in Section 4.6.4.

There is one other computationally significant step in SCF calculations, and that is diagonalization of the Fock matrix, once it has been constructed. This step scales with the cube of molecular size (or basis set size), with a small pre-factor. So, for large enough SCF calculations (very roughly in the vicinity of 2000 basis functions and larger), diagonalization becomes the rate-determining step. The cost of cubic scaling with a small pre-factor at this point exceeds the cost of the linear scaling Fock build, which has a very large pre-factor, and the gap rapidly widens thereafter. This sets an effective upper limit on the size of SCF calculation for which Q-CHEM is useful at several thousand basis functions.

4.6.2 Continuous Fast Multipole Method (CFMM)

The quantum chemical Coulomb problem, perhaps better known as the DFT bottleneck, has been at the forefront of many research efforts throughout the 1990s. The quadratic computational scaling behavior conventionally seen in the construction of the Coulomb matrix in DFT or HF calculations has prevented the application of *ab initio* methods to molecules containing many hundreds of atoms. Q-CHEM Inc., in collaboration with White and Head-Gordon at the University of California at Berkeley, and Gill now at the Australian National University, were the first to develop the generalization of Greengard's Fast Multipole Method[37] (FMM) to continuous charged matter distributions in the form of the CFMM, which is the first linear scaling algorithm for DFT calculations. This initial breakthrough has since lead to an increasing number of linear scaling alternatives and analogies, but for Coulomb interactions, the CFMM remains state of the art. There are two computationally intensive contributions to the Coulomb interactions which we discuss in turn:

- Long-range interactions, which are treated by the CFMM
- Short-range interactions, corresponding to overlapping charge distributions, which are treated by a specialized "J-matrix engine" together with Q-CHEM's state-of-the art two-electron integral methods.

The Continuous Fast Multipole Method was the first implemented linear scaling algorithm for the construction of the **J** matrix. In collaboration with Q-CHEM Inc., Dr. Chris White began the development of the CFMM by more efficiently deriving[100] the original Fast Multipole Method before generalizing it to the CFMM.[104] The generalization applied by White *et al.* allowed the principles underlying the success of the FMM to be applied to *arbitrary* (subject to constraints in evaluating the related integrals) continuous, but localized, matter distributions. White and coworkers further improved the underlying CFMM algorithm,[101,102] then implemented it efficiently,[105] achieving performance that is an order of magnitude faster than some competing implementations.

The success of the CFMM follows similarly with that of the FMM, in that the charge system is subdivided into a hierarchy of boxes. Local charge distributions are then systematically organized into multipole representations so that each distribution interacts with local expansions of the potential due to all distant charge distributions. Local and distant distributions are distinguished by a well-separated (WS) index, which is the number of boxes that must separate two collections of charges before they may be considered distant and can interact through multipole expansions; near-field interactions must be calculated directly. In the CFMM each distribution is given its own WS index and is sorted on the basis of the WS index, and the position of their space centers. The implementation in Q-CHEM has allowed the efficiency gains of contracted basis functions to be maintained.

The CFMM algorithm can be summarized in five steps:

1. Form and translate multipoles.
2. Convert multipoles to local Taylor expansions.
3. Translate Taylor information to the lowest level.
4. Evaluate Taylor expansions to obtain the far-field potential.
5. Perform direct interactions between overlapping distributions.

Accuracy can be carefully controlled by due consideration of tree depth, truncation of the multipole expansion and the definition of the extent of charge distributions in accordance with a rigorous mathematical error bound. As a rough guide, 10 poles are adequate for single point energy calculations, while 25 poles yield sufficient accuracy for gradient calculations. Subdivision of boxes to yield a one-dimensional length of about 8 boxes works quite well for systems of up to about one hundred atoms. Larger molecular systems, or ones which are extended along one dimension, will benefit from an increase in this number. The program automatically selects an appropriate number of boxes by default.

For the evaluation of the remaining short-range interactions, Q-CHEM incorporates efficient J-matrix engines, originated by White and Head-Gordon.[103] These are analytically exact methods that are based on standard two-electron integral methods, but with an interesting twist. If one knows that the two-electron integrals are going to be summed into a Coulomb matrix, one can ask whether they are in fact the most efficient intermediates for this specific task. Or, can one instead find a more compact and computationally efficient set of intermediates by folding the density matrix into the recurrence relations for the two-electron integrals. For integrals that are not highly contracted (*i.e.*, are not linear combinations of more than a few Gaussians), the answer is a dramatic yes. This is the basis of the J-matrix approach, and Q-CHEM includes the latest algorithm developed by Yihan Shao working with Martin Head-Gordon at Berkeley for this purpose. Shao's J-engine is employed for both energies[77] and forces,[78] and gives substantial speedups relative to the use of two-electron integrals without any approximation—roughly a factor of 10 for energies and 30 for forces at the level of an uncontracted $dddd$ shell quartet, and increasing with angular momentum). Its use is automatically selected for integrals with low degrees of contraction, while regular integrals are employed when the degree of contraction is high, following the state of the art PRISM approach of Gill and coworkers.[5]

The CFMM is controlled by the following input parameters:

CFMM_ORDER

Controls the order of the multipole expansions in CFMM calculation.

TYPE:

INTEGER

DEFAULT:

15 For single point SCF accuracy

25 For tighter convergence (optimizations)

OPTIONS:

n Use multipole expansions of order n

RECOMMENDATION:

Use the default.

GRAIN

Controls the number of lowest-level boxes in one dimension for CFMM.

TYPE:

INTEGER

DEFAULT:

-1 Program decides best value, turning on CFMM when useful

OPTIONS:

-1 Program decides best value, turning on CFMM when useful

1 Do not use CFMM

$n \geq 8$ Use CFMM with n lowest-level boxes in one dimension

RECOMMENDATION:

This is an expert option; either use the default, or use a value of 1 if CFMM is not desired.

4.6.3 Linear Scaling Exchange (LinK) Matrix Evaluation

Hartree-Fock calculations and the popular hybrid density functionals such as B3LYP also require two-electron integrals to evaluate the exchange energy associated with a single determinant. There is no useful multipole expansion for the exchange energy, because the bra and ket of the two-electron integral are coupled by the density matrix, which carries the effect of exchange. Fortunately, density matrix elements decay exponentially with distance for systems that have a HOMO/LUMO gap.[75] The better the insulator, the more localized the electronic structure, and the faster the rate of exponential decay. Therefore, for insulators, there are only a linear number of numerically significant contributions to the exchange energy. With intelligent numerical thresholding, it is possible to rigorously evaluate the exchange matrix in linear scaling effort. For this purpose, Q-CHEM contains the linear scaling K (LinK) method[67] to evaluate both exchange energies and their gradients[66] in linear scaling effort (provided the density matrix is highly sparse). The LinK method essentially reduces to the conventional direct SCF method for exchange in the small molecule limit (by adding no significant overhead), while yielding large speedups for (very) large systems where the density matrix is indeed highly sparse. For full details, we refer the reader to the original papers.[66,67] LinK can be explicitly requested by the following option (although Q-CHEM automatically switches it on when the program believes it is the preferable algorithm).

LIN_K

Controls whether linear scaling evaluation of exact exchange (LinK) is used.

TYPE:

LOGICAL

DEFAULT:

Program chooses, switching on LinK whenever CFMM is used.

OPTIONS:

TRUE Use LinK

FALSE Do not use LinK

RECOMMENDATION:

Use for HF and hybrid DFT calculations with large numbers of atoms.

Example 4.19 Q-CHEM input for a large single point energy calculation. The CFMM is switched on automatically when LinK is requested.

```
$comment
   HF/3-21G single point calculation on a large molecule
   read in the molecular coordinates from file
$end

$molecule
   read base_pair.inp
$end

$rem
   METHOD     HF      Hartree-Fock
   BASIS      sto-3g  Basis set
   LIN_K      TRUE    Calculate K using LinK
$end
```

4.6.4 Incremental and Variable Thresh Fock Matrix Building

The use of a variable integral threshold, operating for the first few cycles of an SCF, is justifiable on the basis that the MO coefficients are usually of poor quality in these cycles. In Q-CHEM, the integrals in the first iteration are calculated at a threshold of 10^{-6} (for an anticipated final integral threshold greater than, or equal to 10^{-6}) to ensure the error in the first iteration is solely sourced from the poor MO guess. Following this, the integral threshold used is computed as

$$t = 10^{-\text{VARTHRESH}} \times (\text{DIIS error}) \tag{4.48}$$

where the DIIS error is that calculated from the previous cycle, VARTHRESH is the variable threshold set by the program (by default) and t is the temporary threshold used for integral evaluation. Each cycle requires recalculation of all integrals. The variable integral threshold procedure has the greatest impact in early SCF cycles.

In an incremental Fock matrix build,[6,16,41,75] $\mathbf{F}$ is computed recursively as

$$\mathbf{F^m} = \mathbf{F^{m-1}} + \Delta\mathbf{J^{m-1}} - \frac{1}{2}\Delta\mathbf{K^{m-1}} \tag{4.49}$$

where m is the SCF cycle, and $\Delta\mathbf{J}^m$ and $\Delta\mathbf{K}^m$ are computed using the difference density

$$\Delta\mathbf{P^m} = \mathbf{P^m} - \mathbf{P^{m-1}} \tag{4.50}$$

Using Schwartz integrals and elements of the difference density, Q-CHEM is able to determine at each iteration which ERIs are required, and if necessary, recalculated. As the SCF nears convergence, $\Delta\mathbf{P}^m$ becomes sparse and the number of ERIs that need to be recalculated declines dramatically, saving the user large amounts of computational time.

Incremental Fock matrix builds and variable thresholds are only used when the SCF is carried out using the direct SCF algorithm and are clearly complementary algorithms. These options are controlled by the following input parameters, which are only used with direct SCF calculations.

INCFOCK

Iteration number after which the incremental Fock matrix algorithm is initiated

TYPE:

INTEGER

DEFAULT:

1 Start INCFOCK after iteration number 1

OPTIONS:

User-defined (0 switches INCFOCK off)

RECOMMENDATION:

May be necessary to allow several iterations before switching on INCFOCK.

VARTHRESH

Controls the temporary integral cut-off threshold, $t = 10^{-\mathrm{VARTHRESH}} \times$ (DIIS error)

TYPE:

INTEGER

DEFAULT:

0 Turns VARTHRESH off

OPTIONS:

n User-defined threshold

RECOMMENDATION:

3 has been found to be a practical level, and can slightly speed up SCF evaluation.

Example 4.20 Q-CHEM input for a large single point energy calculation. This would be appropriate for a medium-sized molecule, but for truly large calculations, the CFMM and LinK algorithms are far more efficient.

```
$comment
   HF/3-21G single point calculation on a large molecule
   read in the molecular coordinates from file
$end

$molecule
   read base_pair.inp
$end

$rem
   METHOD          HF      Hartree-Fock
   BASIS           3-21G   Basis set
   INCFOCK         5       Incremental Fock after 5 cycles
   VARTHRESH       3       1.0d-03 variable threshold
   MAX_SCF_CYCLES  100
$end
```

4.6.5 Fourier Transform Coulomb Method

The Coulomb part of the DFT calculations using ordinary Gaussian representations can be sped up dramatically using plane waves as a secondary basis set by replacing the most costly analytical electron repulsion integrals with numerical integration techniques. The main advantages to keeping the Gaussians as the primary basis set is that the diagonalization step is much faster than using plane waves as the primary basis set, and all electron calculations can be performed analytically.

The Fourier Transform Coulomb (FTC) technique[33,34] is precise and tunable and all results are practically identical with the traditional analytical integral calculations. The FTC technique is at least 2–3 orders of magnitude more accurate then other popular plane wave based methods using the same energy cutoff. It is also at least 2–3 orders of magnitude more accurate than the density fitting (resolution-of-identity) technique. Recently, an efficient way to implement the forces of the Coulomb energy was introduced,[31] and a new technique to localize filtered core functions. Both of these features have been implemented within Q-CHEM and contribute to the efficiency of the method.

The FTC method achieves these spectacular results by replacing the analytical integral calculations, whose computational costs scales as $\mathcal{O}(N^4)$ (where N is the number of basis function) with procedures that scale as only $\mathcal{O}(N^2)$. The asymptotic scaling of computational costs with system size is linear versus the analytical integral evaluation which is quadratic. Research at Q-CHEM Inc. has yielded a new, general, and very efficient implementation of the FTC method which work in tandem with the J-engine and the CFMM (Continuous Fast Multipole Method) techniques.[32]

In the current implementation the speed-ups arising from the FTC technique are moderate when small or medium Pople basis sets are used. The reason is that the J-matrix engine and CFMM techniques provide an already highly efficient solution to the Coulomb problem. However, increasing the number of polarization functions and, particularly, the number of diffuse functions allows the FTC to come into its own and gives the most significant improvements. For instance, using the 6-311G+(df,pd) basis set for a medium-to-large size molecule is more affordable today then before. We found also significant speed ups when non–Pople basis sets are used such as cc-pvTZ. The FTC energy and gradients calculations are implemented to use up to f-type basis functions.

FTC

Controls the overall use of the FTC.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use FTC in the Coulomb part

1 Use FTC in the Coulomb part

RECOMMENDATION:

Use FTC when bigger and/or diffuse basis sets are used.

FTC_SMALLMOL

Controls whether or not the operator is evaluated on a large grid and stored in memory to speed up the calculation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use a big pre-calculated array to speed up the FTC calculations

0 Use this option to save some memory

RECOMMENDATION:

Use the default if possible and use 0 (or buy some more memory) when needed.

FTC_CLASS_THRESH_ORDER

Together with FTC_CLASS_THRESH_MULT, determines the cutoff threshold for included a shell-pair in the dd class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Logarithmic part of the FTC classification threshold. Corresponds to 10^{-5}

OPTIONS:

n User specified

RECOMMENDATION:

Use the default.

FTC_CLASS_THRESH_MULT

Together with FTC_CLASS_THRESH_ORDER, determines the cutoff threshold for included a shell-pair in the dd class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Multiplicative part of the FTC classification threshold. Together with the default value of the FTC_CLASS_THRESH_ORDER this leads to the 5×10^{-5} threshold value.

OPTIONS:

n User specified.

RECOMMENDATION:

Use the default. If diffuse basis sets are used and the molecule is relatively big then tighter FTC classification threshold has to be used. According to our experiments using Pople-type diffuse basis sets, the default 5×10^{-5} value provides accurate result for an alanine5 molecule while 1×10^{-5} threshold value for alanine10 and 5×10^{-6} value for alanine15 has to be used.

4.6.6 Resolution of the Identity Fock Matrix Methods

Evaluation of the Fock matrix (both Coulomb, J, and exchange, K, pieces) can be sped up by an approximation known as the resolution-of-the-identity (RI-JK). Essentially, the full complexity in common basis sets required to describe chemical bonding is not necessary to describe the mean-field Coulomb and exchange interactions between electrons. That is, ρ in the left side of

$$(\mu\nu|\rho) = \sum_{\lambda\sigma}(\mu\nu|\lambda\sigma)\mathbf{P}_{\lambda\sigma} \tag{4.51}$$

is much less complicated than an individual $\lambda\sigma$ function pair. The same principle applies to the FTC method in subsection 4.6.5, in which case the slowly varying piece of the electron density is replaced with a plane-wave expansion.

With the RI-JK approximation, the Coulomb interactions of the function pair $\rho(r) = \lambda\sigma(r)P_{\lambda\sigma}$ are fit by a smaller set of atom-centered basis functions. In terms of J:

$$\sum_{\lambda\sigma}\int \mathrm{d}^3\mathbf{r}_1 P_{\lambda\sigma}\lambda\sigma(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1-\mathbf{r}|} \approx \sum_{K}\int \mathrm{d}^3\mathbf{r}_1 P_K K(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1-\mathbf{r}|} \tag{4.52}$$

The coefficients P_K must be determined to accurately represent the potential. This is done by performing a least-squared minimization of the difference between $P_{\lambda\sigma}\lambda\sigma(\mathbf{r}_1)$ and $P_K K(\mathbf{r}_1)$, *with differences measured by the Coulomb metric*. This requires a matrix inversion over the space of auxiliary basis functions, which may be done rapidly by Cholesky decomposition.

The RI-J can be invoked by either setting RI_J to be true, or (since Q-CHEM 5.2) specifying auxiliary basis set for J using AUX_BASIS_J.

The RI method applied to the Fock matrix may be further enhanced by performing *local* fitting of a density or function pair element. This is the basis of the atomic-RI method (ARI), which has been developed for both Coulomb (J) matrix[82] and exchange (K) matrix evaluation.[83] In ARI, only nearby auxiliary functions $K(\mathbf{r})$ are employed to fit the target function. This reduces the asymptotic scaling of the matrix-inversion step as well as that of many intermediate steps in the digestion of RI integrals. Briefly, atom-centered auxiliary functions on nearby atoms are only used if they are within the "outer" radius (R_1) of the fitting region. Between R_1 and the "inner" radius (R_0), the amplitude of interacting auxiliary functions is smoothed by a function that goes from zero to one and has continuous derivatives. To optimize efficiency, the van der Waals radius of the atom is included in the cutoff so that smaller atoms are dropped from the fitting radius sooner. The values of R_0 and R_1 are specified as REM variables as described below.

RI_J

Toggles the use of the RI algorithm to compute J.

TYPE:

LOGICAL

DEFAULT:

FALSE RI will not be used to compute J.

OPTIONS:

TRUE Turn on RI for J.

RECOMMENDATION:

For large (especially 1D and 2D) molecules the approximation may yield significant improvements in Fock evaluation time when used with ARI.

RI_K

Toggles the use of the RI algorithm to compute K.

TYPE:

LOGICAL

DEFAULT:

FALSE RI will not be used to compute K.

OPTIONS:

TRUE Turn on RI for K.

RECOMMENDATION:

For large (especially 1D and 2D) molecules the approximation may yield significant improvements in Fock evaluation time when used with ARI.

ARI

Toggles the use of the atomic resolution-of-the-identity (ARI) approximation.

TYPE:

LOGICAL

DEFAULT:

FALSE ARI will not be used by default for an RI-JK calculation.

OPTIONS:

TRUE Turn on ARI.

RECOMMENDATION:

For large (especially 1D and 2D) molecules the approximation may yield significant improvements in Fock evaluation time.

ARI_R0

Determines the value of the inner fitting radius (in Ångstroms)

TYPE:

INTEGER

DEFAULT:

4 A value of 4 Å will be added to the atomic van der Waals radius.

OPTIONS:

n User defined radius.

RECOMMENDATION:

For some systems the default value may be too small and the calculation will become unstable.

ARI_R1

Determines the value of the outer fitting radius (in Ångstroms)

TYPE:

INTEGER

DEFAULT:

5 A value of 5 Å will be added to the atomic van der Waals radius.

OPTIONS:

n User defined radius.

RECOMMENDATION:

For some systems the default value may be too small and the calculation will become unstable. This value also determines, in part, the smoothness of the potential energy surface.

4.6.7 PARI-K Fast Exchange Algorithm

PARI-K[61] is an algorithm that significantly accelerates the construction of the exchange matrix in Hartree-Fock and hybrid density functional theory calculations with large basis sets. The speedup is made possible by fitting products of atomic orbitals using only auxiliary basis functions found on their respective atoms. The PARI-K implementation in Q-CHEM is an efficient MO-basis formulation similar to the AO-basis formulation of Merlot *et al.*[63] PARI-K is highly recommended for calculations using basis sets of size augmented triple-zeta or larger, and should be used in conjunction with the standard RI-J algorithm for constructing the Coulomb matrix.[96] The exchange fitting basis sets of Weigend[96] (cc-pVTZ-JK and cc-pVQZ-JK) are recommended for use in conjunction with PARI-K. The errors associated with the PARI-K approximation appear to be only slightly worse than standard RI-HF.[63]

PARI_K

Controls the use of the PARI-K approximation in the construction of the exchange matrix

TYPE:

LOGICAL

DEFAULT:

FALSE Do not use PARI-K.

OPTIONS:

TRUE Use PARI-K.

RECOMMENDATION:

Use for basis sets aug-cc-pVTZ and larger.

4.6.8 occ-RI-K Exchange Algorithm

The occupied orbital RI-K (occ-RI-K) algorithm[60] is a new scheme for building the exchange matrix (K) partially in the MO basis using the RI approximation. occ-RI-K typically matches current alternatives in terms of both the accuracy (energetics identical to standard RI-K) and convergence (essentially unchanged relative to conventional methods). On the other hand, this algorithm exhibits significant speedups over conventional integral evaluation (14x) and standard RI-K (3.3x) for a test system, a graphene fragment ($C_{68}H_{22}$) using cc-pVQZ basis set (4400 basis functions), whereas the speedup increases with the size of the AO basis set. Thus occ-RI-K helps to make larger basis set hybrid DFT calculations more feasible, which is quite desirable for achieving improved accuracy in DFT calculations with modern functionals.

The idea of the occ-RI-K formalism comes from a simple observation that the exchange energy E_K and its gradient can be evaluated from the diagonal elements of the exchange matrix in the occupied-occupied block K_{ii}, and occupied-virtual block K_{ia}, respectively, rather than the full matrix in the AO representation, $K_{\mu\nu}$. Mathematically,

$$\begin{aligned} E_K &= -\sum_{\mu\nu} P_{\mu\nu} K_{\mu\nu} \\ &= -\sum_{\mu\nu} c_{\mu i} K_{\mu\nu} c_{\nu i} \\ &= -\sum_{i} K_{ii} \end{aligned} \tag{4.53}$$

and

$$\frac{\partial E_K}{\partial \Delta_{ai}} = 2K_{ai} \tag{4.54}$$

where Δ is a skew-symmetric matrix used to parameterize the unitary transformation U, which represents the variations of the MO coefficients as follows:

$$U = e^{(\Delta - \Delta^T)}. \tag{4.55}$$

From Eq. 4.53 and 4.54 it is evident that the exchange energy and gradient need just $K_{i\nu}$ rather than $K_{\mu\nu}$.

In regular RI-K one has to compute two quartic terms,[96] whereas there are three quartic terms for the occ-RI-K algorithm. The speedup of the latter with respect to former can be explained from the following ratio of operations; refer to Ref. 60 for details.

$$\frac{\text{\# of RI-K quartic operations}}{\text{\# of occ-RI-K quartic operations}} = \frac{oNX^2 + oN^2X}{o^2X^2 + o^2NX + o^2NX} = \frac{N(X+N)}{o(X+2N)} \tag{4.56}$$

With a conservative approximation of $X \approx 2N$, the speedup is $\frac{3}{4}(N/o)$. The occ-RI-K algorithm also involves some cubic steps which should be negligible in the very large molecule limit. Tests in the Ref. 60 suggest that occ-RI-K for small systems with large basis will gain less speed than a large system with small basis, because the cubic terms will be more dominant for the former than the latter case.

In the course of SCF iteration, the occ-RI-K method does not require us to construct the exact Fock matrix explicitly. Rather, $k_{i\nu}$ contributes to the Fock matrix in the mixed MO and AO representations ($F_{i\nu}$) and yields orbital gradient and DIIS error vectors for converging SCF. On the other hand, since occ-RI-K does not provide exactly the same unoccupied eigenvalues, the diagonalization updates can differ from the conventional SCF procedure. In Ref. 60, occ-RI-K was found to require, on average, the same number of SCF iterations to converge and to yield accurate energies.

The occ-RI-K can be invoked by either setting OCC_RI_K to be true, or (since Q-CHEM 5.2) specifying auxiliary basis set for K using AUX_BASIS_K.

OCC_RI_K

Controls the use of the occ-RI-K approximation for constructing the exchange matrix

TYPE:

LOGICAL

DEFAULT:

False Do not use occ-RI-K.

OPTIONS:

True Use occ-RI-K.

RECOMMENDATION:

Larger the system, better the performance

4.6.8.1 occ-RI-K for exchange energy gradient evaluation

A very attractive feature of occ-RI-K framework is that one can compute the exchange energy gradient with respect to nuclear coordinates with the same leading quartic-scaling operations as the energy calculation.

The occ-RI-K formulation yields the following formula for the gradient of exchange energy in global Coulomb-metric RI:

$$\begin{aligned} E_K^x &= (ij|ij)^x \\ &= \sum_{\mu\nu P}\sum_{ij} c_{\mu i}c_{\nu j}C_{ij}^P(\mu\nu|P)^x - \sum_{RS}\sum_{ij} C_{ij}^R C_{ij}^S (R|S)^x. \end{aligned} \tag{4.57}$$

The superscript x represents the derivative with respect to a nuclear coordinate. Note that the derivatives of the MO coefficients $c_{\mu i}$ are not included here, because they are already included in the total energy derivative calculation by Q-CHEM via the derivative of the overlap matrix.

In Eq. 4.57, the construction of the density fitting coefficients ($C_{\mu\nu}^P$) has the worst scaling of $\mathcal{O}(M^4)$ because it involves MO to AO back transformations:

$$C_{\mu\nu}^P = \sum_{ij} c_{\mu i}c_{\nu j}C_{ij}^P \tag{4.58}$$

where the operation cost is $o^2NX + o[\mathrm{NB2}]X$.

RI_K_GRAD

Turn on the nuclear gradient calculations

TYPE:

LOGICAL

DEFAULT:

FALSE Do not invoke occ-RI-K based gradient

OPTIONS:

TRUE Use occ-RI-K based gradient

RECOMMENDATION:

Use "RI_J false"

Example 4.21 Q-CHEM input for a energy and gradient calculations with occ-RI-K method.

```
$molecule
   0   1
   C     0.0000000     0.3057430     5.7138876
   C     0.0000000    -0.5442831     4.4256275
   C     0.0000000     0.3675825     3.1787857
   C     0.0000000    -0.4853210     1.8908707
   C     0.0000000     0.4264823     0.6439435
   C     0.0000000    -0.4264823    -0.6439435
   C     0.0000000     0.4853210    -1.8908707
   C     0.0000000    -0.3675825    -3.1787857
   C     0.0000000     0.5442831    -4.4256275
   C     0.0000000    -0.3057430    -5.7138876
   H     0.8973039     0.9418895     5.7433715
   H    -0.8973039     0.9418895     5.7433715
   H    -0.8965960    -1.1827891     4.4155828
   H     0.8965960    -1.1827891     4.4155828
   H     0.8966512     1.0057416     3.1940146
   H    -0.8966512     1.0057416     3.1940146
   H    -0.8966512    -1.1234851     1.8761493
   H     0.8966512    -1.1234851     1.8761493
   H     0.8966507     1.0646443     0.6587590
   H    -0.8966507     1.0646443     0.6587590
   H    -0.8966507    -1.0646443    -0.6587590
   H     0.8966507    -1.0646443    -0.6587590
   H     0.8966512     1.1234851    -1.8761493
   H    -0.8966512     1.1234851    -1.8761493
   H    -0.8966512    -1.0057416    -3.1940146
   H     0.8966512    -1.0057416    -3.1940146
   H     0.8965960     1.1827891    -4.4155828
   H    -0.8965960     1.1827891    -4.4155828
   H     0.8973039    -0.9418895    -5.7433715
   H     0.0000000     0.3580832    -6.5913113
   H    -0.8973039    -0.9418895    -5.7433715
   H     0.0000000    -0.3580832     6.5913113
$end

$rem
   JOBTYPE          force
   EXCHANGE         HF
   BASIS            cc-pVTZ
   AUX_BASIS        cc-pVTZ-JK
   OCC_RI_K         1
   RI_K_GRAD        1
   INCFOCK          0
   PURECART         1111
$end
```

4.7 Dual-Basis Self-Consistent Field Calculations

4.7.1 Introduction

The dual-basis approximation[26,58,84,86–88] to self-consistent field (HF or DFT) energies provides an efficient means for obtaining large basis set effects at vastly less cost than a full SCF calculation in a large basis set. First, a full SCF calculation is performed in a chosen small basis (specified by BASIS2). Second, a single SCF-like step in the larger,

target basis (specified, as usual, by BASIS) is used to perturbatively approximate the large basis energy. This correction amounts to a first-order approximation in the change in density matrix, after the single large-basis step:

$$E_{\text{total}} = E_{\text{small basis}} + \text{tr}[(\Delta\mathbf{P})\mathbf{F}]_{\text{large basis}} \ . \tag{4.59}$$

Here $\mathbf{F}$ (in the large basis) is built from the converged (small basis) density matrix. Thus, only a single Fock build is required in the large basis set. Currently, HF and DFT energies (SP) as well as analytic first derivatives (FORCE or OPT) are available.

Note: As of version 4.0, first derivatives of unrestricted dual-basis DFT energies—though correct—require a code-efficiency fix. We do not recommend use of these derivatives until this improvement has been made.

Across the G3 set[18,20,21] of 223 molecules, using cc-pVQZ, dual-basis errors for B3LYP are 0.04 kcal/mol (energy) and 0.03 kcal/mol (atomization energy per bond) and are at least an order of magnitude less than using a smaller basis set alone. These errors are obtained at roughly an order of magnitude savings in cost, relative to the full, target-basis calculation.

4.7.2 Dual-Basis MP2

The dual-basis approximation can also be used for the reference energy of a correlated second-order Møller-Plesset (MP2) calculation.[86,88] When activated, the dual-basis HF energy is first calculated as described above; subsequently, the MO coefficients and orbital energies are used to calculate the correlation energy in the large basis. This technique is particularly effective for RI-MP2 calculations (see Section 6.6), in which the cost of the underlying SCF calculation often dominates.

Furthermore, efficient analytic gradients of the DB-RI-MP2 energy have been developed[26] and added to Q-CHEM. These gradients allow for the optimization of molecular structures with RI-MP2 near the basis set limit. Typical computational savings are on the order of 50% (aug-cc-pVDZ) to 71% (aug-cc-pVTZ). Resulting dual-basis errors are only 0.001 Å in molecular structures and are, again, significantly less than use of a smaller basis set alone.

4.7.3 Dual-Basis Dynamics

The ability to compute SCF and MP2 energies and forces at reduced cost makes dual-basis calculations attractive for *ab initio* molecular dynamics simulations, which are described in Section 9.9. Dual-basis BOMD has demonstrated[89] savings of 58%, even relative to state-of-the-art, Fock-extrapolated BOMD. Savings are further increased to 71% for dual-basis RI-MP2 dynamics. Notably, these timings outperform estimates of extended Lagrangian ("Car-Parrinello") dynamics, without detrimental energy conservation artifacts that are sometimes observed in the latter.[43]

Two algorithm improvements make modest but worthwhile improvements to dual-basis dynamics. First, the iterative, small-basis calculation can benefit from Fock matrix extrapolation.[43] Second, extrapolation of the response equations ("Z-vector" equations) for nuclear forces further increases efficiency.[85] (See Section 9.9.) Q-CHEM automatically adjusts to extrapolate in the proper basis set when DUAL_BASIS_ENERGY is activated.

4.7.4 Basis-Set Pairings

We recommend using basis pairings in which the small basis set is a proper subset of the target basis (6-31G into 6-31G*, for example). They not only produce more accurate results; they also lead to more efficient integral screening in both energies and gradients. Subsets for many standard basis sets (including Dunning-style cc-pVXZ basis sets and their augmented analogs) have been developed and thoroughly tested for these purposes. A summary of the pairings is provided in Table 4.7.4; details of these truncations are provided in Figure 4.1.

A new pairing for 6-31G*-type calculations is also available. The 6-4G subset (named r64G in Q-CHEM) is a subset by *primitive* functions and provides a smaller, faster alternative for this basis set regime.[84] A case-dependent switch in the projection code (still OVPROJECTION) properly handles 6-4G. For DB-HF, the calculations proceed as described above. For DB-DFT, empirical scaling factors (see Ref. 84 for details) are applied to the dual-basis correction. This scaling is handled automatically by the code and prints accordingly.

As of Q-CHEM version 3.2, the basis set projection code has also been adapted to properly account for linear dependence,[88] which can often be problematic for large, augmented (aug-cc-pVTZ, *etc.*) basis set calculations. The same standard keyword (LIN_DEP_THRESH) is used to determine linear dependence in the projection code. Because of the scheme used to account for linear dependence, only proper-subset pairings are now allowed.

Like single-basis calculations, user-specified general or mixed basis sets may be employed (see Chapter 8) with dual-basis calculations. The target basis specification occurs in the standard *$basis* section. The smaller, secondary basis is placed in a similar *$basis*2 section; the syntax within this section is the same as the syntax for *$basis*. General and mixed small basis sets are activated by BASIS2 = BASIS2_GEN and BASIS2 = BASIS2_MIXED, respectively.

BASIS	**BASIS2**
cc-pVTZ	rcc-pVTZ
cc-pVQZ	rcc-pVQZ
aug-cc-pVDZ	racc-pVDZ
aug-cc-pVTZ	racc-pVTZ
aug-cc-pVQZ	racc-pVQZ
6-31G*	r64G, 6-31G
6-31G**	r64G, 6-31G
6-31++G**	6-31G*
6-311++G(3df,3pd)	6-311G*, 6-311+G*

Table 4.2: Summary and nomenclature of recommended dual-basis pairings

4.7.5 Job Control and Example

Dual-basis calculations are controlled with the following *$rem*. DUAL_BASIS_ENERGY turns on the dual-basis approximation. Note that use of BASIS2 without DUAL_BASIS_ENERGY *only* uses basis set projection to generate the initial guess and does not invoke the dual-basis approximation (see Section 4.4.5). OVPROJECTION is used as the default projection mechanism for dual-basis calculations; it is not recommended that this be changed. Specification of SCF variables (*e.g.*, THRESH) will apply to calculations in both basis sets.

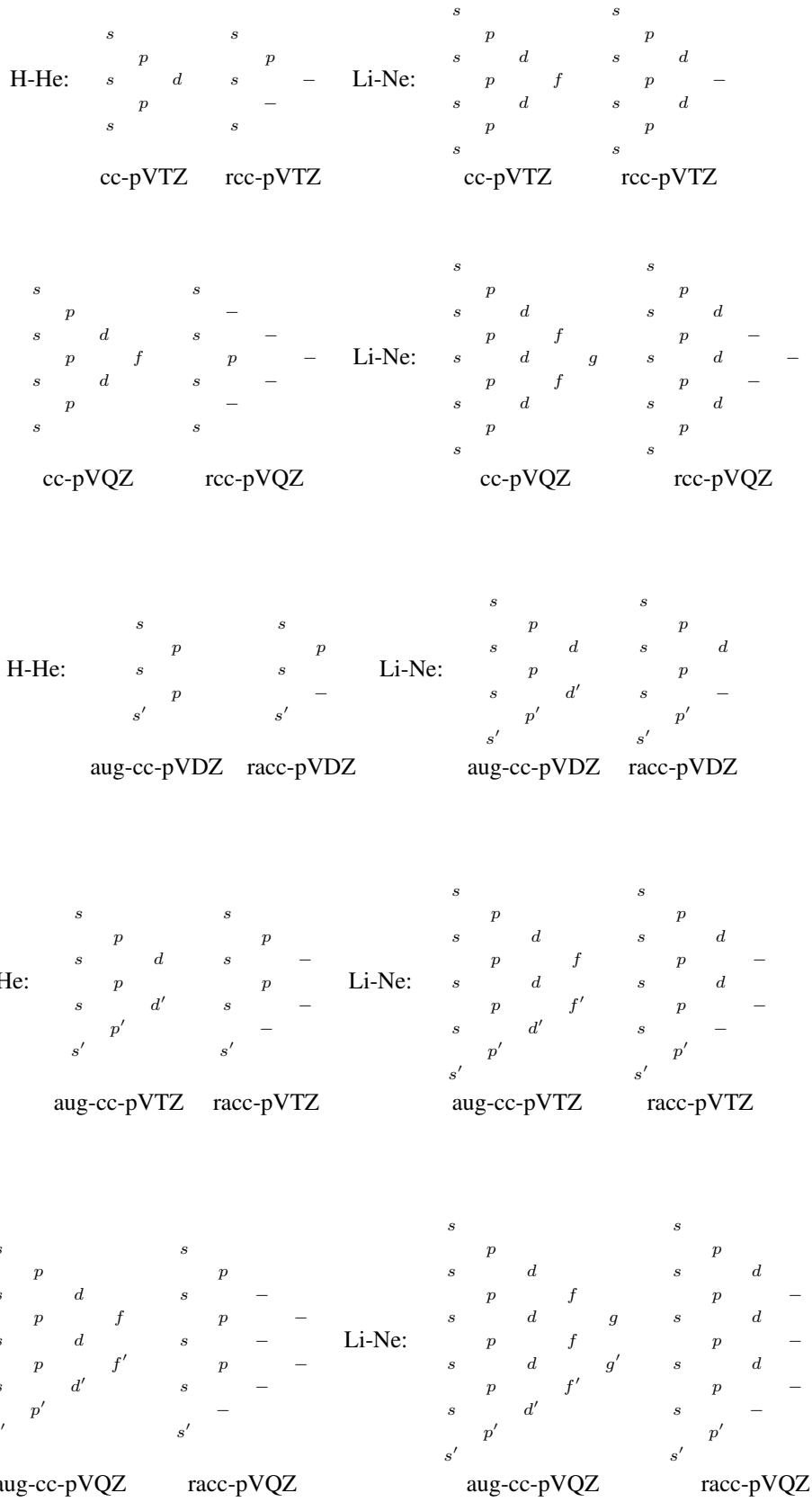


Figure 4.1: Structure of the truncated basis set pairings for cc-pV(T,Q)Z and aug-cc-pV(D,T,Q)Z. The most compact functions are listed at the top. Primed functions depict diffuse function augmentation. Dashes indicate eliminated functions, relative to the paired standard basis set. In each case, the truncations for hydrogen and heavy atoms are shown, along with the nomenclature used in Q-CHEM.

DUAL_BASIS_ENERGY

Activates dual-basis SCF (HF or DFT) energy correction.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Analytic first derivative available for HF and DFT (see JOBTYPE)

Can be used in conjunction with MP2 or RI-MP2

See BASIS, BASIS2, BASISPROJTYPE

RECOMMENDATION:

Use dual-basis to capture large-basis effects at smaller basis cost. Particularly useful with RI-MP2, in which HF often dominates. Use only proper subsets for small-basis calculation.

4.7.5.1 Examples

Example 4.22 Input for a dual-basis B3LYP single-point calculation.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   METHOD              b3lyp
   BASIS               6-31G*
   BASIS2              r64G
   DUAL_BASIS_ENERGY   true
$end
```

Example 4.23 Input for a dual-basis B3LYP single-point calculation with a minimal 6-4G small basis.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   JOBTYPE             opt
   METHOD              rimp2
   AUX_BASIS           rimp2-aug-cc-pVDZ
   BASIS               aug-cc-pVDZ
   BASIS2              racc-pVDZ
   DUAL_BASIS_ENERGY   true
$end
```

Example 4.24 Input for a dual-basis RI-MP2 geometry optimization.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   JOBTYPE             opt
   METHOD              rimp2
   AUX_BASIS           rimp2-aug-cc-pVDZ
   BASIS               aug-cc-pVDZ
   BASIS2              racc-pVDZ
   DUAL_BASIS_ENERGY   true
$end
```

Example 4.25 Input for a dual-basis RI-MP2 single-point calculation with mixed basis sets.

```
$molecule
   0 1
   H
   O   1   1.1
   H   2   1.1  1  104.5
$end

$rem
   JOBTYPE             opt
   METHOD              rimp2
   AUX_BASIS           aux_mixed
   BASIS               mixed
   BASIS2              basis2_mixed
   DUAL_BASIS_ENERGY   true
$end

$basis
 H 1
 cc-pVTZ
 ****
 O 2
 aug-cc-pVTZ
 ****
 H 3
 cc-pVTZ
 ****
$end

$basis2
 H 1
 rcc-pVTZ
 ****
 O 2
 racc-pVTZ
 ****
 H 3
 rcc-pVTZ
 ****
$end

$aux_basis
 H 1
 rimp2-cc-pVTZ
 ****
 O 2
 rimp2-aug-cc-pVTZ
 ****
 H 3
 rimp2-cc-pVTZ
 ****
$end
```

4.8 Hartree-Fock and Density-Functional Perturbative Corrections

4.8.1 Introduction

Closely related to the dual-basis approach of Section 4.7, but somewhat more general, is the Hartree-Fock perturbative correction (HFPC) developed by Deng *et al.*.[23,24] An HFPC calculation consists of an iterative HF calculation in a small primary basis followed by a single Fock matrix formation, diagonalization, and energy evaluation in a larger, secondary basis. In the following, we denote a conventional HF calculation by HF/basis, and a HFPC calculation by HFPC/primary/secondary. Using a primary basis of n functions, the restricted HF matrix elements for a $2m$-electron system are

$$F_{\mu\nu} = h_{\mu\nu} + \sum_{\lambda\sigma}^{n} P_{\lambda\sigma} \left[(\mu\nu|\lambda\sigma) - \frac{1}{2}(\mu\lambda|\nu\sigma) \right] \tag{4.60}$$

Solving the Roothaan-Hall equation in the primary basis results in molecular orbitals and an associated density matrix, $\mathbf{P}$. In an HFPC calculation, $\mathbf{P}$ is subsequently used to build a new Fock matrix, $\mathbf{F}^{[1]}$, in a larger secondary basis of N functions

$$F^{[1]}_{ab} = h_{ab} + \sum_{\lambda\sigma}^{n} P_{\lambda\sigma} \left[(ab|\lambda\sigma) - \frac{1}{2}(a\lambda|b\sigma) \right] \tag{4.61}$$

where λ, σ indicate primary basis functions and a, b represent secondary basis functions. Diagonalization of $\mathbf{F}^{[1]}$ affords improved molecular orbitals and an associated density matrix $\mathbf{P}^{[1]}$. The HFPC energy is given by

$$E^{\text{HFPC}} = \sum_{ab}^{N} P^{[1]}_{ab} h_{ab} + \frac{1}{2} \sum_{abcd}^{N} P^{[1]}_{ab} P^{[1]}_{cd} \left[2(ab|cd) - (ac|bd) \right] \tag{4.62}$$

where a, b, c and d represent secondary basis functions. This differs from the DBHF energy evaluation where $\mathbf{PP}^{[1]}$, rather than $\mathbf{P}^{[1]}\mathbf{P}^{[1]}$, is used. The inclusion of contributions that are quadratic in $\mathbf{PP}^{[1]}$ is the key reason for the fact that HFPC is more accurate than DBHF.

Unlike dual-basis HF, HFPC does not require that the small basis be a proper subset of the large basis, and is therefore able to jump between any two basis sets. Benchmark study of HFPC on a large and diverse data set of total and reaction energies demonstrate that, for a range of primary/secondary basis set combinations, the HFPC scheme can reduce the error of the primary calculation by around two orders of magnitude at a cost of about one third that of the full secondary calculation.[23,24]

A density-functional version of HFPC ("DFPC")[25] seeks to combine the low cost of pure DFT calculations using small bases and grids, with the high accuracy of hybrid calculations using large bases and grids. The DFPC approach is motivated by the dual-functional method of Nakajima and Hirao[64] and the dual-grid scheme of Tozer *et al.*[93] Combining these features affords a triple perturbation: to the functional, to the grid, and to the basis set. We call this approach density-functional "triple jumping".

4.8.2 Job Control

HFPC/DFPC calculations are controlled with the following *$rem*. HFPT turns on the HFPC/DFPC approximation. Note that HFPT_BASIS specifies the secondary basis set.

HFPT

Activates HFPC/DFPC calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Single-point energy only

RECOMMENDATION:

Use Dual-Basis to capture large-basis effects at smaller basis cost. See reference for recommended basis set, functional, and grid pairings.

HFPT_BASIS

Specifies the secondary basis in a HFPC/DFPC calculation.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

None

RECOMMENDATION:

See reference for recommended basis set, functional, and grid pairings.

DFPT_XC_GRID

Specifies the secondary grid in a HFPC/DFPC calculation.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

None

RECOMMENDATION:

See reference for recommended basis set, functional, and grid pairings.

DFPT_EXCHANGE

Specifies the secondary functional in a HFPC/DFPC calculation.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

None

RECOMMENDATION:

See reference for recommended basis set, functional, and grid pairings.

4.8.3 Examples

Example 4.26 Input for a HFPC single-point calculation.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   EXCHANGE     hf
   BASIS        cc-pVDZ  ! primary basis
   HFPT_BASIS   cc-pVQZ  ! secondary basis
   PURECART     1111     ! set to purecart of the target basis
   HFPT         true
   GEN_SCFMAN   false    ! runs in the old SCF code
$end
```

Example 4.27 Input for a DFPC single-point calculation.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   METHOD           blyp               ! primary functional
   DFPT_EXCHANGE    b3lyp              ! secondary functional
   DFPT_XC_GRID     00075000302        ! secondary grid
   XC_GRID          0                  ! primary grid
   HFPT_BASIS       6-311++G(3df,3pd)  ! secondary basis
   BASIS            6-311G*            ! primary basis
   PURECART         1111
   HFPT             true
   GEN_SCFMAN       false
$end
```

4.9 Unconventional SCF Calculations

4.9.1 Polarized Atomic Orbital (PAO) Calculations

Polarized atomic orbital (PAO) calculations are an interesting unconventional SCF method, in which the molecular orbitals and the density matrix are not expanded directly in terms of the basis of atomic orbitals. Instead, an intermediate molecule-optimized minimal basis of polarized atomic orbitals (PAOs) is used.[51] The polarized atomic orbitals are defined by an atom-blocked linear transformation from the fixed atomic orbital basis, where the coefficients of the transformation are optimized to minimize the energy, at the same time as the density matrix is obtained in the PAO representation. Thus a PAO-SCF calculation is a constrained variational method, whose energy is above that of a full SCF calculation in the same basis. However, a molecule optimized minimal basis is a very compact and useful representation for purposes of chemical analysis, and it also has potential computational advantages in the context of MP2 or local MP2 calculations, as can be done after a PAO-HF calculation is complete to obtain the PAO-MP2 energy.

PAO-SCF calculations tend to systematically underestimate binding energies (since by definition the exact result is obtained for atoms, but not for molecules). In tests on the G2 database, PAO-B3LYP/6-311+G(2df,p) atomization

energies deviated from full B3LYP/6-311+G(2df,p) atomization energies by roughly 20 kcal/mol, with the error being essentially extensive with the number of bonds. This deviation can be reduced to only 0.5 kcal/mol with the use of a simple non-iterative second order correction for "beyond-minimal basis" effects.[52] The second order correction is evaluated at the end of each PAO-SCF calculation, as it involves negligible computational cost. Analytical gradients are available using PAOs, to permit structure optimization. For additional discussion of the PAO-SCF method and its uses, see the references cited above.

Calculations with PAOs are determined controlled by the following *$rem* variables. PAO_METHOD = PAO invokes PAO-SCF calculations, while the algorithm used to iterate the PAOs can be controlled with PAO_ALGORITHM.

PAO_ALGORITHM

Algorithm used to optimize polarized atomic orbitals (see PAO_METHOD)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use efficient (and riskier) strategy to converge PAOs.

1 Use conservative (and slower) strategy to converge PAOs.

RECOMMENDATION:

None

PAO_METHOD

Controls evaluation of polarized atomic orbitals (PAOs).

TYPE:

STRING

DEFAULT:

EPAO For local MP2 calculations Otherwise no default.

OPTIONS:

PAO Perform PAO-SCF instead of conventional SCF.

EPAO Obtain EPAOs after a conventional SCF.

RECOMMENDATION:

None

4.9.2 SCF Metadynamics

As the SCF equations are non-linear in the electron density, there are in theory very many solutions, *i.e.*, sets of orbitals where the energy is stationary with respect to changes in the orbital subset. Most often sought is the solution with globally minimal energy as this is a variational upper bound to the true eigenfunction in this basis. The SCF methods available in Q-CHEM allow the user to converge upon an SCF solution, and (using STABILITY_ANALYSIS) ensure it is a minimum, but there is no known method of ensuring that the found solution is a global minimum; indeed in systems with many low-lying energy levels the solution converged upon may vary considerably with initial guess.

SCF metadynamics[92] is a technique which can be used to locate multiple SCF solutions, and thus gain some confidence that the calculation has converged upon the global minimum. It works by searching out a solution to the SCF equations. Once found, the solution is stored, and a biasing potential added so as to avoid re-converging to the same solution. More formally, the distance between two solutions, w and x, can be expressed as $d^2_{wx} = \langle {}^w\Psi | {}^w\hat{\rho} - {}^x\hat{\rho} | {}^w\Psi \rangle$, where ${}^w\Psi$ is a Slater determinant formed from the orthonormal orbitals, ${}^w\phi_i$, of solution w, and ${}^w\hat{\rho}$ is the one-particle density operator for ${}^w\Psi$. This definition is equivalent to $d^2_{wx} = N - {}^wP^{\mu\nu} S_{\nu\sigma} \cdot {}^xP^{\sigma\tau} S_{\tau\mu}$. and is easily calculated. The function d^2_{wx} is

between zero and the number of electrons, and can be taken as the distance between two solutions. As an example, any singly-excited determinant (which will not in general be another SCF solution) is a distance 1 away from the reference (unexcited) determinant.

In a manner analogous to classical metadynamics, to bias against the set of previously located solutions, x, we create a new Lagrangian,

$$\tilde{E} = E + \sum_x N_x e^{-\lambda_x d_{0x}^2} \tag{4.63}$$

where 0 represents the present density. From this we may derive a new effective Fock matrix,

$$\tilde{F}_{\mu\nu} = F_{\mu\nu} + \sum_x {}^x P_{\mu\nu} N_x \lambda_x e^{-\lambda_x d_{0x}^2} \tag{4.64}$$

This may be used with very little modification within a standard DIIS procedure to locate multiple solutions. When close to a new solution, the biasing potential is removed so the location of that solution is not affected by it. If the calculation ends up re-converging to the same solution, N_x and λ_x can be modified to avert this. Once a solution is found it is added to the list of solutions, and the orbitals mixed to provide a new guess for locating a different solution.

This process can be customized by the REM variables below. Both DIIS and GDM methods can be used, but it is advisable to turn on MOM when using DIIS to maintain the orbital ordering. Post-HF correlation methods can also be applied. By default they will operate for the last solution located, but this can be changed with the SCF_MINFIND_RUNCORR variable.

SCF_SAVEMINIMA

Turn on SCF metadynamics and specify how many solutions to locate.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use SCF metadynamics

n Attempt to find n distinct SCF solutions.

RECOMMENDATION:

Perform SCF Orbital metadynamics and attempt to locate n different SCF solutions. Note that these may not all be minima. Many saddle points are often located. The last one located will be the one used in any post-SCF treatments. In systems where there are infinite point groups, this procedure cannot currently distinguish between spatial rotations of different densities, so will likely converge on these multiply.

SCF_READMINIMA

Read in solutions from a previous SCF metadynamics calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Read in n previous solutions and attempt to locate them all.

$-n$ Read in n previous solutions, but only attempt to locate solution n (not available in LIBNOCI).

RECOMMENDATION:

This may not actually locate all solutions required and will probably locate others too. The SCF will also stop when the number of solutions specified in SCF_SAVEMINIMA are found. Solutions from other geometries may also be read in and used as starting orbitals. If a solution is found and matches one that is read in within SCF_MINFIND_READDISTTHRESH, its orbitals are saved in that position for any future calculations. The algorithm works by restarting from the orbitals and density of a the minimum it is attempting to find. After 10 failed restarts (defined by SCF_MINFIND_RESTARTSTEPS), it moves to another previous minimum and attempts to locate that instead. If there are no minima to find, the restart does random mixing (with 10 times the normal random mixing parameter). Note that in LIBNOCI, previous minima are read using NOCI_REFGEN = 1, whilst the exact solutions are specified as described in Section 4.9.3

SCF_MINFIND_WELLTHRESH

Specify what SCF_MINFIND believes is the basin of a solution

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n for a threshold of 10^{-n}

RECOMMENDATION:

When the DIIS error is less than 10^{-n}, penalties are switched off to see whether it has converged to a new solution.

SCF_MINFIND_RESTARTSTEPS

Restart with new orbitals if no minima have been found within this many steps

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

n Restart after n steps.

RECOMMENDATION:

If the SCF calculation spends many steps not finding a solution, lowering this number may speed up solution-finding. If the system converges to solutions very slowly, then this number may need to be raised.

SCF_MINFIND_INCREASEFACTOR

Controls how the height of the penalty function changes when repeatedly trapped at the same solution

TYPE:

INTEGER

DEFAULT:

10100 meaning 1.01

OPTIONS:

$abcde$ corresponding to $a.bcde$

RECOMMENDATION:

If the algorithm converges to a solution which corresponds to a previously located solution, increase both the normalization N and the width lambda of the penalty function there. Then do a restart.

SCF_MINFIND_INITLAMBDA

Control the initial width of the penalty function.

TYPE:

INTEGER

DEFAULT:

02000 meaning 2.000

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The initial inverse-width (*i.e.*, the inverse-variance) of the Gaussian to place to fill solution's well. Measured in electrons$^{(}-1)$. Increasing this will repeatedly converging on the same solution.

SCF_MINFIND_INITNORM

Control the initial height of the penalty function.

TYPE:

INTEGER

DEFAULT:

01000 meaning 1.000

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The initial normalization of the Gaussian to place to fill a well. Measured in hartrees.

SCF_MINFIND_RANDOMMIXING

Control how to choose new orbitals after locating a solution

TYPE:

INTEGER

DEFAULT:

00200 meaning .02 radians

OPTIONS:

$abcde$ corresponding to $a.bcde$ radians

RECOMMENDATION:

After locating an SCF solution, the orbitals are mixed randomly to move to a new position in orbital space. For each occupied and virtual orbital pair picked at random and rotate between them by a random angle between 0 and this. If this is negative then use exactly this number, *e.g.*, -15708 will almost exactly swap orbitals. Any number< -15708 will cause the orbitals to be swapped exactly.

SCF_MINFIND_NRANDOMMIXES

Control how many random mixes to do to generate new orbitals

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Perform n random mixes.

RECOMMENDATION:

This is the number of occupied/virtual pairs to attempt to mix, per separate density (*i.e.*, for unrestricted calculations both alpha and beta space will get this many rotations). If this is negative then only mix the highest 25% occupied and lowest 25% virtuals.

SCF_MINFIND_READDISTTHRESH

The distance threshold at which to consider two solutions the same

TYPE:

INTEGER

DEFAULT:

00100 meaning 0.1

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The threshold to regard a minimum as the same as a read in minimum. Measured in electrons. If two minima are closer together than this, reduce the threshold to distinguish them.

SCF_MINFIND_MIXMETHOD

Specify how to select orbitals for random mixing

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Random mixing: select from any orbital to any orbital.

1 Active mixing: select based on energy, decaying with distance from the Fermi level.

2 Active Alpha space mixing: select based on energy, decaying with distance from the Fermi level only in the alpha space.

RECOMMENDATION:

Random mixing will often find very high energy solutions. If lower energy solutions are desired, use 1 or 2.

SCF_MINFIND_MIXENERGY

Specify the active energy range when doing Active mixing

TYPE:

INTEGER

DEFAULT:

00200 meaning 00.200

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The standard deviation of the Gaussian distribution used to select the orbitals for mixing (centered on the Fermi level). Measured in Hartree. To find less-excited solutions, decrease this value

SCF_MINFIND_RUNCORR

Run post-SCF correlated methods on multiple SCF solutions

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

If this is set > 0, then run correlation methods for all found SCF solutions.

RECOMMENDATION:

Post-HF correlation methods should function correctly with excited SCF solutions, but their convergence is often much more difficult owing to intruder states.

4.9.3 Multiple SCF Solutions for Non-Orthogonal CI

The solutions found through metadynamics often appear to be good approximations to diabatic surfaces, where the electronic structure does not significantly change with geometry. In situations where there are such multiple electronic states close in energy, an adiabatic state may be produced by diagonalizing a matrix of these states, *i.e.*, through a configuration interaction (CI) procedure. As they are distinct solutions of the SCF equations, these states are non-orthogonal (*i.e.* one cannot be constructed as a single determinant made out of the orbitals of another), and so the CI is a little more complicated and corresponds to a non-orthogonal CI (NOCI). More information on NOCI can be found in Section 7.5.

Version 5.2 of Q-CHEM introduces a new NOCI package, LIBNOCI, for locating multiple SCF solutions and running NOCI calculations (see Section 7.5.2), including a new implementation of SCF metadynamics. The LIBNOCI implementation of SCF metadynamics can be accessed using USE_LIBNOCI = TRUE in combination with NOCI_DETGEN = 3. In addition to the original SCF metadynamics features available in Q-CHEM, this new implementation includes:

- An active space approach where orbital mixing and optimization occurs only in a user-defined subset of orbitals.
- Full support for restricted, unrestricted and generalized orbital types, along with complex (Hermitian) and holomorphic (non-Hermitian) orbitals; see Section 4.9.4.

Multiple Hartree-Fock states of particular relevance for NOCI are often related to varying orbital occupations in a dominant subset of molecular orbitals. For example, important multiple solutions may correspond to excited determinants whose orbitals have been individually relaxed at the SCF level, or symmetry-broken states formed from strong mixing in a dominant active space. LIBNOCI allows multiple solutions to be identified by allowing orbital mixing and relaxation only in a subset of orbitals defined using the *$active_orbitals* input section By default, the multiple solutions located are then subsequently optimized in the full orbital space, although this can be skipped using SKIP_SCFMAN = TRUE.

Finally, LIBNOCI introduces easier control over reading initial guesses from previous calculations. Using the input NOCI_REFGEN = 1, all previous solutions are read from file (if available), while a particular subset can be chosen using SCF_READMINIMA.

Example 4.28 Example of using the LIBNOCI implementation of SCF metadynamics.

```
$molecule
   0 1
   H    0.0000000   0.0000000   0.0000000
   H    0.0000000   0.0000000   4.0000000
$end

$rem
   METHOD                     hf
   UNRESTRICTED               true
   BASIS                      sto-3g
   SCF_CONVERGENCE            10
   MAX_SCF_CYCLES             1000
   MOM_START                  1
   USE_LIBNOCI                true
   NOCI_DETGEN                3
   SCF_SAVEMINIMA             4
   SCF_MINFIND_RANDOMMIXING   30000
   SCF_MINFIND_MIXMETHOD      1
$end
```

Active orbitals can be specified for SCF metadynamics in LIBNOCI. Indices for β orbitals are offset by the number of α MOs, *i.e.* the case selects α orbitals 1 and 2, and β orbitals 1 and 2, with a total of 10 α molecular orbitals (including occupied and virtual).

```
$active_orbitals
    1 2 11 12
$end
```

The initial guess coefficients can also be read in as follows:

```
$scf_read
    1 2 4 ...
$end
```

4.9.4 Holomorphic Hartree-Fock Theory

4.9.4.1 Theory

To use multiple SCF solutions in NOCI (see Section 7.5), it is essential that all solutions exist across all geometries of interest to prevent discontinuities in the NOCI energies. However, it is well known that symmetry-broken SCF solutions can disappear at certain points along a potential energy surface, for example at the Coulson-Fischer point in H_2. The holomorphic Hartree-Fock approach provides a means of analytically continuing solutions across all geometries.[10,11,45]

In holomorphic Hartree-Fock theory, the real Hartree-Fock equations are analytically continued into the complex plane *without* introducing the complex conjugation of molecular orbital coefficients. Multiple solutions are then identified as the stationary points of the holomorphic energy[45]

$$\widetilde{E} = \frac{\langle \Psi^* | \hat{H} | \Psi \rangle}{\langle \Psi^* | \Psi \rangle}, \tag{4.65}$$

where $\hat{H}$ defines the conventional electronic Hamiltonian

$$\hat{H} = V_{\text{nuc}} + \sum_{i}^{N_e} \hat{h}(i) + \sum_{i<j}^{N_e} \frac{1}{r_{ij}}. \tag{4.66}$$

As a result, the holomorphic Hartree-Fock equations are complex-analytic in the orbital coefficients, satisfying the Cauchy-Riemann conditions, and the number of stationary points is found to be constant across all geometries.[11] Real Hartree-Fock states remain stationary points of the holomorphic Hartree-Fock energy, and where real solutions vanish, their holomorphic counterparts continue to exist with complex orbital coefficients.[10,11]

Holomorphic Hartree-Fock stationary points can be located using minor modifications to conventional SCF algorithms.[10]. Most significantly, by removing the complex conjugate of the wave function in Eq. (4.65), the required complex holomorphic density $\mathbf{P}$ and Fock $\mathbf{F}$ matrices become complex-symmetric (*cf.* Hermitian), satisfying $P^{\mu\nu} = P^{\nu\mu}$ and $F_{\mu\nu} = F_{\nu\mu}$. Moreover, since the complex conjugation must also be removed from the normalisation constraint, the molecular orbital coefficients must form a complex-orthogonal set (*cf.* unitary), *i.e.*,

$$\sum_{\mu\nu} C_{i\cdot}^{\cdot\mu} S_{\mu\nu} C_{\cdot j}^{\nu\cdot} = \delta_{ij}, \tag{4.67}$$

Finally, the holomorphic Hartree-Fock orbital energies and total energy can in general also become complex, and thus selecting the new occupied orbitals on each SCF cycle using the orbital energies is poorly defined. Instead, a complex-symmetric analogue to the Maximum Overlap Method can be employed (see Section 4.5.14).

Following real solutions past the Coulson-Fischer point into the complex plane can often be difficult due to their coalesence with symmetry-pure solutions on the real axis. However, by scaling the electron-electron interaction using a complex parameter λ, *i.e.* introducing the Hamiltonian

$$\hat{H} = V_{\text{nuc}} + \sum_{i}^{N_e} \hat{h}(i) + \lambda \sum_{i<j}^{N_e} \frac{1}{r_{ij}}, \tag{4.68}$$

it is possible to show that Coulson-Fischer points form *isolated* exceptional points on the real axis.[12] Consequently, following a suitable complex λ trajectory allows real solutions to be perturbed off the real axis and followed with ease past the Coulson-Fischer point into their complex holomorphic regimes.[12] These perturbed solutions can then be relaxed onto the real axis to identify the holomorphic Hartree-Fock states required for NOCI.

4.9.4.2 Job Control

Within Q-CHEM, the holomorphic Hartree-Fock approach is implemented in the LIBNOCI package (see Section 7.5.2), accessed using USE_LIBNOCI = TRUE and designed for locating multiple SCF solutions for use in NOCI calculations.

SCF_HOLOMORPHIC

Turn on the use of holomorphic Hartree-Fock orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Holomorphic Hartree-Fock is turned off

TRUE Holomorphic Hartree-Fock is turned on.

RECOMMENDATION:

If TRUE, holomorphic Hartree-Fock complex orbital coefficients will always be used. If FALSE, but COMPLEX = TRUE, complex Hermitian orbitals will be used.

SCF_EESCALE_MAG

Control the magnitude of the λ electron-electron scaling.

TYPE:

INTEGER

DEFAULT:

10000 meaning 1.0000

OPTIONS:

$abcde$ corresponding to $a.bcde$

RECOMMENDATION:

For holomorphic Hartree-Fock orbitals, only the magnitude of the input is used, while for real Hartree-Fock orbitals, the input sign indicates the sign of λ.

SCF_EESCALE_ARG

Control the phase angle of the complex λ electron-electron scaling.

TYPE:

INTEGER

DEFAULT:

00000 meaning 0.0000

OPTIONS:

$abcde$ corresponding to $a.bcde$

RECOMMENDATION:

A complex phase angle of 00500, meaning 0.0500, is usually sufficient to follow a solution safely past the Coulson-Fischer point and onto its complex holomorphic counterpart.

4.9.5 Non-Hermitian SCF with complex basis functions

Metastable electronic states can be characterized by a complex Siegert energy,

$$E = E_r - i\Gamma/2, \tag{4.69}$$

where the width, Γ, is proportional to the inverse lifetime of the state: $\Gamma = \hbar/\tau$. Complex coordinate methods aim to compute this complex energy as an eigenvalue of an effective non-Hermitian Hamiltonian. One such method is the method of complex basis functions (CBFs) where a basis of Gaussians with complex exponents is used in conjunction with a symmetric (not complex-conjugated) inner product to effictively produce a finite-basis representation of a non-Hermitian operators.[62,97–99] In cases, such as temporary anions, where the decay channel is of 1-electron character, a mean-field theory can provide approximate Siegert energies for a many-electron system.

The simplest such approximation is the static-exchange approximation. In this approximation the Siegert energies of an $(N+1)$-electron state are computed by diagonalizing a Fock operator computed from the density of an N-electron state.[97] This approximation neglects orbital relaxation effects which can be included by a non-Hermitian self-consistent-field (NH-SCF) procedure.[62,98] In practice the NH-SCF energy functional is the same as the Holomorphic Hartree-Fock energy functional (Eq. 4.65), though it is used for a different purpose. Both static-exchange and NH-SCF theories using complex basis functions (CBFs) are available in Q-CHEM. Specification of the complex basis set is described in Section 8.7.

COMPLEX_EXPONENTS

Enable a non-Hermitian calculation with CBFs.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a non-Hermitian calculation with CBFs

RECOMMENDATION:

Set to TRUE if a non-Hermitian calculation using CBFs is desired.

COMPLEX_SPIN_STATE

Spin state for non-Hermitian calculation

TYPE:

INTEGER

DEFAULT:

1 Singlet

OPTIONS:

$2S+1$ A state of spin S

RECOMMENDATION:

None

COMPLEX_N_ELECTRON

Add electrons for non-Hermitian calculation.

TYPE:

INTEGER

DEFAULT:

0 Perform the non-Hermitian calculation on N-electrons

OPTIONS:

n Perform the non-Hermitian calculation on an $N+n$ electron system

RECOMMENDATION:

None

COMPLEX_STATIC_EXCHANGE

Perform a CBF static-exchange calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a static exchange calculation

FALSE Do not perform a static exchange calculation

RECOMMENDATION:

Set to TRUE if a static-exchange calculation is desired.

COMPLEX_SCF

Perform a non-Hermitian SCF calculation with CBFs

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform an NH-SCF calculation

1 Perform a restricted NH-SCF calculation

2 Perform an unrestricted NH-SCF calculation

3 Perform a restricted, open-shell NH-SCF calculation

RECOMMENDATION:

None

COMPLEX_METSCF

Specify the NH-SCF solver

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Roothaan iterations

1 DIIS

3 ADIIS

21 Newton-MINRES

RECOMMENDATION:

Use the default (DIIS).

COMPLEX_SCF_GUESS

Specify the NH-SCF guess

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Use a guess from a static-exchange calculation
1	Read real-basis MO coefficients
2	Read real-basis density matrix
1000	Read guess from a previous calculation

RECOMMENDATION:

Use a guess from a static exchange calculation. Note that for temporary anions, this requires the specification of COMPLEX_TARGET.

COMPLEX_TARGET

Specify the orbital index to be occupied for a temporary anion

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n	Orbital index (starting at zero) for the additional electron

RECOMMENDATION:

n should always be greater than $N_{\text{occ}} - 1$.

4.9.6 Relativistic Effects

Relativistic effects play a major role in several physical and chemical phenomena, such as the properties of heavy elements and the proper characterization of the most inner energy levels probed in X-Ray espectroscopy experiments. Solving the four component Dirac equation, which describes both electrons and its anti-particles (positrons), is computationally expensive. Since most chemical proceses can be explained by solely taking the electronic wavefunction into account, several ways of effectively decoupling the electronic and positronic degrees of freedom have been proposed.

The one-electron exact two-component (1eX2C) hamiltonian[47,57,59,73] provides one route for achieving such decoupling. The method relies on solving the more tractable one electron four-component Dirac Hamiltonian in a restricted kinetic balance (RKB)[50] form to obtain the decoupling unitary transformations that will be used to modify the one-electron matrix elements, such as the kinetic energy and nuclear-attraction, to account for relativistic effects. A key ingredient to the X2C transformation matrices is to compute

$$W_{\text{SF},\mu\nu} = \langle \phi_\mu | \vec{p} \cdot (V\vec{p}) \, | \phi_\nu \rangle \tag{4.70}$$

where $W_{\text{SF},\mu\nu}$ is a matrix elements for the small-component of the full Dirac Hamiltonian, only computing the scalar-relativistic component. This is accomplished by noting that the the momentum operator is the generator of translations and its effects on a basis function can be captured by taking appropriate derivatives of such functions. It should be noted that, in order to properly capture the effects of the small components to the electronic wavefunction through X2C, **all-electron decontracted basis sets are required**. Full details of the finite difference X2C algorithm are provided in Ref. 17. An example on how to include scalar relativistic effects to model K-edge X-Ray spectroscopy can be found in Section 7.13.4.

REL_X2C

Enables X2C scalar relativistic calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perform a regular, non-relativistic SCF calculation

1 Perform a scalar relativistic X2C calculation

RECOMMENDATION:

Set to 1 if a scalar relativistic X2C calculation is desired.

REL_X2C_FD_DISPLACEMENT

Controls finite difference step for calulating W

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n Set finite difference step to $n \times 10^{-6}$

RECOMMENDATION:

None

4.9.6.1 Including one-electron Spin-Orbit Coupling Effects

In principle, the full solution to the 1eX2C Hamiltonian can be achieved, which will also include a spin-orbit coupling term. This, however, requires a spin-generealized and complex-valued solution. This can be seen by the additional matrix $W_{\mathrm{SOC},\mu\ nu}$ required to obtain a full W matrix:

$$W_{\mathrm{SOC},\mu\nu} = i\sigma \cdot \langle \phi_\mu | \vec{p} \times (V\vec{p}) \, | \phi_\nu \rangle \tag{4.71}$$

which is complex-valued and requires a cross product of different pauli matrices, resulting in a spin-generalized solution. Thus SOC effects may only be included when REL_X2C = 1 is called in cGHF routine - other calculations (e.g., RHF, UHF, ROHF) will all result in SF-1eX2C and will not include SOC effects. Setting the rem varaible REL_X2C = 1 in cGHF class automatically computes $W_{\mathrm{SOC},\mu\nu}$ unless REL_SF_X2C_CGHF is also set to 1.

REL_SF_X2C_CGHF

Enables X2C scalar relativistic calculation in a cGHF routine.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perform a full relativistic X2C calculation with SOC effects

1 Perform a scalar relativistic X2C calculation

RECOMMENDATION:

Set to 1 if a scalar relativistic X2C calculation in a **cGHF class** is desired. Must be used with REL_X2C = 1. Otherwise, non-relativistic calculation is performed without X2C, and this rem variable is ignored.

4.9.6.2 Screened-Nuclear Spin-Orbit Coupling for two-electron SOC effects

Although two-electron SOC terms in the X2C term can in principle be computed, the relevant matrix elements are often highly complicated and difficult to be evaluated. Instead, a popular approach to approximation the two-electron SOC effects in X2C scheme is to empirically scale the spin-orbit dependent terms of the full 1eX2C hamiltonian. The family of this methods is called screened-nuclear spin-orbit approach (SNSO), first suggested by Boettger[8]. Since then, improved form of the coefficient which accommodate diffuseness/tightness of the p-shell Gaussian-Type-Orbitals (GTOs) as well as occupied and virtual MOs[30,107] were suggested. Particularly, a new parametrization of the original SNSO coefficients based on the full four-component Dirac-Coulomb-Breit (DCB) Hamiltonian orbital splitting results was provided in Ref 29.

REL_SNSO

Enables 1eX2C-SNSO calculation in a cGHF routine.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

- 0 Perform a full relativistic X2C calculation with unadjusted SOC Hamiltonian
- 1 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by original Boetter coefficients
- 2 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by modified SNSO coefficients by Yoshizawa and co-workers
- 3 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by DC-SNSO coefficients from Ehrman and co-workers
- 4 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by universal DCB-SNSO coefficients from Ehrman and co-workers
- 5 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by row-dependent DCB-SNSO coefficients from Ehrman and co-workers

RECOMMENDATION:

Set to 1 for most general cases. Universal DCB-SNSO coefficients are the most recent contribution to the SNSO family and is reported to give accurate doublet splitting of $L_{2,3}$-edges.

References and Further Reading

[1] AOINTS (Appendix A).

[2] Molecular Properties Analysis (Chapter 10).

[3] Basis Sets and Effective Core Potentials (Chapter 8).

[4] Molecular Geometry and Critical Points (Chapter 9).

[5] T. R. Adams, R. D. Adamson, and P. M. W. Gill. *J. Chem. Phys.*, 107:124, 1997. DOI: 10.1063/1.474359.

[6] J. Almlöf, K. Fægri Jr., and K. Korsell. *J. Comput. Chem.*, 3:385, 1982. DOI: 10.1002/jcc.540030314.

[7] G. M. J. Barca, A. T. B. Gilbert, and P. M. W. Gill. *J. Chem. Theory Comput.*, 14:1501, 2018. DOI: 10.1021/acs.jctc.7b00994.

[8] J. C. Boettger. *Phys. Rev. B*, 62:7809, 2000. DOI: 10.1103/PhysRevB.62.7809.

[9] M. Born and R. Oppenheimer. *Ann. Phys.*, 389:457, 1927. DOI: 10.1002/andp.19273892002.

[10] H. G. A. Burton and A. J. W. Thom. *J. Chem. Theory Comput.*, 12:167, 2016. DOI: 10.1021/acs.jctc.5b01005.

[11] H. G. A. Burton, M. Gross, and A. J. W. Thom. *J. Chem. Theory Comput.*, 14:607, 2018. DOI: 10.1021/acs.jctc.7b00980.

[12] H. G. A. Burton, A. J. W. Thom, and P.-F. Loos. *J. Chem. Phys.*, 150:041103, 2019. DOI: 10.1063/1.5085121.

[13] E. Cancès. *J. Chem. Phys.*, 114:10616, 2001. DOI: 10.1063/1.1373430.

[14] E. Cancès and C. Le Bris. *Int. J. Quantum Chem.*, 79:82, 2000. DOI: 10.1002/1097-461X(2000)79:2<82::AID-QUA3>3.0.CO;2-I.

[15] K. Carter-Fenk and J. M. Herbert. *J. Chem. Theory Comput.*, 16:5067, 2020. DOI: 10.1021/acs.jctc.0c00502.

[16] D. Cremer and J. Gauss. *J. Comput. Chem.*, 7:274, 1986. DOI: 10.1002/jcc.540070305.

[17] L. A. Cunha, D. Hait, R. Kang, Y. Mao, and M. Head-Gordon. *J. Phys. Chem. Lett.*, 13:3438, 2022. DOI: 10.1021/acs.jpclett.2c00578.

[18] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople. *J. Chem. Phys.*, 94:7221, 1991. DOI: 10.1063/1.460205.

[19] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople. *J. Chem. Phys.*, 106:1063, 1997. DOI: 10.1063/1.473182.

[20] L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov, and J. A. Pople. *J. Chem. Phys.*, 109:7764, 1998. DOI: 10.1063/1.477422.

[21] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople. *J. Chem. Phys.*, 112:7374, 2000. DOI: 10.1063/1.481336.

[22] E. R. Davidson. *J. Comput. Phys.*, 17:87, 1975. DOI: 10.1016/0021-9991(75)90065-0.

[23] J. Deng, A. T. B. Gilbert, and P. M. W. Gill. *J. Chem. Phys.*, 130:231101, 2009. DOI: 10.1063/1.3152864.

[24] J. Deng, A. T. B. Gilbert, and P. M. W. Gill. *J. Chem. Phys.*, 133:044116, 2009. DOI: 10.1063/1.3463800.

[25] J. Deng, A. T. B. Gilbert, and P. M. W. Gill. *Phys. Chem. Chem. Phys.*, 12:10759, 2010. DOI: 10.1039/c0cp00242a.

[26] R. A. DiStasio, Jr., R. P. Steele, and M. Head-Gordon. *Mol. Phys.*, 105:2731, 2007. DOI: 10.1080/00268970701624687.

[27] M. Dupuis and H. F. King. *Int. J. Quantum Chem.*, 11:613, 1977. DOI: 10.1002/qua.560110408.

[28] M. Dupuis and H. F. King. *J. Chem. Phys.*, 68:3998, 1978. DOI: 10.1063/1.436313.

[29] J. Ehrman, E. Martinez-Baez, J. A. Jenkins, and X. Li. *J. Chem. Theory Comput.*, 19:5785, 2023. DOI: 10.1021/acs.jctc.3c00479.

[30] M. Filatov, W. Zou, and D. Cremer. *J. Chem. Phys.*, 139:014106, 2013. DOI: 10.1063/1.4811776.

[31] L. Fusti-Molnar. *J. Chem. Phys.*, 119:11080, 2003. DOI: 10.1063/1.1622922.

[32] L. Fusti-Molnar and J. Kong. *J. Chem. Phys.*, 122:074108, 2005. DOI: 10.1063/1.1849168.

[33] L. Fusti-Molnar and P. Pulay. *J. Chem. Phys.*, 116:7795, 2002. DOI: 10.1063/1.1467901.

[34] L. Fusti-Molnar and P. Pulay. *J. Chem. Phys.*, 117:7827, 2002. DOI: 10.1063/1.1510121.

[35] A. T. B. Gilbert, N. A. Besley, and P. M. W. Gill. *J. Phys. Chem. A*, 112:13164, 2008. DOI: 10.1021/jp801738f.

[36] M. Gray, P. E. Bowling, and J. M. Herbert. *J. Phys. Chem. A*, 128:7739, 2024. DOI: 10.1021/acs.jpca.4c00283.

[37] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. MIT Press, London, 1987.

[38] M. F. Guest and V. R. Saunders. *Mol. Phys.*, 28:819, 1974. DOI: 10.1080/00268977400102171.

[39] D. Hait and M. Head-Gordon. *J. Chem. Theory Comput.*, 16:1699, 2020. DOI: 10.1021/acs.jctc.9b01127.

[40] D. H. Hartree. *The Calculation of Atomic Structures*. John Wiley and Sons, 1957.

[41] M. Häser and R. Ahlrichs. *J. Comput. Chem.*, 10:104, 1989. DOI: 10.1002/jcc.540100111.

[42] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople. *Ab Initio Molecular Orbital Theory*. Wiley, New York, 1986.

[43] J. M. Herbert and M. Head-Gordon. *J. Chem. Phys.*, 121:11542, 2004. DOI: 10.1063/1.1814934.

[44] I. H. Hillier and V. R. Saunders. *Proc. Roy. Soc. Ser. A*, 320:161, 1970. DOI: 10.1098/rspa.1970.0203.

[45] H. G. Hiscock and A. J. W. Thom. *J. Chem. Theory Comput.*, 10:4795, 2014. DOI: 10.1021/ct5007696.

[46] X. Hu and W. Yang. *J. Chem. Phys.*, 132:054109, 2010. DOI: 10.1063/1.3304922.

[47] M. Ilias and T. Saue. *J. Chem. Phys.*, 126:064102, 2007. DOI: 10.1063/1.2436882.

[48] F. Jensen. *Introduction to Computational Chemistry*. Wiley, New York, 1994.

[49] K. N. Kudin, G. E. Scuseria, and E. Cancès. *J. Chem. Phys.*, 116:8255, 2002. DOI: 10.1063/1.1470195.

[50] W. Kutzelnigg. *Int. J. Quantum Chem.*, 25:107, 1984. DOI: 10.1002/qua.560250112.

[51] M. S. Lee and M. Head-Gordon. *J. Chem. Phys.*, 107:9085, 1997. DOI: 10.1063/1.475199.

[52] M. S. Lee and M. Head-Gordon. *Comp. Chem.*, 24:295, 2000. DOI: 10.1016/S0097-8485(99)00086-8.

[53] S. Lehtola. *J. Chem. Theory Comput.*, 15:1593, 2019. DOI: 10.1021/acs.jctc.8b01089.

[54] S. Lehtola. *Int. J. Quantum Chem.*, 119:e25945, 2019. DOI: 10.1002/qua.25945.

[55] S. Lehtola. *Phys. Rev. A*, 101:012516, 2020. DOI: 10.1103/PhysRevA.101.012516.

[56] J. H. Van Lenthe, R. Zwaans, H. J. J. Van Dam, and M. F. Guest. *J. Comput. Chem.*, 27:926, 2006. DOI: 10.1002/jcc.20393.

[57] Z. Li, Y. Xiao, and W. Liu. *J. Chem. Phys.*, 137:154114, 2012. DOI: 10.1063/1.4758987.

[58] W. Z. Liang and M. Head-Gordon. *J. Phys. Chem. A*, 108:3206, 2004. DOI: 10.1021/jp0374713.

[59] W. Liu and D. Peng. *J. Chem. Phys.*, 131:031104, 2009. DOI: 10.1063/1.3159445.

[60] S. Manzer, P. R. Horn, N. Mardirossian, and M. Head-Gordon. *J. Chem. Phys.*, 143:024113, 2015. DOI: 10.1063/1.4923369.

[61] S. F. Manzer, E. Epifanovsky, and M. Head-Gordon. *J. Chem. Theory Comput.*, 11:518, 2015. DOI: 10.1021/ct5008586.

[62] C. W. McCurdy and T. N. Recigno. *Phys. Rev. A*, 41:1364, 1978. DOI: 10.1103/PhysRevA.17.1931.

[63] P. Merlot, T. Kjaergaard, T. Helgaker, R. Lindh, F. Aquilante, S. Reine, and T. B. Pedersen. *J. Comput. Chem.*, 34:1486, 2013. DOI: 10.1002/jcc.23284.

[64] T. Nakajima and K. Hirao. *J. Chem. Phys.*, 124:184108, 2006. DOI: 10.1063/1.2198529.

[65] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 2006.

[66] C. Ochsenfeld. *Chem. Phys. Lett.*, 327:216, 2000. DOI: 10.1016/S0009-2614(00)00865-4.

[67] C. Ochsenfeld, C. A. White, and M. Head-Gordon. *J. Chem. Phys.*, 109:1663, 1998. DOI: 10.1063/1.476741.

[68] B. N. Plakhutin and E. R. Davidson. *J. Chem. Phys.*, 140:014102, 2014. DOI: 10.1063/1.4849615.

[69] J. A. Pople and R. K. Nesbet. *J. Chem. Phys.*, 22:571, 1954. DOI: 10.1063/1.1740120.

[70] P. Pulay. *Chem. Phys. Lett.*, 73:393, 1980. DOI: 10.1016/0009-2614(80)80396-4.

[71] P. Pulay. *J. Comput. Chem.*, 3:556, 1982. DOI: 10.1002/jcc.540030413.

[72] A. D. Rabuck and G. E. Scuseria. *J. Chem. Phys.*, 110:695, 1999. DOI: 10.1063/1.478177.

[73] T. Saue. *ChemPhysChem*, 12:3077, 2011. DOI: 10.1002/cphc.201100682.

[74] V. R. Saunders and I. H. Hillier. *Int. J. Quantum Chem.*, 7:699, 1973. DOI: 10.1002/qua.560070407.

[75] E. Schwegler, M. Challacombe, and M. Head-Gordon. *J. Chem. Phys.*, 106:9708, 1997. DOI: 10.1063/1.473833.

[76] R. Seeger and J. A. Pople. *J. Chem. Phys.*, 66:3045, 1977. DOI: 10.1063/1.434318.

[77] Y. Shao and M. Head-Gordon. *Chem. Phys. Lett.*, 323:425, 2000. DOI: 10.1016/S0009-2614(00)00524-8.

[78] Y. Shao and M. Head-Gordon. *J. Chem. Phys.*, 114:6572, 2001. DOI: 10.1063/1.1357441.

[79] S. M. Sharada, D. Stück, E. J. Sundstrom, A. T. Bell, and M. Head-Gordon. *Mol. Phys.*, 113:1802, 2015. DOI: 10.1080/00268976.2015.1014442.

[80] J. C. Slater. *Phys. Rev.*, 34:1293, 1929. DOI: 10.1103/PhysRev.34.1293.

[81] J. C. Slater. *Phys. Rev.*, 35:509, 1930. DOI: 10.1103/PhysRev.35.509.

[82] A. Sodt and M. Head-Gordon. *J. Chem. Phys.*, 125:074116, 2006. DOI: 10.1063/1.2370949.

[83] A. Sodt and M. Head-Gordon. *J. Chem. Phys.*, 128:104106, 2008. DOI: 10.1063/1.2828533.

[84] R. P. Steele and M. Head-Gordon. *Mol. Phys.*, 105:2455, 2007. DOI: 10.1080/00268970701519754.

[85] R. P. Steele and J. C. Tully. *Chem. Phys. Lett.*, 500:167, 2010. DOI: 10.1016/j.cplett.2010.10.003.

[86] R. P. Steele, R. A. DiStasio, Jr., Y. Shao, J. Kong, and M. Head-Gordon. *J. Chem. Phys.*, 125:074108, 2006. DOI: 10.1063/1.2234371.

[87] R. P. Steele, Y. Shao, R. A. DiStasio, Jr., and M. Head-Gordon. *J. Phys. Chem. A*, 110:13915, 2006. DOI: 10.1021/jp065444h.

[88] R. P. Steele, R. A. DiStasio, Jr., and M. Head-Gordon. *J. Chem. Theory Comput.*, 5:1560, 2009. DOI: 10.1021/ct900058p.

[89] R. P. Steele, M. Head-Gordon, and J. C. Tully. *J. Phys. Chem. A*, 114:11853, 2010. DOI: 10.1021/jp107342g.

[90] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry*. Dover, 1996.

[91] T. Takada, M. Dupuis, and H. F. King. *J. Chem. Phys.*, 75:332, 1981. DOI: 10.1063/1.441785.

[92] A. J. W. Thom and M. Head-Gordon. *Phys. Rev. Lett.*, 101:193001, 2008. DOI: 10.1103/PhysRevLett.101.193001.

[93] D. J. Tozer, M. E. Mura, R. D. Amos, and N. C. Handy. In *Computational Chemistry*, AIP Conference Proceedings, page 3, 1994.

[94] T. Van Voorhis and M. Head-Gordon. *Mol. Phys.*, 100:1713, 2002. DOI: 10.1080/00268970110103642.

[95] O. A. Vydrov and T. Van Voorhis. *J. Chem. Phys.*, 133:244103, 2010. DOI: 10.1063/1.3521275.

[96] F. Weigend. *Phys. Chem. Chem. Phys.*, 4:4285, 2002. DOI: 10.1039/b204199p.

[97] A. F. White, M. Head-Gordon, and C. W. McCurdy. *J. Chem. Phys.*, 142:054103, 2015. DOI: 10.1063/1.4906940.

[98] A. F. White, C. W. McCurdy, and M. Head-Gordon. *J. Chem. Phys.*, 143:074103, 2015. DOI: 10.1063/1.4928529.

[99] A. F. White, E. Epifanovsky, C. W. McCurdy, and M. Head-Gordon. *J. Chem. Phys.*, 146:234107, 2017. DOI: 10.1063/1.4986950.

[100] C. A. White and M. Head-Gordon. *J. Chem. Phys.*, 101:6593, 1994. DOI: 10.1063/1.468354.

[101] C. A. White and M. Head-Gordon. *J. Chem. Phys.*, 105:5061, 1996. DOI: 10.1063/1.472369.

[102] C. A. White and M. Head-Gordon. *Chem. Phys. Lett.*, 257:647, 1996. DOI: 10.1016/0009-2614(96)00574-X.

[103] C. A. White and M. Head-Gordon. *J. Chem. Phys.*, 104:2620, 1996. DOI: 10.1063/1.470986.

[104] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon. *Chem. Phys. Lett.*, 230:8, 1994. DOI: 10.1016/0009-2614(94)01128-1.

[105] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon. *Chem. Phys. Lett.*, 253:268, 1996. DOI: 10.1016/0009-2614(96)00175-3.

[106] M. Wolfsberg and L. Helmholtz. *J. Chem. Phys.*, 20:837, 1952. DOI: 10.1063/1.1700580.

[107] T. Yoshizawa, Zou W, and D. Cremer. *J. Chem. Phys.*, 145:184104, 2016. DOI: 10.1063/1.4964765.

Chapter 5

Density Functional Theory

5.1 Introduction

DFT[134,143,187,312] has emerged as an accurate, alternative first-principles approach to quantum mechanical molecular investigations. DFT calculations account for the overwhelming majority of all quantum chemistry calculations, not only because of its proven chemical accuracy, but also because of its relatively low computational expense, comparable to Hartree-Fock theory but with treatment of electron correlation that is neglected in a HF calculation. These two features suggest that DFT is likely to remain a leading method in the quantum chemist's toolkit well into the future. Q-CHEM contains fast, efficient and accurate algorithms for all popular density functionals, making calculations on large molecules possible and practical.

DFT is primarily a theory of electronic ground state structures based on the electron density, $\rho(\mathbf{r})$, as opposed to the many-electron wave function, $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$. (Its excited-state extension, time-dependent DFT, is discussed in Section 7.3.) There are a number of distinct similarities and differences between traditional wave function approaches and modern DFT methodologies. First, the essential building blocks of the many-electron wave function Ψ are single-electron orbitals, which are directly analogous to the Kohn-Sham orbitals in the DFT framework. Second, both the electron density and the many-electron wave function tend to be constructed via a SCF approach that requires the construction of matrix elements that are conveniently very similar.

However, traditional *ab initio* approaches using the many-electron wave function as a foundation must resort to a post-SCF calculation (Chapter 6) to incorporate correlation effects, whereas DFT approaches incorporate correlation at the SCF level. Post-SCF methods, such as perturbation theory or coupled-cluster theory are extremely expensive relative to the SCF procedure. On the other hand, while the DFT approach is exact in principle, in practice it relies on modeling an unknown exchange-correlation energy functional. While more accurate forms of such functionals are constantly being developed, there is no systematic way to improve the functional to achieve an arbitrary level of accuracy. Thus, the traditional approaches offer the possibility of achieving a systematically-improvable level of accuracy, but can be computationally demanding, whereas DFT approaches offer a practical route, but the theory is currently incomplete.

5.2 Kohn-Sham Density Functional Theory

The density functional theory by Hohenberg, Kohn, and Sham[113,133] stems from earlier work by Dirac,[75] who showed that the exchange energy of a uniform electron gas can be computed exactly from the charge density along. However, while this traditional *density* functional approach, nowadays called "orbital-free" DFT, makes a direct connection to the density alone, in practice it is constitutes a direct approach where the necessary equations contain only the electron

density, difficult to obtain decent approximations for the kinetic energy functional. Kohn and Sham sidestepped this difficulty via an indirect approach in which the kinetic energy is computed exactly for a noninteracting reference system, namely, the Kohn-Sham determinant.[133] It is the Kohn-Sham approach that first made DFT into a practical tool for calculations.

Within the Kohn-Sham formalism,[133] the ground state electronic energy, E, can be written as

$$E = E_{\mathrm{T}} + E_{\mathrm{V}} + E_{\mathrm{J}} + E_{\mathrm{XC}} \tag{5.1}$$

where E_{T} is the kinetic energy, E_{V} is the electron–nuclear interaction energy, E_{J} is the Coulomb self-interaction of the electron density, $\rho(\mathbf{r})$ and E_{XC} is the exchange-correlation energy. Adopting an unrestricted format, the α and β total electron densities can be written as

$$\rho_\alpha(\mathbf{r}) = \sum_{i=1}^{n_\alpha} |\psi_i^\alpha|^2 \tag{5.2a}$$

$$\rho_\beta(\mathbf{r}) = \sum_{i=1}^{n_\beta} |\psi_i^\beta|^2 \tag{5.2b}$$

where n_α and n_β are the number of alpha and beta electron respectively, and ψ_i are the Kohn-Sham orbitals. Thus, the total electron density is

$$\rho(\mathbf{r}) = \rho_\alpha(\mathbf{r}) + \rho_\beta(\mathbf{r}) \tag{5.3}$$

Within a finite basis set, the density is represented by[213]

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}\phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r}) \ , \tag{5.4}$$

where the $P_{\mu\nu}$ are the elements of the one-electron density matrix; see Eq. (4.24) in the discussion of Hartree-Fock theory. The various energy components in Eq. (5.1) can now be written

$$\begin{aligned} E_{\mathrm{T}} &= \sum_{i=1}^{n_\alpha} \left\langle \psi_i^\alpha \left| -\frac{1}{2}\hat{\nabla}^2 \right| \psi_i^\alpha \right\rangle + \sum_{i=1}^{n_\beta} \left\langle \psi_i^\beta \left| -\frac{1}{2}\hat{\nabla}^2 \right| \psi_i^\beta \right\rangle \\ &= \sum_{\mu\nu} P_{\mu\nu} \left\langle \phi_\mu(\mathbf{r}) \left| -\frac{1}{2}\hat{\nabla}^2 \right| \phi_\nu(\mathbf{r}) \right\rangle \end{aligned} \tag{5.5}$$

$$\begin{aligned} E_{\mathrm{V}} &= -\sum_{A=1}^{M} Z_A \int \frac{\rho(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_A|} d\mathbf{r} \\ &= -\sum_{\mu\nu} P_{\mu\nu} \sum_A \left\langle \phi_\mu(\mathbf{r}) \left| \frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|} \right| \phi_\nu(\mathbf{r}) \right\rangle \end{aligned} \tag{5.6}$$

$$\begin{aligned} E_{\mathrm{J}} &= \frac{1}{2} \left\langle \rho(\mathbf{r}_1) \left| \frac{1}{|\mathbf{r}_1-\mathbf{r}_2|} \right| \rho(\mathbf{r}_2) \right\rangle \\ &= \frac{1}{2} \sum_{\mu\nu} \sum_{\lambda\sigma} P_{\mu\nu} P_{\lambda\sigma} \, (\mu\nu|\lambda\sigma) \end{aligned} \tag{5.7}$$

$$E_{\mathrm{XC}} = \int f\left[\rho(\mathbf{r}), \hat{\boldsymbol{\nabla}}\rho(\mathbf{r}), \ldots\right] \rho(\mathbf{r}) \, d\mathbf{r} \ . \tag{5.8}$$

Minimizing E with respect to the unknown Kohn-Sham orbital coefficients yields a set of matrix equations exactly analogous to Pople-Nesbet equations of the UHF case, Eq. (4.13), but with modified Fock matrix elements [*cf.* Eq. (4.27)]

$$F_{\mu\nu}^\alpha = H_{\mu\nu}^{\mathrm{core}} + J_{\mu\nu} - F_{\mu\nu}^{\mathrm{XC}\alpha} \tag{5.9a}$$

$$F_{\mu\nu}^\beta = H_{\mu\nu}^{\mathrm{core}} + J_{\mu\nu} - F_{\mu\nu}^{\mathrm{XC}\beta} \ . \tag{5.9b}$$

Here, $\mathbf{F}^{\mathrm{XC}\alpha}$ and $\mathbf{F}^{\mathrm{XC}\beta}$ are the exchange-correlation parts of the Fock matrices and depend on the exchange-correlation functional used. UHF theory is recovered as a special case simply by taking $F_{\mu\nu}^{\mathrm{XC}\alpha} = K_{\mu\nu}^{\alpha}$, and similarly for β. Thus, the density and energy are obtained in a manner analogous to that for the HF method. Initial guesses are made for the MO coefficients and an iterative process is applied until self-consistency is achieved.

5.3 Overview of Available Functionals

5.3.1 Introduction

Q-CHEM currently has more than 30 exchange functionals as well as more than 30 correlation functionals, and in addition over 150 exchange-correlation (XC) functionals, which refer to functionals that are not separated into exchange and correlation parts, either because the way in which they were parameterized renders such a separation meaningless (*e.g.*, B97-D[95] or ωB97X[56]) or because they are a standard linear combination of exchange and correlation (*e.g.*, PBE[197] or B3LYP[23,241]). User-defined XC functionals can be created as specified linear combinations of any of the 30+ exchange functionals and/or the 30+ correlation functionals.

KS-DFT functionals can be organized onto a ladder with five rungs, in a classification scheme ("Jacob's Ladder") proposed by John Perdew in 2001.[192,199] The first rung contains a functional that only depends on the (spin-) density ρ_σ, namely, the local spin-density approximation (LSDA). These functionals are exact for the infinite uniform electron gas (UEG), but are highly inaccurate for molecular properties whose densities exhibit significant inhomogeneity. To improve upon the weaknesses of the LSDA, it is necessary to introduce an ingredient that can account for inhomogeneities in the density: the density gradient, $\hat{\boldsymbol{\nabla}}\rho_\sigma$. These generalized gradient approximation (GGA) functionals define the second rung of Jacob's Ladder and tend to improve significantly upon the LSDA. Two additional ingredients that can be used to further improve the performance of GGA functionals are either the Laplacian of the density $\hat{\nabla}^2\rho_\sigma$, and/or the kinetic energy density,

$$\tau_\sigma = \sum_{i}^{n_\sigma} \left\| \hat{\boldsymbol{\nabla}}\psi_{i,\sigma} \right\|^2 . \tag{5.10}$$

While functionals that employ both of these options are available in Q-CHEM, the kinetic energy density is by far the more popular ingredient and has been used in many modern functionals to add flexibility to the functional form with respect to both constraint satisfaction (non-empirical functionals) and least-squares fitting (semi-empirical parameterization). Functionals that depend on either of these two ingredients belong to the third rung of the Jacob's Ladder and are called meta-GGAs. These meta-GGAs often further improve upon GGAs in areas such as thermochemistry, kinetics (reaction barrier heights), and even non-covalent interactions.

Functionals on the fourth rung of Jacob's Ladder are called hybrid density functionals. This rung contains arguably the most popular density functional of our time, B3LYP, the first functional to see widespread application in chemistry. "Global" hybrid (GH) functionals such as B3LYP (as distinguished from the "range-separated hybrids" introduced below) add a constant fraction of "exact" (Hartree-Fock) exchange to any of the functionals from the first three rungs. Thus, hybrid LSDA, hybrid GGA, and hybrid meta-GGA functionals can be constructed, although the latter two types are much more common. As an example, the formula for the B3LYP functional, as implemented in Q-CHEM, is[92]

$$E_{\mathrm{xc}}^{\mathrm{B3LYP}} = c_{\mathrm{x}}E_{\mathrm{x}}^{\mathrm{HF}} + (1 - c_{\mathrm{x}} - a_{\mathrm{x}})E_{\mathrm{x}}^{\mathrm{Slater}} + a_{\mathrm{x}}E_{\mathrm{x}}^{\mathrm{B88}} + (1 - a_{\mathrm{c}})E_{\mathrm{c}}^{\mathrm{VWN1RPA}} + a_{\mathrm{c}}E_{\mathrm{c}}^{\mathrm{LYP}} \tag{5.11}$$

where $c_{\mathrm{x}} = 0.20$, $a_{\mathrm{x}} = 0.72$, and $a_{\mathrm{c}} = 0.81$.

A more recent approach to introducing exact exchange into the functional form is via range separation. Range-separated hybrid (RSH) functionals split the exact exchange contribution into a short-range (SR) component and a long-range

(LR) component, often by means of the error function (erf) and complementary error function ($\mathrm{erfc} \equiv 1 - \mathrm{erf}$):

$$\frac{1}{r_{12}} = \underbrace{\frac{\mathrm{erfc}(\omega r_{12})}{r_{12}}}_{\mathrm{SR}} + \underbrace{\frac{\mathrm{erf}(\omega r_{12})}{r_{12}}}_{\mathrm{LR}} \tag{5.12}$$

The first term on the right in Eq. (5.12) is singular but short-range, and decays to zero on a length scale of $\sim 1/\omega$, while the second term constitutes a non-singular, long-range background. An RSH XC functional can be expressed generically as

$$E_{\mathrm{xc}}^{\mathrm{RSH}} = c_{\mathrm{x}} E_{\mathrm{x,SR}}^{\mathrm{HF}} + c_{\mathrm{x,LR}} E_{\mathrm{x,LR}}^{\mathrm{HF}} + (1 - c_{\mathrm{x,SR}}) E_{\mathrm{x,SR}}^{\mathrm{DFT}} + (1 - c_{\mathrm{x,LR}}) E_{\mathrm{x,LR}}^{\mathrm{DFT}} + E_{\mathrm{c}}^{\mathrm{DFT}} \,, \tag{5.13}$$

where the SR and LR parts of the Coulomb operator are used, respectively, to evaluate the HF exchange energies $E_{\mathrm{x,SR}}^{\mathrm{HF}}$ and $E_{\mathrm{x,LR}}^{\mathrm{HF}}$. The corresponding DFT exchange functional is partitioned in the same manner, but the correlation energy $E_{\mathrm{c}}^{\mathrm{DFT}}$ is evaluated using the full Coulomb operator, r_{12}^{-1}. Of the two linear parameters in Eq. (5.13), $c_{\mathrm{x,LR}}$ is usually either set to 1 to define long-range corrected (LRC) RSH functionals (see Section 5.6) or else set to zero, which defines screened-exchange (SE) RSH functionals. On the other hand, the fraction of short-range exact exchange ($c_{\mathrm{x,SR}}$) can either be determined via least-squares fitting, theoretically justified using the adiabatic connection, or simply set to zero. As with the global hybrids, RSH functionals can be fashioned using all of the ingredients from the lower three rungs. The rate at which the local DFT exchange is turned off and the non-local exact exchange is turned on is controlled by the parameter ω. Large values of ω tend to lead to attenuators that are less smooth (unless the fraction of short-range exact exchange is very large), while small values of (*e.g.*, $\omega = 0.2$–0.4 bohr^{-1}) are the most common in semi-empirical RSH functionals.

The final rung on Jacob's Ladder contains functionals that use not only occupied orbitals (via exact exchange), but virtual orbitals as well (via methods such as MP2 or the random phase approximation, RPA). These double hybrids (DH) are the most expensive density functionals available in Q-CHEM, but can also be very accurate. The most basic form of a DH functional is

$$E_{\mathrm{xc}}^{\mathrm{DH}} = c_{\mathrm{x}} E_{\mathrm{x}}^{\mathrm{HF}} + (1 - c_{\mathrm{x}}) E_{\mathrm{x}}^{\mathrm{DFT}} + c_{\mathrm{c}} E_{c}^{\mathrm{MP2}} + (1 - c_{\mathrm{c}}) E_{\mathrm{c}}^{\mathrm{DFT}} \,. \tag{5.14}$$

As with hybrids, the coefficients can either be theoretically motivated or empirically determined. In addition, double hybrids can use exact exchange both globally or via range-separation, and their components can be as primitive as LSDA or as advanced as in meta-GGA functionals. More information on double hybrids can be found in Section 5.9.

Finally, the last major advance in KS-DFT in recent years has been the development of methods that are capable of accurately describing non-covalent interactions, particularly dispersion. All of the functionals from Jacob's Ladder can technically be combined with these dispersion corrections, although in some cases the combination is detrimental, particularly for semi-empirical functionals that were parameterized in part using data sets of non-covalent interactions, and already tend to overestimate non-covalent interaction energies. The most popular such methods available in Q-CHEM are:

- Non-local correlation (NLC) functionals (Section 5.7.2), including those of Vydrov and Van Voorhis[269,271] (VV09 and VV10) and of Lundqvist and Langreth[73,74] (vdW-DF-04 and vdW-DF-10). The revised VV10 NLC functional of Sabatini and coworkers (rVV10) is also available[230].
- Damped, atom–atom pairwise empirical dispersion potentials from Grimme and others[57,95,97,98,232,237] [DFT-D2, DFT-CHG, DFT-D3(0), DFT-D3(BJ), DFT-D3(CSO), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op)]; see Section 5.7.3.
- The exchange-dipole models (XDM) of Johnson and Becke (XDM6 and XDM10); see Section 5.7.4.
- The Tkatchenko-Scheffler (TS) method for dispersion interactions;[251] see Section 5.7.5.
- The Many-Body Dispersion (MBD) method for van der Waals interactions[12,252]; see Section 5.7.6.

	Single-Point	**Optimization**	**Frequency**
Ground State	LSDA	LSDA	LSDA
	GGA	GGA	GGA
	meta-GGA	meta-GGA	meta-GGA
	GH	GH	GH
	RSH	RSH	RSH
	NLC	NLC	VV10
	DFT-D	DFT-D	DFT-D
	SRC	—	—
	XDM	—	—
TDDFT	LSDA	LSDA	LSDA
	GGA	GGA	GGA
	meta-GGA	meta-GGA	meta-GGA
	GH	GH	GH
	RSH	RSH	RSH
	NLC	—	—
	DFT-D	DFT-D	DFT-D
	SRC	—	—
	—	—	—

Table 5.1: Available analytic properties for SCF calculations.

Below, we categorize the functionals that are available in Q-CHEM, including exchange functionals (Section 5.3.3), correlation functionals (Section 5.3.4), and exchange-correlation functionals (Section 5.3.5). Within each category the functionals will be categorized according to Jacob's Ladder. Exchange and correlation functionals can be invoked using the *$rem* variables EXCHANGE and CORRELATION, while the exchange-correlation functionals can be invoked either by setting the *$rem* variable METHOD or alternatively (in most cases, and for backwards compatibility with earlier versions of Q-CHEM) by using the *$rem* variable EXCHANGE. Some caution is warranted here. While setting METHOD to PBE, for example, requests the Perdew-Burke-Ernzerhof (PBE) exchange-correlation functional,[197] which includes both PBE exchange and PBE correlation, setting EXCHANGE = PBE requests only the exchange component and setting CORRELATION = PBE requests only the correlation component. Setting both of these values is equivalent to specifying METHOD = PBE.

Finally, Table 5.1 provides a summary, arranged according to Jacob's Ladder, of which categories of functionals are available with analytic first derivatives (for geometry optimizations) or second derivatives (for vibrational frequency calculations). If analytic derivatives are not available for the requested job type, Q-CHEM will automatically generate them via finite difference. Tests of the finite-difference procedure, in cases where analytic second derivatives *are* available, suggest that finite-difference frequencies are accurate to < 1 cm^{-1}, except for very low-frequency, non-bonded modes that are not well-described by harmonic frequencies anyway.[159] Also listed in Table 5.1 are which functionals are available for excited-state time-dependent DFT (TDDFT) calculations, as described in Section 7.3.

5.3.2 Suggested Density Functionals

Q-CHEM contains over 150 exchange-correlation functionals, not counting those that can be straightforwardly appended with a dispersion correction (such as B3LYP-D3). Therefore, we suggest a few functionals from the second through fourth rungs of Jacob's Ladder in order to guide functional selection. Most of these suggestions come from a benchmark of over 200 density functionals on a vast database of nearly 5000 data points, covering non-covalent inter-

actions, isomerization energies, thermochemistry, and barrier heights. **The single recommended method from each category is indicated in bold.**

From the GGAs on Rung 2, we recommend:

- **B97-D3(BJ)**: METHOD = B97-D3 and DFT_D = D3_BJ
- revPBE-D3(BJ): METHOD = revPBE and DFT_D = D3_BJ
- BLYP-D3(BJ): METHOD = BLYP and DFT_D = D3_BJ
- PBE: METHOD = PBE

From the meta-GGAs on Rung 3, we recommend:

- **B97M-rV**: METHOD = B97M-rV
- MS1-D3(0): METHOD = MGGA_MS1 and DFT_D = D3_ZERO
- MS2-D3(0): METHOD = MGGA_MS2 and DFT_D = D3_ZERO
- M06-L-D3(0): METHOD = M06-L and DFT_D = D3_ZERO
- TPSS-D3(BJ): METHOD = TPSS and DFT_D = D3_BJ

From the hybrid GGAs on Rung 4, we recommend:

- ω**B97X-V**: METHOD = wB97X-V
- ωB97X-D3: METHOD = wB97X-D3
- ωB97X-D: METHOD = wB97X-D
- B3LYP-D3(BJ): METHOD = B3LYP and DFT_D = D3_BJ
- revPBE0-D3(BJ): METHOD = revPBE0 and DFT_D = D3_BJ

From the hybrid meta-GGAs on Rung 4, we recommend:

- ω**B97M-V**: METHOD = wB97M-V
- ωM05-D: METHOD = wM05-D
- M06-2X-D3(0): METHOD = M06-2X and DFT_D = D3_ZERO
- TPSSh-D3(BJ): METHOD = TPSSh and DFT_D = D3_BJ

From the double-hybrid GGAs on Rung 5, we recommend:

- ω**B97X-2(LP)**: METHOD = wB97X-2(LP)
- ω**B97X-2(TQZ)**: METHOD = wB97X-2(TQZ)
- **DSD-PBEPBE-D3**: METHOD = DSD-PBEPBE-D3

From the double-hybrid mGGAs on Rung 5, we recommend:

- ω**B97M-(2)**: METHOD = wB97M-(2)
- **PTPSS-D3**: METHOD = PTPSS-D3

5.3.3 Exchange Functionals

Note: All exchange functionals in this section can be invoked using the *$rem* variable EXCHANGE. Popular and/or recommended functionals within each class are listed first and indicated in bold. The rest are in alphabetical order.

- Local Spin-Density Approximation (LSDA)
  - **Slater**: Slater-Dirac exchange functional (Xα method with $\alpha = 2/3$)[75]
  - SR_LSDA (BNL): Short-range version of the Slater-Dirac exchange functional[85]
- Generalized Gradient Approximation (GGA)
  - **PBE**: Perdew, Burke, and Ernzerhof exchange functional[197]
  - **B88**: Becke exchange functional from 1988[22]
  - **revPBE**: Zhang and Yang one-parameter modification of the PBE exchange functional[298]
  - AK13: Armiento-Kümmel exchange functional from 2013[13]
  - B86: Becke exchange functional (X$\alpha\beta\gamma$) from 1986[19]
  - G96: Gill exchange functional from 1996[82]
  - mB86: Becke "modified gradient correction" exchange functional from 1986[20]
  - mPW91: modified version (Adamo and Barone) of the 1991 Perdew-Wang exchange functional[6]
  - muB88 (μB88): Short-range version of the B88 exchange functional by Hirao and coworkers[115]
  - muPBE (μPBE): Short-range version of the PBE exchange functional by Hirao and coworkers[115]
  - srPBE: Short-range version of the PBE exchange functional by Goll and coworkers[88,89]
  - optB88: Refit version of the original B88 exchange functional (for use with vdW-DF-04) by Michaelides and coworkers[132]
  - OPTX: Two-parameter exchange functional by Handy and Cohen[103]
  - PBEsol: PBE exchange functional modified for solids[201]
  - PW86: Perdew-Wang exchange functional from 1986[193]
  - PW91: Perdew-Wang exchange functional from 1991[196]
  - RPBE: Hammer, Hansen, and Norskov exchange functional (modification of PBE)[101]
  - rPW86: Revised version (Murray *et al.*) of the 1986 Perdew-Wang exchange functional[180]
  - SOGGA: Second-order GGA functional by Zhao and Truhlar[305]
  - wPBE (ωPBE): Henderson *et al.* model for the PBE GGA short-range exchange hole[105]
- Meta-Generalized Gradient Approximation (meta-GGA)
  - **TPSS**: Tao, Perdew, Staroverov, and Scuseria exchange functional[249]
  - **revTPSS**: Revised version of the TPSS exchange functional[203]
  - BLOC: Minor modification of the TPSS exchange functional that works best with TPSSloc correlation (both by Della Sala and coworkers)[68]
  - modTPSS: One-parameter version of the TPSS exchange functional[200]
  - oTPSS: TPSS exchange functional with 5 refit parameters (for use with oTPSS correlation) by Grimme and coworkers[86]

- PBE-GX: First exchange functional based on a finite uniform electron gas (rather than an infinite UEG) by Pierre-François Loos [162]
- PKZB: Perdew, Kurth, Zupan, and Blaha exchange functional [198]
- regTPSS: Regularized (fixed order of limits issue) version of the TPSS exchange functional [229]
- SCAN: Strongly Constrained and Appropriately Normed exchange functional [246]
- rSCAN: Regularized SCAN exchange [17,80]
- r++SCAN: Regularized SCAN with uniform density limit and coordinate scaling behavior [80]
- r2SCAN: Re-Regularized SCAN exchange [78–80]
- r4SCAN: Regularized SCAN with exact constraints obeyed by SCAN [80]
- revSCAN: Revised SCAN exchange [172]
- TM: Tao-Mo exchange functional derived via an accurate modeling of the conventional exchange hole [248]
- regTM: Regularized TM exchange [188]
- revTM: Revised TM exchange [116]
- TASK: TASK exchange functional [14]
- mTASK: Modified TASK exchange functional [181]

5.3.4 Correlation Functionals

Note: All correlation functionals in this section can be invoked using the *$rem* variable CORRELATION. Popular and/or recommended functionals within each class are listed first and indicated in bold. The rest are in alphabetical order.

- Local Spin-Density Approximation (LSDA)
  - **PW92**: Perdew-Wang parameterization of the LSDA correlation energy from 1992 [194]
  - **VWN5** (VWN): Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #5 [265]
  - srVWN: Short-range version of the VWN correlation functional by Toulouse and coworkers [254]
  - Liu-Parr: Liu-Parr $\rho^{1/3}$ model from the functional expansion formulation [160]
  - PK09: Proynov-Kong parameterization of the LSDA correlation energy from 2009 [216]
  - PW92RPA: Perdew-Wang parameterization of the LSDA correlation energy from 1992 with RPA values [194]
  - srPW92: Short-range version of the PW92 correlation functional by Paziani and coworkers [189]
  - PZ81: Perdew-Zunger parameterization of the LSDA correlation energy from 1981 [195]
  - VWN1: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #1 [265]
  - VWN1RPA: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #1 with RPA values [265]
  - VWN2: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #2 [265]
  - VWN3: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #3 [265]
  - VWN4: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #4 [265]
  - Wigner:Wigner correlation functional (simplification of LYP) [242,280]
- Generalized Gradient Approximation (GGA)
  - BOP: “one-parameter progressive” correlation functional with parameters appropriate for B88 exchange [256]

- **PBE**: Perdew, Burke, and Ernzerhof correlation functional[197]
- **LYP**: Lee-Yang-Parr opposite-spin correlation functional[152]
- P86: Perdew-Wang correlation functional from 1986 based on the PZ81 LSDA functional[190]
- P86VWN5: Perdew-Wang correlation functional from 1986 based on the VWN5 LSDA functional[190]
- PBEloc: PBE correlation functional with a modified beta term by Della Sala and coworkers[67]
- PBEBOP: "one-parameter progressive" correlation functional with parameters appropriate for PBE exchange[256]
- PBEsol: PBE correlation functional modified for solids[201]
- srPBE: Short-range version of the PBE correlation functional by Goll and coworkers[88,89]
- PW91: Perdew-Wang correlation functional from 1991[196]
- regTPSS: Slight modification of the PBE correlation functional (also called vPBEc)[229]

◦ Meta-Generalized Gradient Approximation (meta-GGA)

- **TPSS**:Tao, Perdew, Staroverov, and Scuseria correlation functional[249]
- **revTPSS**: Revised version of the TPSS correlation functional[203]
- B95: Becke's two-parameter correlation functional from 1995[25]
- oTPSS: TPSS correlation functional with 2 refit parameters (for use with oTPSS exchange) by Grimme and coworkers[86]
- PK06: Proynov-Kong "tLap" functional with τ and Laplacian dependence[214]
- PKZB: Perdew, Kurth, Zupan, and Blaha correlation functional[198]
- SCAN: Strongly Constrained and Appropriately Normed correlation functional[246]
- rSCAN: Regularized SCAN correlation[17,80]
- r++SCAN: Regularized SCAN with uniform density limit and coordinate scaling behavior[80]
- r2SCAN: Re-Regularized SCAN correlation[78–80]
- revSCAN: Revised SCAN correlation[172]
- TM: Tao-Mo correlation functional, representing a minor modification to the TPSS correlation functional[248]
- revTM: Revised TM correlation[116]
- rregTM: Revised regularized TM correlation[117]
- TPSSloc: The TPSS correlation functional with the PBE component replaced by the PBEloc correlation functional[67]

5.3.5 Exchange-Correlation Functionals

Note: All exchange-correlation functionals in this section can be invoked using the *$rem* variable METHOD. For backwards compatibility, all of the exchange-correlation functionals *except* for the ones marked with an asterisk can be used with the *$rem* variable EXCHANGE. Popular and/or recommended functionals within each class are listed first and indicated in bold. The rest are in alphabetical order.

◦ Local Spin-Density Approximation (LSDA)

- **SPW92***: Slater LSDA exchange + PW92 LSDA correlation
- LDA: Slater LSDA exchange + VWN5 LSDA correlation

- SVWN5*: Slater LSDA exchange + VWN5 LSDA correlation

◦ Generalized Gradient Approximation (GGA)

- **B97-D3(0)**: B97-D with a fitted DFT-D3(0) tail instead of the original DFT-D2 tail[97]
- **B97-D**: 9-parameter dispersion-corrected (DFT-D2) functional by Grimme[95]
- **PBE***: PBE GGA exchange + PBE GGA correlation
- **BLYP***: B88 GGA exchange + LYP GGA correlation
- **revPBE***: revPBE GGA exchange + PBE GGA correlation
- BEEF-vdW: 31-parameter semi-empirical exchange functional developed via a Bayesian error estimation framework paired with PBE correlation and vdW-DF-10 NLC[277]
- BOP: B88 GGA exchange + BOP "one-parameter progressive" GGA correlation[256]
- BP86*: B88 GGA exchange + P86 GGA correlation
- BP86VWN*: B88 GGA exchange + P86VWN5 GGA correlation
- BPBE*: B88 GGA exchange + PBE GGA correlation
- EDF1: Modification of BLYP to give good performance in the 6-31+G* basis set[9]
- EDF2: Modification of B3LYP to give good performance in the cc-pVTZ basis set for frequencies[155]
- GAM: 21-parameter non-separable gradient approximation functional by Truhlar and coworkers[294]
- HCTH93 (HCTH/93): 15-parameter functional trained on 93 systems by Handy and coworkers[102]
- HCTH120 (HCTH/120): 15-parameter functional trained on 120 systems by Boese *et al.*[41]
- HCTH147 (HCTH/147): 15-parameter functional trained on 147 systems by Boese *et al.*[41]
- HCTH407 (HCTH/407): 15-parameter functional trained on 407 systems by Boese and Handy[38]
- HLE16 – HCTH/407 exchange functional enhanced by a factor of 1.25 + HCTH/407 correlation functional enhanced by a factor of 0.5[263]
- KT1: GGA functional designed specifically for shielding constant calculations[127]
- KT2: GGA functional designed specifically for shielding constant calculations[127]
- KT3: GGA functional with improved results for main-group nuclear magnetic resonance shielding constants[128]
- mPW91*: mPW91 GGA exchange + PW91 GGA correlation
- N12: 21-parameter non-separable gradient approximation functional by Peverati and Truhlar[209]
- OLYP*: OPTX GGA exchange + LYP GGA correlation
- PBEOP: PBE GGA exchange + PBEOP "one-parameter progressive" GGA correlation[256]
- PBEsol*: PBEsol GGA exchange + PBEsol GGA correlation
- PW91*: PW91 GGA exchange + PW91 GGA correlation
- RPBE*: RPBE GGA exchange + PBE GGA correlation
- rVV10*: rPW86 GGA exchange + PBE GGA correlation + rVV10 non-local correlation[230]
- SOGGA*: SOGGA GGA exchange + PBE GGA correlation
- SOGGA11: 20-parameter functional by Peverati, Zhao, and Truhlar[212]
- VV10: rPW86 GGA exchange + PBE GGA correlation + VV10 non-local correlation[271]

◦ Meta-Generalized Gradient Approximation (meta-GGA)

- **B97M-V**: 12-parameter combinatorially-optimized, dispersion-corrected (VV10) functional by Mardirossian and Head-Gordon[168]
- **B97M-rV***: B97M-V density functional with the VV10 NLC functional replaced by the rVV10 NLC functional[171]
- **M06-L**: 34-parameter functional by Zhao and Truhlar[303]
- **TPSS***: TPSS meta-GGA exchange + TPSS meta-GGA correlation
- **revTPSS***: revTPSS meta-GGA exchange + revTPSS meta-GGA correlation
- BLOC*: BLOC meta-GGA exchange + TPSSloc meta-GGA correlation
- M11-L: 44-parameter dual-range functional by Peverati and Truhlar[208]
- mBEEF: 64-parameter exchange functional paired with the PBEsol correlation functional[278]
- MGGA_MS0: MGGA_MS0 meta-GGA exchange + regTPSS GGA correlation[243]
- MGGA_MS1: MGGA_MS1 meta-GGA exchange + regTPSS GGA correlation[244]
- MGGA_MS2: MGGA_MS2 meta-GGA exchange + regTPSS GGA correlation[244]
- MGGA_MVS: MGGA_MVS meta-GGA exchange + regTPSS GGA correlation[245]
- MN12-L: 58-parameter meta-nonseparable gradient approximation functional by Peverati and Truhlar[210]
- MN15-L: 58-parameter meta-nonseparable gradient approximation functional by Yu, He, and Truhlar[296]
- oTPSS*: oTPSS meta-GGA exchange + oTPSS meta-GGA correlation
- PKZB*: PKZB meta-GGA exchange + PKZB meta-GGA correlation
- revM06-L: 31-parameter revised M06-L functional[273]
- SCAN*: SCAN meta-GGA exchange + SCAN meta-GGA correlation
- rSCAN: rSCAN exchange + rSCAN correltaion
- r++SCAN: r++SCAN exchange + r++SCAN correlation
- r2SCAN: r2SCAN exchange + r2SCAN correlation
- r4SCAN: r4SCAN exchange + r2SCAN correlation
- revSCAN: revSCAN exchange + revSCAN correlation
- t-HCTH (τ-HCTH): 16-parameter functional by Boese and Handy[39]
- TM*: TM meta-GGA exchange + TM meta-GGA correlation[248]
- revTM: revTM exchange + revTM correlation
- regTM: regTM exchange + regTPSS correlation
- rregTM: regTM exchange + rregTM correlation
- TASK: TASK exchange + PW92 correlation
- mTASK: mTASK exchange + PW92 correlation
- VSXC: 21-parameter functional by Voorhis and Scuseria[260]

◦ Global Hybrid Generalized Gradient Approximation (GH GGA)

- **B3LYP**: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange + 19% VWN1RPA LSDA correlation + 81% LYP GGA correlation[23,241]
- **PBE0**: 25% HF exchange + 75% PBE GGA exchange + PBE GGA correlation[7]
- **revPBE0**: 25% HF exchange + 75% revPBE GGA exchange + PBE GGA correlation
- **B97**: Becke's original 10-parameter density functional with 19.43% HF exchange[26]

- B1LYP: 25% HF exchange + 75% B88 GGA exchange + LYP GGA correlation[5]
- B1PW91: 25% HF exchange + 75% B88 GGA exchange + PW91 GGA correlation[5]
- B3LYP5: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[23,241]
- B3P86: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange+ 19% VWN1RPA LSDA correlation + 81% P86 GGA correlation
- B1LYP: 25% HF exchange + 75% B88 GGA exchange + LYP GGA correlation[5]
- B1PW91: 25% HF exchange + 75% B88 GGA exchange + PW91 GGA correlation[5]
- B3LYP5: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[23,241]
- B3P86: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange+ 19% VWN1RPA LSDA correlation + 81% P86 GGA correlation
- B3PW91: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange+ 19% PW92 LSDA correlation + 81% PW91 GGA correlation[23]
- B5050LYP: 50% HF exchange + 8% Slater LSDA exchange + 42% B88 GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[234]
- B97-1: Self-consistent parameterization of Becke's B97 density functional with 21% HF exchange[102]
- B97-2: Re-parameterization of B97 by Tozer and coworkers with 21% HF exchange[282]
- B97-3: 16-parameter version of B97 by Keal and Tozer with $\approx$ 26.93% HF exchange[129]
- B97-K: Re-parameterization of B97 for kinetics by Boese and Martin with 42% HF exchange[40]
- BHHLYP: 50% HF exchange + 50% B88 GGA exchange + LYP GGA correlation, sometimes called BH&H-LYP[92]
- HFLYP*: 100% HF exchange + LYP GGA correlation
- MPW1K: 42.8% HF exchange + 57.2% mPW91 GGA exchange + PW91 GGA correlation[164]
- MPW1LYP: 25% HF exchange + 75% mPW91 GGA exchange + LYP GGA correlation[6]
- MPW1PBE: 25% HF exchange + 75% mPW91 GGA exchange + PBE GGA correlation[6]
- MPW1PW91: 25% HF exchange + 75% mPW91 GGA exchange + PW91 GGA correlation[6]
- O3LYP: 11.61% HF exchange + $\approx$ 7.1% Slater LSDA exchange + 81.33% OPTX GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[112]
- PBEh-3c: Low-cost composite scheme of Grimme and coworkers for use with the def2-mSVP basis set only[99]
- PBE50: 50% HF exchange + 50% PBE GGA exchange + PBE GGA correlation[34]
- SOGGA11-X: 21-parameter functional with 40.15% HF exchange by Peverati and Truhlar[206]
- WC04: Hybrid density functional optimized for the computation of ^{13}C chemical shifts[281]
- WP04: Hybrid density functional optimized for the computation of ^{1}H chemical shifts[281]
- X3LYP: 21.8% HF exchange + 7.3% Slater LSDA exchange + $\approx$ 54.24% B88 GGA exchange + $\approx$ 16.66% PW91 GGA exchange + 12.9% VWN1RPA LSDA correlation + 87.1% LYP GGA correlation[292]

◦ Global Hybrid Meta-Generalized Gradient Approximation (GH meta-GGA)

- **M06-2X**: 29-parameter functional with 54% HF exchange by Zhao and Truhlar[307]
- **M08-HX**: 47-parameter functional with 52.23% HF exchange by Zhao and Truhlar[306]

- **TPSSh**: 10% HF exchange + 90% TPSS meta-GGA exchange + TPSS meta-GGA correlation[240]
- **revTPSSh**: 10% HF exchange + 90% revTPSS meta-GGA exchange + revTPSS meta-GGA correlation[71]
- B1B95: 28% HF exchange + 72% B88 GGA exchange + B95 meta-GGA correlation[25]
- B3TLAP: 17.13% HF exchange + 9.66% Slater LSDA exchange + 72.6% B88 GGA exchange + PK06 meta-GGA correlation[214,215]
- BB1K: 42% HF exchange + 58% B88 GGA exchange + B95 meta-GGA correlation[309]
- BMK: Boese-Martin functional for kinetics with 42% HF exchange[40]
- dlDF: Dispersion-less density functional (based on the M05-2X functional form) by Szalewicz and coworkers[205]
- M05: 22-parameter functional with 28% HF exchange by Zhao, Schultz, and Truhlar[310]
- M05-2X: 19-parameter functional with 56% HF exchange by Zhao, Schultz, and Truhlar[311]
- M06: 33-parameter functional with 27% HF exchange by Zhao and Truhlar[307]
- M06-HF: 32-parameter functional with 100% HF exchange by Zhao and Truhlar[304]
- M08-SO: 44-parameter functional with 56.79% HF exchange by Zhao and Truhlar[306]
- MGGA_MS2h: 9% HF exchange + 91 % MGGA_MS2 meta-GGA exchange + regTPSS GGA correlation[244]
- MGGA_MVSh: 25% HF exchange + 75 % MGGA_MVS meta-GGA exchange + regTPSS GGA correlation[245]
- MN15: 59-parameter functional with 44% HF exchange by Truhlar and coworkers[295]
- MPW1B95: 31% HF exchange + 69% mPW91 GGA exchange + B95 meta-GGA correlation[301]
- MPWB1K: 44% HF exchange + 56% mPW91 GGA exchange + B95 meta-GGA correlation[301]
- PW6B95: 6-parameter combination of 28 % HF exchange, 72 % optimized PW91 GGA exchange, and re-optimized B95 meta-GGA correlation by Zhao and Truhlar[302]
- PWB6K: 6-parameter combination of 46 % HF exchange, 54 % optimized PW91 GGA exchange, and re-optimized B95 meta-GGA correlation by Zhao and Truhlar[302]
- revM06: 32-parameter functional with 40.41% HF exchange[274]
- SCAN0: 25% HF exchange + 75% SCAN meta-GGA exchange + SCAN meta-GGA correlation[114]
- t-HCTHh (τ-HCTHh): 17-parameter functional with 15% HF exchange by Boese and Handy[39]
- TPSS0: 25% HF exchange + 75% TPSS meta-GGA exchange + TPSS meta-GGA correlation[93]

◦ Range-Separated Hybrid Generalized Gradient Approximation (RSH GGA)

- **wB97X-V** (ωB97X-V): 10-parameter combinatorially-optimized, dispersion-corrected (VV10) functional with 16.7% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$[167]
- **wB97X-D3** (ωB97X-D3): 16-parameter dispersion-corrected (DFT-D3(0)) functional with $\approx$ 19.57% SR HF exchange, 100% LR HF exchange, and $\omega = 0.25$[157]
- **wB97X-D** (ωB97X-D): 15-parameter dispersion-corrected (DFT-CHG) functional with $\approx$ 22.2% SR HF exchange, 100% LR HF exchange, and $\omega = 0.2$[57]
- CAM-B3LYP: Coulomb-attenuating method functional by Handy and coworkers[293]
- CAM-QTP00: Re-parameterized CAM-B3LYP designed to satisfy the IP-theorem for all occupied orbitals of the water molecule[262]
- CAM-QTP01: Re-parameterized CAM-B3LYP optimized to satisfy the valence IPs of the water molecule, 34 excitation states, and G2-1 atomization energies[121]

- HSE-HJS: Screened-exchange "HSE06" functional with 25% SR HF exchange, 0% LR HF exchange, and ω=0.11, using the updated HJS PBE exchange hole model[105,141]
- LC-rVV10*: LC-VV10 density functional with the VV10 NLC functional replaced by the rVV10 NLC functional[171]
- LC-VV10: 0% SR HF exchange + 100% LR HF exchange + ωPBE GGA exchange + PBE GGA correlation + VV10 non-local correlation (ω=0.45)[271]
- LC-wPBE08 (LC-ωPBE08): 0% SR HF exchange + 100% LR HF exchange + ωPBE GGA exchange + PBE GGA correlation (ω=0.45)[276]
- LRC-BOP (LRC-μBOP): 0% SR HF exchange + 100% LR HF exchange + μB88 GGA exchange + BOP GGA correlation (ω=0.47)[223,238]
- LRC-wPBE (LRC-ωPBE): 0% SR HF exchange + 100% LR HF exchange + ωPBE GGA exchange + PBE GGA correlation (ω=0.3)[224]
- LRC-wPBEh (LRC-ωPBEh): 20% SR HF exchange + 100% LR HF exchange + 80% ωPBE GGA exchange + PBE GGA correlation (ω=0.2)[225]
- N12-SX: 26-parameter non-separable GGA with 25% SR HF exchange, 0% LR HF exchange, and $\omega = 0.11$[211]
- rCAM-B3LYP: Re-fit CAM-B3LYP with the goal of minimizing many-electron self-interaction error[64]
- wB97 (ωB97): 13-parameter functional with 0% SR HF exchange, 100% LR HF exchange, and $\omega = 0.4$[56]
- wB97X (ωB97X): 14-parameter functional with $\approx$ 15.77% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$[56]
- wB97X-rV* (ωB97X-rV): ωB97X-V density functional with the VV10 NLC functional replaced by the rVV10 NLC functional[171]

◦ Range-Separated Hybrid Meta-Generalized Gradient Approximation (RSH meta-GGA)

- **wB97M-V** (ωB97M-V): 12-parameter combinatorially-optimized, dispersion-corrected (VV10) functional with 15% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$[169]
- M06-SX: local revM06-L functional with 33.5% SR HF exchange[275]
- M11: 40-parameter functional with 42.8% SR HF exchange, 100% LR HF exchange, and $\omega = 0.25$[207]
- MN12-SX: 58-parameter non-separable meta-GGA with 25% SR HF exchange, 0% LR HF exchange, and $\omega = 0.11$[211]
- revM11: 22-parameter functional with 22.5% SR HF exchange, 100% LR HF exchange, and $\omega = 0.4$[264]
- wB97M-rV* (ωB97X-rV): ωB97M-V density functional with the VV10 NLC functional replaced by the rVV10 NLC functional[171]
- wM05-D (ωM05-D): 21-parameter dispersion-corrected (DFT-CHG) functional with $\approx$ 36.96% SR HF exchange, 100% LR HF exchange, and $\omega = 0.2$[156]
- wM06-D3 (ωM06-D3): 25-parameter dispersion-corrected [DFT-D3(0)] functional with $\approx$ 27.15% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$[157]

◦ Double Hybrid Generalized Gradient Approximation (DH GGA)

Note: In order to use the resolution-of-the-identity approximation for the MP2 component, specify an auxiliary basis set with the *$rem* variable AUX_BASIS

- **DSD-PBEPBE-D3**: 68% HF exchange + 32% PBE GGA exchange + 49% PBE GGA correlation + 13% SS MP2 correlation + 55% OS MP2 correlation with DFT-D3(BJ) tail[138]

- **wB97X-2(LP)** (ωB97X-2(LP)): 13-parameter functional with $\approx$ 67.88% SR HF exchange, 100% LR HF exchange, $\approx$ 58.16% SS MP2 correlation, $\approx$ 47.80% OS MP2 correlation, and $\omega = 0.3$[58]
- **wB97X-2(TQZ)** (ωB97X-2(TQZ)): 13-parameter functional with $\approx$ 63.62% SR HF exchange, 100% LR HF exchange, $\approx$ 52.93% SS MP2 correlation, $\approx$ 44.71% OS MP2 correlation, and $\omega = 0.3$[58]
- **XYG3**: 80.33% HF exchange - 1.4% Slater LSDA exchange + 21.07% B88 GGA exchange + 67.89% LYP GGA correlation + 32.11% MP2 correlation (evaluated with B3LYP orbitals)[300]
- **XYGJ-OS**: 77.31% HF exchange + 22.69% Slater LSDA exchange + 23.09% VWN1RPA LSDA correlation + 27.54% LYP GGA correlation + 43.64% OS MP2 correlation (evaluated with B3LYP orbitals)[297]
- B2PLYP: 53% HF exchange + 47% B88 GGA exchange + 73% LYP GGA correlation + 27% MP2 correlation[94]
- B2GPPLYP: 65% HF exchange + 35% B88 GGA exchange + 64% LYP GGA correlation + 36% MP2 correlation[126]
- DSD-PBEP86-D3: 69% HF exchange + 31% PBE GGA exchange + 44% P86 GGA correlation + 22% SS MP2 correlation + 52% OS MP2 correlation with DFT-D3(BJ) tail[138]
- LS1DH-PBE: 75% HF exchange + 25% PBE GGA exchange + 57.8125% PBE GGA correlation + 42.1875% MP2 correlation[255]
- PBE-QIDH: 69.3361% HF exchange + 30.6639% PBE GGA exchange + 66.6667% PBE GGA correlation + 33.3333% MP2 correlation[44]
- PBE0-2: $\approx$ 79.37% HF exchange + $\approx$ 20.63% PBE GGA exchange + 50% PBE GGA correlation + 50% MP2 correlation[59]
- PBE0-DH: 50% HF exchange + 50% PBE GGA exchange + 87.5% PBE GGA correlation + 12.5% MP2 correlation[43]

◦ Double Hybrid Meta-Generalized Gradient Approximation (DH MGGA)

- **wB97M(2)**: 14-parameter functional form by wB97M-V + MP2 correlation.[170]
- **PTPSS-D3**: 50% HF exchange + 50% Re-Fit TPSS meta-GGA exchange + 62.5% Re-Fit TPSS meta-GGA correlation + 37.5% OS MP2 correlation with DFT-D3(0) tail[87]
- DSD-PBEB95-D3: 66% HF exchange + 34% PBE GGA exchange + 55% B95 GGA correlation + 9% SS MP2 correlation + 46% OS MP2 correlation with DFT-D3(BJ) tail[138]
- PWPB95-D3: 50% HF exchange + 50% Re-Fit PW91 GGA exchange + 73.1% Re-Fit B95 meta-GGA correlation + 26.9% OS MP2 correlation with DFT-D3(0) tail[87]

5.3.6 Specialized Functionals

- SRC1-R1: TDDFT short-range corrected functional [Eq. (1) in Ref. 35, 1st row atoms]
- SRC1-R2: TDDFT short-range corrected functional [Eq. (1) in Ref. 35, 2nd row atoms]
- SRC2-R1: TDDFT short-range corrected functional [Eq. (2) in Ref. 35, 1st row atoms]
- SRC2-R2: TDDFT short-range corrected functional [Eq. (2) in Ref. 35, 2nd row atoms]
- BR89: Becke-Roussel meta-GGA exchange functional modeled after the hydrogen atom[32]
- B94: meta-GGA correlation functional by Becke that uses the BR89 exchange functional to compute the Coulomb potential[24]

- B94hyb: modified version of the B94 correlation functional for use with the BR89B94hyb exchange-correlation functional[24]
- BR89B94h: 15.4% HF exchange + 84.6% BR89 meta-GGA exchange + BR89hyb meta-GGA correlation[24]
- BRSC: Exchange component of the original B05 exchange-correlation functional[27]
- MB05: Exchange component of the modified B05 (BM05) exchange-correlation functional[218]
- B05: A full exact-exchange Kohn-Sham scheme of Becke that uses the exact-exchange energy density (RI) and accounts for static correlation[27,217,219]
- BM05 (XC): Modified B05 hyper-GGA scheme that uses MB05 instead of BRSC as the exchange functional[218]
- PSTS: Hyper-GGA (100% HF exchange) exchange-correlation functional of Perdew, Staroverov, Tao, and Scuseria[202]
- MCY2: Mori-Sánchez-Cohen-Yang adiabatic connection-based hyper-GGA exchange-correlation functional[63,158,178]

This example illustrate the use of the RI-B05 and RI-PSTS functionals. These are presently available only for single-point calculations, and convergence is greatly facilitated by obtaining converged SCF orbitals from, *e.g.*, an LDA or HF calculation first. (LDA is used in the example below but HF can be substituted.) Use of the RI approximation

(Section 6.6) requires specification of an auxiliary basis set.

Example 5.1 Q-CHEM input of H_2 using RI-B05.

```
$comment
   H2, example of SP RI-B05.  First do a well-converged LSD, G3LARGE is the
   basis of choice for good accuracy.
$end

$molecule
   0 1
   H   0.  0.   0.0
   H   0.  0.   0.7414
$end

$rem
   METHOD           lda
   BASIS            g3large
   SCF_GUESS        core     ! required
   PURECART         2222     ! required
   THRESH           14
   INCDFT           false
   SCF_CONVERGENCE  9
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

@@@

$molecule
   read
$end

$rem
   EXCHANGE        b05            ! or set to psts for ri-psts
   BASIS           g3large
   AUX_BASIS       rib05-cc-pvtz  ! the aux basis for both RI-B05 and RI-PSTS
   SCF_GUESS       read
   PURECART        2222           ! required
   THRESH          4
   PRINT_INPUT     true
   INCDFT          false
   MAX_SCF_CYCLES  0              ! required
   DFT_CUTOFFS     0              ! required
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end
```

5.3.7 User-Defined Density Functionals

Users can also request a customized density functional consisting of any linear combination of exchange and/or correlation functionals available in Q-CHEM. A "general" density functional of this sort is requested by setting EXCHANGE = GEN and then specifying the functional by means of an *$xc_functional* input section consisting of one line for each

desired exchange (X) or correlation (C) component of the functional, and having the format shown below.

```
$xc_functional
   X   exchange_symbol   coefficient
   X   exchange_symbol   coefficient
   ...
   C   correlation_symbol   coefficient
   C   correlation_symbol   coefficient
   ...
   K   coefficient
$end
```

Each line requires three variables: X or C to designate whether this is an exchange or correlation component; the symbolic representation of the functional, as would be used for the EXCHANGE or CORRELATION keywords variables as described above; and a real number coefficient for each component. Note that Hartree-Fock exchange can be

designated either as “X" or as “K". Examples are shown below.

Example 5.2 Q-CHEM input for H_2O with the B3tLap functional.

```
$molecule
   0  1
   O
   H1  O  oh
   H2  O  oh  H1  hoh

   oh  =   0.97
   hoh = 120.0
$end

$rem
   EXCHANGE     gen
   CORRELATION  none
   BASIS        g3large   ! recommended for high accuracy
   THRESH       14        ! and better convergence
$end

$xc_functional
   X    Becke     0.726
   X    S         0.0966
   C    PK06      1.0
   K    0.1713
$end
```

Example 5.3 Q-CHEM input for H_2O with the BR89B94hyb functional.

```
$molecule
   0  1
   O
   H1  O  oh
   H2  O  oh  H1  hoh

   oh  =   0.97
   hoh = 120.0
$end

$rem
   EXCHANGE     gen
   CORRELATION  none
   BASIS        g3large   ! recommended for high accuracy
   THRESH       14        ! and better convergence
$end

$xc_functional
   X    BR89      0.846
   C    B94hyb    1.0
   K              0.154
$end
```

5.3.8 Semi-Empirical Functionals

The following semi-empirical methods are available in Q-CHEM:

- **HF-3c**: It is an HF-based semi-empirical method which is used with a minimal basis called MINIX. All the elements from H to Xe are supported. For the elements from Rb ($Z = 37$) to Xe ($Z = 54$), def2-ECP must be used.[247]
- **PBEh-3c**: It is a DFT based method which is used with def2-mSVP basis set.This is available for all the elements from H to Rn. For the elements from Rb ($Z = 37$) to Rn ($Z = 86$), def2-ECP has to be used. It is a more accurate method than HF-3c as it has correlation,[99] although the double-ζ basis set makes it more expensive for large molecules.

Note: The HF-3c and PBEh-3c methods were parameterized for use with the aforementioned basis sets. Although Q-CHEM does not enforce this, other basis sets should not be used unless there is a good reason for doing so, and only then with caution as the results may not match published accuracy.

Example 5.4 Q-CHEM input for HF-3c method.

```
$rem
 method  hf-3c
 basis   minix
 ecp     def2-ecp
$end

$molecule
-1 1
        I         0.6539056222    0.0300979939    0.0000000000
        O        -2.8593081518    0.0835488510    0.0000000000
        H        -1.8947008157    0.2596704547    0.0000000000
        H        -2.8641221666   -0.8778240298    0.0000000000
$end
```

Example 5.5 Q-CHEM input for PBEh-3c method.

```
$molecule
   0 1
   O  -1.551007  -0.114520   0.000000
   H  -1.934259   0.762503   0.000000
   H  -0.599677   0.040712   0.000000
   O   1.350625   0.111469   0.000000
   H   1.680398  -0.373741  -0.758561
   H   1.680398  -0.373741   0.758561
$end

$rem
   METHOD           PBEh-3c
   BASIS            def2-mSVP
   XC_GRID          000099000590
   INCDFT           0
   INCFOCK          0
   SCF_CONVERGENCE  8
   THRESH           14
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

5.4 Basic DFT Job Control

Basic SCF job control was described in Section 4.3 in the context of Hartree-Fock theory and is largely the same for DFT. The keywords METHOD and BASIS are required, although for DFT the former could be substituted by specifying EXCHANGE and CORRELATION instead.

METHOD

Specifies the exchange-correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use METHOD = *NAME*, where *NAME* is either HF for Hartree-Fock theory or else one of the DFT methods listed in Section 5.3.5.

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations for DFT are indicated in bold in Section 5.3.5.

EXCHANGE

Specifies the exchange functional (or most exchange-correlation functionals for backwards compatibility).

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use EXCHANGE = *NAME*, where *NAME* is either:

1) One of the exchange functionals listed in Section 5.3.3

2) One of the XC functionals listed in Section 5.3.5 that is not marked with an asterisk.

3) GEN, for a user-defined functional (see Section 5.3.7).

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations are indicated in bold in Sections 5.3.5 and 5.3.3.

CORRELATION

Specifies the correlation functional.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

NAME Use CORRELATION = *NAME*, where *NAME* is one of the correlation functionals listed in Section 5.3.4.

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations are indicated in bold in Section 5.3.4.

The following *$rem* variables are related to the choice of the quadrature grid required to integrate the XC part of the functional, which does not appear in Hartree-Fock theory. DFT quadrature grids are described in Section 5.5.

FAST_XC

Controls direct variable thresholds to accelerate exchange-correlation (XC) in DFT.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Turn FAST_XC on.

FALSE Do not use FAST_XC.

RECOMMENDATION:

Caution: FAST_XC improves the speed of a DFT calculation, but may occasionally cause the SCF calculation to diverge.

XC_GRID

Specifies the type of grid to use for DFT calculations.

TYPE:

INTEGER

DEFAULT:

Functional-dependent; see Table 5.3.

OPTIONS:

0 Use SG-0 for H, C, N, and O; SG-1 for all other atoms.

n Use SG-n for all atoms, $n = 1, 2$, or 3

XY A string of two six-digit integers X and Y, where X is the number of radial points and Y is the number of angular points where possible numbers of Lebedev angular points, which must be an allowed value from Table 5.2 in Section 5.5.

$-XY$ Similar format for Gauss-Legendre grids, with the six-digit integer X corresponding to the number of radial points and the six-digit integer Y providing the number of Gauss-Legendre angular points, $Y = 2N^2$.

RECOMMENDATION:

Use the default unless numerical integration problems arise. Larger grids may be required for optimization and frequency calculations.

NL_GRID

Specifies the grid to use for non-local correlation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

Same as for XC_GRID

RECOMMENDATION:

Use the default unless computational cost becomes prohibitive, in which case SG-0 may be used. XC_GRID should generally be finer than NL_GRID.

XC_SMART_GRID

Uses SG-0 (where available) for early SCF cycles, and switches to the (larger) target grid specified by XC_GRID for final cycles of the SCF.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE (or 1) Use the smaller grid for the initial cycles.

FALSE (or 0) Use the target grid for all SCF cycles.

RECOMMENDATION:

The use of the smart grid can save some time on initial SCF cycles.

5.5 DFT Numerical Quadrature

5.5.1 Introduction

In practical DFT calculations, the forms of the approximate exchange-correlation functionals used are quite complicated, such that the required integrals involving the functionals generally cannot be evaluated analytically. Q-CHEM evaluates these integrals through numerical quadrature directly applied to the exchange-correlation integrand. Several standard quadrature grids are available ("SG-n", $n = 0, 1, 2, 3$), with a default value that is automatically set according to the complexity of the functional in question.

The quadrature approach in Q-CHEM is generally similar to that found in many DFT programs. The multi-center XC integrals are first partitioned into "atomic" contributions using a nuclear weight function. Q-CHEM uses the nuclear partitioning of Becke,[21] though without the "atomic size adjustments" of Ref. 21. The atomic integrals are then evaluated through standard one-center numerical techniques. Thus, the exchange-correlation energy is obtained as

$$E_{\mathrm{XC}} = \sum_{A}^{\text{atoms}} \sum_{i \in A}^{\text{points}} w_{Ai} f(\mathbf{r}_{Ai}) \,, \tag{5.15}$$

where the function f is the aforementioned XC integrand and the quantities w_{Ai} are the quadrature weights. The sum over i runs over grid points belonging to atom A, which are located at positions $\mathbf{r}_{Ai} = \mathbf{R}_A + \mathbf{r}_i$, so this approach requires only the choice of a suitable one-center integration grid (to define the $\mathbf{r}_i$), which is independent of nuclear configuration. These grids are implemented in Q-CHEM in a way that ensures that the E_{XC} is rotationally-invariant, *i.e.*, that is does not change when the molecule undergoes rigid rotation in space.[122]

Quadrature grids are further separated into radial and angular parts. Within Q-CHEM, the radial part is usually treated by the Euler-Maclaurin scheme proposed by Murray *et al.*,[179] which maps the semi-infinite domain $[0, \infty)$ onto $[0, 1)$ and applies the extended trapezoid rule to the transformed integrand. Alternatively, Gill and Chien proposed a radial scheme based on a Gaussian quadrature on the interval $[0, 1]$ with a different weight function.[61] This "MultiExp" radial quadrature is exact for integrands that are a linear combination of a geometric sequence of exponential functions, and is therefore well suited to evaluating atomic integrals. However, the task of generating the MultiExp quadrature points becomes increasingly ill-conditioned as the number of radial points increases, so that a "double exponential" radial quadrature[173,174] is used for the largest standard grids in Q-CHEM,[173,174] namely SG-2 and SG-3.[72] (See Section 5.5.3.)

No. Points	Degree ($\ell_{\max}$)	No. Points	Degree ($\ell_{\max}$)	No. Points	Degree ($\ell_{\max}$)
6	3	230	25	1730	71
18	5	266	27	2030	77
26	7	302	29	2354	83
38	9	350	31	2702	89
50	11	434	35	3074	95
74	13	590	41	3470	101
86	15	770	47	3890	107
110	17	974	53	4334	113
146	19	1202	59	4802	119
170	21	1454	65	5294	125
194	23				

Table 5.2: Lebedev angular quadrature grids available in Q-CHEM.

5.5.2 Angular Grids

For a fixed value of the radial spherical-polar coordinate r, a function $f(\mathbf{r}) \equiv f(r,\theta,\phi)$ has an exact expansion in spherical harmonic functions,

$$f(r,\theta,\phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} c_{\ell m} \mathrm{Y}_{\ell m}(\theta,\phi) \; . \tag{5.16}$$

Angular quadrature grids are designed to integrate $f(r,\theta,\phi)$ for fixed r, and are often characterized by their degree, meaning the maximum value of ℓ for which the quadrature is exact, as well as by their efficiency, meaning the number of spherical harmonics exactly integrated per degree of freedom in the formula. Q-CHEM supports the following two types of angular grids.

- **Lebedev grids.** These are specially-constructed grids for quadrature on the surface of a sphere,[148–151] based on the octahedral point group. Lebedev grids available in Q-CHEM are listed in Table 5.2. These grids typically have near-unit efficiencies, with efficiencies exceeding unity in some cases. A Lebedev grid is selected by specifying the number of grid points (from Table 5.2) using the *$rem* keyword XC_GRID, as discussed below.

- **Gauss-Legendre grids.** These are spherical direct-product grids in the two spherical-polar angles, θ and ϕ. Integration in over θ is performed using a Gaussian quadrature derived from the Legendre polynomials, while integration over ϕ is performed using equally-spaced points. A Gauss-Legendre grid is selected by specifying the total number of points, $2N^2$, to be used for the integration, which specifies a grid consisting of $2N_\phi$ points in ϕ and N_θ in θ, for a degree of $2N-1$. Gauss-Legendre grids exhibit efficiencies of only 2/3, and are thus lower in quality than Lebedev grids for the same number of grid points, but have the advantage that they are defined for arbitrary (and arbitrarily-large) numbers of grid points. This offers a mechanism to achieve arbitrary accuracy in the angular integration, if desired.

Combining these radial and angular schemes yields an intimidating selection of quadratures, so it is useful to standardize the grids. This is done for the convenience of the user, to facilitate comparisons in the literature, and also for developers wishing to compared detailed results between different software programs, because the total electronic energy is sensitive to the details of the grid, just as it is sensitive to details of the basis set. Standard quadrature grids are discussed next.

Pruned Grid	Ref.	Parent Grid (N_r, N_Ω)	No. Grid Points (C atom)[a]	Default Grid for Which Functionals?[b]
SG-0	62	(23, 170)	1,390 (36%)	None
SG-1	84	(50, 194)	3,816 (39%)	LDA, most GGAs and hybrids
SG-2	72	(75, 302)	7,790 (34%)	Meta-GGAs; B95- and B97-based functionals
SG-3	72	(99, 590)	17,674 (30%)	Minnesota functionals

[a]Number in parenthesis is the fraction of points retained from the parent grid

[b]Reflects Q-CHEM versions since v. 4.4.2

Table 5.3: Standard quadrature grids available in Q-CHEM, along with the number of grid points for a carbon atom, showing the reduction in grid points due to pruning.

5.5.3 Standard Quadrature Grids

Four different "standard grids" are available in Q-CHEM, designated SG-n, for $n = 0, 1, 2$, or 3; both quality and the computational cost of these grids increases with n. These grids are constructed starting from a "parent" grid $(N_r,\ N_\Omega)$ consisting of N_r radial spheres with N_Ω angular (Lebedev) grid points on each, then systematically pruning the number of angular points in regions where sophisticated angular quadrature is not necessary, such as near the nuclei where the charge density is nearly spherically symmetric and at long distance from the nuclei where it varies slowly. A large number of points are retained in the valence region where angular accuracy is critical. The SG-n grids are summarized in Table 5.3. While many electronic structure programs use some kind of procedure to delete unnecessary grid points in the interest of computational efficiency, Q-CHEM's SG-n grids are notable in that the complete grid specifications are available in the peer-reviewed literature,[62,72,84] to facilitate reproduction of Q-CHEM DFT calculations using other electronic structure programs. Just as computed energies may vary quite strongly with the choice of basis set, so too in DFT they may vary strongly with the choice of quadrature grid. In publications, users should always specify the grid that is used, and it is suggested to cite the appropriate literature reference from Table 5.3.

The SG-0 and SG-1 grids are designed for calculations on large molecules using GGA functionals. SG-1 affords integration errors on the order of ∼0.2 kcal/mol for medium-sized molecules and GGA functionals, including for demanding test cases such as reaction enthalpies for isomerizations. (Integration errors in total energies are no more than a few μhartree, or ∼0.01 kcal/mol.) The SG-0 grid was derived in similar fashion, and affords a root-mean-square error in atomization energies of 72 μhartree with respect to SG-1, while relative energies are reproduced well.[62] In either case, errors of this magnitude are typically considerably smaller than the intrinsic errors in GGA energies, and hence acceptable. As seen in Table 5.3, SG-1 retains $< 40\%$ of the grid points of its parent grid, which translates directly into cost savings.

Both SG-0 and SG-1 were optimized so that the integration error in the energy falls below a target threshold, but derivatives of the energy (including such properties as (hyper)polarizabilities[51]) are often more sensitive to the quality of the integration grid. Special care is required, for example, when imaginary vibrational frequencies are encountered, as low-frequency (but real) vibrational frequencies can manifest as imaginary if the grid is sparse. If imaginary frequencies are found, or if there is some doubt about the frequencies reported by Q-CHEM, the recommended procedure is to perform the geometry optimization and vibrational frequency calculations again using a higher-quality grid. (The optimization should converge quite quickly if the previously-optimized geometry is used as an initial guess.)

SG-1 was the default DFT integration grid for all density functionals for Q-CHEM versions 3.2–4.4. Beginning with Q-CHEM v. 4.4.2, however, the default grid is functional-dependent, as summarized in Table 5.3. This is a reflection of the fact that although SG-1 is adequate for energy calculations using most GGA and hybrid functionals (although care must be taken for some other properties, as discussed below), it is *not* adequate to integrate many functionals developed since ∼2005. These include meta-GGAs, which are more complicated due to their dependence on the kinetic

energy density (τ_σ in Eq. (5.10)) and/or the Laplacian of the density ($\nabla^2\rho_\sigma$). Functionals based on B97, along with the Minnesota suite of functionals,[307,308] contain relatively complicated expressions for the exchange inhomogeneity factor, and are therefore also more sensitive to the quality of the integration grid.[72,167,279] To integrate these modern density functionals, the SG-2 and SG-3 grids were developed,[72] which are pruned versions of the medium-quality (75, 302) and high-quality (99, 590) integration grids, respectively. Tests of properties known to be highly sensitive to the quality of the integration grid, such as vibrational frequencies, hyper-polarizabilities, and potential energy curves for non-bonded interactions, demonstrate that SG-2 is usually adequate for meta-GGAs and B97-based functionals, and in many cases is essentially converged with respect to an unpruned (250, 974) grid.[72] The Minnesota functionals are more sensitive to the grid, and while SG-3 is often adequate, it is not completely converged in the case of non-bonded interactions.[72]

Note:

1. SG-0 was re-optimized for Q-CHEM v. 3.0, so results may differ slightly as compared to older versions of the program.

2. The SG-2 and SG-3 grids use a double-exponential radial quadrature,[72] whereas a general grid (selected by setting XC_GRID = XY, as described in Section 5.4) uses an Euler-MacLaurin radial quadrature. As such, *absolute* energies cannot be compared between, *e.g.*, SG-2 and XC_GRID = 000075000302, even though SG-2 uses a pruned (75, 302) grid. However, energy *differences* should be quite similar between the two.

3. As noted in Ref. 72, for Minnesota functionals some wiggles in potential energy surfaces may persist with the SG-3 grid, especially for longer-range non-bonded interactions. Although these are rarely problematic for energy differences, if the user wants to eliminate these oscillations then we recommend an unpruned (99, 590) grid, *i.e.*, XC_GRID = 000099000590.

4. Although much has been said in the literature regarding the grid sensitivity of the SCAN functional,[17,78] the conventional wisdom that SCAN is extremely sensitive to grid quality is largely based on calculations in condensed-matter (periodic DFT) codes. In Q-CHEM, the SG-2 and SG-3 grids appear to be of sufficient quality for SCAN calculations.[45] (That said, regularized functionals such as r^{2}SCAN[78] do reduce the grid dependence.)

5.5.4 Consistency Check and Cutoffs

Whenever Q-CHEM calculates numerical density functional integrals, the electron density itself is also integrated numerically as a test of the quality of the numerical quadrature. The extent to which this numerical result differs from the number of electrons is an indication of the accuracy of the other numerical integrals. A warning message is printed whenever the relative error in the numerical electron count reaches 0.01%, indicating that the numerical XC results may not be reliable. If the warning appears on the first SCF cycle it is probably not serious, because the initial-guess density matrix is sometimes not idempotent. This is the case with the SAD guess discussed in Section 4.4, and also with a density matrix that is taken from a previous geometry optimization cycle, and in such cases the problem will likely correct itself in subsequent SCF iterations. If the warning persists, however, then one should consider either using a finer grid or else selecting an alternative initial guess.

By default, Q-CHEM will estimate the magnitude of various XC contributions on the grid and eliminate those determined to be numerically insignificant. Q-CHEM uses specially-developed cutoff procedures which permits evaluation of the XC energy and potential in only $\mathcal{O}(N)$ work for large molecules. This is a significant improvement over the formal $\mathcal{O}(N^3)$ scaling of the XC cost, and is critical in enabling DFT calculations to be carried out on very large systems. In rare cases, however, the default cutoff scheme can be too aggressive, eliminating contributions that should be retained; this is almost always signaled by an inaccurate numerical density integral. An example of when this could

occur is in calculating anions with multiple sets of diffuse functions in the basis. A remedy may be to increase the size of the quadrature grid.

5.5.5 Multi-resolution Exchange-Correlation (MRXC) Method

The multi-resolution exchange-correlation (MRXC) method is a new approach, courtesy of the Q-CHEM development team,[60,135,227] for accelerating computation of the exchange-correlation (XC) energy and matrix for any given density functional. As explained in Section 4.6.5, XC functionals are sufficiently complicated integration of them is usually performed by numerical quadrature. There are two basic types of quadrature. One is the atom-centered grid (ACG), a superposition of atomic quadrature described in Section 4.6.5. The ACG has high density of points near the nucleus to handle the compact core density and low density of points in the valence and non-bonding region where the electron density is smooth. The other type is even-spaced cubic grid (ESCG), which is typically used together with pseudopotentials and plane-wave basis functions where only the valence and non-bonded electron density is assumed smooth. In quantum chemistry, an ACG is more often used as it can handle accurately all-electron calculations of molecules. MRXC combines those two integration schemes seamlessly to achieve an optimal computational efficiency by placing the calculation of the smooth part of the density and XC matrix onto the ESCG. The computation associated with the smooth fraction of the electron density is the major bottleneck of the XC part of a DFT calculation and can be done at a much faster rate on the ESCG due to its low resolution. Fast Fourier transform and B-spline interpolation are employed for the accurate transformation between the two types of grids such that the final results remain the same as they would be on the ACG alone, yet a speedup of several times is achieved for the XC matrix. The smooth part of the calculation with MRXC can also be combined with FTC (see Section 4.6.5) to achieve a further gain in efficiency.

MRXC

Controls the use of MRXC.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use MRXC

1 Use MRXC in the evaluation of the XC part

RECOMMENDATION:

MRXC is very efficient for medium and large molecules, especially when medium and large basis sets are used.

The following two keywords control the smoothness precision. The default value is carefully selected to maintain high accuracy.

MRXC_CLASS_THRESH_MULT

Controls the of smoothness precision

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

im An integer

RECOMMENDATION:

A prefactor in the threshold for MRXC error control: $im \times 10^{-io}$

MRXC_CLASS_THRESH_ORDER

Controls the of smoothness precision

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

io An integer

RECOMMENDATION:

The exponent in the threshold of the MRXC error control: $im \times 10^{-io}$

The next keyword controls the order of the B-spline interpolation:

LOCAL_INTERP_ORDER

Controls the order of the B-spline

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n An integer

RECOMMENDATION:

The default value is sufficiently accurate

5.5.6 Incremental DFT

Incremental DFT (IncDFT) uses the difference density and functional values to improve the performance of the DFT quadrature procedure by providing a better screening of negligible values. Using this option will yield improved efficiency at each successive iteration due to more effective screening.

INCDFT

Toggles the use of the IncDFT procedure for DFT energy calculations.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not use IncDFT

TRUE Use IncDFT

RECOMMENDATION:

Turning this option on can lead to faster SCF calculations, particularly towards the end of the SCF. Please note that for some systems use of this option may lead to convergence problems.

INCDFT_DENDIFF_THRESH

Sets the threshold for screening density matrix values in the IncDFT procedure.

TYPE:

INTEGER

DEFAULT:

SCF_CONVERGENCE + 3

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

INCDFT_GRIDDIFF_THRESH

Sets the threshold for screening functional values in the IncDFT procedure

TYPE:

INTEGER

DEFAULT:

SCF_CONVERGENCE + 3

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

INCDFT_DENDIFF_VARTHRESH

Sets the lower bound for the variable threshold for screening density matrix values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT_DENDIFF_THRESH is reached. This means this value must be set lower than INCDFT_DENDIFF_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

INCDFT_GRIDDIFF_VARTHRESH

Sets the lower bound for the variable threshold for screening the functional values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT_GRIDDIFF_THRESH is reached. This means that this value must be set lower than INCDFT_GRIDDIFF_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

5.6 Range-Separated Hybrid Density Functionals

5.6.1 Introduction

Whereas RSH functionals such as LRC-ωPBE are attempts to add 100% LR Hartree-Fock exchange with minimal perturbation to the original functional (PBE, in this example), other RSH functionals are of a more empirical nature and their range-separation parameters have been carefully parameterized along with all of the other parameters in the functional. These cases are functionals are discussed first, in Section 5.6.2, because their range-separation parameters should be taken as fixed. User-defined values of the range-separation parameter are discussed in Section 5.6.3, and Section 5.6.4 discusses a procedure for which an optimal, system-specific value of this parameter (ω or μ) can be chosen for functionals such as LRC-ωPBE or LRC-μPBE.

5.6.2 Semi-Empirical RSH Functionals

Semi-empirical RSH functionals for which the range-separation parameter should be considered fixed include the ωB97, ωB97X, and ωB97X-D functionals developed by Chai and Head-Gordon;[56,57] ωB97X-V and ωB97M-V from Mardirossian and Head-Gordon;[167,169] M11 from Peverati and Truhlar;[207] ωB97X-D3, ωM05-D, and ωM06-D3 from Chai and coworkers;[156,157] and the screened exchange functionals N12-SX and MN12-SX from Truhlar and coworkers.[211] More recently, Mardirossian and Head-Gordon developed two RSH functionals, ωB97X-V and ωB97M-V, via a combinatorial approach by screening over 100,000 possible functionals in the first case and over 10 billion possible functionals in the second case. Both of the latter functionals use the VV10 non-local correlation functional in order to improve the description of non-covalent interactions and isomerization energies. ωB97M-V is a 12-parameter meta-GGA with 15% short-range exact exchange and 100% long-range exact exchange and is one of the most accurate functionals available through rung 4 of Jacob's Ladder, across a wide variety of applications. This has been verified by benchmarking the functional on nearly 5000 data points against over 200 alternative functionals available in Q-CHEM.[169]

5.6.3 User-Defined RSH Functionals

As pointed out in Ref. 77 and elsewhere, the description of charge-transfer excited states within density functional theory (or more precisely, time-dependent DFT, which is discussed in Section 7.3) requires full (100%) non-local HF

exchange, at least in the limit of large donor–acceptor distance. Hybrid functionals such as B3LYP[23,241] and PBE0[8] that are well-established and in widespread use, however, employ only 20% and 25% HF exchange, respectively. While these functionals provide excellent results for many ground-state properties, they cannot correctly describe the distance dependence of charge-transfer excitation energies, which are enormously underestimated by most common density functionals. This is a serious problem in any case, but it is a *catastrophic* problem in large molecules and in non-covalent clusters, where TDDFT often predicts a near-continuum of spurious, low-lying charge transfer states.[144,145] The problems with TDDFT's description of charge transfer are not limited to large donor–acceptor distances, but have been observed at $\sim$2 Å separation, in systems as small as uracil–$(H_2O)_4$.[144] Rydberg excitation energies also tend to be substantially underestimated by standard TDDFT.

One possible avenue by which to correct such problems is to parameterize functionals that contain 100% HF exchange, though few such functionals exist to date. An alternative option is to attempt to preserve the form of common GGAs and hybrid functionals at short range (*i.e.*, keep the 25% HF exchange in PBE0) while incorporating 100% HF exchange at long range, which provides a rigorously correct description of the long-range distance dependence of charge-transfer excitation energies, but aims to avoid contaminating short-range exchange-correlation effects with additional HF exchange. The separation is accomplished using the range-separation *ansatz* that was introduced in Section 5.3. In particular, functionals that use 100% HF exchange at long range ($c_{x,\mathrm{LR}} = 1$ in Eq. (5.13)) are known as "long-range-corrected" (LRC) functionals. An LRC version of PBE0 would, for example, have $c_{x,\mathrm{SR}} = 0.25$.

To fully specify an LRC functional, one must choose a value for the range separation parameter ω in Eq. (5.12). In the limit $\omega \to 0$, the LRC functional in Eq. (5.13) reduces to a non-RSH functional where there is no "SR" or "LR", because all exchange and correlation energies are evaluated using the full Coulomb operator, r_{12}^{-1}. Meanwhile the $\omega \to \infty$ limit corresponds to a new functional, $E_{xc}^{\mathrm{RSH}} = E_c + E_x^{\mathrm{HF}}$. Full HF exchange is inappropriate for use with most contemporary GGA correlation functionals, so the latter limit is expected to perform quite poorly. Values of $\omega > 1.0$ bohr^{-1} are likely not worth considering, according to benchmark tests.[146,224]

Evaluation of the short- and long-range HF exchange energies is straightforward,[10] so the crux of any RSH functional is the form of the short-range GGA exchange functional, and several such functionals are available in Q-CHEM. These include short-range variants of the B88 and PBE exchange described by Hirao and co-workers,[115,238] called μB88 and μPBE in Q-CHEM,[223] and an alternative formulation of short-range PBE exchange proposed by Scuseria and co-workers,[105] which is known as ωPBE. These functionals are available in Q-CHEM thanks to the efforts of the Herbert group.[224,225] By way of notation, the terms "μPBE", "ωPBE", *etc.*, refer only to the short-range exchange functional, $E_{x,\mathrm{SR}}^{\mathrm{DFT}}$ in Eq. (5.13). These functionals could be used in "screened exchange" mode, as described in Section 5.3, as for example in the HSE03 functional,[110] therefore the designation "LRC-ωPBE", for example, should only be used when the short-range exchange functional ωPBE is combined with 100% Hartree-Fock exchange in the long range.

In general, LRC-DFT functionals have been shown to remove the near-continuum of spurious charge-transfer excited states that appear in large-scale TDDFT calculations.[146] However, certain results depend sensitively upon the value of the range-separation parameter ω,[145,146,224,225,257] especially in TDDFT calculations (Section 7.3) and therefore the results of LRC-DFT calculations must therefore be interpreted with caution, and probably for a range of ω values. This can be accomplished by requesting a functional that contains some short-range GGA exchange functional (ωPBE or μPBE, in the examples mentioned above), in combination with setting the *$rem* variable LRC_DFT = TRUE, which requests the addition of 100% Hartree-Fock exchange in the long-range. Basic job-control variables and an example can be found below. The value of the range-separation parameter is then controlled by the variable OMEGA, as shown in the examples below.

LRC_DFT

Controls the application of long-range-corrected DFT

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply long-range correction.

TRUE (or 1) Add 100% long-range Hartree-Fock exchange to the requested functional.

RECOMMENDATION:

The *$rem* variable OMEGA must also be specified, in order to set the range-separation parameter.

OMEGA

Sets the range-separation parameter, ω, also known as μ, in functionals based on Hirao's RSH scheme.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

COMBINE_K

Controls separate or combined builds for short-range and long-range K

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Build short-range and long-range K separately (twice as expensive as a global hybrid)

TRUE (or 1) Build short-range and long-range K together ($\approx$ as expensive as a global hybrid)

RECOMMENDATION:

Most pre-defined range-separated hybrid functionals in Q-CHEM use this feature by default. However, if a user-specified RSH is desired, it is necessary to manually turn this feature on.

HFK_SR_COEF

Sets the coefficient for short-range HF exchange

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $n/100000000$

RECOMMENDATION:

None

HFK_LR_COEF

Sets the coefficient for long-range HF exchange

TYPE:

INTEGER

DEFAULT:

100000000

OPTIONS:

n Corresponding to $n/100000000$

RECOMMENDATION:

None

Example 5.6 Application of LRC-μBOP to $(H_2O)_2^-$.

```
$comment
   The value of omega is 0.47 by default but can
   be overwritten by specifying OMEGA.
$end

$molecule
   -1 2
   O           1.347338    -0.017773    -0.071860
   H           1.824285     0.813088     0.117645
   H           1.805176    -0.695567     0.461913
   O          -1.523051    -0.002159    -0.090765
   H          -0.544777    -0.024370    -0.165445
   H          -1.682218     0.174228     0.849364
$end

$rem
   EXCHANGE      LRC-BOP
   BASIS         6-311(1+,2+)G*
   XC_GRID       2
   LRC_DFT       TRUE
   OMEGA         300       ! = 0.300 bohr**(-1)
$end
```

Rohrdanz *et al.*[225] published a thorough benchmark study of both ground- and excited-state properties using the LRC-ωPBEh functional, in which the "h" indicates a short-range hybrid (*i.e.*, the presence of some short-range HF exchange). Empirically-optimized parameters of $c_{x,\mathrm{SR}} = 0.2$ (see Eq. (5.13)) and $\omega = 0.2$ bohr^{-1} were obtained,[225] and these parameters are taken as the defaults for LRC-ωPBEh. Caution is warranted, however, especially in TDDFT calculations for large systems, as excitation energies for states that exhibit charge-transfer character can be rather sensitive to the precise value of ω.[145,225] In such cases (and maybe in general), the "tuning" procedure described in Section 5.6.4 is

recommended.

Example 5.7 Application of LRC-ωPBEh to the C_2H_4–C_2F_4 dimer at 5 Å separation.

```
$comment
    This example uses the "optimal" parameter set discussed above.
    It can also be run by setting METHOD = LRC-wPBEh.
$end

$molecule
   0 1
   C           0.670604    0.000000    0.000000
   C          -0.670604    0.000000    0.000000
   H           1.249222    0.929447    0.000000
   H           1.249222   -0.929447    0.000000
   H          -1.249222    0.929447    0.000000
   H          -1.249222   -0.929447    0.000000
   C           0.669726    0.000000    5.000000
   C          -0.669726    0.000000    5.000000
   F           1.401152    1.122634    5.000000
   F           1.401152   -1.122634    5.000000
   F          -1.401152   -1.122634    5.000000
   F          -1.401152    1.122634    5.000000
$end

$rem
   EXCHANGE       GEN
   BASIS          6-31+G*
   LRC_DFT        TRUE
   OMEGA          200       ! = 0.2 a.u.
   CIS_N_ROOTS    4
   CIS_TRIPLETS   FALSE
$end

$xc_functional
   C  PBE   1.00
   X  wPBE  0.80
   X  HF    0.20
$end
```

By adding 100% Hartree-Fock exchange to the asymptotic Coulomb operator, LRC functionals guarantee that an electron and hole experience an asymptotic interaction potential $1/r$. This is correct for a molecule in the gas phase, but to simulate a material one might desire an asymptotic behavior of $1/(\varepsilon r)$, where ε is the (static) dielectric constant of the material. In conjunction with "optimal tuning" of the range-separation parameter, as described in Section 5.6.4, such functionals have been shown to afford accurate fundamental gaps for organic photovoltaic materials,[139] and are naturally combined with polarizable continuum models (Section 11.2.3) that employ the same dielectric constant.[36] These have come to be called *screened* RSH (sRSH) functionals.[139] An XC function of this type can be expressed generically as[11]

$$E_{xc}^{\text{sRSH}} = c_{x,\text{SR}} E_{x,\text{SR}}^{\text{HF}} + \varepsilon^{-1} E_{x,\text{LR}}^{\text{HF}} + (\varepsilon^{-1} - c_{x,\text{SR}}) E_{x,\text{SR}}^{\text{DFT}} + (1 - \varepsilon^{-1}) E_{x,\text{LR}}^{\text{DFT}} + E_c^{\text{DFT}} \,, \tag{5.17}$$

which should be compared to Eq. (5.13) that provides the generic form for an RSH functional. Although the RSH formalism allows for an arbitrary coefficient $c_{x,\text{LR}}$ for the long-range Hartree-Fock exchange term, as in Eq. (5.13), this implies that the asymptotic electron–hole interaction has the form $c_{x,\text{LR}}/r$ rather than $1/r$.[77] As such, LRC functionals are a particular class of RSH functionals where $c_{x,\text{LR}} = 1$, ensuring proper asymptotic behavior in vacuum. Along the same lines, sRSH functionals set $c_{x,\text{LR}} = \varepsilon^{-1}$ to ensure proper asymptotic behavior in a dielectric material. Using Eq. (5.17), users may construct sRSH functionals by means of a *$xc_functional* input section.

5.6.4 Tuned RSH Functionals

Whereas the range-separation parameters for the functionals described in Section 5.6.2 are wholly empirical in nature and should not be adjusted, for the functionals described in Section 5.6.3 some adjustment *was* suggested, especially for TDDFT calculations and for any properties that require interpretation of orbital energies, such as HOMO/LUMO gaps. This adjustment can be performed in a non-empirical (albeit system-specific) way,[16] by "tuning" the value of ω in order to satisfy the Koopmans-like ionization energy criterion

$$-\epsilon_{\mathrm{HOMO}}(\omega) = \mathrm{IE}(\omega) \tag{5.18}$$

where $\mathrm{IE} = E(N-1) - E(N)$ is the ΔSCF value of the ionization energy for the N-electron system of interest. The condition $\epsilon_{\mathrm{HOMO}} = -\mathrm{IE}$ is a theorem in exact DFT,[191] but this condition is often badly violated by approximate functionals. When an RSH functional is used, both sides of Eq. (5.18) are ω-dependent and this parameter is adjusted until the condition in Eq. (5.18) is met, which requires a sequence of SCF calculations on both the neutral and ionized species, using different values of ω. The value that is obtained has come to be called the "optimally tuned" value of ω. Formally speaking, there is no guarantee that an approximate density functional can be made to satisfy Eq. (5.18) for any given molecule, thus the optimally-tuned value need not exist. In practice it is usually possible to find such a value, although it should be noted that the optimally-tuned value of ω depends on system size, and as a result this tuning procedure formally violates size-consistency.[125]

A few variations on the simple "IE tuning" criterion in Eq. (5.18) are possible. For proper description of charge-transfer states, Baer and co-workers[16] suggest finding the value of ω that (to the extent possible) satisfies Eq. (5.18) for both the neutral donor molecule and (separately) for the *anion* of the acceptor species. Along similar lines, in an effort to set both the HOMO and LUMO energy levels such that the fundamental gap (IE $-$ EA) is equal to the HOMO/LUMO gap, Kronik *et al.*[140] suggest minimizing the function

$$J(\omega) = \left[\epsilon_{\mathrm{HOMO}}(\omega) + \mathrm{IE}(\omega)\right]^2 + \left[\epsilon_{\mathrm{LUMO}}(\omega) + \mathrm{EA}(\omega)\right]^2 \tag{5.19}$$

with respect to ω, where $\mathrm{EA} = E(N+1) - E(N)$. Minimization of $J(\omega)$ represents an attempt to satisfy the IE theorem of Eq. (5.18) for both the N-electron molecule and its $(N+1)$-electron anion, assuming that the latter is bound. Published benchmarks suggest that these system-specific approaches afford the most accurate values of IEs and TDDFT excitation energies.[16,140,161,231]

A script that optimizes ω, called `OptOmegaIPEA.pl`, is located in the `$QC/bin/tools` directory. The script scans ω over the range 0.1–0.8 bohr^{-1}, corresponding to values of the *$rem* variable OMEGA in the range 100–800. See the script for the instructions how to modify the script to scan over a wider range. To execute the script, the user must create three inputs for an RSH single-point energy calculation, using the same geometry and basis set: one for a neutral molecule (`N.in`), one for its anion (`M.in`), and one for the molecule's cation (`P.in`). The user should then run the command

```
OptOmegaIPEA.pl >& optomega
```

This command both generates the input files (`N_*`, `P_*`, `M_*`) and also runs Q-CHEM on these input files, writing the optimization output into `optomega`. This script applies the IE condition to both the neutral molecule and its anion, minimizing $J(\omega)$ in Eq. (5.19). A similar script, `OptOmegaIP.pl`, uses Eq. (5.18) for the neutral molecule only.

Note:

1. If the system does not have positive EA, then the tuning should be done according to the IP condition only. The IP/EA script will yield an incorrect value of ω in such cases.

2. In order for the scripts to work, one must specify SCF_FINAL_PRINT = 1 in the inputs. The scripts look for specific regular expressions and will not work correctly without this keyword.

Although the tuning procedure was originally developed by Baer and co-workers using the BNL functional,[16,161,231] it can equally well be applied to any RSH functional, as for example LRC-ωPBE (see, Ref. 257). The aforementioned scripts will work with these other RSH functionals as well.

Unfortunately, optimally-tuned values of "ω_{IE}", obtained by satisfying the criterion in Eq. (5.18) exhibit a troubling dependence on system size,[70,81,137,184,257,261] leading to a loss of size-extensivity.[125] For example, the optimally-tuned value for the cluster anion $(H_2O)_6^-$ is very different than the one tuned for $(H_2O)_{70}^-$,[257] and the optimally-tuned value also varies with conjugation length for π-conjugated systems.[137] An alternative to the IE-based criterion in Eq. (5.18) is the global density-dependent (GDD) tuning procedure,[175] in which the optimal value

$$\omega_{\text{GDD}} = C\langle d_{\text{x}}^2\rangle^{-1/2} \tag{5.20}$$

is related to the average of the distance d_{x} between an electron in the outer regions of a molecule and the exchange hole in the region of localized valence orbitals. The quantity C is an empirical parameter for a given LRC functional, which was determined for LRC-ωPBE ($C = 0.90$) and LRC-ωPBEh ($C = 0.75$),[175] in order to ensure that $\omega_{\text{GDD}} \approx \omega_{\text{IE}}$ in small molecules. (A slightly different value, $C = 0.885$, was determined for Q-CHEM's implementation of LRC-ωPBE.[147]) For TD-DFT calculations in small molecules, the choice of ω_{GDD} versus ω_{IE} makes essentially no difference, while the former can be considerably cheaper (and more convenient) as compared to searching for the value ω_{IE}.[165]

Since LRC-ω_{GDD}PBE provides a better description of polarizabilities in polyacetylene as compared to ω_{IE},[104] it is anticipated that using ω_{GDD} in place of ω_{IE} may afford more accurate molecular properties, especially in conjugated systems where IE tuning sometimes encounters problems related to vanishing HOMO/LUMO gaps.[90,165] GDD tuning of an RSH functional is involving by setting the *$rem* variable OMEGA_GDD = TRUE. The electron density is needed to compute ω_{GDD} in Eq. (5.20) and this is accomplished using the converged SCF density computed using the RSH functional with the value of ω given by the *$rem* variable OMEGA. The value of ω_{GDD} therefore depends, in principle, upon the value of OMEGA, although in practice it is not very sensitive to this value and results obtained in this way are similar to fully self-consistent values of ω_{GDD}.[175]

OMEGA_GDD

Controls the application of ω_{GDD} tuning for long-range corrected DFT

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply GDD tuning.

TRUE (or 1) Use GDD tuning.

RECOMMENDATION:

The *$rem* variable OMEGA must also be specified, in order to set the initial range-separation parameter.

OMEGA_GDD_SCALING

Sets the empirical constant C in the GDD tuning procedure.

TYPE:

INTEGER

DEFAULT:

885

OPTIONS:

n Corresponding to $C = n/1000$.

RECOMMENDATION:

The quantity $C = 0.885$ ($n = 885$) was determined in Ref. 147 using LRC-ωPBE/hpTZVPP calculations.

Example 5.8 Sample input illustrating a calculation to determine the ω value for LRC-ωPBE based on the GDD tuning procedure.

```
$comment
   The initial omega value has to set.
$end

$molecule
   0 1
   O   -0.042500   0.091700   0.110000
   H    0.749000   0.556800   0.438700
   H   -0.825800   0.574700   0.432500
$end


$rem
   EXCHANGE          gen
   BASIS             aug-cc-pvdz
   LRC_DFT           true
   OMEGA             300
   OMEGA_GDD         true
$end

$xc_functional
   X   wPBE   1.0
   C    PBE   1.0
$end
```

However the tuning is accomplished, these tuned functionals are generally thought to work by reducing self-interaction error in approximate DFT. A convenient way to quantify—or at least depict—this error is by plotting the DFT energy as a function of the (fractional) number of electrons, N, because $E(N)$ should in principle consist of a sequence of line segments with abrupt changes in slope (the so-called derivative discontinuity[65,177]) at integer values of N, but in practice these $E(N)$ plots bow away from straight-line segments.[65] Examination of such plots has been suggested as a means to adjust the fraction of short-range exchange in an RSH functional,[15] while the ω parameter is set by tuning.

FRACTIONAL_ELECTRON

Add or subtract a fraction of an electron.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use an integer number of electrons.

n Add $n/1000$ electrons to the system.

RECOMMENDATION:

Use only if trying to generate $E(N)$ plots. If $n < 0$, a fraction of an electron is removed from the system.

Example 5.9 Example of a DFT job with a fractional number of electrons. Here, we make the $-1.x$ anion of fluoride by subtracting a fraction of an electron from the HOMO of F^{2-}.

```
$comment
   Subtracting a whole electron recovers the energy of F-.
   Adding electrons to the LUMO is possible as well.
$end

$molecule
   -2 2
   F
$end

$rem
   EXCHANGE              b3lyp
   BASIS                 6-31+G*
   FRACTIONAL_ELECTRON  -500   ! divide by 1000 to get the fraction, -0.5 here.
   GEN_SCFMAN            FALSE ! not yet available in new scf code
$end
```

5.7 DFT Methods for van der Waals Interactions

5.7.1 Introduction

This section describes five different procedures for obtaining a better description of dispersion (van der Waals) interactions in DFT calculations: non-local correlation functionals (Section 5.7.2), empirical atom–atom dispersion potentials ("DFT-D", Section 5.7.3), the Becke-Johnson exchange-dipole model (XDM, Section 5.7.4), the Tkatchenko-Scheffler van der Waals method (TS-vdW, Section 5.7.5), and finally the many-body dispersion method (MBD, Section 5.7.6). For a pedagogical introduction to all of these methods, discussing the commonalities between them, see Ref. 91.

5.7.2 Non-Local Correlation (NLC) Functionals

From the standpoint of the electron density, the vdW interaction is a non-local one: even for two non-overlapping, spherically-symmetric charge densities (two argon atoms, say), the presence of molecule B in the non-covalent A$\cdots$B complex induces ripples in the tail of A's charge distribution, which are the hallmarks of non-covalent interactions.[69] (This is the fundamental idea behind the non-covalent interaction plots described in Section 10.5.6; the vdW interaction manifests as large density gradients in regions of space where the density itself is small.) Semi-local GGAs that depend only on the density and its gradient cannot describe this long-range, correlation-induced interaction, and meta-GGAs at best describe it at middle-range via the Laplacian of the density and/or the kinetic energy density. A proper description of *long-range* electron correlation requires a non-local functional, *i.e.*, an exchange-correlation potential having the form

$$v_{\mathrm{c}}^{\mathrm{nl}}(\mathbf{r}) = \int f(\mathbf{r},\mathbf{r}')\,d\mathbf{r}' \ . \tag{5.21}$$

In this way, a perturbation at a point $\mathbf{r}'$ (due to B, say) then induces an exchange-correlation potential at a (possibly far-removed) point $\mathbf{r}$ (on A).

Q-CHEM includes four such functionals that can describe dispersion interactions:

- vdW-DF-04, developed by Langreth, Lundqvist, and coworkers,[73,74] implemented as described in Ref. 272.

- vdW-DF-10 (also known as vdW-DF2), which is a re-parameterization of vdW-DF-04.[153]
- VV09, developed[269] and implemented[270] by Vydrov and Van Voorhis.
- VV10 by Vydrov and Van Voorhis.[271]
- rVV10 by Sabatini and coworkers.[230]

Each of these functionals is implemented in a self-consistent manner, and analytic gradients with respect to nuclear displacements are available.[270–272] The non-local correlation is governed by the *$rem* variable NL_CORRELATION, which can be set to one of the four values: vdW-DF-04, vdW-DF-10, VV09, or VV10. The vdW-DF-04, vdW-DF-10, and VV09 functionals are used in combination with LSDA correlation, which must be specified explicitly. For instance, vdW-DF-10 is invoked by the following keyword combination:

```
$rem
   CORRELATION         PW92
   NL_CORRELATION      vdW-DF-10
   ...
$end
```

VV10 is used in combination with PBE correlation, which must be added explicitly. In addition, the values of two parameters, C and b (see Ref. 272), must be specified for VV10. These parameters are controlled by the *$rem* variables NL_VV_C and NL_VV_B, respectively. For instance, to invoke VV10 with $C = 0.0093$ and $b = 5.9$, the following input is used:

```
$rem
   CORRELATION         PBE
   NL_CORRELATION      VV10
   NL_VV_C             93
   NL_VV_B             590
   ...
$end
```

The variable NL_VV_C may also be specified for VV09, where it has the same meaning. By default, $C = 0.0089$ is used in VV09 (*i.e.* NL_VV_C is set to 89). However, in VV10 neither C nor b are assigned a default value and must always be provided in the input.

Unlike local (LSDA) and semi-local (GGA and meta-GGA) functionals, for non-local functionals evaluation of the correlation energy requires a double integral over the spatial variables, as compared to the single integral [Eq. (5.8)] required for semi-local functionals:

$$E_{\mathrm{c}}^{\mathrm{nl}} = \int v_{\mathrm{c}}^{\mathrm{nl}}(\mathbf{r})\, d\mathbf{r} = \int f(\mathbf{r},\mathbf{r}')\, \rho(\mathbf{r})\, d\mathbf{r}\, d\mathbf{r}'. \tag{5.22}$$

In practice, this double integration is performed numerically on a quadrature grid.[270–272] By default, the SG-1 quadrature (described in Section 5.5.3 below) is used to evaluate $E_{\mathrm{c}}^{\mathrm{nl}}$, but a different grid can be requested via the *$rem* variable NL_GRID. The non-local energy is rather insensitive to the fineness of the grid such that SG-1 or even SG-0 grids can be used in most cases, but a finer grid may be required to integrate other components of the functional. This is controlled by the XC_GRID variable discussed in Section 5.5.3.

The two functionals originally developed by Vydrov and Van Voorhis can be requested by specifying METHOD = VV10 or METHOD LC-VV10. In addition, the combinatorially-optimized functionals of Mardirossian and Head-Gordon (ωB97X-V, B97M-V, and ωB97M-V) make use of non-local correlation and can be invoked by setting METHOD to

wB97X-V, B97M-V, or wB97M-V. Now, the VV10 codes have been rewrited and the feature of NMR chemical shielding and analytical second derivative are added. If you want to use the old codes, please set USE_LIBNLQ to false.

Example 5.10 Geometry optimization of the methane dimer using VV10 with rPW86 exchange.

```
$molecule
   0 1
   C   0.000000  -0.000140   1.859161
   H  -0.888551   0.513060   1.494685
   H   0.888551   0.513060   1.494685
   H   0.000000  -1.026339   1.494868
   H   0.000000   0.000089   2.948284
   C   0.000000   0.000140  -1.859161
   H   0.000000  -0.000089  -2.948284
   H  -0.888551  -0.513060  -1.494685
   H   0.888551  -0.513060  -1.494685
   H   0.000000   1.026339  -1.494868
$end

$rem
   JOBTYPE            opt
   BASIS              aug-cc-pVTZ
   EXCHANGE           rPW86
   CORRELATION        PBE
   XC_GRID            2
   NL_CORRELATION     VV10
   NL_GRID            1
   NL_VV_C            93
   NL_VV_B            590
$end
```

In the above example, the SG-2 grid is used to evaluate the rPW86 exchange and PBE correlation, but a coarser SG-1 grid is used for the non-local part of VV10. Furthermore, the above example is identical to specifying METHOD = VV10.

NL_CORRELATION

Specifies a non-local correlation functional that includes non-empirical dispersion.

TYPE:

STRING

DEFAULT:

None No non-local correlation.

OPTIONS:

None	No non-local correlation
vdW-DF-04	the non-local part of vdW-DF-04
vdW-DF-10	the non-local part of vdW-DF-10 (also known as vdW-DF2)
VV09	the non-local part of VV09
VV10	the non-local part of VV10

RECOMMENDATION:

Do not forget to add the LSDA correlation (PW92 is recommended) when using vdW-DF-04, vdW-DF-10, or VV09. VV10 should be used with PBE correlation. Choose exchange functionals carefully: HF, rPW86, revPBE, and some of the LRC exchange functionals are among the recommended choices.

NL_VV_C

Sets the parameter C in VV09 and VV10. This parameter is fitted to asymptotic van der Waals C_6 coefficients.

TYPE:

INTEGER

DEFAULT:

89 for VV09

No default for VV10

OPTIONS:

n Corresponding to $C = n/10000$

RECOMMENDATION:

$C = 0.0093$ is recommended when a semi-local exchange functional is used. $C = 0.0089$ is recommended when a long-range corrected (LRC) hybrid functional is used. For further details see Ref. 271.

NL_VV_B

Sets the parameter b in VV10. This parameter controls the short range behavior of the non-local correlation energy.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $b = n/100$

RECOMMENDATION:

The optimal value depends strongly on the exchange functional used. $b = 5.9$ is recommended for rPW86. For further details see Ref. 271.

USE_RVV10

Used to turn on the rVV10 NLC functional

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use VV10 NLC (the default for NL_CORRELATION)

TRUE Use rVV10 NLC

RECOMMENDATION:

Set to TRUE if the rVV10 NLC is desired.

USE_LIBNLQ

Turn on the use of LIBNLQ for calculating nonlocal correlation funcitonal.

TYPE:

LOGICAL

DEFAULT:

True For VV10.

FALSE For all other nonlocal funcitonals.

OPTIONS:

False

True

RECOMMENDATION:

Use the default

5.7.3 Empirical Dispersion Corrections: DFT-D

A major development in DFT during the mid-2000s was the recognition that, first of all, semi-local density functionals do not properly capture dispersion (van der Waals) interactions, a problem that has been addressed only much more recently by the non-local correlation functionals discussed in Section 5.7.2; and second, that a cheap and simple solution to this problem is to incorporate empirical potentials of the form $-C_6/R^6$, where the C_6 coefficients are pairwise atomic parameters. This approach, which is an alternative to the use of a non-local correlation functional, is known as *dispersion-corrected DFT* (DFT-D).[96,100]

There are currently three unique DFT-D methods in Q-CHEM. These are requested via the *$rem* variable DFT_D and are discussed below.

DFT_D

Controls the empirical dispersion correction to be added to a DFT calculation.

TYPE:

LOGICAL

DEFAULT:

None

OPTIONS:

FALSE	(or 0) Do not apply the DFT-D2, DFT-CHG, or DFT-D3 scheme
EMPIRICAL_GRIMME	DFT-D2 dispersion correction from Grimme[95]
EMPIRICAL_CHG	DFT-CHG dispersion correction from Chai and Head-Gordon[57]
EMPIRICAL_GRIMME3	DFT-D3(0) dispersion correction from Grimme (deprecated as of Q-CHEM 5.0)
D3_ZERO	DFT-D3(0) dispersion correction from Grimme *et al.*[97]
D3S_ZERO	DFT-D3S(0) dispersion correction from Tkachenko and Head-Gordon[253]
D3_BJ	DFT-D3(BJ) dispersion correction from Grimme *et al.*[98]
D3S_BJ	DFT-D3S(BJ) dispersion correction from Tkachenko and Head-Gordon[253]
D3_CSO	DFT-D3(CSO) dispersion correction from Schröder *et al.*[232]
D3_ZEROM	DFT-D3M(0) dispersion correction from Smith *et al.*[237]
D3_BJM	DFT-D3M(BJ) dispersion correction from Smith *et al.*[237]
D3_OP	DFT-D3(op) dispersion correction from Witte *et al.*[283]
D3	Automatically select the "best" available D3 dispersion correction
D4	DFT-D4 dispersion correction from Caldeweyher *et al.*[46–48]

RECOMMENDATION:

Use D4 if the specified functional is avialable. Currently, only a subset of functionals in DFT-D4 is supported. It includes B3LYP, B97, B1LYP, PBE0, PW6B95, M06L, M06, WB97, WB97X, CAMB3LYP, PBE02, PBE0DH, MPW1K, MPWB1K, B1B95, B1PW91, B2GPPLYP, B2PLYP, B3P86, B3PW91, O3LYP, REVPBE, REVPBE0, REVTPSS, REVTPSSH, SCAN, TPSS0, TPSSH, X3LYP, TPSS, BP86, BLYP, BPBE, MPW1PW91, MPW1LYP, PBE, RPBE, and PW91.

The oldest of these approaches is DFT-D2,[95] in which the empirical dispersion potential has the aforementioned form, namely, pairwise atomic $-C/R^6$ terms:

$$E_{\text{disp}}^{\text{D2}} = -s_6 \sum_{A}^{\text{atoms}} \sum_{B<A}^{\text{atoms}} \left(\frac{C_{6,AB}}{R_{AB}^6} \right) f_{\text{damp}}^{\text{D2}}(R_{AB}) \, . \tag{5.23}$$

This function is damped at short range, where R_{AB}^{-6} diverges, via

$$f_{\text{damp}}^{\text{D2}}(R_{AB}) = \left[1 + e^{-d(R_{AB}/R_{0,AB}-1)} \right]^{-1} \tag{5.24}$$

which also helps to avoid double-counting of electron correlation effects, since short- to medium-range correlation is included via the density functional. (The quantity $R_{0,AB}$ is the sum of the van der Waals radii for atoms A and B, and d is an additional parameter.) The primary parameters in Eq. (5.23) are atomic coefficients $C_{6,A}$, from which the pairwise parameters in Eq. (5.23) are obtained as geometric means, as is common in classical force fields:

$$C_{6,AB} = \left(C_{6,A} C_{6,B} \right)^{1/2} \tag{5.25}$$

The total energy in DFT-D2 is of course $E_{\text{DFT-D2}} = E_{\text{KS-DFT}} + E_{\text{disp}}^{\text{D2}}$.

DFT-D2 is available in Q-CHEM including analytic gradients and frequencies, thanks to the efforts of David Sherrill's group. The D2 correction can be used with any density functional that is available in Q-CHEM, although its use with

the non-local correlation functionals discussed in Section 5.7.2 seems inconsistent and is not recommended. The global parameter s_6 in Eq. (5.23) was optimized by Grimme for four different functionals,[95] and Q-CHEM uses these as the default values: $s_6 = 0.75$ for PBE, $s_6 = 1.2$ for BLYP, $s_6 = 1.05$ for BP86, and $s_6 = 1.05$ for B3LYP. For all other functionals, $s_6 = 1$ by default. The D2 parameters, including the $C_{6,A}$ coefficients and the atomic van der Waals radii, can be modified using a *$empirical_dispersion* input section. For example:

```
$empirical_dispersion
   S6 1.1
   D 10.0
   C6 Ar 4.60 Ne 0.60
   VDW_RADII Ar 1.60 Ne 1.20
$end
```

Values not specified explicitly default to the values optimized by Grimme.

Note:

1. DFT-D2 is only defined for elements up to Xe.

2. B97-D is an exchange-correlation functional that automatically employs the DFT-D2 dispersion correction when used via METHOD = B97-D.

An alternative to Grimme's DFT-D2 is the empirical dispersion correction of Chai and Head-Gordon,[57] which uses the same form as Eq. (5.23) but with a slightly different damping function:

$$f_{\text{damp}}^{\text{CHG}}(R_{AB}) = \left[1 + a(R_{AB}/R_{0,AB})^{-12}\right]^{-1} \tag{5.26}$$

This version is activated by setting DFT_D = EMPIRICAL_CHG, and the damping parameter a is controlled by the keyword DFT_D_A.

DFT_D_A
Controls the strength of dispersion corrections in the Chai–Head-Gordon DFT-D scheme, Eq. (5.26).

TYPE:
INTEGER

DEFAULT:
600

OPTIONS:
n Corresponding to $a = n/100$.

RECOMMENDATION:
Use the default.

Note:

1. DFT-CHG is only defined for elements up to Xe.

2. The ωB97X-D and ωM05-D functionals automatically employ the DFT-CHG dispersion correction when used via METHOD = wB97X-D or wM05-D.

Grimme's DFT-D3 method[97] constitutes an improvement on his D2 approach, and is also available along with analytic first and second derivatives, for any density functional that is available in Q-CHEM. The D3 correction includes a

potential akin to that in D2 but including atomic C_8 terms as well:

$$E_{\text{D3,2-body}} = -\sum_{A}^{\text{atoms}} \sum_{B<A}^{\text{atoms}} \left[s_6 \left(\frac{C_{6,AB}}{R_{AB}^6} \right) f_{\text{damp},6}(R_{AB}) + s_8 \left(\frac{C_{8,AB}}{R_{AB}^8} \right) f_{\text{damp},8}(R_{AB}) \right] . \tag{5.27}$$

The total D3 dispersion correction consists of this plus a three-body term of the Axilrod-Teller-Muto (ATM) triple-dipole variety, so that the total D3 energy is $E_{\text{DFT-D3}} = E_{\text{KS-DFT}} + E_{\text{D3,2-body}} + E_{\text{ATM,3-body}}$

Several versions of DFT-D3 are available as of Q-CHEM 5.0, which differ in the choice of the two damping functions. Grimme's formulation,[97] which is now known as the "zero-damping" version [DFT-D3(0)], uses damping functions of the form

$$f_{\text{damp},n}^{\text{D3(0)}}(R_{AB}) = \left[1 + 6 \left(\frac{R_{AB}}{s_{r,n} R_{0,AB}} \right)^{-\beta_n} \right]^{-1} \tag{5.28}$$

for $n = 6$ or 8, $\beta_6 = 14$, and $\beta_8 = 16$.[97,157] The parameters $R_{0,AB}$ come from atomic van der Waals radii, $s_{r,6}$ is a functional-dependent parameter, and $s_{r,8} = 1$. Typically s_6 is set to unity and s_8 is optimized for the functional in question.

The more recent Becke–Johnson-damping version of DFT-D3,[98] DFT-D3(BJ), is designed to be finite (but non-zero) as $R_{AB} \to 0$. The damping functions used in DFT-D3(BJ) are

$$f_{\text{damp},n}^{\text{D3(BJ)}}(R_{AB}) = \frac{R_{AB}^n}{R_{AB}^n + (\alpha_1 R_{0,AB} + \alpha_2)^n} \tag{5.29}$$

where α_1 and α_2 are adjustable parameters fit for each density functional. As in DFT-D3(0), s_6 is generally fixed to unity and s_8 is optimized for each functional. DFT-D3(BJ) generally outperforms the original DFT-D3(0) version.[98]

In addition to these original DFT-D3(BJ) and DFT-D3(0) variants, modified DFT-D3S versions have also been implemented.[253] The D3S scheme redefines the C_6^{AB} coefficients by introducing atom-pair-dependent k_3. The C_6^{AB} coefficients in original DFT-D3 are introduced through the following expression:

$$C_6^{AB}(CN^A, CN^B) = \frac{\sum_{i=1}^{N_A} \sum_{j=1}^{N_B} C_{6,\text{ref}}^{AB}(CN_i^A, CN_j^B) L_{ij}}{\sum_{i=1}^{N_A} \sum_{j=1}^{N_B} L_{ij}} \tag{6}$$

Here, N_A and N_B are the number of reference molecules for atoms A and B, $C_{6,\text{ref}}^{AB}(CN_i^A, CN_j^B)$ are precomputed dispersion coefficients for the reference molecules, and CN_i^A, CN_j^B are the coordination numbers of the atoms A and B in their respective reference molecules. The term L_{ij} modulates the weight based on how different the coordination numbers are from their reference points and is defined in the original D3 method as:

$$L_{ij}^{D3} = e^{-k_3\left[(CN^A - CN_i^A)^2 + (CN^B - CN_j^B)^2\right]} \tag{7}$$

where CN^A and CN^B are the coordination numbers of atoms A and B in the system of interest, and k_3 is fixed at a value of 4. However, this fixed value can lead to unphysical local minima on the potential energy surface, especially when the coordination numbers of atoms vary. To address this issue, the D3S scheme introduces atom-pair-dependent k_3^{AB} parameters, resulting in an expression:

$$L_{ij}^{D3S} = e^{-k_3^{AB}\left[(CN^A - CN_i^A)^2 + (CN^B - CN_j^B)^2\right]} \tag{8}$$

Here, k_3^{AB} is no longer a universal constant but is instead optimized individually for each atomic pair to reduce the sharpness of the $C_6^{AB}(CN^A, CN^B)$ function. This optimization mitigates the formation of artifact local minima on the PES that can arise with the original D3 formulation, even for purely organic molecules. The introduction of these

pairwise k_3^{AB} parameters in the D3S scheme enhances the accuracy of PES particularly for processes where the atom coordination numbers change significantly.[253]

The C_6-only (CSO) approach of Schröder *et al.*[232] discards the C_8 term in Eq. (5.27) and uses a damping function with one parameter, α_1:

$$f_{\text{damp},6}^{\text{D3(CSO)}}(R_{AB}) = \frac{C_{AB}^6}{R_{AB}^6 + (2.5\text{Å})^6}\left(s_6 + \frac{\alpha_1}{1+\exp[R_{AB} - (2.5\text{Å})R_{0,AB}]}\right) . \tag{5.30}$$

The DFT-D3(BJ) approach was re-parameterized by Smith *et al.*[237] to yield the "modified" DFT-D3(BJ) approach, DFT-D3M(BJ), whose parameterization relied heavily on non-equilibrium geometries. The same authors also introduces a modification DFT-D3M(0) of the original zero-damping correction, which introduces one additional parameter (α_1) as compared to DFT-D3(0):

$$f_{\text{damp},n}^{\text{D3M(0)}}(R_{AB}) = \left[1 + 6\left(\frac{R_{AB}}{s_{r,n}R_{0,AB}} + \alpha_1 R_{0,AB}\right)^{-\beta_n}\right]^{-1} . \tag{5.31}$$

Finally, optimized power approach of Witte *et al.*[283] treats the exponent, β_6, as an optimizable parameter, given by

$$f_{\text{damp},n}^{\text{D3(op)}}(R_{AB}) = \frac{R_{AB}^{\beta_n}}{R_{AB}^{\beta_n} + (\alpha_1 R_{0,AB} + \alpha_2)^{\beta_n}} . \tag{5.32}$$

Note that $\beta_8 = \beta_6 + 2$.

To summarize this bewildering array of D3 damping functions:

- **DFT-D3(0)** is requested by setting DFT_D = D3_ZERO. The model depends on four scaling parameters (s_6, $s_{r,6}$, s_8, and $s_{r,8}$), as defined in Eq. (5.28).
- **DFT-D3(BJ)** is requested by setting DFT_D = D3_BJ. The model depends on four scaling parameters (s_6, s_8, α_1, and α_2), as defined in Eq. (5.29).
- **DFT-D3(CSO)** is requested by setting DFT_D = D3_CSO. The model depends on two scaling parameters (s_6 and α_1), as defined in Eq. (5.30).
- **DFT-D3M(0)** is requested by setting DFT_D = D3_ZEROM. The model depends on five scaling parameters (s_6, s_8, $s_{r,6}$, $s_{r,8}$, and α_1), as defined in Eq. (5.31).
- **DFT-D3M(BJ)** is requested by setting DFT_D = D3_BJM. The model depends on four scaling parameters (s_6, s_8, α_1, and α_2), as defined in Eq. (5.29).
- **DFT-D3(op)** is requested by setting DFT_D = D3_OP. The model depends on four scaling parameters (s_6, s_8, α_1, α_2, and β_6), as defined in Eq. (5.29).

The scaling parameters in these damping functions can be modified using the *$rem* variables described below. Alternatively, one may simply set DFT_D = D3, and a D3 dispersion correction will be selected automatically, if one is available for the selected functional.

Note:

1. DFT-D3(0) is defined for elements up to Pu ($Z = 94$).

2. The B97-D3(0), ωB97X-D3, ωM06-D3 functionals automatically employ the DFT-D3(0) dispersion correction when invoked by setting METHOD equal to B97-D3, wB97X-D3, or wM06-D3.

3. The old way of invoking DFT-D3, namely through the use of EMPIRICAL_GRIMME3, is still supported, though its use is discouraged since D3_ZERO accomplishes the same thing but with additional precision for the relevant parameters.

4. When DFT_D = D3, parameters may not be overwritten, with the exception of DFT_D3_3BODY; this is intended as a user-friendly option. This is also the case when EMPIRICAL_GRIMME3 is employed for a functional parameterized in Q-CHEM. When any of D3_ZERO, D3_BJ, etc. are chosen, Q-CHEM will automatically populate the parameters with their default values, if they available for the desired functional, but these defaults can still be overwritten by the user.

DFT_D3_S6

The linear parameter s_6 in eq. (5.27). Used in all forms of DFT-D3.

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS6

The nonlinear parameter $s_{r,6}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,6} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_S8

The linear parameter s_8 in Eq. (5.27). Used in DFT-D3(0), DFT-D3(BJ), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_8 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS8

The nonlinear parameter $s_{r,8}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,8} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A1

The nonlinear parameter α_1 in Eqs. (5.29), (5.30), (5.31), and (5.32). Used in DFT-D3(BJ), DFT-D3(CSO), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_1 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A2

The nonlinear parameter α_2 in Eqs. (5.29) and (5.32). Used in DFT-D3(BJ), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_2 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_POWER

The nonlinear parameter β_6 in Eq. (5.32). Used in DFT-D3(op). Must be greater than or equal to 6 to avoid divergence.

TYPE:

INTEGER

DEFAULT:

600000

OPTIONS:

n Corresponding to $\beta_6 = n/100000$.

RECOMMENDATION:

NONE

The three-body interaction term, $E^{(3)}$,[97] must be explicitly turned on, if desired.

DFT_D3_3BODY

Controls whether the three-body interaction in Grimme's DFT-D3 method should be applied (see Eq. (14) in Ref. 97).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply the three-body interaction term

TRUE Apply the three-body interaction term

RECOMMENDATION:

NONE

More recently, Grimme published an extended D3 model, D4.[46–48] The main feature of D4 is that the coefficients are generated through Casimir-Polder integration of the dynamic atomic polarizabilities $\alpha(i\omega)$ where electronic density information is employed via atomic partial charges. Benchmark results show that the proposed D4 model yields significantly lower error bars. The DFT-D4 dispersion energy similar to D3 model is given by

$$E_{\text{D4,2-body}} = -\sum_{A}^{\text{atoms}} \sum_{B<A}^{\text{atoms}} \left[s_6 \left(\frac{C_{6,AB}}{R_{AB}^6} \right) f_{\text{BJ damp,6}}(R_{AB}) + s_8 \left(\frac{C_{8,AB}}{R_{AB}^8} \right) f_{\text{BJ damp,8}}(R_{AB}) \right] . \tag{5.33}$$

The Becke-Johnson damping is utilized as default. The coordination number dependent C_6^{AB} coefficients are obtained on-the-fly via Casimir-Polder integration

$$C_6^{\text{AB}} = \sum_{A,\text{ref}=1}^{\text{N}^{A,\text{ref}}} \sum_{B,\text{ref}=1}^{\text{N}^{B,\text{ref}}} \frac{3}{\pi} \int_0^\infty d\omega \alpha^{A,\text{ref}}(i\omega, z^A) \times W_A^{A,\text{ref}} \alpha^{B,\text{ref}}(i\omega, z^B) W_B^{B,\text{ref}} , \tag{5.34}$$

where

$$\alpha^{A,\text{ref}}(i\omega, z^A) = \frac{1}{m} \left[\alpha^{A_m H_n}(i\omega) - \frac{n}{2} \alpha^{H_2}(i\omega) \times \zeta(z^{H_A,\text{ref}}, z^{H_2}) \right] \zeta(z^A, z^{A,\text{ref}}) \tag{5.35}$$

and

$$\zeta(z^A, z^{A,\text{ref}}) = ba^{[1.47\exp(z^A/z^{A,\text{ref}})\log_{10}(z^{A,\text{ref}}/z^A)]} . \tag{5.36}$$

$\alpha^{A_m H_n}$ denotes the reference polarizailities which represents the molecular polarizability of symmetric hydride systems $A_m H_n$. $W_{A/B}^{A,\mathrm{ref}/B,\mathrm{ref}}$ are weighting factors determining the contributions of all element specific reference systems $N^{A,\mathrm{ref}/B,\mathrm{ref}}$. z^{H_A} describes the effective charge of hydrogen connectd to atom A in the reference system $A_m H_n$. The effective charge z^A is computed self-consistently via Mulliken charge q^A,

$$z^A = Z^A + q^A. \tag{5.37}$$

The coefficients a and b are parametrized to match cationic static polarzizabilies and TD-DFT derived molecular dispersion coefficients, respectively.

DFT_D4_S6

The linear parameter s_6. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_6 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S8

The linear parameter s_8. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_8 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S10

The linear parameter s_{10}. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_{10} = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_A1

The nonlinear parameter α_1. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_1 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_A2

The nonlinear parameter α_2. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_2 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S9

The linear parameter s_9. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_9 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_WF

Weighting factor for Gaussian weighting.

TYPE:

INTEGER

DEFAULT:

600000000

OPTIONS:

n Corresponding to $wf = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_GA

Charge scaling

TYPE:

INTEGER

DEFAULT:

300000000

OPTIONS:

n Corresponding to $ga = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_GC

Charge scaling

TYPE:

INTEGER

DEFAULT:

200000000

OPTIONS:

n Corresponding to $gc = n/100000000$.

RECOMMENDATION:

Use default

Example 5.11 Applications of B3LYP-D3(0) with custom parameters to a methane dimer.

```
$comment
   Geometry optimization, followed by single-point calculations using a larger
   basis set.
$end

$molecule
   0 1
   C       0.000000    -0.000323     1.755803
   H      -0.887097     0.510784     1.390695
   H       0.887097     0.510784     1.390695
   H       0.000000    -1.024959     1.393014
   H       0.000000     0.001084     2.842908
   C       0.000000     0.000323    -1.755803
   H       0.000000    -0.001084    -2.842908
   H      -0.887097    -0.510784    -1.390695
   H       0.887097    -0.510784    -1.390695
   H       0.000000     1.024959    -1.393014
$end

$rem
   JOBTYPE          opt
   EXCHANGE         B3LYP
   BASIS            6-31G*
   DFT_D            D3_ZERO
   DFT_D3_S6        100000
   DFT_D3_RS6       126100
   DFT_D3_S8        170300
   DFT_D3_3BODY     FALSE
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE          sp
   EXCHANGE         B3LYP
   BASIS            6-311++G**
   DFT_D            D3_ZERO
   DFT_D3_S6        100000
   DFT_D3_RS6       126100
   DFT_D3_S8        170300
   DFT_D3_3BODY     FALSE
$end
```

5.7.4 Exchange-Dipole Model (XDM)

5.7.4.1 Theory

Becke and Johnson have proposed an *exchange dipole model* (XDM) of dispersion.[28–31,123,124] The attractive dispersion energy arises in this model via the interaction between the instantaneous dipole moment of the exchange hole in one molecule, and the induced dipole moment in another. This is a conceptually simple yet powerful approach that has been shown to yield very accurate dispersion coefficients without fitting parameters. This allows the calculation of both

intermolecular and intramolecular dispersion interactions within a single DFT framework. The implementation and validation of this method in the Q-CHEM code is described in Ref. 136, with an updated set of damping parameters[183] added in Q-CHEM v. 6.1.1.[91]

The dipole moment of the exchange hole function $h_\sigma(\mathbf{r},\mathbf{r}')$ is given at point $\mathbf{r}$ by

$$d_\sigma(\mathbf{r}) = -\mathbf{r} - \int h_\sigma(\mathbf{r},\mathbf{r}')\ \mathbf{r}'\ d\mathbf{r}'\ , \tag{5.38}$$

where $\sigma = \alpha, \beta$. This depends on a model of the exchange hole, and the implementation in Q-CHEM uses the Becke-Roussel (BR) model.[32] In most implementations the BR model, h_σ is not available in analytic form and its value must be numerically at each grid point. Q-CHEM developed for the first time an analytical expression for this function,[136] based on non-linear interpolation and spline techniques, which greatly improves efficiency as well as the numerical stability.

Two different damping functions have been used with XDM. One of them relies only the intermolecular C_6 coefficient, and its implementation in Q-CHEM is denoted as "XDM6". In this version, the dispersion energy is[28]

$$E_{\rm vdW}^{\rm XDM6} = \sum_A^{\rm atoms}\sum_{B<A}^{\rm atoms} E_{{\rm vdW},AB} = -\sum_A^{\rm atoms}\sum_{B<A}^{\rm atoms} \frac{C_{6,AB}}{R_{AB}^6 + \kappa\, C_{6,AB}/E_{AB}^{\rm corr}}\ . \tag{5.39}$$

The term $\kappa\, C_{6,AB}/E_{AB}^{\rm corr}$ in the denominator prevents short-range divergence. The quantity $E_{AB}^{\rm corr}$ is the sum of the absolute values of the correlation energies of the free atoms A and B, whereas κ is a damping parameter that is universal in the sense that it is independent of the choice of functional.[28] The dispersion coefficients $C_{6,AB}$ is computed according to[28]

$$C_{6,ij} = \frac{\langle d_{\rm X}^2\rangle_A \langle d_{\rm X}^2\rangle_B\ \alpha_A\,\alpha_B}{\langle d_{\rm X}^2\rangle_A\,\alpha_B + \langle d_{\rm X}^2\rangle_B\,\alpha_A} \tag{5.40}$$

where $\langle d_{\rm X}^2\rangle_A$ is the square of the exchange-hole dipole moment of atom A, whose effective polarizability (in the molecule) is α_A.

The XDM6 scheme can be further generalized to include higher-order dispersion coefficients,[30,124] which leads to the "XDM10" model in Q-CHEM:

$$E_{\rm vdW}^{\rm XDM10} = -\sum_A^{\rm atoms}\sum_{B<A}^{\rm atoms}\left(\frac{C_{6,AB}}{R_{{\rm vdW},AB}^6 + R_{AB}^6} + \frac{C_{8,AB}}{R_{{\rm vdW},AB}^8 + R_{AB}^8} + \frac{C_{10,AB}}{R_{{\rm vdW},AB}^{10} + R_{AB}^{10}}\right)\ . \tag{5.41}$$

The higher-order dispersion coefficients are computed using higher-order multipole moments of the exchange hole.[124] The quantity $R_{{\rm vdW},AB}$ prevents short-range divergence and is nominally equal to the sum of effective atomic radii for atoms A and B. In practice it is determined from the formula[124]

$$R_{{\rm vdW},AB} = a_1 R_{{\rm crit},AB} + a_2 \tag{5.42}$$

with a critical distance

$$R_{{\rm crit},AB} = \frac{1}{3}\left[\left(\frac{C_{8,AB}}{C_{6,AB}}\right)^{1/2} + \left(\frac{C_{10,AB}}{C_{6,AB}}\right)^{1/4} + \left(\frac{C_{10,AB}}{C_{8,AB}}\right)^{1/2}\right] \tag{5.43}$$

and two parameters, a_1 and a_2. These were originally fitted using Hartree-Fock exchange[124] but were later optimized for several different exchange-correlation functionals: PW86PBE, PBE, BLYP, B97-1, B3LYP, B3P86, B3PW91, PBE0, BHHLYP, LRC-ωPBE, and CAM-B3LYP.[183] Parameters for this set of functionals are implemented in Q-CHEM.[91]

Note: For functionals other than the ones specified above, Hartree-Fock values of the parameters a_1 and a_2 are used, although this may not be optimal.

5.7.4.2 Job Control

As in DFT-D, the van der Waals energy is added as a post-SCF correction. Analytic gradients and Hessians are available for both XDM6 and XDM10. The dispersion correction is requested by setting XDM = TRUE in the *$rem* section. All other job control variables belong in the *$xdm* input section, as described below. Of these, only **NAtoms_Mol1** is required, as this determined which part of the *$molecule* section corresponds to monomer A, with the rest corresponding to monomer B. It is required that all atoms for A are grouped together in the *$molecule* section.

XDM

Controls whether to add XDM dispersion to an SCF calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not apply the XDM scheme.

1 Add XDM dispersion as a correction to the SCF energy (and gradient, if appropriate).

2 Add dispersion as a DFT functional and do full SCF.

RECOMMENDATION:

The second (self-consistent) option is only available for XDM6.

NAtomsMol1

Sets the size of monomer A.

INPUT SECTION: *$xdm*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Monomer A consists of the first N atoms in the *$molecule* section.

RECOMMENDATION:

This is the only required keyword in the *$xdm* section. The two monomers must be grouped together in the *$molecule* section; note that Q-CHEM's fragment-based input format does not work with XDM.

Method

Controls which XDM method to use.

INPUT SECTION: *$xdm*

TYPE:

STRING

DEFAULT:

XDM10

OPTIONS:

XDM6 Use the C_6-only model of Eq. (5.39).

XDM10 Use the model that includes C_8 and C_{10}, Eq. (5.41).

RECOMMENDATION:

The XDM10 is generally more accurate and is now the preferred approach.

Damp_C6_K

Set the damping parameter κ in Eq. (5.39).

INPUT SECTION: *$xdm*

TYPE:

FLOAT

DEFAULT:

0.8

OPTIONS:

x Set $\kappa = x$.

RECOMMENDATION:

The default value is taken from Ref. Becke:2005a and is intended to be universal.

Damp_A1

Sets the parameter a_1 in Eq. (5.42).

INPUT SECTION: *$xdm*

TYPE:

FLOAT

DEFAULT:

Various

OPTIONS:

x Set $a_1 = x$.

RECOMMENDATION:

Functional-specific defaults are available for HF, PW86PBE, PBE, BLYP, B97-1, B3LYP, B3P86, B3PW91, PBE0, BHHLYP, LRC-ωPBE, and CAM-B3LYP, taken from Ref. 183. For other functionals, the HF values (Ref. 124) are used by default, although Q-CHEM will print a warning that this may not be optimal.

Damp_A2

Sets the parameter a_2 in Eq. (5.42).

INPUT SECTION: *$xdm*

TYPE:

FLOAT

DEFAULT:

Various

OPTIONS:

x Set $a_2 = x$.

RECOMMENDATION:

The same comments apply (regarding functional-specific values) as in the case of **Damp_A1**.

Print

Controls the print level for the XDM procedure.

INPUT SECTION: *$xdm*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 No printing.

1 Minimal printing.

2 Debug-level printing.

RECOMMENDATION:

None

Use_Elec_Drv

Specify whether to add the gradient correction to the XDM energy.

INPUT SECTION: *$xdm*

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Use the gradient correction.

FALSE Do not use the gradient correction.

RECOMMENDATION:

This is only valid with Becke's C_6 damping function using the interpolated BR89 model.

Example 5.12 Sample input illustrating a frequency calculation of a vdW complex He$\cdots$N$_2$ using the XDM dispersion correction.

```
$molecule
   0 1
   He 0.000000  0.00000   3.800000
   N  0.000000  0.000000  0.546986
   N  0.000000  0.000000 -0.546986
$end

$rem
   JOBTYPE             FREQ
   IDERIV              2
   EXCHANGE            B3LYP
   INCDFT              0
   SCF_CONVERGENCE     8
   BASIS               6-31G*
   XDM                 TRUE
$end

$xdm
  method      xdm6
  natoms_mol1 1
$end
```

	PBE	PBE0	BLYP	B3LYP	revPBE	M06L	M06
s_R	0.94	0.96	0.62	0.84	0.60	1.26	1.16

Table 5.4: Optimized damping parameters [Eq. (5.44)] for the TS-vdW model, from Ref. 251.

5.7.5 Tkatchenko-Scheffler van der Waals Model (TS-vdW)

Tkatchenko and Scheffler[251] have developed a pairwise method for van der Waals (vdW, *i.e.*, dispersion) interactions, based on a scaling approach that yields *in situ* atomic polarizabilities (α), dispersion coefficients (C_6), and vdW radii (R_{vdW}) that reflect the local electronic environment. These are based on scaling the free-atom values of these parameters in order to account for how the volume of a given atom is modified by its molecular environment. The size of an atom in a molecule is determined using the Hirshfeld partition of the electron density. (Hirshfeld or "stockholder" partitioning, which also affords one measure of atomic charges in a molecule, is described in Section 10.2.2). In the resulting "TS-vdW" approach, only a single empirical range-separation parameter (s_R) is required, which depends upon the underlying exchange-correlation functional.

Note: The parameter s_R is currently implemented only for the PBE, PBE0, BLYP, B3LYP, revPBE, M06L, and M06 functionals.

The TS-vdW energy expression is based on a pairwise-additive model for the dispersion energy,

$$E_{\text{vdW}}^{\text{TS}} = -\frac{1}{2}\sum_{A}^{\text{atoms}}\sum_{B\neq A}^{\text{A}}\left(\frac{C_{6,AB}^{\text{eff}}}{R_{AB}^{6}}\right) f_{\text{damp}}(R_{AB})\ . \tag{5.44}$$

As in DFT-D the R^{-6} potentials in Eq. (5.44) must be damped at short range, and the TS-vdW model uses the damping function

$$f_{\text{damp}}(R_{AB}) = \frac{1}{1+\exp\left[-d(R_{AB}/s_R R_{\text{vdW},AB}^{\text{eff}}-1)\right]} \tag{5.45}$$

with $d = 20$ and an empirical parameter s_R that is optimized in a functional-specific way to reproduce intermolecular interaction energies.[251] Optimized values for several different functionals are listed in Table 5.4.

The pairwise coefficients $C_{6,AB}^{\text{eff}}$ in Eq. (5.44) are constructed from the corresponding atomic parameters $C_{6,A}^{\text{eff}}$ via

$$C_{6,AB}^{\text{eff}} = \frac{2C_{6,A}^{\text{eff}}C_{6,B}^{\text{eff}}}{\left(\alpha_B^{0,\text{eff}}/\alpha_A^{0,\text{eff}}\right)C_{6,A}^{\text{eff}} + \left(\alpha_A^{0,\text{eff}}/\alpha_B^{0,\text{eff}}\right)C_{6,B}^{\text{eff}}}\ , \tag{5.46}$$

as opposed to the simple geometric mean that is used for $C_{6,AB}$ parameters in the empirical DFT-D methods [Eq. (5.25)]. These are "effective" C_6 coefficients in the sense that they account for the local electronic environment. As indicated above, this is accomplished by scaling the corresponding free-atom values, *i.e.*,

$$C_{6,A}^{\text{eff}} = C_{6,A}^{\text{free}}\left(\frac{V_{A,\text{eff}}}{V_{A,\text{free}}}\right)^2 \tag{5.47}$$

where $V_{A,\text{eff}}$ is the effective volume of atom A in the molecule, as determined using Hirshfeld partitioning. Effective atomic polarizabilities and vdW radii are obtained analogously:

$$\alpha_A^{0,\text{eff}} = \alpha_A^{0,\text{free}}\left(\frac{V_{A,\text{eff}}}{V_{A,\text{free}}}\right) \tag{5.48}$$

$$R_{\text{vdW},A}^{\text{eff}} = R_{\text{vdW},A}^{\text{free}}\left(\frac{V_{A,\text{eff}}}{V_{A,\text{free}}}\right)^{1/3}\ . \tag{5.49}$$

All three of these atom-specific parameters are therefore functionals of the electron density.

As with DFT-D, the cost to evaluate the dispersion correction in Eq. (5.44) is essentially zero in comparison to the cost of a DFT calculation. A recent review[108] shows that the performance of the TS-vdW model is on par with that of

other pairwise dispersion corrections. For example, for intermolecular interaction energies in the S66 data set,[267] the TS-vdW correction added to PBE affords a mean absolute error of 0.4 kcal/mol and a maximum error of 1.5 kcal/mol, whereas the corresponding errors for PBE alone are 2.2 kcal/mol (mean) and 7.2 kcal/mol (maximum).

During the implementation of the TS-vdW scheme in Q-CHEM, it was noted that evaluation of the free-atom volumes affords substantially different results as compared to the implementations in the FHI-AIMS and QUANTUM ESPRESSO codes, *e.g.*, $V_{\mathrm{H,free}}$ = 8.68 a.u. (Q-CHEM), 10.32 a.u. (FHI-AIMS), and 10.39 a.u. (QUANTUM ESPRESSO) for hydrogen atom using the PBE functional.[18] These discrepancies were traced to different implementations of Hirshfeld partitioning. In Q-CHEM, the free-atom volumes are computed from an unrestricted atomic SCF calculation and then spherically averaged to obtain spherically-symmetric atomic densities. In FHI-AIMS and QUANTUM ESPRESSO they are obtained by solving a one-dimensional radial Schrödinger equation, which automatically affords spherically-symmetric atomic densities but must be used with fractional occupation numbers for open-shell atoms. These differences could likely be ameliorated by reparameterizing the damping function in Eq. (5.45) for use with atomic volumes calculated self-consistently using Q-CHEM, wherein the representation of the electronic structure is quite different as compared to that in either FHI-AIMS or QUANTUM ESPRESSO. This has not been done, however, and the parameters were simply taken from a previous implementation.[251] In order to reproduced TS-vdW dispersion energies obtained with FHI-AIMS or QUANTUM ESPRESSO, it is possible to use this code in Q-CHEM with scaling factors for the atomic Hirshfeld volumes, recommended values for which are obtained by linear regression, comparing Q-CHEM atomic volumes to those obtained in FHI-AIMS. For full self-consistency, however, these scaling factors should not be used.

The TS-vdW dispersion energy is requested by setting TSVDW = TRUE. Energies and analytic gradients are available.

TSVDW

Flag to switch on the TS-vdW method

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not apply TS-vdW.

1 Apply the TS-vdW method to obtain the TS-vdW energy.

2 Apply the TS-vdW method to obtain the TS-vdW energy and corresponding gradients.

RECOMMENDATION:

Since TS-vdW is itself a form of dispersion correction, it should *not* be used in conjunction with any of the dispersion corrections described in Section 5.7.3.

TSVDW_SR

Set custom value of the s_R damping parameter

TYPE:

INTEGER

DEFAULT:

no default value defined

OPTIONS:

n Corresponding to $n \cdot 10^{-4}$

RECOMMENDATION:

Use predefined values for supported functionals, otherwise consult Ref. 251 and other relevant literature.

HIRSHFELD_CONV

Set different SCF convergence criterion for the calculation of the single-atom Hirshfeld calculations

TYPE:

INTEGER

DEFAULT:

same as SCF_CONVERGENCE

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

5

HIRSHMOD

Apply modifiers to the free-atom volumes used in the calculation of the scaled TS-vdW parameters

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

0 Do not apply modifiers to the Hirshfeld volumes.

1 Apply built-in modifier to H.

2 Apply built-in modifier to H and C.

3 Apply built-in modifier to H, C and N.

4 Apply built-in modifier to H, C, N and O

RECOMMENDATION:

Use the default

Example 5.13 Sample input illustrating a calculation of a water molecule, including the TS-vdW energy.

```
$molecule
   0 1
   O
   H 1 0.95
   H 1 0.95 2 104.5
$end

$rem
   BASIS             6-31G*
   METHOD            PBE
   TSVDW             TRUE  !vdw settings
   HIRSHFELD_CONV    6 ! sets SCF_CONVERGENCE for single atom calculations
   HIRSHMOD          4 ! Apply modifiers to the free-atom volumes for H, C, N, and O
$end
```

5.7.6 Many-Body Dispersion (MBD) Method

Unlike earlier DFT-D methods that were strictly (atomic) pairwise-additive, DFT-D3 includes three-body (triatomic) corrections. These terms are significant for non-covalent complexes assembled from large monomers,[147] especially those that contain a large number of polarizable centers.[76] The *many-body dispersion* (MBD) method of Tkatchenko *et al.*[12,252] represents a more general and less empirical approach that goes beyond the pairwise-additive treatment

of dispersion. This is accomplished by including n-body contributions to the dispersion energy up to the number of atoms, and polarization screening contributions to infinite order. Even in small systems such as benzene dimer, the MBD approach consistently outperforms other popular vdW methods.[37]

The essential idea behind MBD is to approximate the dynamic response of a system by that of dipole-coupled quantum harmonic oscillators (QHOs), each of which represents a fragment of the system of interest. The correlation energy of such a system can then be evaluated exactly by diagonalizing the corresponding Hamiltonian:[108]

$$\hat{H}_{\text{MBD}} = \frac{1}{2}\sum_{A}^{\text{atoms}} \hat{\nabla}^2_{\xi_A} + \frac{1}{2}\sum_{A}^{\text{atoms}} \omega_A^2 \xi_A^2 + \frac{1}{2}\sum_{A,B}^{\text{atoms}} \omega_A \omega_B (\alpha_A^0 \alpha_B^0)^{1/2} \xi_A T_{AB} \xi_B. \tag{5.50}$$

Here, $\xi_A = m_A^{1/2}|\mathbf{r}_A - \mathbf{R}_A|$ is the mass-weight displacement of oscillator A from its center $\mathbf{R}_A$, ω_A is the characteristic frequency, and α_A^0 is the static polarizability. T_{AB} is the dipole potential between the oscillators A and B. The MBD Hamiltonian is obtained through coarse-graining of the long-range correlation (through the long-range dipole tensor $\mathbf{T}_{\text{lr}}$) and approximating the short-range polarizability via the adiabatic connection fluctuation-dissipation formula:

$$E_{c,\text{lr}}^{\text{MBD}} = -\sum_{n=2}^{\infty} \frac{(-1)^n}{n} \int_0^{\infty} \frac{\mathrm{d}u}{2\pi} \sum_{AB} \text{tr}\Big[\big\langle (\alpha_{\text{eff}} \mathbf{T}_{\text{lr}})^n \big\rangle_{AB}(iu)\Big] \,. \tag{5.51}$$

This approximation expresses the dynamic polarizability α_{eff} (of a given fragment) in terms of the polarizability of the corresponding QHO,

$$\alpha_A^{\text{QHO}}(u) = \frac{q_A^2}{m_A(\omega_A^2 - u^2 - i\delta u)} \tag{5.52}$$

in which q_A is the charge, m_A the mass, and ω_A the characteristic frequency of the oscillator. The integration in the frequency domain in Eq. (5.51) can be done analytically, leading to the so-called plasmon pole formula for the correlation energy,

$$E_c = \frac{1}{2}\sum_{p=1}^{3N} (\bar{\omega}_p - \omega_p) \tag{5.53}$$

in which N is the number of fragments, $\bar{\omega}_p$ are the frequencies of the interacting (dipole-coupled) system, and ω_p are the frequencies of the non-interacting system (*i.e.*, the collection of independent QHOs). The sum runs over all $3N$ characteristic frequencies of the system.

A particular method within the MBD framework is defined by the models for the static polarizability ($\alpha^0_{\text{eff},A}$), the non-interacting characteristic frequencies (ω_A), and the damping function [$f(R)$] used to define $\mathbf{T}_{\text{lr}}$. In Q-CHEM, the MBD method is implemented following the "MBD@rsSCS" approach, where "rsSCS" stands for range-separated self-consistent screening.[12] In this approach, $\alpha^0_{\text{eff},A}$ is obtained in a two-step process:

1. The free-volume scaling approach is applied to the free-atom polarizabilities, using the Hirshfeld-partitioned molecular electron density. This is the same procedure used in the TS-vdW method described in Section 5.7.5.

2. The short-range atomic polarizabilities $\alpha_{\text{sr},AB}(iu)$ are obtained by applying a Dyson-like screening on only the short range part of the polarizabilities. The same range-separation will later be used to define $\mathbf{T}_{\text{lr}}$.

The short-range atomic polarizabilities are summed up along one fragment coordinate to obtain the local effective dynamic polarizability, *i.e.*, $\alpha^0_{\text{eff},A} = \sum_B \alpha_{\text{sr},AB}$, and are then spherically averaged. The range-separation (damping) function $f(R)$ used to construct $\alpha_{\text{sr},AB}(iu)$ and $\mathbf{T}_{\text{lr}}$ is the same as that in Eq. (5.45), except with $d = 6$ instead of $d = 20$, and again s_R for a given functional obtained by fitting to interaction energies for non-bonded complexes. The MBD energy is than calculated by diagonalizing the Hamiltonian Eq. (5.50) and using the plasmon-pole formula, Eq. (5.53).

The MBD-vdW approach greatly improves the accuracy of the interaction energies for S66[267] test set, even if a simple functional like PBE is used, with a mean absolute error of 0.3 kcal/mol and a maximum error of 1.3 kcal/mol, as

	BOP	B97-3	B97	BPW91	revPBE	TPSS	BP86	mPW91	PBE	PBE0	PW91	SVWN
s_R	0.42	0.45	0.50	0.53	0.54	0.56	0.64	0.68	0.83	0.85	0.91	1.19

Table 5.5: Optimized values of the dimensionless range-separation parameter s_R [Eq. (5.45)] for the MBD-vdW model, from Ref. 12. (This is the parameter called β in Ref. 12.)

compared to 2.3 kcal/mol (mean) and 7.2 kcal/mol (max) for plain PBE. In general, the MBD-vdW method is superior to pairwise *a posteriori* dispersion corrections.[108]

As mentioned above in the context of the TS-vdW method (Section 5.7.5), the FHI-AIMS or QUANTUM ESPRESSO codes cannot perform exact unrestricted SCF calculations for the atoms and this leads to inconsistent free-atom volumes as compared to the spherical ones computed in Q-CHEM, and thus inconsistent values for the vdW correction. Since the parameters of the TS-vdW and MBD-vdW models were fitted for use with FHI-AIMS and QUANTUM ESPRESSO, results obtained using these codes are slightly closer to S66 benchmarks and thus scalar modifiers are available for the internally-computed Hirshfeld volume ratios in Q-CHEM. For S66, the use of these modifiers leads to negligible differences between results obtained from all three codes.[18]

The MBD-vdW correction is requested by setting MBDVDM = TRUE in the *$rem* section. Other job control variables, including the aforementioned modifiers for the free-atom volume ratios, are the same as those for the TS-vdW method and are described in Section 5.7.5.

The current implementation of MBD in Q-Chem is based on LibMBD,[109] a portable Fortran library interfaced to a multitude of quantum chemical codes, with calculations requesting SCF gradients are done by a Q-CHEM-specific code.[18]

MBDVDW

Flag to switch on the MBD-vdW method

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0	Do not calculate MBD.
MBD_SP	Calculate the MBD-vdW contribution to the energy.
MBD_FORCES	Calculate the MBD-vdW contribution to the energy and the nuclear gradient.
MBD_SCF	Calculate the MBD-vdW contribution to the energy, the nuclear as well as the SCF gradient.

RECOMMENDATION:

NONE

MBDVDW_BETA

Set custom value of the s_R (β) damping parameter

TYPE:

INTEGER

DEFAULT:

no default value defined

OPTIONS:

n Corresponding to $n \cdot 10^{-4}$

RECOMMENDATION:

Use predefined values for supported functionals, otherwise consult Ref. 12 and other relevant literature.

Example 5.14 Sample input illustrating a calculation of a water molecule, including the MBD-vdW energy.

```
$molecule
   0 1
   O
   H 1 0.95
   H 1 0.95 2 104.5
$end

$rem
   BASIS             6-31G*
   METHOD            PBE
   MBDVDW            MBD_SP !vdw settings, single point MBD correction
   HIRSHFELD_CONV    6 ! sets SCF_CONVERGENCE for single atom calculations
   HIRSHMOD          4 ! Apply modifiers to the free-atom volumes for H, C, N, and O
$end
```

5.8 Empirical Corrections for Basis Set Superposition Error

This section describes DFT-C,[284] an empirical correction for basis set superposition error (BSSE) in DFT calculations that is an adaptation of Grimme's geometrical counterpoise (gCP) correction.[142] Unlike the traditional Boys-Bernardi counterpoise correction (Section 8.9),[42] the cost of the DFT-C correction is essentially zero (on the scale of a DFT calculation), and the latter provides an estimate of both inter- and *intra*molecular BSSE. The form of this correction is

$$E_{\text{DFT-C}} = \sigma \sum_{A}^{\text{atoms}} c_A \sum_{B \neq A}^{\text{atoms}} g_{AB^*}^{\text{DFT-C}}(R_{AB})\; h_{AB^*}(\{A, B, \ldots\}) \tag{5.54}$$

where $g_{AB^*}^{\text{DFT-C}}$ is a damped, pairwise BSSE correction,

$$g_{AB^*}^{\text{DFT-C}}(R_{AB}) = d(R_{AB})\; f_{AB^*}^{\text{DFT-C}}(R_{AB}) + \big[1 - d(R_{AB})\big] f_{AB^*}^{\text{DFT-C}}(R_{\text{cov},AB})\;. \tag{5.55}$$

The quantity

$$f_{AB^*}^{\text{DFT-C}}(R_{AB}) = c_{AB} \exp\!\big(-\alpha_{AB} R_{AB}^2 + \beta_{AB} R_{AB}\big) \tag{5.56}$$

is the undamped pairwise BSSE and

$$d(R_{AB}) = \frac{1}{1 + k_{1,AB}(R_{AB}/R_{0,AB})^{-k_{2,AB}}} \tag{5.57}$$

is a damping function. The quantity $h_{AB^*}(\{A, B, ...\})$ is a many-body correction to the two-body BSSE correction, given by

$$h_{AB^*}(\{A, B, ...\}) = \left[1 + \sum_{C \neq A,B} \frac{N_C^{\text{virt}}}{N_B^{\text{virt}}} \operatorname{terfc}\left(R_{AC}, R_{AB}\right) \operatorname{terfc}\left(R_{BC}, R_{AB}\right)\right]^{-1} \tag{5.58}$$

where

$$\operatorname{terfc}(x, y) = 1 - \frac{1}{2}\big[\operatorname{erf}(x + y) + \operatorname{erf}(x - y)\big]\;. \tag{5.59}$$

The parameters c_A, c_{AB}, α_{AB}, and β_{AB} are basis-set-dependent, and the overall scaling parameter σ is loosely method-dependent. All of these parameters are set internally based on the method and basis *$rem* specifications.

Note: Currently, only the def2-SVPD basis set is supported for use with DFT-C.

The DFT-C correction is governed by the following *$rem* variable:

DFT_C

Controls whether the DFT-C empirical BSSE correction should be added.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply the DFT-C correction

TRUE (or 1) Apply the DFT-C correction

RECOMMENDATION:

NONE

The DFT-C method can be applied to any local, GGA, or meta-GGA density functional, as in the following example.

Example 5.15 Geometry optimization of the methane dimer using B97M-V-C/def2-SVPD, *i.e.*, the B97M-V functional with the DFT-C BSSE correction in the def2-SVPD basis set.

```
$molecule
   0 1
   C    0.000000  -0.000140   1.859161
   H   -0.888551   0.513060   1.494685
   H    0.888551   0.513060   1.494685
   H    0.000000  -1.026339   1.494868
   H    0.000000   0.000089   2.948284
   C    0.000000   0.000140  -1.859161
   H    0.000000  -0.000089  -2.948284
   H   -0.888551  -0.513060  -1.494685
   H    0.888551  -0.513060  -1.494685
   H    0.000000   1.026339  -1.494868
$end

$rem
   JOBTYPE            opt
   BASIS              def2-SVPD
   METHOD             b97m-v
   DFT_C              true
$end
```

5.9 Double-Hybrid Density Functional Theory

Double-hybrid density functional theory[33,94,233,250,300] (DH-DFT) has demonstrated tremendous potential for approaching the chemical accuracy with a computational cost comparable to the second-order Møller-Plesset perturbation theory (MP2). In a DH-DFT, a Kohn-Sham (KS) DFT calculation is performed first, followed by a treatment of non-local orbital correlation energy at the level of second-order Møller-Plesset perturbation theory (MP2).[176] This MP2 correlation correction includes a a same-spin (ss) component, $E_c^{\rm ss}$, as well as an opposite-spin (os) component, $E_c^{\rm os}$, which are added to the total energy obtained from the KS-DFT calculation. Two scaling parameters, $c_{\rm ss}$ and $c_{\rm os}$, are introduced in order to avoid double-counting correlation:

$$E_{\text{DH-DFT}} = E_{\text{KS-DFT}} + c_{\rm ss}E_c^{\rm ss} + c_{\rm os}E_c^{\rm os} \tag{5.60}$$

A starting point for understanding where a functional form like Eq. (5.60) might come from is the adiabatic connection formula that provides a rigorous way to define double-hybrid functionals. One considers an adiabatic path between the

fictitious non-interacting KS reference system ($\lambda = 0$) and the real physical system ($\lambda = 1$), while holding the electron density fixed at its value for the real system, for all λ. Then

$$E_{\mathrm{XC}}\left[\rho\right] = \int_0^1 U_{\mathrm{XC},\lambda}[\rho]\; d\lambda\,, \tag{5.61}$$

where $U_{\mathrm{XC},\lambda}$ is the exchange-correlation energy at a coupling strength λ, meaning that the exchange-correlation energy if the electron–electron terms in the Hamiltonian had the form λ/r_{ij} rather than $1/r_{ij}$. Using a linear model of this quantity,

$$U_{\mathrm{XC},\lambda} = a + b\lambda\,, \tag{5.62}$$

one obtains the popular hybrid functional that includes the HF exchange (or occupied orbitals) such as B3LYP.[23,241] If one further uses the Gorling-Levy's perturbation theory (GL2) to define the initial slope at λ= 0, one obtains the doubly hybrid functional form in Eq. (5.60) that includes MP2 type perturbative terms (PT2) involving virtual orbitals:[130]

$$U_{\mathrm{XC},\lambda} = \left.\frac{\partial U_{\mathrm{XC},\lambda}}{\lambda}\right|_{\lambda=0} = 2E_{\mathrm{C}}^{\mathrm{GL2}}\,. \tag{5.63}$$

The adiabatic connection formula has been used to develop double hybrid functionals such as XYG3.[300] Note that XYG3 as implemented in Q-CHEM uses B3LYP orbitals to generate the density and evaluate the PT2 terms. This is different from B2PLYP, an earlier doubly hybrid functional from Grimme.[94] The latter uses truncated KS orbitals while XYG3 uses converged KS orbitals to evaluate the PT2 terms. The performance of XYG3 is not only comparable to that of the G2 or G3 theory for thermochemistry, but barrier heights and non-covalent interactions, including stacking interactions, are also very well described by XYG3.[300]

Note: The recommended basis set for XYG3 is 6-311+G(3df,2p).

Due to the inclusion of PT2 terms, the cost of double-hybrid calculations is formally $\mathcal{O}(N^5)$, as in conventional MP2, thereby not applicable to large systems and partly losing DFT's cost advantages. However, the highly successful SOS-MP2 and local SOS-MP2 algorithms in Q-CHEM can be leveraged to develop double-hybrid functionals based on these $\mathcal{O}(N^4)$ methods. A version of XYG3 that uses SOS-MP2 is the XYGJ-OS functional.[297] This functional has 4 parameters that are optimized using thermochemical data. It is not only faster than XYG3, but comparable to XYG3 (or perhaps slightly better) in accuracy. If the local SOS-MP2 algorithm is applied, the scaling of XYGJ-OS is further reduced to $\mathcal{O}(N^3)$. Recently, XYGJ-OS became the first double-hybrid functional whose analytic energy gradient has been derived and implemented.[120]

Other more empirical double-hybrid functionals have been implemented in Q-CHEM. Among the ωB97 series of functionals, ωB97X-2[58] is a long-range corrected double hybrid that can greatly reduce the self-interaction errors (due to its high fraction of HF exchange), and has been shown significantly superior to other functionals for systems with both bonded and non-bonded interactions. Due to the sensitivity of PT2 correlation energy with respect to the choices of basis sets, ωB97X-2 was parameterized with two different basis sets: the ωB97X-2(LP) was parameterized for use with 6-311++G(3df,3pd), while ωB97X-2(TQZ) was parameterized with the T/Q (triple-ζ/quadruple-ζ) extrapolation to the basis set limit. A careful reading of Ref. 58 is highly advised before using either of these two functionals.

Job control variables for double-hybrid DFT are described below. Note that the PT2 correlation energy can also be computed with the efficient resolution-of-identity (RI) methods. Since Q-CHEM 5.2, RIMP2 can be invoked simply by specifying auxiliary basis set using AUX_BASIS_CORR. See Section 6.6.

DH

Controls the application of DH-DFT scheme.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply the DH-DFT scheme

TRUE (or 1) Apply DH-DFT scheme

RECOMMENDATION:

NONE

The following to *$rem* variables pertain to the ωB97X-2(LP) and ωB97X-2(TQZ) functionals.

SSS_FACTOR

Controls the strength of the same-spin component of PT2 correlation energy.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $c_{\mathrm{ss}} = n/10^6$ in Eq. (5.60).

RECOMMENDATION:

NONE

SOS_FACTOR

Controls the strength of the opposite-spin component of PT2 correlation energy.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $c_{\mathrm{os}} = n/10^6$ in Eq. (5.60).

RECOMMENDATION:

NONE

Example 5.16 Applications of the B2PLYP functional to LiH with and without using RI.

```
$molecule
   0 1
   H
   Li H 1.6
$end

$rem
   EXCHANGE    B2PLYP
   BASIS       cc-pvtz
   AUX_BASIS_CORR   rimp2-cc-pvtz
$end

@@@

$molecule
   read
$end

$rem
   EXCHANGE    B2PLYP
   BASIS       cc-pvtz
$end
```

Example 5.17 Application of the ωB97X-2(TQZ) functional to LiH with and without RI

```
$molecule
   0 1
   H
   Li H 1.6
$end

$rem
   EXCHANGE    omegaB97X-2(TQZ)
   BASIS       cc-pvqz
   AUX_BASIS_CORR   rimp2-cc-pvqz
$end

@@@

$molecule
   read
$end

$rem
   EXCHANGE     omegaB97X-2(TQZ)
   BASIS        cc-pvqz
$end
```

In the following example of XYG3, Q-CHEM automatically performs a B3LYP calculation first, then uses the resulting orbitals to evaluate the PT2 correlation terms. Once can also use XYG3 combined with the RI approximation; use

EXCHANGE = XYG3RI to do so, along with an appropriate choice of auxiliary basis set.

Example 5.18 XYG3 calculation of N_2

```
$molecule
   0 1
   N      0.000000   0.000000   0.547775
   N      0.000000   0.000000  -0.547775
$end

$rem
   EXCHANGE  xyg3
   BASIS     6-311+G(3df,2p)
$end
```

The next example illustrates XYGJ-OS. This functional uses the RI approximation by default, so it is necessary to specify an auxiliary basis set.

Example 5.19 XYGJ-OS calculation of N_2

```
$molecule
   0 1
   N      0.000000   0.000000   0.547775
   N      0.000000   0.000000  -0.547775
$end

$rem
   EXCHANGE    xygjos
   BASIS       6-311+G(3df,2p)
   AUX_BASIS   rimp2-cc-pVTZ
   PURECART    1111
   TIME_MP2    true
$end
```

The final example uses the local version of XYGJ-OS, which is the same as the original XYGJ-OS but with the use of the attenuated Coulomb metric to solve the RI coefficients. Here, the keyword omega determines the locality of the metric.

Example 5.20 Local XYGJ-OS calculation of N_2

```
$molecule
   0 1
   N    0.000    0.000   0.54777500
   N    0.000    0.000  -0.54777500
$end

$rem
   EXCHANGE   lxygjos
   BASIS      6-311+G(3df,2p)
   AUX_BASIS  rimp2-cc-pVTZ
   OMEGA      200
   PURECART   1111
$end
```

5.10 Asymptotically Corrected Exchange-Correlation Potentials

5.10.1 Introduction

No GGA exchange functional can simultaneously produce the correct contribution to the exchange energy density and exchange potential in the asymptotic region of molecular systems,[259] and existing GGA exchange-correlation (XC) potentials decay much faster than the correct $-1/r$ XC potential in the asymptotic region.[49] High-lying occupied orbitals and low-lying virtual orbitals are therefore too loosely bound, and $-\varepsilon_{\mathrm{HOMO}}$ becomes far smaller than the ionization energy, despite the exact condition that these should be the same for the exact functional.[258,289] Moreover, response properties may be poorly predicted from TDDFT calculations with GGA functionals.[258] Long-range corrected hybrid DFT (LRC-DFT), described in Section 5.6, has greatly remedied this situation, but is more expensive that KS-DFT with GGA functionals due to the use of Hatree-Fock exchange. The asymptotic corrections described in this section are designed to remedy the same problems but within the GGA framework.

5.10.2 LB94 Scheme

An asymptotically corrected (AC) exchange potential proposed by van Leeuwen and Baerends is[259]

$$v_x^{\mathrm{LB}} = -\beta \left(\frac{x^2}{1 + 3\beta \mathrm{sinh}^{-1}(x)} \right) \tag{5.64}$$

where $x = \|\hat{\boldsymbol{\nabla}}\rho\|/\rho^{4/3}$ is the reduced density gradient. For an exponentially-decaying density, this potential reduces to $-1/r$ in the asymptotic region of molecular systems. The LB94 xc potential is formed by a linear combination of LDA XC potential and the LB exchange potential:[259]

$$v_{xc}^{\mathrm{LB94}} = v_{xc}^{\mathrm{LDA}} + v_x^{\mathrm{LB}} \ . \tag{5.65}$$

The parameter β in Eq. (5.64) was determined by fitting to the exact XC potential for Be atom. As mentioned in Refs. 50 and 111, for TDDFT calculations, it is sufficient to include the AC XC potential for ground-state calculations followed by TDDFT calculations with an adiabatic LDA XC kernel. The implementation of the LB94 XC potential in Q-CHEM takes this approach, using the LB94 XC potential for the ground state calculations, followed by a TDDFT calculation with an adiabatic LDA XC kernel. This TDLDA/LB94 approach has been widely applied to study excited-state properties of large molecules.

Since the LB exchange potential in Eq. (5.64) does not come from the functional derivative of an exchange energy functional, the Levy-Perdew virial relation[154] is used instead to obtain the exchange energy:

$$E_x^{\mathrm{LB}} = -\int v_x^{\mathrm{LB}}[\rho](\mathbf{r}) \left[3\rho(\mathbf{r}) + \mathbf{r} \cdot \hat{\boldsymbol{\nabla}}\rho(\mathbf{r}) \right] d\mathbf{r} \tag{5.66}$$

An LB94 calculation is requested by setting EXCHANGE = LB94 in the *$rem* section. Additional job control and examples appear below.

LB94_BETA

Sets the β parameter for the LB94 XC potential

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Corresponding to $\beta = n/10000$.

RECOMMENDATION:

Use the default.

Example 5.21 Applications of LB94 XC potential to N_2 molecule.

```
$comment
   TDLDA/LB94 calculation is performed for excitation energies.
$end

$molecule
   0 1
   N    0.0000    0.0000    0.0000
   N    1.0977    0.0000    0.0000
$end

$rem
   EXCHANGE      lb94
   BASIS         6-311(2+,2+)G**
   CIS_N_ROOTS   30
   RPA           true
$end
```

5.10.3 Localized Fermi-Amaldi (LFA) Schemes

Another alternative, proposed by Pan, Fang and Chai,[186] is to use a *localized* version of Fermi-Amaldi exchange-correlation functional. The resulting exchange density functional, whose functional derivative has the correct $-1/r$ asymptotic behavior, can be directly added to any semi-local density functional. Three variants of this method were proposed in Ref. 186. The simplest of these, the strictly-localized Fermi-Amaldi (LFAs) scheme, is implemented in Q-CHEM, for molecules consisting of atoms with $Z \leq 55$.

Example 5.22 LFAs-PBE single-point TD-DFT calculation with water molecule

```
$comment
   Use LFAs-PBE potential for ground-state calculations, followed by
   TDDFT calculations with an adiabatic PBE XC kernel.
$end

$molecule
   0  1
   O
   H1  O  oh
   H2  O  oh  H1  hoh

   oh  =   1.0
   hoh = 110.0
$end

$rem
   EXCHANGE      gen
   BASIS         6-311(2+,2+)G**
   CIS_N_ROOTS   30
   RPA           true
$end

$xc_functional
   X   PBE   1.0
   C   PBE   1.0
   X   LFAs  1.0
$end
```

5.11 Methods Based on "Constrained" DFT

5.11.1 Introduction

Under certain circumstances it is desirable to apply constraints to the electron density during an SCF calculation. For example, in a transition metal complex it may be desirable to constrain the net spin density on a particular metal atom to integrate to a value consistent with the M_S value expected from ligand field theory. Similarly, in a donor/acceptor complex one might be interested in constraining the total density on the acceptor group so that the formal charge on the acceptor is either neutral or negatively charged, depending as the molecule is in its neutral or its charge-transfer state. In these situations, one is interested in controlling the average value of some observable, $O(\mathbf{r})$, to take on a given value, N:

$$\int \rho(\mathbf{r})\, O(\mathbf{r})\; d\mathbf{r} = N \tag{5.67}$$

There are of course many states that satisfy such a constraint, but in practice one is usually looking for the lowest energy such state. To solve the resulting constrained minimization problem, one introduces a Lagrange multiplier, V, and solves for the stationary point of

$$V[\rho, V] = E[\rho] - V\left(\int \rho(\mathbf{r})\, O(\mathbf{r})\; d\mathbf{r} - N\right) \tag{5.68}$$

where $E[\rho]$ is the energy of the system described using density functional theory (DFT). At convergence, the functional W gives the density, ρ, that satisfies the constraint exactly (*i.e.*, it has exactly the prescribed number of electrons on the acceptor or spins on the metal center) but has the lowest energy possible. The resulting self-consistent procedure can be efficiently solved by ensuring at every SCF step the constraint is satisfied exactly. The Q-CHEM implementation of these equations closely parallels those in Ref. 285.

The first step in any constrained DFT calculation is the specification of the constraint operator, $O(\mathbf{r})$. Within Q-CHEM, the user is free to specify any constraint operator that consists of a linear combination of the Becke's atomic partitioning functions[21] or else the fragment-based Hirshfeld partition:[106,266]

$$O(\mathbf{r}) = \sum_{A}^{\text{atoms}} \sum_{\sigma=\alpha,\beta} C_A^{\sigma}\, w_A(\mathbf{r}) \tag{5.69}$$

Here the summation runs over the atoms in the system and over electron spins. The weight function w_A is designed to be ≈ 1 near the nucleus of atom A and rapidly fall to zero near the nucleus of any other atom in the system.[21] The original implementation of cDFT in Q-CHEM used a Becke partition in which the Voronoi polyhedra are defined by the midpoints of the internuclear vectors,[290] but this affords unphysical results.[106,107] As such,the default value of BECKE_SHIFT is now set to use the "atomic size corrections" suggested by Becke,[21] in which a set of empirical atomic radii[236] are used to shift the Voronoi polyhedra away from the bond midpoints and towards something more realistic for bonds between atoms of very different size. This correction is most important for cDFT calculations that are sensitive to the charges on the hydrogen atoms, and cDFT with Becke populations should probably never be used with these atomic size corrections.[107]

The user-specified coefficients C_A^σ are input using a *$cdft* input section having the following format.

```
$cdft
   CONSTRAINT_VALUE_X
   COEFFICIENT1_X       FIRST_ATOM1_X       LAST_ATOM1_X    TYPE1_X
   COEFFICIENT2_X       FIRST_ATOM2_X       LAST_ATOM2_X    TYPE2_X
   ...
   CONSTRAINT_VALUE_Y
   COEFFICIENT1_Y       FIRST_ATOM1_Y       LAST_ATOM1_Y    TYPE1_Y
   COEFFICIENT2_Y       FIRST_ATOM2_Y       LAST_ATOM2_Y    TYPE2_Y
   ...
$end
```

Here, each CONSTRAINT_VALUE is a real number that specifies the desired average value (N) of the ensuing linear combination of atomic partition functions. Each COEFFICIENT specifies the coefficient of a partition function or group of partition functions in the constraint operator O. For each coefficient, all the atoms between the integers FIRST_ATOM and LAST_ATOM contribute with the specified weight in the constraint operator. Finally, TYPE specifies the type of constraint being applied—either "CHARGE" or "SPIN". For a CHARGE constraint the spin up and spin down densities contribute equally ($C_A^\alpha = C_A^\beta = C_A$) yielding the total number of electrons on the atom A. For a SPIN constraint, the spin up and spin down densities contribute with opposite sign ($C_A^\alpha - C_A^\beta = C_A$) resulting in a measure of the net spin on the atom A. Each separate CONSTRAINT_VALUE creates a new operator whose average is to be constrained—for instance, the example above includes several independent constraints: X, Y, ... Q-CHEM can handle an arbitrary number of constraints and will minimize the energy subject to all of these constraints simultaneously.

If an atom is not included in a particular operator, then the coefficient of that atoms partition function is set to zero for that operator. The TYPE specification is optional, and the default is to perform a charge constraint. Further, note that any charge constraint is on the *net* atomic charge. That is, the constraint is on the difference between the average number of electrons on the atom and the nuclear charge. Thus, to constrain CO to be negative, the constraint value would be 1 and not 15.

Note: Constraint in *$cdft* specifies the number of excess electrons on a fragment, not the total charge, *i.e.*, the value 1.0 means charge $= -1$, whereas charge constraint of -1.0 corresponds to the total $+1$ charge.

The choice of which atoms to include in different constraint regions is left entirely to the user and in practice must be based somewhat on chemical intuition. Thus, for example, in an electron transfer reaction the user must specify which atoms are in the "donor" and which are in the "acceptor". In practice, the most stable choice is typically to make the constrained region as large as physically possible. Thus, for the example of electron transfer again, it is best to assign *every* atom in the molecule to one or the other group (donor or acceptor), recognizing that it makes no sense to assign any atoms to both groups. On the other end of the spectrum, constraining the formal charge on a single atom is highly discouraged. The problem is that while our chemical intuition tells us that the lithium atom in LiF should have a formal charge of +1, in practice the quantum mechanical charge is much closer to +0.5 than +1. Only when the fragments are far enough apart do our intuitive pictures of formal charge actually become quantitative.

Note that the atomic populations that Q-CHEM prints out are Mulliken populations, not the Becke weight populations. As a result, the printed populations will not generally add up to the specified constrained values, even though the constraint is exactly satisfied. The *$rem* variable CDFT_BECKE_POP requests that the Becke populations be printed as well, so that the user may confirm that the computed states have the desired charge and/or spin character.

Finally, we note that SCF convergence is typically more challenging in constrained DFT calculations as compared to their unconstrained counterparts. This effect arises because applying the constraint typically leads to a broken symmetry, diradicaloid state. As SCF convergence for these cases is known to be difficult even for unconstrained states, it is perhaps not surprising that there are additional convergence difficulties in this case. See Section 4.5 on SCF

convergence algorithms for ideas on how to improve convergence for constrained calculations. Also, CDFT is more sensitive to grid size than ground-state DFT, so sometimes increasing the integration grid improves the convergence. If convergence is problematic, try using the SG-2 or SG-3 grid (Section 5.5.3).

Note:

1. To improve convergence, use the fewest possible constraints. For example, if your system consists of two fragments, specify the constrains for one of them only. The overall charge and multiplicity will force the "unconstrained" fragment to attain the right charge and multiplicity.

2. The direct minimization methods are not available for constrained calculations. Hence, some combination of DIIS and RCA must be used to obtain convergence. Further, it is often necessary to break symmetry in the initial guess (using SCF_GUESS_MIX) to ensure that the lowest energy solution is obtained.

Analytic gradients are available for constrained DFT calculations.[286] Second derivatives are only available by finite difference of analytic gradients. For details on how to apply constrained DFT to compute magnetic exchange couplings, see Ref. 226. For details on using constrained DFT to compute electron transfer parameters, see Ref. 287,288.

5.11.2 Manner of Counting Electrons

Becke Partition: In density functional theory calculations, the integration over the total density is evaluated on a molecular grid that is systematically broken up into interlocking multi-center atomic quadrature grids.[21] This atomic quadrature scheme is predicated on the definition of atomic cell functions $P_a(\mathbf{r})$, that define smoothed Voronoi polyhedra centered about each atom. These cell functions are products of switching functions that define the atomic cell of atom a, and fall rapidly from ≈ 1 near the nucleus of a, to ≈ 0 near any other nucleus. The integration weights provided by this scheme are multiplied into the Lebedev quadrature weights in any practical DFT calculation:

$$w_n(\mathbf{r}) = \frac{P_n(\mathbf{r})}{\sum_m P_m(\mathbf{r})} \tag{5.70}$$

In some cases, it may be useful to print out the atomic Becke populations that are defined by these atomic cell functions. Becke population analysis may be requested by setting POP_BECKE to TRUE in the input file.

POP_BECKE

Controls the printing of atomic Becke populations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print atomic Becke populations.

FALSE Do not print atomic Becke populations.

RECOMMENDATION:

None

The default quadrature scheme uses atomic cell functions that intersect precisely at bond midpoints. Consequently, the default atomic cell functions will yield physically meaningless atomic populations. However, it is possible to shift the intersect of the atomic cell functions using an atomic radius criterion.[21] In shifting the intersect of neighboring atomic cell functions, the point at which the Becke weights begin to fall from ≈ 1 to ≈ 0 changes depending on the

atomic radius of each atom. While the choice of atomic radius is arbitrary, these atomic cell shifts introduce a physical basis for the partitioning of the underlying atomic quadrature. Two choices for atomic radii exist in Q-CHEM for use with Becke weights, namely the empirically derived radii introduced by Bragg and Slater[236] and the *ab initio*-derived weights due to Pacios.[185]

BECKE_SHIFT

Controls atomic cell shifting in determination of Becke weights.

TYPE:

STRING

DEFAULT:

BRAGG_SLATER

OPTIONS:

UNSHIFTED	Use Becke weighting without atomic size corrections, based on bond midpoints.
BRAGG_SLATER	Use the empirical radii introduced by Bragg and Slater.
UNIVERSAL_DENSITY	Use the *ab initio* radii introduced by Pacios.

RECOMMENDATION:

If interested in the partitioning of the default atomic quadrature, use UNSHIFTED. If using for physical interpretation, choose BRAGG_SLATER or UNIVERSAL_DENSITY. All cDFT calculations and calculations where POP_BECKE = TRUE will default to BRAGG_SLATER radii, otherwise the default grid is UNSHIFTED.

Fragment-Based Hirshfeld Partition: A much less arbitrary scheme with which to count electrons comes from the fragment-based Hirshfeld partition.[106,266] The fragment-based Hirshfeld (FBH) partition uses weights constructed from isolated fragment densities in the form,

$$w_n(\mathbf{r}) = \frac{\rho_n(\mathbf{r})}{\sum_m \rho_m(\mathbf{r})} \,, \tag{5.71}$$

where $\rho_n(\mathbf{r})$ is the density of the isolated fragment, n. Note that unlike the atomic Becke partition, the FBH partition is not constructed from linear combinations of atomic weights, but is instead built from whole fragment densities. The FBH partition comes directly from the densities of the isolated fragments, which are not as arbitrary as the choosing the effective atomic radii in the Becke partition. In order to apply FBH partitioning, one must define fragments within the *$molecule* section to host the constraints, but the input for the *$cdft* section remains unchanged and still applies constraints on a per-atom basis.

CDFT_POP

Sets the charge partitioning scheme for cDFT or cDFT-CI jobs.

TYPE:

STRING

DEFAULT:

BECKE

OPTIONS:

BECKE	Linear combination of atomic Becke functions
FBH	Fragment-based Hirshfeld partition

RECOMMENDATION:

None

5.11.3 Job Control

A CDFT calculation is requested by setting CDFT = TRUE in the *$rem* section. A *$cdft* input section needs to be specified as described above. Three SCF algorithms are currently available for CDFT calculations: DIIS, RCA, and the combined RCA-DIIS algorithm. Additional job control variables are described below.

CDFT

Initiates a constrained DFT calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a Constrained DFT Calculation

FALSE No Density Constraint

RECOMMENDATION:

Set to TRUE if a Constrained DFT calculation is desired.

CDFT_POSTDIIS

Controls whether the constraint is enforced after DIIS extrapolation.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Enforce constraint after DIIS

FALSE Do not enforce constraint after DIIS

RECOMMENDATION:

Use the default unless convergence problems arise, in which case it may be beneficial to experiment with setting CDFT_POSTDIIS to FALSE. With this option set to TRUE, energies should be variational after the first iteration.

CDFT_PREDIIS

Controls whether the constraint is enforced before DIIS extrapolation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Enforce constraint before DIIS

FALSE Do not enforce constraint before DIIS

RECOMMENDATION:

Use the default unless convergence problems arise, in which case it may be beneficial to experiment with setting CDFT_PREDIIS to TRUE. Note that it is possible to enforce the constraint both before and after DIIS by setting both CDFT_PREDIIS and CDFT_POSTDIIS to TRUE.

CDFT_THRESH

Threshold that determines how tightly the constraint must be satisfied.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

N Constraint is satisfied to within 10^{-N}.

RECOMMENDATION:

Default value is set to match SCF_CONVERGENCE. Use the default unless problems occur.

CDFT_MAXITER

Maximum number of iterations for converging the constraint.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

N A maximum of N microiterations will be attempted.

RECOMMENDATION:

Default value is expected to be sufficient in most situations.

CDFT_PRINT

Whether detailed information about CDFT iterations should be printed in the output file.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print detailed information.

FALSE Do not print detailed information.

RECOMMENDATION:

Use the default and invoke additional printing for troubleshooting.

CDFT_BECKE_POP

Whether the calculation should print the Becke atomic charges at convergence

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Print populations

FALSE Do not print them

RECOMMENDATION:

Use the default. Note that the Mulliken populations printed at the end of an SCF run will not typically add up to the prescribed constraint value. Only the Becke populations are guaranteed to satisfy the user-specified constraints.

5.11.4 Examples

Example 5.23 Charge separation on FAAQ

```
$molecule
   0 1
   C     -0.64570736     1.37641945    -0.59867467
   C      0.64047568     1.86965826    -0.50242683
   C      1.73542663     1.01169939    -0.26307089
   C      1.48977850    -0.39245666    -0.15200261
   C      0.17444585    -0.86520769    -0.27283957
   C     -0.91002699    -0.02021483    -0.46970395
   C      3.07770780     1.57576311    -0.14660056
   C      2.57383948    -1.35303134     0.09158744
   C      3.93006075    -0.78485926     0.20164558
   C      4.16915637     0.61104948     0.08827557
   C      5.48914671     1.09087541     0.20409492
   H      5.64130588     2.16192921     0.11315072
   C      6.54456054     0.22164774     0.42486947
   C      6.30689287    -1.16262761     0.53756193
   C      5.01647654    -1.65329553     0.42726664
   H     -1.45105590     2.07404495    -0.83914389
   H      0.85607395     2.92830339    -0.61585218
   H      0.02533661    -1.93964850    -0.19096085
   H      7.55839768     0.60647405     0.51134530
   H      7.13705743    -1.84392666     0.71043613
   H      4.80090178    -2.71421422     0.50926027
   O      2.35714021    -2.57891545     0.20103599
   O      3.29128460     2.80678842    -0.23826460
   C     -2.29106231    -0.63197545    -0.53957285
   O     -2.55084900    -1.72562847    -0.95628300
   N     -3.24209015     0.26680616     0.03199109
   H     -2.81592456     1.08883943     0.45966550
   C     -4.58411403     0.11982669     0.15424004
   C     -5.28753695     1.14948617     0.86238753
   C     -5.30144592    -0.99369577    -0.39253179
   C     -6.65078185     1.06387425     1.01814801
   H     -4.73058059     1.98862544     1.26980479
   C     -6.66791492    -1.05241167    -0.21955088
   H     -4.76132422    -1.76584307    -0.92242502
   C     -7.35245187    -0.03698606     0.47966072
   H     -7.18656323     1.84034269     1.55377875
   H     -7.22179827    -1.89092743    -0.62856041
   H     -8.42896369    -0.10082875     0.60432214
$end

$rem
   JOBTYPE         FORCE
   METHOD          B3LYP
   BASIS           6-31G*
   SCF_PRINT       TRUE
   CDFT            TRUE
$end

$cdft
   2
   1   1  25
  -1  26  38
$end
```

The value of 2 under $cdft section in the FAAQ example represents the constraint. The valules, 1 and -1, represent the coefficients for fragment charges. Suppose the first fragment consists of atom 1–25 with A excess electrons, and the

second fragment consists of atom 26–38 with B excess electrons. The $cdft section can be viewd as a system of linear equations as follows:

$$(\mathbf{1}) * A + (-\mathbf{1}) * B = \mathbf{2}$$
$$A + B = 0$$

The second equation is zero because the FAAQ is neutral. by solving the system of linear equations, one gets A = 1, and B = -1. However, the $cdft section is equivalent to the following one.

```
$cdft
1
1 1  25
-1
1 26 38
$end
```

Example 5.24 Cu2-Ox High Spin

```
$molecule
   2 3
   Cu      1.4674     1.6370     1.5762
   O       1.7093     0.0850     0.3825
   O      -0.5891     1.3402     0.9352
   C       0.6487    -0.3651    -0.1716
   N       1.2005     3.2680     2.7240
   N       3.0386     2.6879     0.6981
   N       1.3597     0.4651     3.4308
   H       2.1491    -0.1464     3.4851
   H       0.5184    -0.0755     3.4352
   H       1.3626     1.0836     4.2166
   H       1.9316     3.3202     3.4043
   H       0.3168     3.2079     3.1883
   H       1.2204     4.0865     2.1499
   H       3.8375     2.6565     1.2987
   H       3.2668     2.2722    -0.1823
   H       2.7652     3.6394     0.5565
   Cu     -1.4674    -1.6370    -1.5762
   O      -1.7093    -0.0850    -0.3825
   O       0.5891    -1.3402    -0.9352
   C      -0.6487     0.3651     0.1716
   N      -1.2005    -3.2680    -2.7240
   N      -3.0386    -2.6879    -0.6981
   N      -1.3597    -0.4651    -3.4308
   H      -2.6704    -3.4097    -0.1120
   H      -3.6070    -3.0961    -1.4124
   H      -3.5921    -2.0622    -0.1485
   H      -0.3622    -3.1653    -3.2595
   H      -1.9799    -3.3721    -3.3417
   H      -1.1266    -4.0773    -2.1412
   H      -0.5359     0.1017    -3.4196
   H      -2.1667     0.1211    -3.5020
   H      -1.3275    -1.0845    -4.2152
$end

$rem
   METHOD          B3LYP
   BASIS           6-31G*
   SCF_PRINT       TRUE
   CDFT            TRUE
$end

$cdft
   2
   1   1   3  s
  -1  17  19  s
$end
```

Example 5.25 Constrained DFT with FBH charges applied to $F^- \cdots H_2O$ interaction

```
$molecule
   -1 1
--
   -1 1
   F          1.2344377204   -0.0287603388    0.0000000000
--
   0 1
   O         -1.2152661043    0.1159898799    0.0000000000
   H         -0.1545755250    0.1042552996    0.0000000000
   H         -1.3911772011   -0.8334364448    0.0000000000
$end

$rem
   METHOD        hf
   BASIS         aug-cc-pvtz
   XC_GRID       3
   CDFT          true
   CDFT_POP      fbh
   NO_REORIENT   true
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

$cdft
   1.0
   1 1 1
   0.0
   1 1 1 s
$end
```

5.11.5 Configuration Interaction with Constrained DFT (CDFT-CI)

There are some situations in which a system is not well-described by a DFT calculation on a single configuration. For example, transition states are known to be poorly described by most functionals, with the computed barrier being too low. We can, in particular, identify homolytic dissociation of diatomic species as situations where static correlation becomes extremely important. Existing DFT functionals have proved to be very effective in capturing dynamic correlation, but frequently exhibit difficulties in the presence of strong static correlation. Configuration Interaction, well known in wave function methods, is a multi-reference method that is quite well-suited for capturing static correlation; the CDFT-CI technique allows for CI calculations on top of DFT calculations, harnessing both static and dynamic correlation methods.

Constrained DFT is used to compute densities (and Kohn-Sham wave functions) for two or more diabatic-like states; these states are then used to build a CI matrix. Diagonalizing this matrix yields energies for the ground *and* excited states within the configuration space. The coefficients of the initial diabatic states are printed, to show the characteristics of the resultant states.

Since DFT only gives converged densities, not actual wave functions, computing the off-diagonal coupling elements H_{12} is not completely straightforward, as the physical meaning of the Kohn-Sham wave function is not entirely clear. We can, however, perform the following manipulation:[288]

$$\begin{aligned} H_{12} &= \frac{1}{2}\Big[\langle 1|H + V_{C_1}\omega_{C_1} - V_{C1}\omega_{C_1}|2\rangle + \langle 1|H + V_{C_2}\omega_{C_2} - V_{C_2}\omega_{C_2}|2\rangle\Big] \\ &= \frac{1}{2}\Big[\left(E_1 + V_{C_1}N_{C_1} + E_2 + V_{C_2}N_{C_2}\right)\langle 1|2\rangle - V_{C_1}\langle 1|\omega_{C_1}|2\rangle - V_{C_2}\langle 1|\omega_{C_2}|2\rangle\Big] \end{aligned} \tag{5.72}$$

where the converged states $|i\rangle$ are assumed to be the ground state of $H + V_{C_i}\omega_{C_i}$ with eigenvalue $E_i + V_{C_i}N_{C_i}$). This manipulation eliminates the two-electron integrals from the expression, and experience has shown that the use of Slater determinants of Kohn-Sham orbitals is a reasonable approximation for the quantities $\langle 1|2\rangle$ and $\langle 1|\omega_{C_i}|2\rangle$.

Note that since these constrained states are eigenfunctions of different Hamiltonians (due to different constraining potentials), they are *not* orthogonal states, and we must set up our CI matrix as a generalized eigenvalue problem. Symmetric orthogonalization is used by default, though the overlap matrix and Hamiltonian in non-orthogonal basis are also printed at higher print levels so that other orthogonalization schemes can be used after-the-fact. In a limited number of cases, it is possible to find an orthogonal basis for the CDFT-CI Hamiltonian, where a physical interpretation can be assigned to the orthogonal states. In such cases, the matrix representation of the Becke weight operator is diagonalized, and the (orthogonal) eigenstates can be characterized.[290] This matrix is printed as the "CDFT-CI Population Matrix" at increased print levels.

In order to perform a CDFT-CI calculation, the N interacting states must be defined, which is accomplished using a *$cdft* input section in a fashion similar to the specification of CDFT states:

```
$cdft
STATE_1_CONSTRAINT_VALUE_X
COEFFICIENT1_X       FIRST_ATOM1_X       LAST_ATOM1_X    TYPE1_X
COEFFICIENT2_X       FIRST_ATOM2_X       LAST_ATOM2_X    TYPE2_X
...
STATE_1_CONSTRAINT_VALUE_Y
COEFFICIENT1_Y       FIRST_ATOM1_Y       LAST_ATOM1_Y    TYPE1_Y
COEFFICIENT2_Y       FIRST_ATOM2_Y       LAST_ATOM2_Y    TYPE2_Y
...
---
STATE_2_CONSTRAINT_VALUE_X
COEFFICIENT1_X       FIRST_ATOM1_X       LAST_ATOM1_X    TYPE1_X
COEFFICIENT2_X       FIRST_ATOM2_X       LAST_ATOM2_X    TYPE2_X
...
STATE_2_CONSTRAINT_VALUE_Y
COEFFICIENT1_Y       FIRST_ATOM1_Y       LAST_ATOM1_Y    TYPE1_Y
COEFFICIENT2_Y       FIRST_ATOM2_Y       LAST_ATOM2_Y    TYPE2_Y
...
$end
```

Each state is specified with the CONSTRAINT_VALUE and the corresponding weights on sets of atoms whose average value should be the constraint value. Different states are separated by a single line containing three or more dash characters.

If it is desired to use an unconstrained state as one of the interacting configurations, charge and spin constraints of zero may be applied to the atom range from 0 to 0.

Note: It is mandatory to specify a spin constraint corresponding to every charge constraint (and it must be immediately following that charge constraint in the input deck), for reasons described below.

In addition to the *$cdft* input section of the input file, a CDFT-CI calculation must also set the CDFTCI flag to TRUE for the calculation to run. Note, however, that the CDFT flag is used internally by CDFT-CI, and should *not* be set in the input deck. The variable CDFTCI_PRINT may also be set manually to control the level of output. The default is 0, which will print the energies and weights (in the diabatic basis) of the N CDFT-CI states. Setting it to 1 or above will also print the CDFT-CI overlap matrix, the CDFT-CI Hamiltonian matrix before the change of basis, and the CDFT-CI Population matrix. Setting it to 2 or above will also print the eigenvectors and eigenvalues of the CDFT-CI Population

matrix. Setting it to 3 will produce more output that is only useful during application debugging.
For convenience, if CDFTCI_PRINT is not set in the input file, it will be set to the value of SCF_PRINT.

As mentioned in the previous section, there is a disparity between our chemical intuition of what charges should be and the actual quantum-mechanical charge. The example was given of LiF, where our intuition gives the lithium atom a formal charge of $+1$; we might similarly imagine performing a CDFT-CI calculation on H_2, with two ionic states and two spin-constrained states. However, this would result in attempting to force both electrons of H_2 onto the same nucleus, and this calculation is impossible to converge (since by the nature of the Becke weight operators, there will be some non-zero amount of the density that gets proportioned onto the other atom, at moderate internuclear separations). To remedy problems such as this, we have adopted a mechanism by which to convert the formal charges of our chemical intuition into reasonable quantum-mechanical charge constraints. We use the formalism of "promolecule" densities, wherein the molecule is divided into fragments (based on the partitioning of constraint operators), and a DFT calculation is performed on these fragments, completely isolated from each other.[290] (This step is why both spin and charge constraints are required, so that the correct partitioning of electrons for each fragment may be made.) The resulting promolecule densities, converged for the separate fragments, are then added together, and the value of the various weight operators as applied to this new density, is used as a constraint for the actual CDFT calculations on the interacting states. The promolecule density method compensates for the effect of nearby atoms on the actual density that will be constrained.

The comments about SCF convergence for CDFT calculations also apply to the calculations used for CDFT-CI, with the addition that if the SCF converges but CDFT does not, it may be necessary to use a denser integration grid or reduce the value of CDFT_THRESH.

Analytic gradients are not available. Many of the CDFT-related *$rem* variables are also applicable to CDFT-CI calculations. For details on using CDFT-CI to calculate reaction barrier heights, see Ref. 291.

5.11.6 CDFT-CI Job Control and Examples

CDFTCI

Initiates a constrained DFT-configuration interaction calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a CDFT-CI Calculation

FALSE No CDFT-CI

RECOMMENDATION:

Set to TRUE if a CDFT-CI calculation is desired.

CDFTCI_PRINT

Controls level of output from CDFT-CI procedure to Q-CHEM output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Only print energies and coefficients of CDFT-CI final states

1 Level 0 plus CDFT-CI overlap, Hamiltonian, and population matrices

2 Level 1 plus eigenvectors and eigenvalues of the CDFT-CI population matrix

3 Level 2 plus promolecule orbital coefficients and energies

RECOMMENDATION:

Level 3 is primarily for program debugging; levels 1 and 2 may be useful for analyzing the coupling elements

CDFT_LAMBDA_MODE

Allows CDFT potentials to be specified directly, instead of being determined as Lagrange multipliers.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard CDFT calculations are used.

TRUE Instead of specifying target charge and spin constraints, use the values from the input deck as the value of the Becke weight potential

RECOMMENDATION:

Should usually be set to FALSE. Setting to TRUE can be useful to scan over different strengths of charge or spin localization, as convergence properties are improved compared to regular CDFT(-CI) calculations.

CDFTCI_SKIP_PROMOLECULES

Skips promolecule calculations and allows fractional charge and spin constraints to be specified directly.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard CDFT-CI calculation is performed.

TRUE Use the given charge/spin constraints directly, with no promolecule calculations.

RECOMMENDATION:

Setting to TRUE can be useful for scanning over constraint values.

Note: CDFT_LAMBDA_MODE and CDFTCI_SKIP_PROMOLECULES are mutually incompatible.

CDFTCI_SVD_THRESH

By default, a symmetric orthogonalization is performed on the CDFT-CI matrix before diagonalization. If the CDFT-CI overlap matrix is nearly singular (*i.e.*, some of the diabatic states are nearly degenerate), then this orthogonalization can lead to numerical instability. When computing $\mathbf{S}^{-1/2}$, eigenvalues smaller than $10^{-\text{CDFTCI\_SVD\_THRESH}}$ are discarded.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

Can be decreased if numerical instabilities are encountered in the final diagonalization.

CDFTCI_STOP

The CDFT-CI procedure involves performing independent SCF calculations on distinct constrained states. It sometimes occurs that the same convergence parameters are not successful for all of the states of interest, so that a CDFT-CI calculation might converge one of these diabatic states but not the next. This variable allows a user to stop a CDFT-CI calculation after a certain number of states have been converged, with the ability to restart later on the next state, with different convergence options.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Stop after converging state n (the first state is state 1)

0 Do not stop early

RECOMMENDATION:

Use this setting if some diabatic states converge but others do not.

CDFTCI_RESTART

To be used in conjunction with CDFTCI_STOP, this variable causes CDFT-CI to read already-converged states from disk and begin SCF convergence on later states. Note that the same *$cdft* section must be used for the stopped calculation and the restarted calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Start calculations on state $n + 1$

RECOMMENDATION:

Use this setting in conjunction with CDFTCI_STOP.

Example 5.5.26 CDFT-CI calculation of couplings between the anionic GFP chromophore (CHR:1-27) and a tyrosine (TYR:28-43) residue. The two diabatic states are CHR($M_S = 0$)$\cdots$TYR($M_S = 0$) and CHR($M_S = 1/2$)$\cdots$TYR($M_S = 1/2$).

View input online

5.12 Unconventional DFT Methods

5.12.1 Density-Corrected DFT

It is well known that self-interaction error (SIE) in DFT leads to over-delocalization of unpaired spins in open-shell molecules.[66] This has a variety of deleterious effects, including overstabilization of three-center, two-electron "hemi-bonds",[131,220] fractional charges in well-separated chemical moieties (*i.e.*, upon dissociation),[166,228,299] and too-low reaction barriers, the latter of which was largely the motivation for the introduction of hybrid density functionals.[118,163] Although various *ad hoc* self-interaction correction schemes have been introduced over the years, none of them is entirely satisfactory.[118,204] Density-corrected (DC-)DFT[235,239,268] represents a revival of an old idea[83,182] to avoid SIE by evaluating a DFT functional non-self-consistently using self-consistent Hartree-Fock density, which is SIE-free. Self-consistent iterations at the DFT level are avoided as this would re-introduce SIE into the density. If $E_{\text{DFT}}[\rho]$ represents the user's chosen density functional and $E_{\text{HF}}[\rho]$ represents the Hartree-Fock functional, then the DC-DFT energy functional is[222]

$$E_{\text{DC-DFT}}[\rho] = E_{\text{DFT}}\left[\arg\min_{\rho}\left(E_{\text{HF}}[\rho]\right)\right] \tag{5.73}$$

DC-DFT affords barrier heights that are comparable in accuracy to those obtained with hybrid functionals, even if $E_{\text{DFT}}[\rho]$ is a semilocal functional.[119,221,262] This does not really reduce the cost of hybrid DFT calculations since the Hartree-Fock calculation must be iterated to self-consistency, nevertheless DC-DFT may serve as a useful diagnostic tool. If the DC-DFT result with a given functional is qualitatively different than the self-consistent DFT result with the same functional, then density-driven SIE may be affecting the results. This diagnostic capacity has been used, for example, to detect unrealistic delocalization of polaron (spin) defects in metal oxides.[221]

Users of Q-CHEM's implementation of DC-DFT are asked to cite Ref. 221. Analytic energy gradients for DC-DFT are available,[221] but because the functional $E_{\text{DC-DFT}}$ is not iterated to self-consistency evaluation of the gradient $dE_{\text{DC-DFT}}/dx$ requires solution of coupled-perturbed (Z-vector) equations,[221,262] which makes the gradient somewhat more expensive than a traditional DFT gradient.

Note: At present, the coupled-perturbed eqution for DC-DFT are solved in serial, meaning that while the SCF iterations are parallelized the Z-vector iterations are not.

To perform a DC-DFT calculation, set use METHOD in the *$rem* section to select the functional of choice, and also set DC_DFT = TRUE. Note that because $E_{\text{DFT}}[\rho]$ is never diagonalized, any subsequent properties that are computed at the end of the SCF procedure are based on the Hartree-Fock density. This includes one-particle energy levels, Mulliken charges, multipole moments, etc.

DC_DFT

Controls whether to use DC-DFT.

TYPE:

Boolean

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do DC-DFT.

TRUE Iterate the density to self-consistency at the Hartree-Fock level and then perform evaluate $E_{\text{DFT}}[\rho_{\text{HF}}]$ using the functional specified with METHOD.

RECOMMENDATION:

Use if desired. Analytic gradients are available but are a serial bottleneck in the present implementation.

5.12.2 Derivative Discontinuity Restoration

From the perspective of perturbation theory, Chai and Chen[55] proposed a systematic procedure for the evaluation of the derivative discontinuity of the exchange-correlation energy functional in Kohn-Sham DFT, wherein the exact derivative discontinuity can in principle be obtained by summing up all the perturbation corrections to infinite order. Truncation of the perturbation series at low order yields an efficient scheme for obtaining the approximate derivative discontinuity. In particular, the first-order correction term is equivalent to the frozen-orbital approximation method. Its implementation in Q-CHEM supports only local and GGA functionals at present, not meta-GGA, hybrid, or non-local functionals. Job control variables and examples appear below.

FOA_FUNDGAP

Compute the frozen-orbital approximation of the fundamental gap.

TYPE:

Boolean

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute FOA derivative discontinuity and fundamental gap.

TRUE Compute and print FOA fundamental gap information. Implies KS_GAP_PRINT.

RECOMMENDATION:

Use in conjunction with KS_GAP_UNIT if true.

KS_GAP_PRINT

Control printing of (generalized Kohn-Sham) HOMO-LUMO gap information.

TYPE:

Boolean

DEFAULT:

false

OPTIONS:

false (default) do not print gap information

true print gap information

RECOMMENDATION:

Use in conjunction with KS_GAP_UNIT if true.

KS_GAP_UNIT

Unit for KS_GAP_PRINT and FOA_FUNDGAP (see Section 5.12.2)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 (default) hartrees

1 eV

RECOMMENDATION:

none

Example 5.27 Frozen-orbital approximation of derivative discontinuity with PBE and LFAs-PBE functionals on carbon atom.

```
$comment
    Frozen-orbital derivative discontinuity, C atom, PBE
$end

$molecule
   0 3
   C
$end

$rem
   BASIS          6-31G*
   METHOD         PBE
   FOA_FUNDGAP    true
   KS_GAP_UNIT    1    ! print gap info in eV
   THRESH         14
$end

@@@

$comment
   with LFAs-PBE functional instead
$end

$molecule
   READ
$end

$rem
   BASIS          6-31G*
   SCF_GUESS      READ
   EXCHANGE       gen
   FOA_FUNDGAP    true
   KS_GAP_UNIT    1
   THRESH         14
$end

$xc_functional
   X   PBE    1.0
   X   LFAs   1.0
   C   PBE    1.0
$end
```

5.12.3 Thermally-Assisted-Occupation (TAO) DFT

Aiming to study the ground-state properties of large, strongly correlated systems with minimum computational complexity, Prof. Jeng-Da Chai recently developed thermally-assisted-occupation density functional theory (TAO-DFT).[52] Unlike conventional multi-reference methods, the computational complexity of TAO-DFT increases very insignificantly with the size of the active space (*i.e.*, an active space restriction is not needed for TAO-DFT calculations), and TAO-DFT appears to be very promising for the study of large poly-radical systems. TAO-DFT is a DFT scheme with fractional orbital occupations produced by the Fermi-Dirac distribution, controlled by a fictitious temperature θ, and existing XC functionals (*e.g.*, LDA[52], GGAs[53] or global hybrid GGAs[54]) can be used in TAO-DFT. The computational cost of the method is similar to that of Kohn-Sham DFT for single-point energy calculations and analytical nuclear gradients, and reduces to the cost of Kohn-Sham DFT in the absence of strong static correlation effects.

There are several *$rem* variables that are used for TAO-DFT.

TAO_DFT

Controls whether to use TAO-DFT.

TYPE:

Boolean

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use TAO-DFT

TRUE Use TAO-DFT

RECOMMENDATION:

NONE

TAO_DFT_THETA

The parameter m (the mantissa) for the value of the fictitious temperature $\theta = m \times 10^{-n}\ E_h$ in TAO-DFT.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

m Customize the mantissa for the fictitious temperature.

RECOMMENDATION:

NONE

TAO_DFT_THETA_NDP

The parameter n (the exponent) for the value of the fictitious temperature $\theta = m \times 10^{-n}\ E_h$ in TAO-DFT.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Customize the exponential power for the fictitious temperature.

RECOMMENDATION:

NONE

Note that setting TAO_DFT_THETA = 0 recovers ordinary Kohn-Sham DFT.[52] In addition to the XC functional, a functional $E_\theta[\rho]$ is needed in TAO-DFT. Currently available in Q-CHEM are an LDA version[52] (the ETheta_LDA functional) as well as a version based on the gradient expansion approximation[53] (GEA) (the ETheta_GEA functional), and the latter may be substituted for the former in the sample jobs below. Furthermore, a functional $E_{x,\theta}[\rho]$ is also needed in TAO-DFT for global hybrid GGAs. Currently available in Q-CHEM is an LDA version (the EThetaX_LDA

functional).[54]

Example 5.28 TAO-LDA calculation on Be atom

```
$molecule
   0 1
   Be
$end

$rem
   BASIS                6-31G*
   EXCHANGE             gen
   TAO_DFT              true
   TAO_DFT_THETA        7    ! default, theta=7 mhartree
   TAO_DFT_THETA_NDP    3    ! default
$end

$xc_functional
   X   S              1.0
   C   PW92           1.0
   X   ETheta_LDA     1.0
$end
```

Example 5.29 TAO-PBE, spin-restricted calculation on stretched N_2

```
$molecule
   0 1
   N1
   N2 N1 4.5
$end

$rem
   BASIS                6-31G*
   EXCHANGE             gen
   TAO_DFT              true
   TAO_DFT_THETA        40  ! theta = 40 mhartree
   TAO_DFT_THETA_NDP    3
$end

$xc_functional
   X PBE          1.0
   C PBE          1.0
   X ETheta_LDA   1.0
$end
```

Example 5.30 TAO-PBE, spin-unrestricted calculation on stretched N_2

```
$molecule
   0 1
   N1
   N2 N1 5.0
$end

$rem
   JOBTYPE             opt
   UNRESTRICTED        true
   BASIS               6-31G*
   EXCHANGE            gen
   TAO_DFT             true
   TAO_DFT_THETA       40  ! theta = 40 mhartrees
   TAO_DFT_THETA_NDP   3   ! can omit this line
   SCF_GUESS           gwh
   SCF_GUESS_MIX       3   ! mix in 30% LUMO in alpha to break symmetry
$end

$xc_functional
   X PBE         1.0
   C PBE         1.0
   X ETheta_LDA  1.0
$end
```

Example 5.31 TAO-PBE0 calculation on H_2 molecule

```
$molecule
   0 1
   H1
   H2 H1 1.00
$end

$rem
   JOBTYPE             opt
   UNRESTRICTED        true
   BASIS               6-31G*
   EXCHANGE            gen
   TAO_DFT             true
   TAO_DFT_THETA       20  ! theta = 20 mhartrees
   TAO_DFT_THETA_NDP   3   ! can omit this line
   SCF_GUESS           gwh
   SCF_GUESS_MIX       3   ! mix in 30% LUMO in alpha to break symmetry
$end

$xc_functional
   X ETheta_LDA    1.00
   X EThetaX_LDA   0.25
   X HF            0.25
   X PBE           0.75
   C PBE           1.00
$end
```

Example 5.32 TAO-B3LYP calculation on H_2 molecule

```
$molecule
   0 1
   H1
   H2 H1 1.00
$end

$rem
   JOBTYPE             opt
   UNRESTRICTED        true
   BASIS               6-31G*
   EXCHANGE            gen
   TAO_DFT             true
   TAO_DFT_THETA       174 ! theta = 17.4 mhartrees
   TAO_DFT_THETA_NDP   4
   SCF_GUESS           gwh
   SCF_GUESS_MIX       3   ! mix in 30% LUMO in alpha to break symmetry
$end

$xc_functional
   X ETheta_LDA    1.00
   X EThetaX_LDA   0.20
   X HF            0.20
   X Slater        0.08
   X Becke88       0.72
   C LYP           0.81
   C VWN1RPA       0.19
$end
```

References and Further Reading

[1] AOINTS (Appendix A).

[2] Molecular Properties Analysis (Chapter 10).

[3] Basis Sets and Effective Core Potentials (Chapter 8).

[4] Molecular Geometry and Critical Points (Chapter 9).

[5] C. Adamo and V. Barone. *Chem. Phys. Lett.*, 274:242, 1997. DOI: 10.1016/S0009-2614(97)00651-9.

[6] C. Adamo and V. Barone. *J. Chem. Phys.*, 108:664, 1998. DOI: 10.1063/1.475428.

[7] C. Adamo and V. Barone. *J. Chem. Phys.*, 110:6158, 1999. DOI: 10.1063/1.478522.

[8] C. Adamo, G. E. Scuseria, and V. Barone. *J. Chem. Phys.*, 111:2889, 1999. DOI: 10.1063/1.479571.

[9] R. D. Adamson, P. M. W. Gill, and J. A. Pople. *Chem. Phys. Lett.*, 284:6, 1998. DOI: 10.1016/S0009-2614(97)01282-7.

[10] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill. *J. Comput. Chem.*, 20:921, 1999. DOI: 10.1002/(SICI)1096-987X(19990715)20:9<921::AID-JCC3>3.0.CO;2-K.

[11] B. Alam, A. F. Morrison, and J. M. Herbert. *J. Phys. Chem. C*, 124:24653, 2020. DOI: 10.1021/acs.jpcc.0c07932.

[12] A. Ambrosetti, A. M. Reilly, R. A. DiStasio, Jr., and A. Tkatchenko. *J. Chem. Phys.*, 140:18A508, 2014. DOI: 10.1063/1.4865104.

[13] R. Armiento and S. Kümmel. *Phys. Rev. Lett.*, 111:036402, 2013. DOI: 10.1103/PhysRevLett.111.036402.

[14] T. Aschebrock and S. Kümmel. *Phys. Rev. Research*, 1:033082, 2019. DOI: 10.1103/PhysRevResearch.1.033082.

[15] J. Autschbach and M. Srebro. *Acc. Chem. Res.*, 47:2592, 2014. DOI: 10.1021/ar500171t.

[16] R. Baer, E. Livshits, and U. Salzner. *Annu. Rev. Phys. Chem.*, 61:85, 2010. DOI: 10.1146/annurev.physchem.012809.103321.

[17] A. P. Bartok and J. R. Yates. *J. Chem. Phys.*, 150:161101, 2019. DOI: 10.1063/1.5094646.

[18] D. Barton, S. Goger, K. U. Lao, R. A. DiStasio, Jr., and A. Tkatchenko. Implementation, validation and reproducibility of the Tkatchenko-Scheffler and many-body dispersion methods in three electronic-structure codes. (in preparation).

[19] A. D. Becke. *J. Chem. Phys.*, 84:4524, 1986. DOI: 10.1063/1.450025.

[20] A. D. Becke. *J. Chem. Phys.*, 85:7184, 1986. DOI: 10.1063/1.451353.

[21] A. D. Becke. *J. Chem. Phys.*, 88:2547, 1988. DOI: 10.1063/1.454033.

[22] A. D. Becke. *Phys. Rev. A*, 38:3098, 1988. DOI: 10.1103/PhysRevA.38.3098.

[23] A. D. Becke. *J. Chem. Phys.*, 98:5648, 1993. DOI: 10.1063/1.464913.

[24] A. D. Becke. *Int. J. Quantum Chem. Symp.*, 28:625, 1994. DOI: 10.1002/qua.560520855.

[25] A. D. Becke. *J. Chem. Phys.*, 104:1040, 1996. DOI: 10.1063/1.470829.

[26] A. D. Becke. *J. Chem. Phys.*, 107:8554, 1997. DOI: 10.1063/1.475007.

[27] A. D. Becke. *J. Chem. Phys.*, 122:064101, 2005. DOI: 10.1063/1.1844493.

[28] A. D. Becke and E. R. Johnson. *J. Chem. Phys.*, 123:154101, 2005. DOI: 10.1063/1.2065267.

[29] A. D. Becke and E. R. Johnson. *J. Chem. Phys.*, 122:154104, 2005. DOI: 10.1063/1.1884601.

[30] A. D. Becke and E. R. Johnson. *J. Chem. Phys.*, 124:014104, 2006. DOI: 10.1063/1.2139668.

[31] A. D. Becke and E. R. Johnson. *J. Chem. Phys.*, 127:154108, 2007. DOI: 10.1063/1.2795701.

[32] A. D. Becke and M. R. Roussel. *Phys. Rev. A*, 39:3761, 1989. DOI: 10.1103/PhysRevA.39.3761.

[33] T. Benighaus, R. A. DiStasio, Jr., R. C. Lochan, J.-D. Chai, and M. Head-Gordon. *J. Phys. Chem. A*, 112:2702, 2008. DOI: 10.1021/jp710439w.

[34] Y. A. Bernard, Y. Shao, and A. I. Krylov. *J. Chem. Phys.*, 136:204103, 2012. DOI: 10.1063/1.4714499.

[35] N. A. Besley, M. J. G. Peach, and D. J. Tozer. *Phys. Chem. Chem. Phys.*, 11:10350, 2009. DOI: 10.1039/b912718f.

[36] S. Bhandari, M. S. Cheung, E. Geva, L. Kronik, and B. D. Dunietz. *J. Chem. Theory Comput.*, 14:6287, 2019. DOI: 10.1021/acs.jctc.8b00876.

[37] M. A. Blood-Forsythe, T. Markovich, R. A. DiStasio, Jr., R. Car, and A. Aspuru-Guzik. *Chem. Sci.*, 7:1712, 2016. DOI: 10.1039/C5SC03234B.

[38] A. D. Boese and N. C. Handy. *J. Chem. Phys.*, 114:5497, 2001. DOI: 10.1063/1.1347371.

[39] A. D. Boese and N. C. Handy. *J. Chem. Phys.*, 116:9559, 2002. DOI: 10.1063/1.1476309.

[40] A. D. Boese and J. M. L. Martin. *J. Chem. Phys.*, 121:3405, 2004. DOI: 10.1063/1.1774975.

[41] A. D. Boese, N. L. Doltsinis, N. C. Handy, and M. Sprik. *J. Chem. Phys.*, 112:1670, 2000. DOI: 10.1063/1.480732.

[42] S. F. Boys and F. Bernardi. *Mol. Phys.*, 19:553, 1970. DOI: 10.1080/00268977000101561.

[43] E. Brémond and C. Adamo. *J. Chem. Phys.*, 135:024106, 2011. DOI: 10.1063/1.3604569.

[44] É. Brémond, J. C. Sancho-García, Á. J. Pérez-Jiménez, and C. Adamo. *J. Chem. Phys.*, 141:031101, 2014. DOI: 10.1063/1.4890314.

[45] D. R. Broderick and J. M. Herbert. *J. Phys. Chem. Lett.*, 16:2793, 2025. DOI: 10.1021/acs.jpclett.4c03619.

[46] E. Caldeweyher, C. Bannwarth, and S. Grimme. *J. Chem. Phys.*, 147:034112, 2017. DOI: 10.1063/1.4993215.

[47] E. Caldeweyher, S. Ehlert, A. Hansen, H. Neugebauer, S. Spicher, C. Bannwarth, and S. Grimme. *J. Chem. Phys.*, 150:154122, 2019. DOI: 10.1063/1.5090222.

[48] E. Caldeweyher, J.-M. Mewes, S. Ehlert, and S. Grimme. *Phys. Chem. Chem. Phys.*, 22:8499, 2020. DOI: 10.1039/d0cp00502a.

[49] M. E. Casida and D. R. Salahub. *J. Chem. Phys.*, 113:8918, 2000. DOI: 10.1063/1.1319649.

[50] M. E. Casida, C. Jamorski, K. C. Casida, and D. R. Salahub. *J. Chem. Phys.*, 108:4439, 1998. DOI: 10.1063/1.475855.

[51] F. Castet and B. Champagne. *J. Chem. Theory Comput.*, 8:2044, 2012. DOI: 10.1021/ct300174z.

[52] J.-D. Chai. *J. Chem. Phys.*, 136:154104, 2012. DOI: 10.1063/1.3703894.

[53] J.-D. Chai. *J. Chem. Phys.*, 140:18A521, 2014. DOI: 10.1063/1.4867532.

[54] J.-D. Chai. *J. Chem. Phys.*, 146:044102, 2017. DOI: 10.1063/1.4974163.

[55] J.-D. Chai and P.-T. Chen. *Phys. Rev. Lett.*, 110:033002, 2013. DOI: 10.1103/PhysRevLett.110.033002.

[56] J.-D. Chai and M. Head-Gordon. *J. Chem. Phys.*, 128:084106, 2008. DOI: 10.1063/1.2834918.

[57] J.-D. Chai and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 10:6615, 2008. DOI: 10.1039/b810189b.

[58] J.-D. Chai and M. Head-Gordon. *J. Chem. Phys.*, 131:174105, 2009. DOI: 10.1063/1.3244209.

[59] J.-D. Chai and S.-P. Mao. *Chem. Phys. Lett.*, 538:121, 2012. DOI: 10.1016/j.cplett.2012.04.045.

[60] C.-M. Chang, N. J. Russ, and J. Kong. *Phys. Rev. A*, 84:022504, 2011. DOI: 10.1103/PhysRevA.84.022504.

[61] S.-H. Chien and P. M. W. Gill. *J. Comput. Chem.*, 24:732, 2003. DOI: 10.1002/jcc.10211.

[62] S.-H. Chien and P. M. W. Gill. *J. Comput. Chem.*, 27:730, 2006. DOI: 10.1002/jcc.20383.

[63] A. J. Cohen, P. Mori-Sánchez, and W. Yang. *J. Chem. Phys.*, 127:034101, 2007. DOI: 10.1063/1.2749510.

[64] A. J. Cohen, P. Mori-Sánchez, and W. Yang. *J. Chem. Phys.*, 126:191109, 2007. DOI: 10.1063/1.2741248.

[65] A. J. Cohen, P. Mori-Sánchez, and W. Yang. *Science*, 321:792, 2008. DOI: 10.1126/science.1158722.

[66] A. J. Cohen, P. Mori-Sanchez, and W. Yang. *Chem. Rev.*, 112:289, 2012. DOI: 10.1021/cr200107z.

[67] L. A. Constantin, E. Fabiano, and F. Della Sala. *Phys. Rev. B*, 86, 2012. DOI: 10.1103/PhysRevB.86.035130.

[68] L. A. Constantin, E. Fabiano, and F. Della Sala. *J. Chem. Theory Comput.*, 9:2256, 2013. DOI: 10.1021/ct400148r.

[69] J. Contreras-García, E. R. Johnson, S. Keinan, B. Chaudret, J.-P. Piquemal, D. N. Beratan, and W. Yang. *J. Chem. Theory Comput.*, 7:625, 2011. DOI: 10.1021/ct100641a.

[70] M. P. Coons, Z.-Q. You, and J. M. Herbert. *J. Am. Chem. Soc.*, 138:10879, 2016. DOI: 10.1021/jacs.6b06715.

[71] G. I. Csonka, J. P. Perdew, and A. Ruzsinszky. *J. Chem. Theory Comput.*, 6:3688, 2010. DOI: 10.1021/ct100488v.

[72] S. Dasgupta and J. M. Herbert. *J. Comput. Chem.*, 38:869, 2017. DOI: 10.1002/jcc.24761.

[73] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist. *Phys. Rev. Lett.*, 92:246401, 2004. DOI: 10.1103/PhysRevLett.92.246401.

[74] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist. *Phys. Rev. Lett.*, 95:109902, 2005. DOI: 10.1103/PhysRevLett.95.109902.

[75] P. A. M. Dirac. *P. Camb. Philos. Soc.*, 26:376, 1930. DOI: 10.1017/S0305004100016108.

[76] J. F. Dobson. *Int. J. Quantum Chem.*, 114:1157, 2014. DOI: 10.1002/qua.24635.

[77] A. Dreuw, J. L. Weisman, and M. Head-Gordon. *J. Chem. Phys.*, 119:2943, 2003. DOI: 10.1063/1.1590951.

[78] J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun. *J. Phys. Chem. Lett.*, 11:8208, 2020. DOI: 10.1021/acs.jpclett.0c02405.

[79] J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun. *J. Phys. Chem. Lett.*, 11:9248, 2020. DOI: 10.1021/acs.jpclett.0c02405.

[80] J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun. *J. Chem. Phys.*, 156:034109, 2022. DOI: 10.1063/5.0073623.

[81] K. Garrett, X. A. S. Vazquez, S. B. Egri, J. Wilmer, L. E. Johnson, B. H. Robinson, and C. M. Isborn. *J. Chem. Theory Comput.*, 10:3821, 2014. DOI: 10.1021/ct500528z.

[82] P. M. W. Gill. *Mol. Phys.*, 89:433, 1996. DOI: 10.1080/002689796173813.

[83] P. M. W. Gill, B. G. Johnson, J. A. Pople, and M. J. Frisch. *Int. J. Quantum Chem. Symp.*, 26:319, 1992.

[84] P. M. W. Gill, B. G. Johnson, and J. A. Pople. *Chem. Phys. Lett.*, 209:506, 1993. DOI: 10.1016/0009-2614(93)80125-9.

[85] P. M. W. Gill, R. D. Adamson, and J. A. Pople. *Mol. Phys.*, 88:1005, 1996. DOI: 10.1080/00268979609484488.

[86] L. Goerigk and S. Grimme. *J. Chem. Theory Comput.*, 6:107, 2010. DOI: 10.1021/ct900489g.

[87] L. Goerigk and S. Grimme. *J. Chem. Theory Comput.*, 7:291, 2011. DOI: 10.1021/ct100466k.

[88] E. Goll, H.-J. Werner, and H. Stoll. *Phys. Chem. Chem. Phys.*, 7:3917, 2005. DOI: 10.1039/B509242F.

[89] E. Goll, H.-J. Werner, H. Stoll, T. Leininger, P. Gori-Giorgi, and A. Savin. *Chem. Phys.*, 329:276, 2006. DOI: 10.1016/j.chemphys.2006.05.020.

[90] M. Gray and J. M. Herbert. *J. Chem. Phys.*, 155:034103, 2021. DOI: 10.1063/5.0059364.

[91] M. Gray and J. M. Herbert. *Annu. Rep. Comput. Chem.*, 20:1, 2024. DOI: 10.1016/bs.arcc.2024.03.001.

[92] M. Gray, A. Mandal, and J. M. Herbert. *J. Phys. Chem. A*, 129:3969, 2025. DOI: 10.1021/acs.jpca.5c01402.

[93] S. Grimme. *J. Phys. Chem. A*, 109:3067, 2005. DOI: 10.1021/jp050036j.

[94] S. Grimme. *J. Chem. Phys.*, 124:034108, 2006. DOI: 10.1063/1.2148954.

[95] S. Grimme. *J. Comput. Chem.*, 27:1787, 2006. DOI: 10.1002/jcc.20495.

[96] S. Grimme. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 1:211, 2011. DOI: 10.1002/wcms.30.

[97] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg. *J. Chem. Phys.*, 132:154104, 2010. DOI: 10.1063/1.3382344.

[98] S. Grimme, S. Ehrlich, and L. Goerigk. *J. Comput. Chem.*, 32:1456, 2011. DOI: 10.1002/jcc.21759.

[99] S. Grimme, J. G. Brandenburg, C. Bannwarth, and A. Hansen. *J. Chem. Phys.*, 143:054107, 2015. DOI: 10.1063/1.4927476.

[100] S. Grimme, A. Hansen, J. G. Brandenburg, and C. Bannwarth. *Chem. Rev.*, 116:5105, 2016. DOI: 10.1021/acs.chemrev.5b00533.

[101] B. Hammer, L. B. Hansen, and J. K. Nørskov. *Phys. Rev. B*, 59:7413, 1999. DOI: 10.1103/PhysRevB.59.7413.

[102] F. A. Hamprecht, A. J. Cohen, D. J. Tozer, and N. C. Handy. *J. Chem. Phys.*, 109:6264, 1998. DOI: 10.1063/1.477267.

[103] N. C. Handy and A. J. Cohen. *Mol. Phys.*, 99:403, 2001. DOI: 10.1080/00268970010018431.

[104] M. Hapka, L. Rajchel, M. Modrzejewski, G. Chałasiǹski, and M. M. Szczęśniak. *J. Chem. Phys.*, 141:134120, 2014. DOI: 10.1063/1.4896608.

[105] T. M. Henderson, B. G. Janesko, and G. E. Scuseria. *J. Chem. Phys.*, 128:194105, 2008. DOI: 10.1063/1.2921797.

[106] J. M. Herbert and K. Carter-Fenk. *J. Phys. Chem. A*, 125:1243, 2021. DOI: 10.1021/acs.jpca.0c11356.

[107] J. M. Herbert and S. K. Paul. *Molecules*, 26:6719, 2021. DOI: 10.3390/molecules26216719.

[108] J. Hermann, R. A. DiStasio Jr., and A. Tkatchanko. *Chem. Rev.*, 117:4714, 2017. DOI: 10.1021/acs.chemrev.6b00446.

[109] J. Hermann, M. Stoehr, S. Goger, S. Chaundhuri, B. Aradi, R. J. Maurer, and A. Tkatchenko. *J. Chem. Phys.*, 159:174802, 2023. DOI: 10.1063/5.0170972.

[110] J. Heyd, G. E. Scuseria, and M. Ernzerhof. *J. Chem. Phys.*, 118:8207, 2003. DOI: 10.1063/1.1564060.

[111] S. Hirata and M. Head-Gordon. *Chem. Phys. Lett.*, 314:291, 1999. DOI: 10.1016/S0009-2614(99)01149-5.

[112] W.-M. Hoe, A. J. Cohen, and N. C. Handy. *Chem. Phys. Lett.*, 341:319, 2001. DOI: 10.1016/S0009-2614(01)00581-4.

[113] P. Hohenberg and W. Kohn. *Phys. Rev. B*, 136:864, 1964. DOI: 10.1103/PhysRev.136.B864.

[114] K. Hui and J.-D. Chai. *J. Chem. Phys.*, 144:044114, 2016. DOI: 10.1063/1.4940734.

[115] H. Iikura, T. Tsuneda, T. Yanai, and K. Hirao. *J. Chem. Phys.*, 115:3540, 2001. DOI: 10.1063/1.1383587.

[116] S. Jana, K. Sharma, and P. Samal. *J. Phys. Chem. A*, 123:6356, 2019. DOI: 10.1021/acs.jpca.9b02921.

[117] S. Jana, S. K. Behera, S. Smiga, L. A. Constantin, and P. Samal. *J. Chem. Phys.*, 155:024103, 2021. DOI: 10.1063/5.0051331.

[118] B. G. Janesko. *Chem. Soc. Rev.*, 50:8470, 2021. DOI: 10.1039/d0cs01074j.

[119] B. G. Janesko and G. E. Scuseria. *J. Chem. Phys.*, 128:244112, 2008. DOI: 10.1063/1.2940738.

[120] H. Ji, Y. Shao, W. A. Goddard, and Y. Jung. *J. Chem. Theory Comput.*, 9:1971, 2013. DOI: 10.1021/ct400050d.

[121] Y. Jin and R. J. Bartlett. *J. Chem. Phys.*, 145:034107, 2016. DOI: 10.1063/1.4955497.

[122] B. G. Johnson, P. M. W. Gill, and J. A. Pople. *Chem. Phys. Lett.*, 220:377, 1994. DOI: 10.1016/0009-2614(94)00199-5.

[123] E. R. Johnson and A. D. Becke. *J. Chem. Phys.*, 123:024101, 2005. DOI: 10.1063/1.1949201.

[124] E. R. Johnson and A. D. Becke. *J. Chem. Phys.*, 124:174104, 2006. DOI: 10.1063/1.2190220.

[125] A. Karolewski, L. Kronik, and S. Kümmel. *J. Chem. Phys.*, 138:204115, 2013. DOI: 10.1063/1.4807325.

[126] A. Karton, A. Tarnopolsky, J.-F. Lamère, G. C. Schatz, and J. M. L. Martin. *J. Phys. Chem. A*, 112:12868, 2008. DOI: 10.1021/jp801805p.

[127] T. W. Keal and D. J. Tozer. *J. Chem. Phys.*, 119:3015, 2003. DOI: 10.1063/1.1590634.

[128] T. W. Keal and D. J. Tozer. *J. Chem. Phys.*, 121:5654, 2004. DOI: 10.1063/1.1784777.

[129] T. W. Keal and D. J. Tozer. *J. Chem. Phys.*, 123:121103, 2005. DOI: 10.1063/1.2061227.

[130] J. Kim and Y. Jung. *J. Chem. Theory Comput.*, 11:45, 2015. DOI: 10.1021/ct500660k.

[131] M.-C. Kim, E. Sim, and K. Burke. *J. Chem. Phys.*, 140:18A528, 2014.

[132] J. Klimeš, D. R. Bowler, and A. Michaelides. *J. Phys. Condens. Matter*, 22:022201, 2010. DOI: 10.1088/0953-8984/22/2/022201.

[133] W. Kohn and L. J. Sham. *Phys. Rev. A*, 140:1133, 1965. DOI: 10.1103/PhysRev.140.A1133.

[134] W. Kohn, A. D. Becke, and R. G. Parr. *J. Phys. Chem.*, 100:12974, 1996. DOI: 10.1021/jp960669l.

[135] J. Kong, S. T. Brown, and L. Fusti-Molnar. *J. Chem. Phys.*, 124:094109, 2006. DOI: 10.1063/1.2173244.

[136] J. Kong, Z. Gan, E. Proynov, M. Freindorf, and T. Furlani. *Phys. Rev. A*, 79:042510, 2009. DOI: 10.1103/PhysRevA.79.042510.

[137] T. Körzdörfer, J. S. Sears, C. Sutton, and J.-L. Brédas. *J. Chem. Phys.*, 135:204107, 2011. DOI: 10.1063/1.3663856.

[138] S. Kozuch and J. M. L. Martin. *J. Comput. Chem.*, 34:2327, 2013. DOI: 10.1002/jcc.23391.

[139] L. Kronik and S. Kümmel. *Adv. Mater.*, 30:1706560, 2018. DOI: 10.1002/adma.201706560.

[140] L. Kronik, T. Stein, S. Refaely-Abramson, and R. Baer. *J. Chem. Theory Comput.*, 8:1515, 2012. DOI: 10.1021/ct2009363.

[141] A. V. Krukau, O. A. Vydrov, A. F. Izmaylov, and G. E. Scuseria. *J. Chem. Phys.*, 125:224106, 2006. DOI: 10.1063/1.2404663.

[142] H. Kruse and S. Grimme. *J. Chem. Phys.*, 136:154101, 2012. DOI: 10.1063/1.3700154.

[143] B. B. Laird, R. B. Ross, and T. Ziegler, editors. *Density-Functional Methods in Chemistry: An Overview*, volume 629 of *ACS Symposium Series*. American Chemical Society, Washington, D.C., 1996. DOI: 10.1021/bk-1996-0629.ch001.

[144] A. Lange and J. M. Herbert. *J. Chem. Theory Comput.*, 3:1680, 2007. DOI: 10.1021/ct700125v.

[145] A. W. Lange and J. M. Herbert. *J. Am. Chem. Soc.*, 131:124115, 2009. DOI: 10.1021/ja808998q.

[146] A. W. Lange, M. A. Rohrdanz, and J. M. Herbert. *J. Phys. Chem. B*, 112:6304, 2008. DOI: 10.1021/jp802058k.

[147] K. U. Lao and J. M. Herbert. *J. Chem. Theory Comput.*, 14:2955, 2018. DOI: 10.1021/acs.jctc.8b00058.

[148] V. I. Lebedev. *Zh. Vychisl. Mat. Mat. Fix.*, 15:48, 1975. DOI: 10.1016/0041-5553(75)90133-0.

[149] V. I. Lebedev. *Zh. Vychisl. Mat. Mat. Fix.*, 16:293, 1976. DOI: 10.1016/0041-5553(76)90100-2.

[150] V. I. Lebedev. *Sibirsk. Mat. Zh.*, 18:132, 1977.

[151] V. I. Lebedev and D. N. Laikov. *Dokl. Math.*, 366:741, 1999.

[152] C. Lee, W. Yang, and R. G. Parr. *Phys. Rev. B*, 37:785, 1988. DOI: 10.1103/PhysRevB.37.785.

[153] K. Lee, É. D. Murray, L. Kong, B. I. Lundqvist, and D. C. Langreth. *Phys. Rev. B*, 82:081101(R), 2010. DOI: 10.1103/PhysRevB.82.081101.

[154] M. Levy and J. P. Perdew. *Phys. Rev. A*, 32:2010, 1985. DOI: 10.1103/PhysRevA.32.2010.

[155] C. Y. Lin, M. W. George, and P. M. W. Gill. *Aust. J. Chem.*, 57:365, 2004. DOI: 10.1071/CH03263.

[156] Y.-S. Lin, C.-W. Tsai, G.-D. Li, and J.-D. Chai. *J. Chem. Phys.*, 136:154109, 2012. DOI: 10.1063/1.4704370.

[157] Y.-S. Lin, G.-D. Li, S.-P. Mao, and J.-D. Chai. *J. Chem. Theory Comput.*, 9:263, 2013. DOI: 10.1021/ct300715s.

[158] F. Liu, E. Proynov, J.-G. Yu, T. R. Furlani, and J. Kong. *J. Chem. Phys.*, 137:114104, 2012. DOI: 10.1063/1.4752396.

[159] K.-Y. Liu, J. Liu, and J. M. Herbert. *J. Comput. Chem.*, 38:1678, 2017. DOI: 10.1002/jcc.24811.

[160] S. Liu and R. G. Parr. *J. Mol. Struct. (Theochem)*, 501:29, 2000. DOI: 10.1016/S0166-1280(99)00410-8.

[161] E. Livshits and R. Baer. *Phys. Chem. Chem. Phys.*, 9:2932, 2007. DOI: 10.1039/b617919c.

[162] P.-F. Loos. *J. Chem. Phys.*, 146:114108, 2017. DOI: 10.1063/1.4978409.

[163] B. J. Lynch and D. G. Truhlar. *J. Phys. Chem. A*, 105:2936, 2001. DOI: 10.1021/jp004262z.

[164] B. J. Lynch, P. L. Fast, M. Harris, and D. G. Truhlar. *J. Phys. Chem. A*, 104:4811, 2000. DOI: 10.1021/jp000497z.

[165] A. Mandal and J. M. Herbert. *J. Phys. Chem. Lett.*, 16:2672, 2025. DOI: 10.1021/acs.jpclett.5c00086.

[166] Y. A. Mantz, F. L. Gervasio, T. Laino, and M. Parrinello. *J. Phys. Chem. A*, 111:105, 2007. DOI: 10.1021/jp063080n.

[167] N. Mardirossian and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 16:9904, 2014. DOI: 10.1039/c3cp54374a.

[168] N. Mardirossian and M. Head-Gordon. *J. Chem. Phys.*, 142:074111, 2015. DOI: 10.1063/1.4907719.

[169] N. Mardirossian and M. Head-Gordon. *J. Chem. Phys.*, 144:214110, 2016. DOI: 10.1063/1.4952647.

[170] N. Mardirossian and M. Head-Gordon. *J. Chem. Phys.*, 148:241736, 2018. DOI: 10.1063/1.5025226.

[171] N. Mardirossian, L. R. Pestana, J. C. Womack, C.-K. Skylaris, T. Head-Gordon, and M. Head-Gordon. *J. Phys. Chem. Lett.*, 8:35, 2017. DOI: 10.1021/acs.jpclett.6b02527.

[172] P. D. Mezei, G. I. Csonka, and M. Kállay. *J. Chem. Theory Comput.*, 14:2469, 2018. DOI: 10.1021/acs.jctc.8b00072.

[173] M. Mitani. *Theor. Chem. Acc.*, 130:645, 2011. DOI: 10.1007/s00214-011-0985-x.

[174] M. Mitani and Y. Yoshioka. *Theor. Chem. Acc.*, 131:1169, 2012. DOI: 10.1007/s00214-012-1169-z.

[175] M. Modrzejewski, L. Rajchel, G. Chalasinski, and M. M. Szczesniak. *J. Phys. Chem. A*, 117:11580, 2013. DOI: 10.1021/jp4088404.

[176] C. Møller and M. S. Plesset. *Phys. Rev.*, 46:618, 1934. DOI: 10.1103/PhysRev.46.618.

[177] P. Mori-Sánchez and A. J. Cohen. *Phys. Chem. Chem. Phys.*, 16:14378, 2014. DOI: 10.1039/C4CP01170H.

[178] P. Mori-Sánchez, A. J. Cohen, and W. Yang. *J. Chem. Phys.*, 124:091102, 2006. DOI: 10.1063/1.2179072.

[179] C. W. Murray, N. C. Handy, and G. J. Laming. *Mol. Phys.*, 78:997, 1993. DOI: 10.1080/00268979300100651.

[180] É. D. Murray, K. Lee, and D. C. Langreth. *J. Chem. Theory Comput.*, 5:2754, 2009. DOI: 10.1021/ct900365q.

[181] B. Neupane, H. Tang, N. K. Nepal, S. Adhikari, and A. Ruzsinszky. *Phys. Rev. Materials*, 5:063803, 2021. DOI: 10.1103/PhysRevMaterials.5.063803.

[182] N. Oliphant and R. Bartlett. *J. Chem. Phys.*, 100:6550, 1994. DOI: 10.1063/1.467064.

[183] A. Otero-de-la-Roza and E. R. Johnson. *J. Chem. Phys.*, 138:204109, 2013. DOI: 10.1063/1.4807330.

[184] M. B. Oviedo, N. V. Ilawe, and B. M. Wong. *J. Chem. Theory Comput.*, 12:3593, 2016. DOI: 10.1021/acs.jctc.6b00360.

[185] L. F. Pacios. *J. Comput. Chem.*, 16:133, 1995. DOI: 10.1002/jcc.540160202.

[186] C.-R. Pan, P.-T. Fang, and J.-D. Chai. *Phys. Rev. A*, 87:052510, 2013. DOI: 10.1103/PhysRevA.87.052510.

[187] R. G. Parr and W. Yang. *Density-Functional Theory of Atoms and Molecules*. Oxford University Press, New York, 1989.

[188] A. Patra, S. Jana, and P. Samal. *J. Chem. Phys.*, 153:184112, 2020. DOI: 10.1063/5.0025173.

[189] S. Paziani, S. Moroni, P. Gori-Giorgi, and G. B. Bachelet. *Phys. Rev. B*, 73:155111, 2006. DOI: 10.1103/PhysRevB.73.155111.

[190] J. P. Perdew. *Phys. Rev. B*, 33:8822, 1986. DOI: 10.1103/PhysRevB.33.8822.

[191] J. P. Perdew and M. Levy. *Phys. Rev. B*, 56:16021, 1997. DOI: 10.1103/PhysRevB.56.16021.

[192] J. P. Perdew and K. Schmidt. Jacob's ladder of density functional approximations for the exchange-correlation energy. In V. Van Doren, C. Van Alsenoy, and P. Geerlings, editors, *Density Functional Theory and Its Applications to Materials*, volume 577 of *AIP Conference Proceedings*, pages 1–20. American Institute of Physics, 2001. DOI: 10.1063/1.1390175.

[193] J. P. Perdew and Y. Wang. *Phys. Rev. B*, 33:8800, 1986. DOI: 10.1103/PhysRevB.33.8800.

[194] J. P. Perdew and Y. Wang. *Phys. Rev. B*, 45:13244, 1992. DOI: 10.1103/PhysRevB.45.13244.

[195] J. P. Perdew and A. Zunger. *Phys. Rev. B*, 23:5048, 1981. DOI: 10.1103/PhysRevB.23.5048.

[196] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais. *Phys. Rev. B*, 46:6671, 1992. DOI: 10.1103/PhysRevB.46.6671.

[197] J. P. Perdew, K. Burke, and M. Ernzerhof. *Phys. Rev. Lett.*, 77:3865, 1996. DOI: 10.1103/PhysRevLett.77.3865.

[198] J. P. Perdew, S. Kurth, A. Zupan, and P. Blaha. *Phys. Rev. Lett.*, 82:2544, 1999. DOI: 10.1103/PhysRevLett.82.2544.

[199] J. P. Perdew, A. Ruzsinszky, J. Tao, V. N. Staroverov, G. E. Scuseria, and G. I. Csonka. *J. Chem. Phys.*, 123: 062201, 2005. DOI: 10.1063/1.1904565.

[200] J. P. Perdew, A. Ruzsinszky, J. Tao, G. I. Csonka, and G. E. Scuseria. *Phys. Rev. A*, 76:042506, 2007. DOI: 10.1103/PhysRevA.76.042506.

[201] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke. *Phys. Rev. Lett.*, 100:136406, 2008. DOI: 10.1103/PhysRevLett.100.136406.

[202] J. P. Perdew, V. N. Staroverov, J. Tao, and G. E. Scuseria. *Phys. Rev. A*, 78:052513, 2008. DOI: 10.1103/PhysRevA.78.052513.

[203] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin, and J. Sun. *Phys. Rev. Lett.*, 103:026403, 2009. DOI: 10.1103/PhysRevLett.103.026403.

[204] J. P. Perdew, A. Ruzsinszky, J. Sun, and M. R. Pederson. *Adv. Atom. Mol. Opt. Phys.*, 64:1, 2015.

[205] K. Pernal, R. Podeszwa, K. Patkowski, and K. Szalewicz. *Phys. Rev. Lett.*, 103:263201, 2009. DOI: 10.1103/PhysRevLett.103.263201.

[206] R. Peverati and D. G. Truhlar. *J. Chem. Phys.*, 135:191102, 2011. DOI: 10.1063/1.3663871.

[207] R. Peverati and D. G. Truhlar. *J. Phys. Chem. Lett.*, 2:2810, 2011. DOI: 10.1021/jz201170d.

[208] R. Peverati and D. G. Truhlar. *J. Phys. Chem. Lett.*, 3:117, 2012. DOI: 10.1021/jz201525m.

[209] R. Peverati and D. G. Truhlar. *J. Chem. Theory Comput.*, 8:2310, 2012. DOI: 10.1021/ct3002656.

[210] R. Peverati and D. G. Truhlar. *Phys. Chem. Chem. Phys.*, 14:13171, 2012. DOI: 10.1039/c2cp42025b.

[211] R. Peverati and D. G. Truhlar. *Phys. Chem. Chem. Phys.*, 14:16187, 2012. DOI: 10.1039/c2cp42576a.

[212] R. Peverati, Y. Zhao, and D. G. Truhlar. *J. Phys. Chem. Lett.*, 2:1991, 2011. DOI: 10.1021/jz200616w.

[213] J. A. Pople, P. M. W. Gill, and B. G. Johnson. *Chem. Phys. Lett.*, 199:557, 1992. DOI: 10.1016/0009-2614(92)85009-Y.

[214] E. Proynov and J. Kong. *J. Chem. Theory Comput.*, 3:746, 2007. DOI: 10.1021/ct600372t.

[215] E. Proynov and J. Kong. In G. Vaysilov and T. Mineva, editors, *Theoretical Aspects of Catalysis*, page 453. Heron Press, Birmingham, 2008.

[216] E. Proynov and J. Kong. *Phys. Rev. A*, 79:014103, 2009. DOI: 10.1103/PhysRevA.79.014103.

[217] E. Proynov, Y. Shao, and J. Kong. *Chem. Phys. Lett.*, 493:381, 2010. DOI: 10.1016/j.cplett.2010.05.029.

[218] E. Proynov, F. Liu, and J. Kong. *Chem. Phys. Lett.*, 525:150, 2012. DOI: 10.1016/j.cplett.2011.12.069.

[219] E. Proynov, F. Liu, Y. Shao, and J. Kong. *J. Chem. Phys.*, 136:034102, 2012. DOI: 10.1063/1.3676726.

[220] B. Rana and J. M. Herbert. *J. Phys. Chem. Lett.*, 12:8053, 2021. DOI: 10.1021/acs.jpclett.1c02283.

[221] B. Rana, M. P. Coons, and J. M. Herbert. *J. Phys. Chem. Lett.*, 13:5275, 2022. DOI: 10.1021/acs.jpclett.2c01187.

[222] B. Rana, G. J. O. Beran, and J. M. Herbert. *Mol. Phys.*, 121:e2138789, 2023. DOI: 10.1080/00268976.2022.2138789.

[223] R. M. Richard and J. M. Herbert. *J. Chem. Theory Comput.*, 7:1296, 2011. DOI: 10.1021/ct100607w.

[224] M. A. Rohrdanz and J. M. Herbert. *J. Chem. Phys.*, 129:034107, 2008. DOI: 10.1063/1.2954017.

[225] M. A. Rohrdanz, K. M. Martins, and J. M. Herbert. *J. Chem. Phys.*, 130:054112, 2009. DOI: 10.1063/1.3073302.

[226] I. Rudra, Q. Wu, and T. Van Voorhis. *J. Chem. Phys.*, 124:024103, 2006. DOI: 10.1063/1.2145878.

[227] N. J. Russ, C.-M. Chang, and J. Kong. *Can. J. Chem.*, 89:657, 2011. DOI: 10.1139/v11-063.

[228] A. Ruzsinszky, J. P. Perdew, G. I. Csonka, O. A. Vydrov, and G. E. Scuseria. *J. Chem. Phys.*, 125:194112, 2006. DOI: 10.1063/1.2387954.

[229] A. Ruzsinszky, J. Sun, B. Xiao, and G. Csonka. *J. Chem. Theory Comput.*, 8:2078, 2012. DOI: 10.1021/ct300269u.

[230] R. Sabatini, T. Gorni, and S. de Gironcoli. *Phys. Rev. B*, 87:041108, 2013. DOI: 10.1103/PhysRevB.87.041108.

[231] U. Salzner and R. Baer. *J. Chem. Phys.*, 131:231101, 2009. DOI: 10.1063/1.3269030.

[232] H. Schröder, A. Creon, and T. Schwabe. *J. Chem. Theory Comput.*, 11:3163, 2015. DOI: 10.1021/acs.jctc.5b00400.

[233] T. Schwabe and S. Grimme. *Phys. Chem. Chem. Phys.*, 9:3397, 2007. DOI: 10.1039/b704725h.

[234] Y. Shao, M. Head-Gordon, and A. I. Krylov. *J. Chem. Phys.*, 118:4807, 2003. DOI: 10.1063/1.1545679.

[235] E. Sim, S. Song, S. Vuckovic, and K. Burke. *J. Am. Chem. Soc.*, 144:6625, 2022. DOI: 10.1021/jacs.1c11506.

[236] J. C. Slater. *J. Chem. Phys.*, 41:3199, 1964. DOI: 10.1063/1.1725697.

[237] D. G. Smith, L. A. Burns, K. Patkowski, and C. D. Sherrill. *J. Phys. Chem. Lett.*, 7:2197, 2016. DOI: 10.1021/acs.jpclett.6b00780.

[238] J. W. Song, T. Hirosawa, T. Tsuneda, and K. Hirao. *J. Chem. Phys.*, 126:154105, 2007. DOI: 10.1063/1.2721532.

[239] S. Song, S. Vuckovic, E. Sim, and K. Burke. *J. Chem. Theory Comput.*, 18:817, 2022. DOI: 10.1021/acs.jctc.1c01045.

[240] V. N. Staroverov, G. E. Scuseria, J. Tao, and J. P. Perdew. *J. Chem. Phys.*, 119:12129, 2003. DOI: 10.1063/1.1626543.

[241] P. J. Stephens, F. J. Devlin, C. F. Chabolowski, and M. J. Frisch. *J. Phys. Chem.*, 98:11623, 1994. DOI: 10.1021/j100096a001.

[242] P. A. Stewart and P. M. W. Gill. *J. Chem. Soc. Faraday Trans.*, 91:4337, 1995. DOI: 10.1039/FT9959104337.

[243] J. Sun, B. Xiao, and A. Ruzsinszky. *J. Chem. Phys.*, 137:051101, 2012. DOI: 10.1063/1.4742312.

[244] J. Sun, R. Haunschild, B. Xiao, I. W. Bulik, G. E. Scuseria, and J. P. Perdew. *J. Chem. Phys.*, 138:044113, 2013. DOI: 10.1063/1.4789414.

[245] J. Sun, J. P. Perdew, and A. Ruzsinszky. *Proc. Natl. Acad. Sci. USA*, 112:685, 2015. DOI: 10.1073/pnas.1423145112.

[246] J. Sun, A. Ruzsinszky, and J. P. Perdew. *Phys. Rev. Lett.*, 115:036402, 2015. DOI: 10.1103/PhysRevLett.115.036402.

[247] R. Sure and S. Grimme. *J. Comput. Chem.*, 34:1672, 2013. DOI: 10.1002/jcc.23317.

[248] J. Tao and Y. Mo. *Phys. Rev. Lett.*, 117:073001, 2016. DOI: 10.1103/PhysRevLett.117.073001.

[249] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria. *Phys. Rev. Lett.*, 91:146401, 2003. DOI: 10.1103/PhysRevLett.91.146401.

[250] A. Tarnopolsky, A. Karton, R. Sertchook, D. Vuzman, and J. M. L. Martin. *J. Phys. Chem. A*, 112:3, 2008. DOI: 10.1021/jp710179r.

[251] A. Tkatchenko and M. Scheffler. *Phys. Rev. Lett.*, 102:073005, 2009. DOI: 10.1103/PhysRevLett.102.073005.

[252] A. Tkatchenko, R. A. DiStasio, Jr., R. Car, and M. Scheffler. *Phys. Rev. Lett.*, 108:236402, 2012. DOI: 10.1103/PhysRevLett.108.236402.

[253] N. V. Tkatchenko and M. Head-Gordon. *J. Chem. Theory Comput.*, 20:9741, 2024. DOI: 10.1021/acs.jctc.4c01105.

[254] J. Toulouse, A. Savin, and H.-J. Flad. *Int. J. Quantum Chem.*, 100:1047, 2004. DOI: 10.1002/qua.20259.

[255] J. Toulouse, K. Sharkas., É. Brémond, and C. Adamo. *J. Chem. Phys.*, 135:101102, 2011. DOI: 10.1063/1.3640019.

[256] T. Tsuneda, T. Suzumura, and K. Hirao. *J. Chem. Phys.*, 110:10664, 1999. DOI: 10.1063/1.479012.

[257] F. Uhlig, J. M. Herbert, M. P. Coons, and P. Jungwirth. *J. Phys. Chem. A*, 118:7507, 2014. DOI: 10.1021/jp5004243.

[258] S. J. A. van Gisbergen, V. P. Osinga, O. V. Gritsenko, R. van Leeuwen, J. G. Snijders, and E. J. Baerends. *J. Chem. Phys.*, 105:3142, 1996. DOI: 10.1063/1.472182.

[259] R. van Leeuwen and E. J. Baerends. *Phys. Rev. A*, 49:2421, 1994. DOI: 10.1103/PhysRevA.49.2421.

[260] T. Van Voorhis and G. E. Scuseria. *J. Chem. Phys.*, 109:400, 1998. DOI: 10.1063/1.476577.

[261] X. A. S. Vazquez and C. M. Isborn. *J. Chem. Phys.*, 143:244105, 2015. DOI: 10.1063/1.4937417.

[262] P. Verma and R. J. Bartlett. *J. Chem. Phys.*, 140:18A534, 2014. DOI: 10.1063/1.4871409.

[263] P. Verma and D. G. Truhlar. *J. Phys. Chem. Lett.*, 8:380, 2017. DOI: 10.1021/acs.jpclett.6b02757.

[264] P. Verma, Y. Wang, S. Ghost, X. He, and D. G. Truhlar. *J. Phys. Chem. A*, 123:2966, 2019. DOI: 10.1021/acs.jpca.8b11499.

[265] S. H. Vosko, L. Wilk, and M. Nusair. *Can. J. Phys.*, 58:1200, 1980. DOI: 10.1139/p80-159.

[266] J. Řezáč and A. de la Lande. *J. Chem. Theory Comput.*, 11:528, 2015. DOI: 10.1021/ct501115m.

[267] J. Řezáč, K. E. Riley, and P. Hobza. *J. Chem. Theory Comput.*, 7:2427, 2011. DOI: 10.1021/ct2002946.

[268] S. Vuckovic, S. Song, J. Kozlowski, E. Sim, and K. Burke. *J. Chem. Theory Comput.*, 15:6336, 2019. DOI: 10.1021/acs.jctc.9b00826.

[269] O. A. Vydrov and T. Van Voorhis. *Phys. Rev. Lett.*, 103:063004, 2009. DOI: 10.1103/PhysRevLett.103.063004.

[270] O. A. Vydrov and T. Van Voorhis. *J. Chem. Phys.*, 132:164113, 2010. DOI: 10.1063/1.3398840.

[271] O. A. Vydrov and T. Van Voorhis. *J. Chem. Phys.*, 133:244103, 2010. DOI: 10.1063/1.3521275.

[272] O. A. Vydrov, Q. Wu, and T. Van Voorhis. *J. Chem. Phys.*, 129:014106, 2008. DOI: 10.1063/1.2948400.

[273] Y. Wang, X. Jin, H. S. Yu, D. G. Truhlar, and X. He. *Proc. Natl. Acad. Sci. USA*, page 201705670, 2017. DOI: 10.1073/pnas.1705670114.

[274] Y. Wang, P. Verma, X. Jin, D. G. Truhlar, and X. He. *Proc. Natl. Acad. Sci. USA*, 115:10257, 2018. DOI: 10.1073/pnas.1810421115.

[275] Y. Wang, P. Verma, L. Zhang, Y. Li, Z. Liu, D. G. Truhlar, and X. He. *Proc. Natl. Acad. Sci. USA*, 117:2294, 2020. DOI: 10.1073/pnas.1913699117.

[276] E. Weintraub, T. M. Henderson, and G. E. Scuseria. *J. Chem. Theory Comput.*, 5:754, 2009. DOI: 10.1021/ct800530u.

[277] J. Wellendorff, K. T. Lundgaard, A. Møgelhøj, V. Petzold, D. D. Landis, J. K. Nørskov, T. Bligaard, and K. W. Jacobsen. *Phys. Rev. B*, 85:235149, 2012. DOI: 10.1103/PhysRevB.85.235149.

[278] J. Wellendorff, K. T. Lundgaard, K. W. Jacobsen, and T. Bligaard. *J. Chem. Phys.*, 140:144107, 2014. DOI: 10.1063/1.4870397.

[279] S. E. Wheeler and K. N. Houk. *J. Chem. Theory Comput.*, 6:395, 2010. DOI: 10.1021/ct900639j.

[280] E. P. Wigner. *Trans. Faraday Soc.*, 34:678, 1938. DOI: 10.1039/tf9383400678.

[281] K. W. Wiitala, T. R. Hoye, and C. J. Cramer. *J. Chem. Theory Comput.*, 2:1085, 2006. DOI: 10.1021/ct6001016.

[282] P. J. Wilson, T. J. Bradley, and D. J. Tozer. *J. Chem. Phys.*, 115:9233, 2001. DOI: 10.1063/1.1412605.

[283] J. Witte, N. Mardirossian, J. B. Neaton, and M. Head-Gordon. *J. Chem. Theory Comput.*, 13:2043, 2017. DOI: 10.1021/acs.jctc.7b00176.

[284] J. Witte, J. B. Neaton, and M. Head-Gordon. *J. Chem. Phys.*, 146:234105, 2017. DOI: 10.1063/1.4986962.

[285] Q. Wu and T. Van Voorhis. *Phys. Rev. A*, 72:024502, 2005. DOI: 10.1103/PhysRevA.72.024502.

[286] Q. Wu and T. Van Voorhis. *J. Phys. Chem. A*, 110:9212, 2006. DOI: 10.1021/jp061848y.

[287] Q. Wu and T. Van Voorhis. *J. Chem. Theory Comput.*, 2:765, 2006. DOI: 10.1021/ct0503163.

[288] Q. Wu and T. Van Voorhis. *J. Chem. Phys.*, 125:164105, 2006. DOI: 10.1063/1.2360263.

[289] Q. Wu, P. W. Ayers, and W. Yang. *J. Chem. Phys.*, 119:2978, 2003. DOI: 10.1063/1.1590631.

[290] Q. Wu, C. L. Cheng, and T. Van Voorhis. *J. Chem. Phys.*, 127:164119, 2007. DOI: 10.1063/1.2800022.

[291] Q. Wu, B. Kaduk, and T. Van Voorhis. *J. Chem. Phys.*, 130:034109, 2009. DOI: 10.1063/1.3059784.

[292] X. Xu and W. A. Goddard III. *Proc. Natl. Acad. Sci. USA*, 101:2673, 2004. DOI: 10.1073/pnas.0308730100.

[293] T. Yanai, D. P. Tew, and N. C. Handy. *Chem. Phys. Lett.*, 393:51, 2004. DOI: 10.1016/j.cplett.2004.06.011.

[294] H. S. Yu, W. Zhang, P. Verma, X. He, and D. G. Truhlar. *Phys. Chem. Chem. Phys.*, 17:12146, 2015. DOI: 10.1039/C5CP01425E.

[295] H. S. Yu, X. He, S. L. Li, and D. G. Truhlar. *Chem. Sci.*, 7:5032, 2016. DOI: 10.1039/C6SC00705H.

[296] H. S. Yu, X. He, and D. G. Truhlar. *J. Chem. Theory Comput.*, 12:1280, 2016. DOI: 10.1021/acs.jctc.5b01082.

[297] I. Y. Zhang, X. Xin, Y. Jung, and W. A. Goddard III. *Proc. Natl. Acad. Sci. USA*, 108:19896, 2011. DOI: 10.1073/pnas.1115123108.

[298] Y. Zhang and W. Yang. *Phys. Rev. Lett.*, 80:890, 1998. DOI: 10.1103/PhysRevLett.80.890.

[299] Y. Zhang and W. Yang. *J. Chem. Phys.*, 109:2604, 1998. DOI: 10.1063/1.476859.

[300] Y. Zhang, X. Xu, and W. A. Goddard III. *Proc. Natl. Acad. Sci. USA*, 106:4963, 2009. DOI: 10.1073/pnas.0901093106.

[301] Y. Zhao and D. G. Truhlar. *J. Phys. Chem. A*, 108:6908, 2004. DOI: 10.1021/jp048147q.

[302] Y. Zhao and D. G. Truhlar. *J. Phys. Chem. A*, 109:5656, 2005. DOI: 10.1021/jp050536c.

[303] Y. Zhao and D. G. Truhlar. *J. Chem. Phys.*, 125:194101, 2006. DOI: 10.1063/1.2370993.

[304] Y. Zhao and D. G. Truhlar. *J. Phys. Chem. A*, 110:13126, 2006. DOI: 10.1021/jp066479k.

[305] Y. Zhao and D. G. Truhlar. *J. Chem. Phys.*, 128:184109, 2006. DOI: 10.1063/1.2912068.

[306] Y. Zhao and D. G. Truhlar. *J. Chem. Theory Comput.*, 4:1849, 2007. DOI: 10.1021/ct800246v.

[307] Y. Zhao and D. G. Truhlar. *Theor. Chem. Acc.*, 120:215, 2008. DOI: 10.1007/s00214-007-0310-x.

[308] Y. Zhao and D. G. Truhlar. *Chem. Phys. Lett.*, 502:1, 2011. DOI: 10.1016/j.cplett.2010.11.060.

[309] Y. Zhao, B. J. Lynch, and D. G. Truhlar. *J. Phys. Chem. A*, 108:2715, 2004. DOI: 10.1021/jp049908s.

[310] Y. Zhao, N. E. Schultz, and D. G. Truhlar. *J. Chem. Phys.*, 123:161103, 2005. DOI: 10.1063/1.2126975.

[311] Y. Zhao, N. E. Schultz, and D. G. Truhlar. *J. Chem. Theory Comput.*, 2:364, 2006. DOI: 10.1021/ct0502763.

[312] T. Ziegler. *Chem. Rev.*, 91:651, 1991. DOI: 10.1021/cr00005a001.

Chapter 6

Wave Function-Based Correlation Methods

6.1 Introduction

The Hartree-Fock procedure, while often qualitatively correct, is frequently quantitatively deficient. The deficiency is due to the underlying assumption of the Hartree-Fock approximation: that electrons move *independently* within molecular orbitals subject to an averaged field imposed by the remaining electrons. The error that this introduces is called the correlation energy and a wide variety of procedures exist for estimating its magnitude. The purpose of this Chapter is to introduce the main wave function-based methods available in Q-CHEM to describe electron correlation.

Wave function-based electron correlation methods concentrate on the design of corrections to the wave function beyond the mean-field Hartree-Fock description. This is to be contrasted with the density functional theory methods discussed in the previous Chapter. While density functional methods yield a description of electronic structure that accounts for electron correlation subject only to the limitations of present-day functionals (which, for example, omit dispersion interactions), DFT cannot be systematically improved if the results are deficient. Wave function-based approaches for describing electron correlation[4,5] offer this main advantage. Their main disadvantage is relatively high computational cost, particularly for the higher-level theories.

There are four broad classes of models for describing electron correlation that are supported within Q-CHEM. The first three directly approximate the full time-independent Schrödinger equation. In order of increasing accuracy, and also increasing cost, they are:

1. Perturbative treatment of pair correlations between electrons, typically capable of recovering 80% or so of the correlation energy in stable molecules.

2. Self-consistent treatment of pair correlations between electrons (most often based on coupled-cluster theory), capable of recovering on the order of 95% or so of the correlation energy.

3. Non-iterative corrections for higher than double substitutions, which can account for more than 99% of the correlation energy. They are the basis of many modern methods that are capable of yielding chemical accuracy for ground state reaction energies, as exemplified by the G2[19] and G3 methods.[20]

These methods are discussed in the following subsections.

There is also a fourth class of methods supported in Q-CHEM, which have a different objective. These active space methods aim to obtain a balanced description of electron correlation in highly correlated systems, such as diradicals, or along bond-breaking coordinates. Active space methods are discussed in Section 6.14. Finally, equation-of-motion

(EOM) methods provide tools for describing open-shell and electronically excited species. Selected configuration interaction (CI) models are also available.

In order to carry out a wave function-based electron correlation calculation using Q-CHEM, three *$rem* variables need to be set:

- BASIS to specify the basis set (see Chapter 8)
- METHOD for treating correlation
- N_FROZEN_CORE frozen core electrons (FC default, optionally FC, or n)

For wave function-based correlation methods, the default option for exchange is Hartree-Fock. If desired, correlated calculations can employ DFT orbitals, which should be set up using a pair of EXCHANGE and CORRELATION keywords. EXCHANGE should be set to a specific DFT method (see Section 6.15.2).

Additionally, for EOM or CI calculations the number of target states of each type (excited, spin-flipped, ionized, attached, *etc.*) in each irreducible representation (irrep) should be specified (see Section 7.10.17). The level of correlation of the target EOM states may be different from that used for the reference, and can be specified by EOM_CORR keyword.

The full range of ground and excited state wave function-based correlation methods available (*i.e.* the recognized options to the METHOD keyword) are as follows. Ground-state methods are also a valid option for the CORRELATION keyword.

METHOD

Specifies the level of theory, either DFT or wave function-based.

TYPE:

STRING

DEFAULT:

HF No correlation, Hartree-Fock exchange

OPTIONS:

MP2	Sections 6.3 and 6.4
RI-MP2	Section 6.6
Local_MP2	Section 6.5
RILMP2	Section 6.6.2
ATTMP2	Section 6.7
ATTRIMP2	Section 6.7
ZAPT2	A more efficient restricted open-shell MP2 method.[59]
MP3	Section 6.3
MP4SDQ	Section 6.3
MP4	Section 6.3
CCD	Section 6.12
CCD(2)	Section 6.13
CCSD	Section 6.12
CCSDT	Section 6.12.3
CC2	Section 6.12
CCSD(T)	Section 6.13
CCSD(2)	Section 6.13
CCSD(fT)	Section 6.13.3
CCSD(dT)	Section 6.13.3
QCISD	Section 6.12
QCISD(T)	Section 6.13
OD	Section 6.12
OD(T)	Section 6.13
OD(2)	Section 6.13
VOD	Section 6.14
VOD(2)	Section 6.14
QCCD	Section 6.12
QCCD(T)	
QCCD(2)	
VQCCD	Section 6.14

RECOMMENDATION:

Consult the literature for guidance.

6.2 Treatment and the Definition of Core Electrons

Treatment of core electrons is controlled by N_FROZEN_CORE. Starting from Q-CHEM v. 5.0, the core electrons are frozen by default in most post-Hartree–Fock calculations. Selected virtual orbitals can also be frozen by using N_FROZEN_VIRTUAL keyword (the default for this is zero).

The number of core electrons in an atom is relatively well-defined, and consists of certain atomic shells. (Note that

effective core potentials are available in both "small-core" and "large-core" varieties; see Section 8.10.) For example, in phosphorus the core consists of $1s$, $2s$, and $2p$ shells, for a total of ten electrons. In molecular systems, the core electrons are usually chosen as those occupying the $n/2$ lowest energy orbitals, where n is the number of core electrons in the constituent atoms. In some cases, particularly in the lower parts of the periodic table, this definition is inappropriate and can lead to significant errors in the correlation energy. Vitaly Rassolov has implemented an alternative definition of core electrons within Q-CHEM which is based on a Mulliken population analysis, and which addresses this problem.[102]

The current implementation is restricted to *n-kl* type basis sets such as 3-21 or 6-31, and related bases such as 6-31+G(d). There are essentially two cases to consider, the outermost 6G functions (or 3G in the case of the 3-21G basis set) for Na, Mg, K and Ca, and the 3d functions for the elements Ga—Kr. Whether or not these are treated as core or valence is determined by the CORE_CHARACTER *$rem*, as summarized in Table 6.2.

CORE_CHARACTER	Outermost 6G (3G) for Na, Mg, K, Ca	3d (Ga–Kr)
1	valence	valence
2	valence	core
3	core	core
4	core	valence

Table 6.1: A summary of the effects of different core definitions

N_FROZEN_CORE

Sets the number of frozen core orbitals in a post-Hartree–Fock calculation.

TYPE:

INTEGER

DEFAULT:

FC

OPTIONS:

FC Frozen Core approximation (all core orbitals frozen).

n Freeze n core orbitals (if set to 0, all electrons will be active).

RECOMMENDATION:

Correlated calculations calculations are more efficient with frozen core orbitals. Use default if possible.

Note: The default setting (N_FROZEN_CORE=FC) does not work correctly in QM/MM calculations. One should specify the number of frozen core orbitals explicitly.

N_FROZEN_VIRTUAL

Sets the number of frozen virtual orbitals in a post-Hartree–Fock calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Freeze n virtual orbitals.

RECOMMENDATION:

None

CORE_CHARACTER

Selects how the core orbitals are determined in the frozen-core approximation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use energy-based definition.

1-4 Use Mulliken-based definition (see Table 6.2 for details).

RECOMMENDATION:

Use the default, unless performing calculations on molecules with heavy elements.

PRINT_CORE_CHARACTER

Determines the print level for the CORE_CHARACTER option.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No additional output is printed.

1 Prints core characters of occupied MOs.

2 Print level 1, plus prints the core character of AOs.

RECOMMENDATION:

Use the default, unless you are uncertain about what the core character is.

6.3 Møller-Plesset Perturbation Theory

Møller-Plesset Perturbation Theory[89] is a widely used method for approximating the correlation energy of molecules. In particular, second-order Møller-Plesset perturbation theory (MP2) is one of the simplest and most useful levels of theory beyond the Hartree-Fock approximation. Conventional and local MP2 methods available in Q-CHEM are discussed in detail in Sections 6.4 and 6.5 respectively. The MP3 method is still occasionally used, while MP4 calculations are quite commonly employed as part of the G2 and G3 thermochemical methods.[19,20] In the remainder of this section, the theoretical basis of Møller-Plesset theory is reviewed.

The Hartree-Fock wave function Ψ_0 and energy E_0 are *approximate* solutions (eigenfunction and eigenvalue) to the exact Hamiltonian eigenvalue problem or Schrödinger's electronic wave equation, Eq. (4.5). The HF wave function and energy are, however, exact solutions for the Hartree-Fock Hamiltonian H_0 eigenvalue problem. If we assume that the Hartree-Fock wave function Ψ_0 and energy E_0 lie near the exact wave function Ψ and energy E, we can now write the exact Hamiltonian operator as

$$H = H_0 + \lambda V \tag{6.1}$$

where V is the small perturbation and λ is a dimensionless parameter. Expanding the exact wave function and energy in terms of the HF wave function and energy yields

$$E = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \lambda^3 E^{(3)} + \ldots \tag{6.2}$$

and

$$\Psi = \Psi_0 + \lambda \Psi^{(1)} + \lambda^2 \Psi^{(2)} + \lambda^3 \Psi^{(3)} + \ldots \tag{6.3}$$

Substituting these expansions into the Schrödinger equation and collecting terms according to powers of λ yields

$$H_0\Psi_0 = E^{(0)}\Psi_0 \tag{6.4}$$

$$H_0\Psi^{(1)} + V\Psi_0 = E^{(0)}\Psi^{(1)} + E^{(1)}\Psi_0 \tag{6.5}$$

$$H_0\Psi^{(2)} + V\Psi^{(1)} = E^{(0)}\Psi^{(2)} + E^{(1)}\Psi^{(1)} + E^{(2)}\Psi_0 \tag{6.6}$$

and so forth. Multiplying each of the above equations by Ψ_0 and integrating over all space yields the following expression for the nth-order (MPn) energy:

$$E^{(0)} = \langle\Psi_0|H_0|\Psi_0\rangle \tag{6.7}$$

$$E^{(1)} = \langle\Psi_0|V|\Psi_0\rangle \tag{6.8}$$

$$E^{(2)} = \left\langle\Psi_0|V|\Psi^{(1)}\right\rangle \tag{6.9}$$

Thus, the Hartree-Fock energy

$$E_0 = \langle\Psi_0|\, H_0 + V\,|\Psi_0\rangle \tag{6.10}$$

is simply the sum of the zeroth- and first- order energies

$$E_0 = E^{(0)} + E^{(1)} \tag{6.11}$$

The correlation energy can then be written

$$E_{\text{corr}} = E_0^{(2)} + E_0^{(3)} + E_0^{(4)} + \ldots \tag{6.12}$$

of which the first term is the MP2 energy.

It can be shown that the MP2 energy can be written (in terms of spin-orbitals) as

$$E_0^{(2)} = -\frac{1}{4}\sum_{ab}^{\text{virt}}\sum_{ij}^{\text{occ}}\frac{|\langle ab|\,|ij\rangle|^2}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} \tag{6.13}$$

where

$$\langle ab||ij\rangle = \langle ab|ij\rangle - \langle ab|ji\rangle \tag{6.14}$$

and

$$\langle ab|cd\rangle = \int\int \psi_a(\mathbf{r}_1)\psi_c(\mathbf{r}_1)\frac{1}{r_{12}}\psi_b(\mathbf{r}_2)\psi_d(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{6.15}$$

which can be written in terms of the two-electron repulsion integrals

$$\langle ab|cd\rangle = \sum_\mu\sum_\nu\sum_\lambda\sum_\sigma C_{\mu a}C_{\nu c}C_{\lambda b}C_{\sigma d}\,(\mu\nu|\lambda\sigma) \tag{6.16}$$

Expressions for higher order terms follow similarly, although with much greater algebraic and computational complexity. MP3 and particularly MP4 (the third and fourth order contributions to the correlation energy) are both occasionally used, although they are increasingly supplanted by the coupled-cluster methods described in the following sections. The disk and memory requirements for MP3 are similar to the self-consistent pair correlation methods discussed in Section 6.12 while the computational cost of MP4 is similar to the (T) corrections discussed in Section 6.13.

6.4 Exact MP2 Methods

6.4.1 Algorithm

Second-order Møller-Plesset theory[89] (MP2) is probably the simplest useful wave function-based electron correlation method. Revived in the mid-1970s, it remains highly popular today, because it offers systematic improvement in optimized geometries and other molecular properties relative to Hartree-Fock (HF) theory.[55] Indeed, in a recent comparative study of small closed-shell molecules,[56] MP2 outperformed much more expensive singles and doubles coupled-cluster theory for such properties! Relative to state-of-the-art Kohn-Sham density functional theory (DFT) methods, which are the most economical methods to account for electron correlation effects, MP2 has the advantage of properly incorporating long-range dispersion forces. The principal weaknesses of MP2 theory are for open shell systems, and other cases where the HF determinant is a poor starting point.

Q-CHEM contains an efficient conventional semi-direct method to evaluate the MP2 energy and gradient.[52] These methods require OVN memory (O, V, N are the numbers of occupied, virtual and total orbitals, respectively), and disk space which is bounded from above by $OVN^2/2$. The latter can be reduced to $IVN^2/2$ by treating the occupied orbitals in batches of size I, and re-evaluating the two-electron integrals O/I times. This approach is tractable on modern workstations for energy and gradient calculations of at least 500 basis functions or so, or molecules of between 15 and 30 first row atoms, depending on the basis set size. The computational cost increases between the 3rd and 5th power of the size of the molecule, depending on which part of the calculation is time-dominant.

The algorithm and implementation in Q-CHEM is improved over earlier methods,[39,53] particularly in the following areas:

- Uses pure functions, as opposed to Cartesians, for all fifth-order steps. This leads to large computational savings for basis sets containing pure functions.
- Customized loop unrolling for improved efficiency.
- The sort-less semi-direct method avoids a read and write operation resulting in a large I/O savings.
- Reduction in disk and memory usage.
- No extra integral evaluation for gradient calculations.
- Full exploitation of frozen core approximation.

The implementation offers the user the following alternatives:

- Direct algorithm (energies only).
- Disk-based sort-less semi-direct algorithm (energies and gradients).
- Local occupied orbital method (energies only).

The semi-direct algorithm is the only choice for gradient calculations. It is also normally the most efficient choice for energy calculations. There are two classes of exceptions:

- If the amount of disk space available is not significantly larger than the amount of memory available, then the direct algorithm is preferred.
- If the calculation involves a very large basis set, then the local orbital method may be faster, because it performs the transformation in a different order. It does not have the large memory requirement (no OVN array needed), and always evaluates the integrals four times.

There are three important options that should be wisely chosen by the user in order to exploit the full efficiency of Q-CHEM's direct and semi-direct MP2 methods (as discussed above, the LOCAL_OCCUPIED method has different requirements).

- MEM_STATIC: The value specified for this *$rem* variable must be sufficient to permit efficient integral evaluation (10-80MB) and to hold a large temporary array whose size is OVN, the product of the number of occupied, virtual and total numbers of orbitals.
- N_FROZEN_CORE: The computational requirements for MP2 are proportional to the number of occupied orbitals for some steps, and the square of that number for other steps. Therefore the CPU time can be significantly reduced if your job employs the frozen core approximation. Additionally the memory and disk requirements are reduced when the frozen core approximation is employed.

6.4.2 Algorithm Control and Customization

The direct and semi-direct integral transformation algorithms used by Q-CHEM (*e.g.*, MP2, CIS(D)) are limited by available disk space, D, and memory, C, the number of basis functions, N, the number of virtual orbitals, V and the number of occupied orbitals, O, as discussed above. The generic description of the key *$rem* variables are:

MEM_STATIC

Sets the memory for Fortran AO integral calculation and transformation modules.

TYPE:

INTEGER

DEFAULT:

192 corresponding to 192 MB.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

For direct and semi-direct MP2 calculations, this must exceed OVN + requirements for AO integral evaluation (32–160 MB), as discussed above.

MEM_TOTAL

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 Corresponding to 2000 MB.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use the default, or set equal to the physical memory of your machine. Note that if the memory allocation total more than 1 GB for a CCMAN job, the memory is allocated as follows

12% MEM_STATIC

50% CC_MEMORY

35% Other memory requirements:

CD_ALGORITHM

Determines the algorithm for MP2 integral transformations.

TYPE:

STRING

DEFAULT:

Program determined.

OPTIONS:

DIRECT	Uses fully direct algorithm (energies only).
SEMI_DIRECT	Uses disk-based semi-direct algorithm.
LOCAL_OCCUPIED	Alternative energy algorithm (see 6.4.1).

RECOMMENDATION:

Semi-direct is usually most efficient, and will normally be chosen by default.

Example 6.1 Example of an MP2/6-31G* calculation employing the frozen core approximation. Note that the EXCHANGE *$rem* variable will default to HF

```
$molecule
   0 1
   O
   H1  O  oh
   H2  O  oh  H1  hoh

   oh  = 1.01
   hoh = 105
$end

$rem
   METHOD          mp2
   BASIS           6-31g*
   N_FROZEN_CORE   fc
$end
```

6.5 Local MP2 Methods

6.5.1 Local Triatomics in Molecules (TRIM) Model

The development of what may be called "fast methods" for evaluating electron correlation is a problem of both fundamental and practical importance, because of the unphysical increases in computational complexity with molecular size which afflict "exact" implementations of electron correlation methods. Ideally, the development of fast methods for treating electron correlation should not impact either model errors or numerical errors associated with the original electron correlation models. Unfortunately this is not possible at present, as may be appreciated from the following rough argument. *Spatial locality* is what permits re-formulations of electronic structure methods that yield the same answer as traditional methods, but faster. The one-particle density matrix decays exponentially with a rate that relates to the HOMO-LUMO gap in periodic systems. When length scales longer than this characteristic decay length are examined, sparsity will emerge in both the one-particle density matrix and also pair correlation amplitudes expressed in terms of localized functions. Very roughly, such a length scale is about 5 to 10 atoms in a line, for good insulators such as alkanes. Hence sparsity emerges beyond this number of atoms in 1-D, beyond this number of atoms squared in 2-D, and this number of atoms cubed in 3-D. Thus for three-dimensional systems, locality only begins to emerge for systems of between hundreds and thousands of atoms.

If we wish to accelerate calculations on systems below this size regime, we must therefore introduce additional errors into the calculation, either as numerical noise through looser tolerances, or by modifying the theoretical model, or perhaps both. Q-CHEM's approach to local electron correlation is based on modifying the theoretical models describing correlation with an additional well-defined local approximation. We do not attempt to accelerate the calculations by introducing more numerical error because of the difficulties of controlling the error as a function of molecule size, and the difficulty of achieving reproducible significant results. From this perspective, local correlation becomes an integral part of specifying the electron correlation treatment. This means that the considerations necessary for a correlation treatment to qualify as a well-defined theoretical model chemistry apply equally to local correlation modeling. The local approximations should be

- *Size-consistent:* meaning that the energy of a super-system of two non-interacting molecules should be the sum of the energy obtained from individual calculations on each molecule.
- *Uniquely defined:* Require no input beyond nuclei, electrons, and an atomic orbital basis set. In other words, the model should be uniquely specified without customization for each molecule.
- *Yield continuous potential energy surfaces:* The model approximations should be smooth, and not yield energies that exhibit jumps as nuclear geometries are varied.

To ensure that these model chemistry criteria are met, Q-CHEM's local MP2 methods[54,76] express the double substitutions (*i.e.*, the pair correlations) in a redundant basis of atom-labeled functions. The advantage of doing this is that local models satisfying model chemistry criteria can be defined by performing an *atomic truncation* of the double substitutions. A general substitution in this representation will then involve the replacement of occupied functions associated with two given atoms by empty (or virtual) functions on two other atoms, coupling together four different atoms. We can force one occupied to virtual substitution (of the two that comprise a double substitution) to occur only between functions on the same atom, so that only three different atoms are involved in the double substitution. This defines the *triatomics in molecules* (TRIM) local model for double substitutions. The TRIM model offers the potential for reducing the computational requirements of exact MP2 theory by a factor proportional to the number of atoms. We could also force each occupied to virtual substitution to be on a given atom, thereby defining a more drastic *diatomics in molecules* (DIM) local correlation model.

The simplest atom-centered basis that is capable of spanning the occupied space is a *minimal basis* of core and valence atomic orbitals on each atom. Such a basis is necessarily redundant because it also contains sufficient flexibility to describe the empty valence anti-bonding orbitals necessary to correctly account for non-dynamical electron correlation effects such as bond-breaking. This redundancy is actually important for the success of the atomic truncations because occupied functions on adjacent atoms to some extent describe the same part of the occupied space. The minimal functions we use to span the occupied space are obtained at the end of a large basis set calculation, and are called *extracted polarized atomic orbitals* (EPAOs).[75] We discuss them briefly below. It is even possible to explicitly perform an SCF calculation in terms of a molecule-optimized minimal basis of *polarized atomic orbitals* (PAOs); see Chapter 4. To span the virtual space, we use the full set of atomic orbitals, appropriately projected into the virtual space.

We summarize the situation. The number of functions spanning the occupied subspace will be the minimal basis set dimension, M, which is greater than the number of occupied orbitals, O, by a factor of up to about two. The virtual space is spanned by the set of projected atomic orbitals whose number is the atomic orbital basis set size N, which is fractionally greater than the number of virtuals VNO. The number of double substitutions in such a redundant representation will be typically three to five times larger than the usual total. This will be more than compensated by reducing the number of retained substitutions by a factor of the number of atoms, A, in the local triatomics in molecules model, or a factor of A^2 in the diatomics in molecules model.

The local MP2 energy in the TRIM and DIM models are given by the following expressions, which can be compared

against the full MP2 expression given earlier in Eq. (6.13). First, for the DIM model:

$$E_{\mathrm{DIM\,MP2}} = -\frac{1}{2}\sum_{\bar{P},\bar{Q}} \frac{(\bar{P}|\bar{Q})(\bar{P}||\bar{Q})}{\Delta_{\bar{P}} + \Delta_{\bar{Q}}} \tag{6.17}$$

The sums run over the linear number of atomic single excitations after they have been canonicalized. Each term in the denominator is thus an energy difference between occupied and virtual levels in this local basis. Similarly, the TRIM model corresponds to the following local MP2 energy:

$$E_{\mathrm{TRIM\,MP2}} = -\sum_{\bar{P},jb} \frac{(\bar{P}|jb)(\bar{P}||jb)}{\Delta_{\bar{P}} + \varepsilon_b - \varepsilon_j} - E_{\mathrm{DIM\,MP2}} \tag{6.18}$$

where the sum is now mixed between atomic substitutions $\bar{P}$, and non-local occupied j to virtual b substitutions. See Refs. 54,76 for a full derivation and discussion.

The accuracy of the local TRIM and DIM models has been tested in a series of calculations.[54,76] In particular, the TRIM model has been shown to be quite faithful to full MP2 theory via the following tests:

- The TRIM model recovers around 99.7% of the MP2 correlation energy for covalent bonding. This is significantly higher than the roughly 98–99% correlation energy recovery typically exhibited by the Saebo-Pulay local correlation method.[108] The DIM model recovers around 95% of the correlation energy.
- The performance of the TRIM model for relative energies is very robust, as shown in Ref. 76 for the challenging case of torsional barriers in conjugated molecules. The RMS error in these relative energies is only 0.031 kcal/mol, as compared to around 1 kcal/mol when electron correlation effects are completely neglected.
- For the water dimer with the aug-cc-pVTZ basis, 96% of the MP2 contribution to the binding energy is recovered with the TRIM model, as compared to 62% with the Saebo-Pulay local correlation method.
- For calculations of the MP2 contribution to the G3 and G3(MP2) energies with the larger molecules in the G3-99 database,[21] introduction of the TRIM approximation results in an RMS error relative to full MP2 theory of only 0.3 kcal/mol, even though the absolute magnitude of these quantities is on the order of tens of kcal/mol.

6.5.2 EPAO Evaluation Options

When a local MP2 job (requested by the LOCAL_MP2 option for CORRELATION) is performed, the first new step after the SCF calculation is converged is to extract a minimal basis of polarized atomic orbitals (EPAOs) that spans the occupied space. There are three valid choices for this basis, controlled by the PAO_METHOD and EPAO_ITERATE keywords described below.

- **Non-iterated EPAOs:** The initial guess EPAOs are the default for local MP2 calculations, and are defined as follows. For each atom, the covariant density matrix (SPS) is diagonalized, giving eigenvalues which are approximate natural orbital occupancies, and eigenvectors which are corresponding atomic orbitals. The m eigenvectors with largest populations are retained (where m is the minimal basis dimension for the current atom). This non-orthogonal minimal basis is symmetrically orthogonalized, and then modified as discussed in Ref. 75 to ensure that these functions rigorously span the occupied space of the full SCF calculation that has just been performed. These orbitals may be denoted as EPAO(0) to indicate that no iterations have been performed after the guess. In general, the quality of the local MP2 results obtained with this option is very similar to the EPAO option below, but it is much faster and fully robust. For the example of the torsional barrier calculations discussed above,[76] the TRIM RMS deviations of 0.03 kcal/mol from full MP2 calculations are increased to only 0.04 kcal/mol when EPAO(0) orbitals are employed rather than EPAOs.

- **EPAOs:** EPAOs are defined by minimizing a localization functional as described in Ref. 75. These functions were designed to be suitable for local MP2 calculations, and have yielded excellent results in all tests performed so far. Unfortunately the functional is difficult to converge for large molecules, at least with the algorithms that have been developed to this stage. Therefore it is not the default, but is switched on by specifying a (large) value for EPAO_ITERATE, as discussed below.

- **PAO:** If the SCF calculation is performed in terms of a molecule-optimized minimal basis, as described in Chapter 4, then the resulting PAO-SCF calculation can be corrected with either conventional or local MP2 for electron correlation. PAO-SCF calculations alter the SCF energy, and are therefore not the default. This can be enabled by specifying PAO_METHOD as PAO, in a job which also requests CORRELATION as LOCAL_MP2.

PAO_METHOD

Controls the type of PAO calculations requested.

TYPE:

STRING

DEFAULT:

EPAO For local MP2, EPAOs are chosen by default.

OPTIONS:

EPAO Find EPAOs by minimizing delocalization function.

PAO Do SCF in a molecule-optimized minimal basis.

RECOMMENDATION:

None

EPAO_ITERATE

Controls iterations for EPAO calculations (see PAO_METHOD).

TYPE:

INTEGER

DEFAULT:

0 Use non-iterated EPAOs based on atomic blocks of SPS.

OPTIONS:

n Optimize the EPAOs for up to n iterations.

RECOMMENDATION:

Use the default. For molecules that are not too large, one can test the sensitivity of the results to the type of minimal functions by the use of optimized EPAOs in which case a value of $n = 500$ is reasonable.

EPAO_WEIGHTS

Controls algorithm and weights for EPAO calculations (see PAO_METHOD).

TYPE:

INTEGER

DEFAULT:

115 Standard weights, use 1^{st} and 2^{nd} order optimization

OPTIONS:

15 Standard weights, with 1^{st} order optimization only.

RECOMMENDATION:

Use the default, unless convergence failure is encountered.

A local MP2 calculation (requested by the LOCAL_MP2 option for CORRELATION) consists of the following steps:

- After the SCF is converged, a minimal basis of EPAOs are obtained.
- The TRIM (and DIM) local MP2 energies are then evaluated (gradients are not yet available).

Details of the efficient implementation of the local MP2 method described above are reported in the recent thesis of Dr. Michael Lee.[74] Here we simply summarize the capabilities of the program. The computational advantage associated with these local MP2 methods varies depending upon the size of molecule and the basis set. As a rough general estimate, TRIM MP2 calculations are feasible on molecule sizes about twice as large as those for which conventional MP2 calculations are feasible on a given computer, and this is their primary advantage. Our implementation is well suited for large basis set calculations. The AO basis two-electron integrals are evaluated four times. DIM MP2 calculations are performed as a by-product of TRIM MP2 but no separately optimized DIM algorithm has been implemented.

The resource requirements for local MP2 calculations are as follows:

- *Memory:* The memory requirement for the integral transformation does not exceed OON, and is thresholded so that it asymptotically grows linearly with molecule size. Additional memory of approximately $32N^2$ is required to complete the local MP2 energy evaluation.
- *Disk:* The disk space requirement is only about $8OVN$, but is not governed by a threshold. This is a very large reduction from the case of a full MP2 calculation, where, in the case of four integral evaluations, OVN^2/4 disk space is required. As the local MP2 disk space requirement is not adjustable.

The evaluation of the local MP2 energy does not require any further customization. An adequate amount of MEM_STATIC (at least 80 to 160 MB) should be specified to permit efficient AO basis two-electron integral evaluation, but all large

scratch arrays are allocated from MEM_TOTAL.

Example 6.2 A relative energy evaluation using the local TRIM model for MP2 with the 6-311G** basis set. The energy difference is the internal rotation barrier in propenal, with the first geometry being planar trans, and the second the transition structure.

```
$molecule
   0 1
   C
   C  1  1.32095
   C  2  1.47845  1  121.19
   O  3  1.18974  2  123.83  1   180.00
   H  1  1.07686  2  121.50  3     0.00
   H  1  1.07450  2  122.09  3   180.00
   H  2  1.07549  1  122.34  3   180.00
   H  3  1.09486  2  115.27  4   180.00
$end

$rem
   METHOD        local_mp2
   BASIS         6-311g**
$end

@@@

$molecule
   0 1
   C
   C  1  1.31656
   C  2  1.49838  1  123.44
   O  3  1.18747  2  123.81  1   92.28
   H  1  1.07631  2  122.03  3   -0.31
   H  1  1.07484  2  121.43  3  180.28
   H  2  1.07813  1  120.96  3  180.34
   H  3  1.09387  2  115.87  4  179.07
$end

$rem
   METHOD        local_mp2
   BASIS         6-311g**
$end
```

6.6 Auxiliary Basis (Resolution of the Identity) MP2 Methods

6.6.1 Introduction

For a molecule of fixed size, increasing the number of basis functions *per atom*, n, leads to $\mathcal{O}(n^4)$ growth in the number of significant four-center two-electron integrals, since the number of non-negligible product charge distributions, $|\mu\nu\rangle$, grows as $\mathcal{O}(n^2)$. As a result, the use of large (high-quality) basis expansions is computationally costly. Perhaps the most practical way around this "basis set quality" bottleneck is the use of auxiliary basis expansions.[29,37,62] The ability to use auxiliary basis sets to accelerate a variety of electron correlation methods, including both energies and analytical gradients, is a major feature of Q-CHEM.

The auxiliary basis $\{|K\rangle\}$ is used to approximate products of Gaussian basis functions:

$$|\mu\nu\rangle \approx |\widetilde{\mu\nu}\rangle = \sum_K |K\rangle C_{\mu\nu}^K \tag{6.19}$$

Auxiliary basis expansions were introduced long ago, and are now widely recognized as an effective and powerful approach, which is sometimes synonymously called resolution of the identity (RI) or density fitting (DF). When using auxiliary basis expansions, the rate of growth of computational cost of large-scale electronic structure calculations with n is reduced to approximately n^3.

If n is fixed and molecule size increases, auxiliary basis expansions reduce the pre-factor associated with the computation, while not altering the scaling. The important point is that the pre-factor can be reduced by 5 or 10 times or more. Such large speedups are possible because the number of auxiliary functions required to obtain reasonable accuracy, X, has been shown to be only about 3 or 4 times larger than N.

The auxiliary basis expansion coefficients, $\mathbf{C}$, are determined by minimizing the deviation between the fitted distribution and the actual distribution, $\langle \mu\nu - \widetilde{\mu\nu} | \mu\nu - \widetilde{\mu\nu} \rangle$, which leads to the following set of linear equations:

$$\sum_L \langle K \,|L\rangle C^L_{\mu\nu} = \langle K \,|\mu\nu\rangle \tag{6.20}$$

Evidently solution of the fit equations requires only two- and three-center integrals, and as a result the (four-center) two-electron integrals can be approximated as the following optimal expression for a given choice of auxiliary basis set:

$$\langle \mu\nu|\lambda\sigma\rangle \approx \langle \widetilde{\mu\nu}|\widetilde{\lambda\sigma}\rangle = \sum_{K,L} C^L_{\mu\nu}\langle L|K\rangle C^K_{\lambda\sigma} \tag{6.21}$$

In the limit where the auxiliary basis is complete (*i.e.* all products of AOs are included), the fitting procedure described above will be exact. However, the auxiliary basis is invariably incomplete (as mentioned above, $X \approx 3N$) because this is essential for obtaining increased computational efficiency.

Standardized auxiliary basis sets have been developed by the Karlsruhe group for second-order perturbation (MP2) calculations of the correlation energy.[131,132] Using these basis sets, absolute errors in the correlation energy are small (*e.g.*, below 60 μHartree per atom), and errors in relative energies are smaller still At the same time, speedups of 3–30$\times$ are realized. This development has made the routine use of auxiliary basis sets for electron correlation calculations possible.

Correlation calculations that can take advantage of auxiliary basis expansions are described in the remainder of this section (MP2, and MP2-like methods) and in Section 6.19 (simplified active space coupled cluster methods such as PP, PP(2), IP, RP). These methods automatically employ auxiliary basis expansions when a valid choice of auxiliary basis set is supplied using the AUX_BASIS_CORR or AUX_BASIS keyword which is used in the same way as the BASIS keyword. The PURECART *$rem* is no longer needed here, even if using a auxiliary basis that does not have a predefined value. There is a built-in automatic procedure that provides the effect of the PURECART *$rem* in these cases by default.

6.6.2 RI-MP2 Energies and Gradients.

Following common convention, the MP2 energy evaluated approximately using an auxiliary basis is referred to as "resolution of the identity" MP2, or RI-MP2 for short. RI-MP2 energy and gradient calculations are enabled simply by specifying the AUX_BASIS keyword discussed above. As discussed above, RI-MP2 energies[37] and gradients[25,130] are significantly faster than the best conventional MP2 energies and gradients, and cause negligible loss of accuracy, when an appropriate standardized auxiliary basis set is employed. Therefore they are recommended for jobs where turnaround time is an issue. Disk requirements are very modest; one merely needs to hold various 3-index arrays. Memory requirements grow more slowly than our conventional MP2 algorithms—only quadratically with molecular size. The minimum memory requirement is approximately $3X^2$, where X is the number of auxiliary basis functions, for both energy and analytical gradient evaluations, with some additional memory being necessary for integral evaluation and other small arrays.

In fact, for molecules that are not too large (perhaps no more than 20 or 30 heavy atoms) the RI-MP2 treatment of electron correlation is so efficient that the computation is dominated by the initial Hartree-Fock calculation. This is despite the fact that as a function of molecule size, the cost of the RI-MP2 treatment still scales more steeply with molecule size (it is just that the pre-factor is so much smaller with the RI approach). Its scaling remains 5th order with the size of the molecule, which only dominates the initial SCF calculation for larger molecules. Thus, for RI-MP2 energy evaluation on moderate size molecules (particularly in large basis sets), it is desirable to use the dual basis HF method to further improve execution times (see Section 4.7).

For the size of required memory, the following need to be considered.

MEM_STATIC

Sets the memory for AO-integral evaluations and their transformations in Q-CHEM 4.1 or older versions.

TYPE:

INTEGER

DEFAULT:

192 corresponding to 192 MB.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

For RI-MP2 calculations using Q-CHEM 4.1 or older versions, $150(ON + V)$ of MEM_STATIC is required. Because a number of matrices with N^2 size also need to be stored, 32–160 MB of additional MEM_STATIC is needed.

MEM_TOTAL

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 2 GB

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use the default, or set to the physical memory of your machine. The minimum requirement is $3X^2$.

Example 6.3 Q-CHEM input for an RI-MP2 geometry optimization.

```
$molecule
   0 1
   O
   H  1  0.9
   F  1  1.4  2  100.
$end

$rem
   JOBTYPE       opt
   METHOD        rimp2
   BASIS         cc-pvtz
   AUX_BASIS     rimp2-cc-pvtz
   INTEGRAL_SYMMETRY false
$end
```

6.6.3 OpenMP Implementation of RI-MP2

An OpenMP RI-MP2 energy algorithm is used by default in Q-CHEM 4.1 onward. This can be invoked by using CORR = PRIMP2 for older versions, but note that in 4.01 and below, only RHF/RI-MP2 was supported. Now UHF/RI-MP2 and ROHF/RI-MP2 is supported, and the formation of the 'B' matrices as well as three center integrals are parallelized. This algorithm uses the remaining memory from the MEM_TOTAL allocation for all computation, which can drastically reduce hard drive reads in the formation of t-amplitudes. Since Q-CHEM 5.2, RI-MP2 can be invoked by using method=mp2 and proper auxiliary basis using AUX_BASIS_CORR. In case of double-hybrid, adding AUX_BASIS_CORR will simply invoke RI-MP2.

Example 6.4 Example of OpenMP-parallel RI-MP2 job.

```
$molecule
   0 1
   C1
   H1  C1    1.077260
   H2  C1    1.077260  H1  131.608240
$end

$rem
   EXCHANGE           HF
   CORRELATION        pRIMP2
   BASIS              cc-pVTZ
   AUX_BASIS          rimp2-cc-pVTZ
   PURECART           1111
   THRESH             12
   SCF_CONVERGENCE    8
   MAX_SUB_FILE_NUM   128
   INTEGRAL_SYMMETRY  false
$end
```

6.6.4 GPU Implementation of RI-MP2

Q-CHEM currently offers the possibility of accelerating RI-MP2 calculations using graphics processing units (GPUs). Currently, this is implemented for CUDA-enabled NVIDIA graphics cards only, such as (in historical order from 2008) the GeForce, Quadro, Tesla and Fermi cards. More information about CUDA-enabled cards is available at

http://www.nvidia.com/object/cuda_gpus.html

It should be noted that these GPUs have specific power and motherboard requirements.

Software requirements include the installation of the appropriate NVIDIA CUDA driver (at least version 1.0, currently 3.2) and linear algebra library, CUBLAS (at least version 1.0, currently 2.0). These can be downloaded jointly in NVIDIA's developer website:

http://developer.nvidia.com/object/cuda_3_2_downloads.html

We have implemented a mixed-precision algorithm in order to get *better than* single precision when users only have single-precision GPUs. This is accomplished by noting that RI-MP2 matrices have a *large* fraction of numerically "small" elements and a *small* fraction of numerically "large" ones. The latter can greatly affect the accuracy of the calculation in single-precision only calculations, but calculation involves a relatively small number of compute cycles. So, given a threshold value δ, we perform a separation between "small" and "large" elements and accelerate the former

compute-intensive operations using the GPU (in single-precision) and compute the latter on the CPU (using double-precision). We are thus able to determine how much double-precision we desire by tuning the δ parameter, and tailoring the balance between computational speed and accuracy.

CUDA_RI-MP2

Enables GPU implementation of RI-MP2

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE GPU-enabled MGEMM off

TRUE GPU-enabled MGEMM on

RECOMMENDATION:

Necessary to set to 1 in order to run GPU-enabled RI-MP2

USECUBLAS_THRESH

Sets threshold of matrix size sent to GPU (smaller size not worth sending to GPU).

TYPE:

INTEGER

DEFAULT:

250

OPTIONS:

n user-defined threshold

RECOMMENDATION:

Use the default value. Anything less can seriously hinder the GPU acceleration

USE_MGEMM

Use the mixed-precision matrix scheme (MGEMM) if you want to make calculations in your card in single-precision (or if you have a single-precision-only GPU), but leave some parts of the RI-MP2 calculation in double precision)

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE MGEMM disabled

TRUE MGEMM enabled

RECOMMENDATION:

Use when having single-precision cards

MGEMM_THRESH

Sets MGEMM threshold to determine the separation between "large" and "small" matrix elements. A larger threshold value will result in a value closer to the single-precision result. Note that the desired factor should be multiplied by 10000 to ensure an integer value.

TYPE:

INTEGER

DEFAULT:

10000 (corresponds to 1)

OPTIONS:

n User-specified threshold

RECOMMENDATION:

For small molecules and basis sets up to triple-ζ, the default value suffices to not deviate too much from the double-precision values. Care should be taken to reduce this number for larger molecules and also larger basis-sets.

Example 6.5 RI-MP2 double-precision calculation

```
$molecule
   0 1
   c
   h1  c  1.089665
   h2  c  1.089665  h1  109.47122063
   h3  c  1.089665  h1  109.47122063  h2   120.
   h4  c  1.089665  h1  109.47122063  h2  -120.
$end

$rem
   METHOD       rimp2
   BASIS        cc-pvdz
   AUX_BASIS    rimp2-cc-pvdz
   CUDA_RIMP2   1
$end
```

Example 6.6 RI-MP2 calculation with MGEMM

```
$molecule
   0 1
   c
   h1  c  1.089665
   h2  c  1.089665  h1  109.47122063
   h3  c  1.089665  h1  109.47122063  h2   120.
   h4  c  1.089665  h1  109.47122063  h2  -120.
$end

$rem
   METHOD         rimp2
   BASIS          cc-pvdz
   AUX_BASIS      rimp2-cc-pvdz
   CUDA_RIMP2     1
   USE_MGEMM      1
   MGEMM_THRESH   10000
$end
```

6.6.5 Spin-Biased MP2 Methods (SCS-MP2, SOS-MP2, and MOS-MP2)

The accuracy of MP2 calculations can be significantly improved by semi-empirically scaling the opposite-spin (OS) and same-spin (SS) correlation components with separate scaling factors, as shown by Grimme.[47] Scaling with 1.2 and 0.33 (or OS and SS components) defines the SCS-MP2 method, but other parameterizations are desirable for systems involving intermolecular interactions, as in the SCS-MI-MP2 method, which uses 0.40 and 1.29 (for OS and SS components).[22]

Results of similar quality for thermochemistry can be obtained by only retaining and scaling the opposite spin correlation (by 1.3), as was recently demonstrated.[61] Furthermore, the SOS-MP2 energy can be evaluated using the RI approximation together with a Laplace transform technique, in effort that scales only with the 4th power of molecular size. Efficient algorithms for the energy[61] and the analytical gradient[81] of this method are available since Q-CHEM v. 3.0, and offer advantages in speed over MP2 for larger molecules, as well as statistically significant improvements in accuracy.

However, we note that the SOS-MP2 method does systematically underestimate long-range dispersion (for which the appropriate scaling factor is 2 rather than 1.3) but this can be accounted for by making the scaling factor distance-dependent, which is done in the modified opposite spin variant (MOS-MP2) that has recently been proposed and tested.[80] The MOS-MP2 energy and analytical gradient are also available in Q-CHEM 3.0 at a cost that is essentially identical with SOS-MP2. Timings show that the 4th-order implementation of SOS-MP2 and MOS-MP2 yields substantial speedups over RI-MP2 for molecules in the 40 heavy atom regime and larger. It is also possible to customize the scale factors for particular applications, such as weak interactions, if required.

A fourth order scaling SOS-MP2/MOS-MP2 energy calculation can be invoked by setting the CORRELATION keyword to either SOSMP2 or MOSMP2. MOS-MP2 further requires the specification of the *$rem* variable OMEGA, which tunes the level of attenuation of the MOS operator:[80]

$$g_\omega(r_{12}) = \frac{1}{r_{12}} + c_{\mathrm{MOS}} \frac{\mathrm{erf}\,(\omega r_{12})}{r_{12}} \tag{6.22}$$

The recommended OMEGA value is $\omega = 0.6$ bohr^{-1}.[80] The fast algorithm makes use of auxiliary basis expansions and therefore, the keyword AUX_BASIS should be set consistently with the user's choice of BASIS. Fourth-order scaling analytical gradient for both SOS-MP2 and MOS-MP2 are also available and is automatically invoked when JOBTYPE is set to OPT or FORCE. The minimum memory requirement is $3X^2$, where X = the number of auxiliary basis functions, for both energy and analytical gradient evaluations. Disk space requirement for closed shell calculations is $\sim 2OVX$ for energy evaluation and $\sim 4OVX$ for analytical gradient evaluation.

Summary of key *$rem* variables to be specified:

CORRELATION	RIMP2
	SOSMP2
	MOSMP2
JOBTYPE	sp (default) single point energy evaluation
	opt geometry optimization with analytical gradient
	force evaluation with analytical gradient
BASIS	user's choice (standard or user-defined: GENERAL or MIXED)
AUX_BASIS	corresponding auxiliary basis (standard or user-defined: AUX_GENERAL or AUX_MIXED
OMEGA	no default n; use $\omega = n/1000$. The recommended value is $n = 600$ ($\omega = 0.6$ bohr^{-1})
N_FROZEN_CORE	Optional
N_FROZEN_VIRTUAL	Optional
SCS	Turns on spin-component scaling with SCS-MP2(1), SOS-MP2(2), and arbitrary SCS-MP2(3)

Example 6.7 Example of SCS-MP2 geometry optimization

```
$molecule
   0 1
   C
   H  1  1.0986
   H  1  1.0986  2  109.5
   H  1  1.0986  2  109.5  3   120.0 0
   H  1  1.0986  2  109.5  3  -120.0 0
$end

$rem
   JOBTYPE             opt
   CORRELATION         rimp2
   BASIS               aug-cc-pvdz
   AUX_BASIS           rimp2-aug-cc-pvdz
   BASIS2              racc-pvdz   ! Optional Secondary basis
   SCS                 1           ! Turn on spin-component scaling
   DUAL_BASIS_ENERGY   true        ! Optional dual-basis approximation
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 6.8 Example of SCS-MI-MP2 energy calculation

```
$molecule
   0 1
   C       0.000000     -0.000140      1.859161
   H      -0.888551      0.513060      1.494685
   H       0.888551      0.513060      1.494685
   H       0.000000     -1.026339      1.494868
   H       0.000000      0.000089      2.948284
   C       0.000000      0.000140     -1.859161
   H       0.000000     -0.000089     -2.948284
   H      -0.888551     -0.513060     -1.494685
   H       0.888551     -0.513060     -1.494685
   H       0.000000      1.026339     -1.494868
$end

$rem
   CORRELATION         rimp2
   BASIS               aug-cc-pvtz
   AUX_BASIS           rimp2-aug-cc-pvtz
   BASIS2              racc-pvtz    Optional Secondary basis
   THRESH              12
   SCF_CONVERGENCE     8
   SCS                 3            Spin-component scale arbitrarily
   SOS_FACTOR          0400000      Specify OS parameter
   SSS_FACTOR          1290000      Specify SS parameter
   DUAL_BASIS_ENERGY   true         Optional dual-basis approximation
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 6.9 Example of SOS-MP2 geometry optimization

```
$molecule
   0 3
   C1
   H1   C1   1.07726
   H2   C1   1.07726   H1  131.60824
$end

$rem
   JOBTYPE        opt
   METHOD         sosmp2
   BASIS          cc-pvdz
   AUX_BASIS      rimp2-cc-pvdz
   UNRESTRICTED   true
   INTEGRAL_SYMMETRY false
$end
```

Example 6.10 Example of MOS-MP2 energy evaluation with frozen core approximation

```
$molecule
   0 1
   Cl
   Cl 1 2.05
$end

$rem
   METHOD            mosmp2
   BASIS             cc-pVTZ
   AUX_BASIS         rimp2-cc-pVTZ
   N_FROZEN_CORE     fc
   OMEGA             600
   THRESH            12
   SCF_CONVERGENCE   8
$end
```

6.6.6 Orbital-Optimized MP2

Brueckner orbitals (BOs) are highly desirable when one is unsure whether artificial symmetry breaking occurs at the Hartree-Fock (HF) level. It is *artificial* because this symmetry breaking merely reflects the lack of dynamic correlation at the HF level, not the lack of strong correlation. On the other hand, it is possible for a single-reference approach to attempt to describe strongly correlated systems by *essential* symmetry breaking. Therefore, it is often crucial to distinguish these two by obtaining orbitals in the presence of electron correlation.[71]

Since orbital-optimized coupled-cluster doubles (OOCCD) can be computationally demanding ($\mathcal{O}(o^2v^4)$), Rohini Lochan working with Martin Head-Gordon proposed to obtain orbitals by optimizing the MP2 correlation energy. To this end, BOs are introduced into SOSMP2 and MOSMP2 methods to resolve the problems of symmetry breaking and spin contamination that are often associated with Hartree-Fock orbitals. The molecular orbitals are optimized with the mean-field energy plus a correlation energy taken as the opposite-spin component of the second-order many-body correlation energy, scaled by an empirically chosen parameter. This "optimized second-order opposite-spin" (O2) method[79] requires fourth-order computation on each orbital iteration. O2 is shown to yield predictions of structure and frequencies for closed-shell molecules that are very similar to scaled MP2 methods. However, it yields substantial improvements for open-shell molecules, where problems with spin contamination and symmetry breaking are shown

to be greatly reduced.

Example 6.11 Example of O2 methodology applied to $\mathcal{O}(N^4)$ SOSMP2

```
$molecule
   1 2
   F
   H 1 1.001
$end

$rem
   EXCHANGE          HF
   BASIS             sto-3g
   UNRESTRICTED      TRUE
   JOBTYPE           FORCE           Options are SP/FORCE/OPT
   DO_O2             1               O2 with O(N^4) SOS-MP2 algorithm
   SOS_FACTOR        1000000         Opposite Spin scaling factor = 1.0
   SCF_ALGORITHM     DIIS_GDM
   SCF_GUESS         GWH
   AUX_BASIS         rimp2-vdz
   SCF_CONVERGENCE   8
   THRESH            14
   PURECART          1111
   INTEGRAL_SYMMETRY FALSE
$end
```

Example 6.12 Example of O2 methodology applied to $\mathcal{O}(N^4)$ MOSMP2

```
$molecule
   1 2
   F
   H 1 1.001
$end

$rem
   UNRESTRICTED      TRUE
   JOBTYPE           FORCE      ! Options are SP/FORCE/OPT
   EXCHANGE          HF
   DO_O2             2          ! O2 with O(N^4) MOS-MP2 algorithm
   OMEGA             600        ! Omega = 600/1000 = 0.6 a.u.
   SCF_ALGORITHM     DIIS_GDM
   SCF_GUESS         GWH
   BASIS             sto-3g
   AUX_BASIS         rimp2-vdz
   SCF_CONVERGENCE   8
   THRESH            14
   PURECART          1111
   INTEGRAL_SYMMETRY FALSE
$end
```

Although O2 (or OOMP2) was successful in numerous applications, there are two limitations of this model. First of all, the energy optimization often runs into a numerical instability caused by the singularity of the MP2 energy due to a small energy denominator. Secondly, the disappearance of Coulson-Fischer point hinders the use of *essential* symmetry breaking. This led David Stück and Martin Head-Gordon to regularized OOMP2 where they employed a linear level shift parameter, δ, to stabilize small energy denominators.[119] The thermochemistry performance of δ-OOMP2 was found to be disappointing when one wishes to keep δ large enough to recover the Coulson-Fischer point.[103]

Joonho Lee working with Martin Head-Gordon developed a new regularized OOMP2 suite of methods that utilizes an energy-dependent regularizer (κ-regularizer) unlike the δ-regularizer.[70] The κ-regularizer modifies the MP2 correlation

energy as follows:

$$E_{\kappa-\mathrm{MP2}} = -\frac{1}{4}\sum_{ijab}\frac{|\langle ij||ab\rangle|^2}{\Delta_{ij}^{ab}}\left(1-\exp(-\kappa\Delta_{ij}^{ab})\right)^2 \tag{6.23}$$

where the energy denominator $\Delta_{ij}^{ab} = \epsilon_a\epsilon_b - \epsilon_i - \epsilon_j$ and κ controls the regularization strength. Evidently, $\kappa = 0$ gives zero correlation energy (*i.e.*, HF) and $\kappa \to \infty$ recovers the unregularized MP2 energy expression. In κ-OOMP2, orbitals are then determined as a minimizer for $E_{\mathrm{HF}}E_{\kappa-\mathrm{MP2}}$. The κ value of 1.45 E_h^{-1} is recommended due to its good balance between the Coulson-Fischer point recovery and thermochemistry performance. It should be noted that κ-OOMP2 runs through Q-CHEM's new SCF library, `libgscf`, and new MP2 library, `libgmbpt`. The older OOMP2 code (written by Rohini Lochan and David Stück) is no longer supported and should be used with a greater caution. Furthermore, the new OOMP2 code can handle restricted (R), complex, restricted (cR), unrestricted (U), generalized (G), and complex, generalized (cG) orbital types. The complex, unrestricted (cU) orbital type is not yet supported due to its limited applicability.

Summary of rem variables relevant to run κ-OOMP2:

CORRELATION	None (default)
JOBTYPE	sp (default) single point energy evaluation; force (force supported); opt (geometry optimization supported
BASIS	user's choice (standard or user-defined: GENERAL or MIXED)
GEN_SCFMAN_FINAL	TRUE (default if κ-OOMP2 is requested)
	FALSE (default for other SCF jobs)
SCF_ALGORITHM	GDM (default)
	DIIS
	GDM-LS
AUX_BASIS	corresponding auxiliary basis (standard or user-defined:
	AUX_GENERAL or AUX_MIXED)
REGULARIZED_O2	0 (no regularizer; default)
	1 (δ-regularizer)
	2 (κ-regularizer; recommended)
	3 (σ-regularizer)
REG_PARAMETER	regularization parameter multiplied by 1e^3; no default
	1450 (Recommended value for κ-OOMP2)
N_FROZEN_CORE	0 (Code supports this functionality but it is not
	recommended due to some convergence issues)
N_FROZEN_VIRTUAL	0 (Code supports this functionality but it is not
	recommended due to some convergence issues)
SCS	0 (default)
	1 Turns on spin-component scaling with SCS-OOMP2,
	2 SOS-OOMP2,
	3 arbitrary SCS-OOMP2
SSS_FACTOR	1000000 (default) Specify same-spin-component scaling factor (multiplied by 1e^6)
SOS_FACTOR	1000000 (default) Specify opposite-spin-component scaling factor (multiplied by 1e^6)
DO_S2	0 (default)
	1 (Compute $\langle\hat{S}^2\rangle$ at the MP2 level)

Example 6.13 Example of κ-OOMP2 with the cG orbital type applied to OH

```
$molecule
   0  2
   O  -2.766559046  0.187082886   0.566917837
   H  -3.696304300  1.179189102  -0.642506882
$end

$rem
   BASIS              cc-pvdz
   AUX_BASIS          rimp2-cc-pvdz
   EXCHANGE           hf
   THRESH             14
   INPUT_BOHR         true
   SCF_CONVERGENCE    8
   SCF_ALGORITHM      gdm
   MAXSCF             1000
   SCF_GUESS          sad
   GEN_SCFMAN         true
   GEN_SCFMAN_FINAL   true
   N_FROZEN_CORE      0      ! no frozen core
   N_FROZEN_VIRTUAL   0      ! no frozen virtual
   DO_O2              3      ! run OOMP2
   REGULARIZED_O2     2      ! use kappa-regularizer
   REG_VARIABLE       1450   ! set kappa = 1.45
   SCS                3      ! use arbitrary SCS
   SOS_FACTOR         883532 ! use cos = 0.883532
   SSS_FACTOR         883532 ! use css = 0.883532
   DO_S2              1      ! compute s^2 at the MP2 level
   UNRESTRICTED       true   ! use unrestricted
   GHF                true   ! use generalized
   COMPLEX            true   ! use complex
   INTEGRAL_SYMMETRY  false
$end
```

Example 6.14 Example of κ-OOMP2 with the R orbital type applied to a water dimer

```
$molecule
   0  1
   O    -2.766559046    0.187082886    0.566917837
   H    -3.696304300    1.179189102   -0.642506882
   H    -3.395837846   -1.509891173    0.389283582
   O     2.587035064    0.275900014   -0.746441819
   H     3.579141280    0.918406897    0.633058252
   H     0.852266482    0.311804811   -0.156847268
$end

$rem
   BASIS              cc-pvdz
   AUX_BASIS_CORR     rimp2-cc-pvdz
   EXCHANGE           hf
   THRESH             14
   INPUT_BOHR         true
   SCF_CONVERGENCE    8
   SCF_ALGORITHM      gdm
   MAXSCF             1000
   SCF_GUESS          sad
   GEN_SCFMAN         true
   UNRESTRICTED       false  ! use restricted
   DO_O2              3      ! run OOMP2
   REGULARIZED_O2     2      ! use kappa-regularizer
   REG_VARIABLE       1450   ! set kappa = 1.45
   INTEGRAL_SYMMETRY false
$end
```

6.6.7 Brueckner CC2

Brueckner orbitals (BOs) are shown to largely ameliorate the artificial symmetry breaking that occurs at the Hartree-Fock (HF) level.[71] This can lead to improved results even in higher order wavefunction theories that typically use HF orbitals as a starting point. Unfortunately, the cheapest traditional Brueckner theory, Brueckner doubles (BD), is still quite computationally demanding. Orbital optimized MP2 (OOMP2) was proposed as a low scaling approximation to BD[79] due to the similarity of orbital optimized coupled cluster doubles (OOCCD) and BD. Another possible source of approximate BD orbitals is Bruckner CC2 (BCC2).[7]

Rather than optimizing the energy as in OOMP2, BCC2 optimizes the orbitals in order to reduce the CC2 t_1 amplitudes to 0. In the absence of t_1 amplitudes, the CC2 doubles amplitudes are exactly the same as MP2, and the singles amplitude equations are as follows:

$$\Omega_i^a = f_{ai} + \sum_{jb} f_{jb} t_{ij}^{ab} + \frac{1}{2} \sum_{jbc} \langle ja||cb\rangle t_{ij}^{bc} + \frac{1}{2} \sum_{jkb} \langle jk||bi\rangle t_{jk}^{ab} \tag{6.24}$$

BCC2 has a computational cost scaling as $\mathcal{O}(o^2v^3)$ - a significant improvement over OOCCD/BD, and can greatly

improve the quality of orbitals over HF in open-shell molecules due to the reduction of artificial spin contamination.

Example 6.15 Example of BCC2 methodology

```
$molecule
   1 2
   F
   H 1 1.001
$end

$rem
   UNRESTRICTED       TRUE
   JOBTYPE            SP
   EXCHANGE           HF
   GEN_SCFMAN_FINAL   TRUE
   DO_BCC2            3                 run BCC2
   SCF_ALGORITHM      DIIS
   SCF_GUESS          GWH
   BASIS              sto-3g
   AUX_BASIS          rimp2-vdz
   SCF_CONVERGENCE    8
   THRESH             14
   PURECART           1111
   INTEGRAL_SYMMETRY FALSE
$end
```

As CC2 is a perturbative approach similar to MP2, breakdowns occur in the presence of orbital degeneracy. While BCC2 does not explicitly optimize the energy and thereby drive orbitals toward degeneracy, results are still severely hindered when orbital energy differences become small. The κ regularizer originally proposed by Joonho Lee and Martin Head-Gordon[70] can be used in to further improve the results of BCC2 in this case. The regularizer is applied by modifying the t amplitudes as follows:

$$t_{ij}^{ab} = \frac{-\langle ab||ij\rangle}{\Delta_{ij}^{ab}} \left(1 - \exp(-\kappa \Delta_{ij}^{ab})\right)^2 \tag{6.25}$$

As in κ-OOMP2, the parameter κ sets the regularization strength, with $\kappa = 0$ yielding HF, and $\kappa = \infty$ yielding unregularized BCC2. A value of $\kappa = 1.2\ E_h^{-1}$ is suggested for most applications. κ-BCC2 runs through `libgscf` and `libgmbpt`. Currently, DIIS should be used to converge BCC2 orbitals, as the singles residual is not an orbital gradient and cannot be used with gradient-based algorithms. The BCC2 code can currently handle restricted (R) and unrestricted (U) orbital types.

Summary of rem variables relevant to run κ-BCC2:

CORRELATION	None (default)
JOBTYPE	sp (default) single point energy evaluation
BASIS	user's choice (standard or user-defined: GENERAL or MIXED)
GEN_SCFMAN_FINAL	TRUE
SCF_ALGORITHM	DIIS (gradient based algorithms currently unsupported)
AUX_BASIS	corresponding auxiliary basis (standard or user-defined: AUX_GENERAL or AUX_MIXED)
DO_BCC2	3 (run BCC2)
REGULARIZED_BCC2	0 (no regularizer; default) 1 (δ-regularizer) 2 (κ-regularizer; recommended) 3 (σ-regularizer)
REG_PARAMETER	regularization parameter multiplied by $1e^3$; no default 1200 (Recommended value for κ-BCC2)
N_FROZEN_CORE	0 (frozen core currently unsupported)
N_FROZEN_VIRTUAL	0 (frozen core currently unsupported)
DO_S2	0 (default) 1 (Compute $\langle \hat{S}^2 \rangle$ at the MP2 level)

Example 6.16 Example of κ-BCC2 with the R orbital type applied to a water dimer

```
$molecule
   0  1
   O    -2.766559046    0.187082886    0.566917837
   H    -3.696304300    1.179189102   -0.642506882
   H    -3.395837846   -1.509891173    0.389283582
   O     2.587035064    0.275900014   -0.746441819
   H     3.579141280    0.918406897    0.633058252
   H     0.852266482    0.311804811   -0.156847268
$end

$rem
   EXCHANGE          hf
   BASIS             cc-pvdz
   AUX_BASIS_CORR    rimp2-cc-pvdz
   THRESH            14
   INPUT_BOHR        true
   SCF_CONVERGENCE   8
   MAXSCF            1000
   SCF_GUESS         sad
   GEN_SCFMAN_FINAL  true
   UNRESTRICTED      false    use restricted
   DO_BCC2           3        run BCC2
   REGULARIZED_O2    2        use kappa-regularizer
   REG_VARIABLE      1450     set kappa = 1.45
   INTEGRAL_SYMMETRY false
$end
```

Example 6.17 Example of κ-BCC2 with the U orbital type applied to an OH radical

```
$molecule
0  2
O -2.766559046 0.187082886 0.566917837
H -3.6963043 1.179189102 -0.642506882
$end

$rem
   EXCHANGE          hf
   BASIS             cc-pvdz
   AUX_BASIS_CORR    rimp2-cc-pvdz
   THRESH            14
   INPUT_BOHR        true
   SCF_CONVERGENCE   8
   MAXSCF            1000
   SCF_GUESS         sad
   GEN_SCFMAN_FINAL  true
   UNRESTRICTED      true   ! use unrestricted
   DO_BCC2           3      ! run BCC2
   REGULARIZED_O2    2      ! use kappa-regularizer
   REG_VARIABLE      1450   ! set kappa = 1.45
   INTEGRAL_SYMMETRY false
$end
```

6.6.8 RI-TRIM MP2 Energies

The triatomics in molecules (TRIM) local correlation approximation to MP2 theory[76] was described in detail in Section 6.5.1, which also discussed our implementation of this approach based on conventional four-center two-electron

integrals. Starting from Q-CHEM v. 3.0, an auxiliary basis implementation of the TRIM model is available. The new RI-TRIM MP2 energy algorithm[23] greatly accelerates these local correlation calculations (often by an order of magnitude or more for the correlation part), which scale with the 4th power of molecule size. The electron correlation part of the calculation is speeded up over normal RI-MP2 by a factor proportional to the number of atoms in the molecule. For a hexadecapeptide, for instance, the speedup is approximately a factor of 4.[23] The TRIM model can also be applied to the scaled opposite spin models discussed above. As for the other RI-based models discussed in this section, we recommend using RI-TRIM MP2 instead of the conventional TRIM MP2 code whenever run-time of the job is a significant issue. As for RI-MP2 itself, TRIM MP2 is invoked by adding AUX_BASIS *$rem*s to the input deck, in addition to requesting CORRELATION = RILMP2.

Example 6.18 Example of RI-TRIM MP2 energy evaluation

```
$molecule
   0 3
   C1
   H1   C1   1.07726
   H2   C1   1.07726   H1   131.60824
$end

$rem
   METHOD          rilmp2
   BASIS           cc-pVDZ
   AUX_BASIS       rimp2-cc-pVDZ
   PURECART        1111
   UNRESTRICTED    true
   INTEGRAL_SYMMETRY false
$end
```

6.6.9 Dual-Basis MP2

The successful computational cost speedups of the previous sections often leave the cost of the underlying SCF calculation dominant. The dual-basis method provides a means of accelerating the SCF by roughly an order of magnitude, with minimal associated error (see Section 4.7). This dual-basis reference energy may be combined with RI-MP2 calculations for both energies[117,118] and analytic first derivatives.[24] In the latter case, further savings (beyond the SCF alone) are demonstrated in the gradient due to the ability to solve the response (Z-vector) equations in the smaller basis set. Refer to Section 4.7 for details and job control options.

6.6.10 RI-MP2 Method for Complex Basis Functions

Based on the general implementation of complex basis functions in libqints by White, Head-Gordon, and McCurdy,[133,134] (see Section 4.9.5) an RI-MP2 method for complex resonance energies has been implemented.[128,129] This method is currently limited to closed-shell cases. The RI approximation can be applied to the complex MP2 energy as well as to the Coulomb and exchange parts of the complex HF energy. The use of the RI approximation is particularly advantageous for electronic resonances since their treatment using complex-scaled methods requires large bases with many diffuse functions. In many cases, RI reduces computation times by a factor of 10 or more. Also, there is no need to include complex-scaled functions in the auxiliary basis set; standard auxiliary bases provide excellent results.[129]

The full basis set is supplied through the keyword COMPLEX_BASIS, while BASIS specifies the unscaled part thereof. This process is described in Section 8.7. In complete analogy, the auxiliary basis set is specified using the keywords COMPLEX_AUX_BASIS and AUX_BASIS. The keyword COMPLEX_RI_JK controls whether the RI approximation is invoked only for the MP2 part or for the HF reference as well.

See Sections 4.9.5 and 7.10.9 for more information about electronic resonances, functionalities offered by Q-CHEM in this context, and the corresponding keywords.

Example 6.19 Q-CHEM input for an RI-MP2 calculation of the complex resonance energy (= Stark-shifted energy and tunnel ionization rate) of N_2 in a static electric field of a strength of 0.1 a.u.

```
$molecule
   0 1
   N  0.00  0.00  0.55
   N  0.00  0.00 -0.55
$end

$rem
   CORRELATION           RIMP2
   BASIS                 6-31G
   COMPLEX_BASIS         6-31G*
   AUX_BASIS             rimp2-aug-cc-pVDZ
   COMPLEX_AUX_BASIS     rimp2-aug-cc-pVDZ
   COMPLEX_RI_JK         true
   COMPLEX_CCMAN         true
   SCF_GUESS             gwh
   SCF_CONVERGENCE       10
   COMPLEX_EXPONENTS     1
   COMPLEX_THETA         80
   COMPLEX_SCF           1
   COMPLEX_SCF_GUESS     1
   COMPLEX_N_ELECTRONS   0
   COMPLEX_METSCF        1
   GEN_SCFMAN            true
   THRESH                14
   PURECART              1111
   INTEGRAL_SYMMETRY     false
   POINT_GROUP_SYMMETRY  false
$end

$complex_ccman
   stark_z   1000
   cs_alpha  1000
   cs_theta  0
$end
```

6.6.11 RI-MP2 Method with the Laplace Transformation

Joonho Lee working with Martin Head-Gordon added a libgmbpt-based Laplace-transformed RI-MP2 implementation for both same-spin and opposite-spin correlations.[73] To use this feature, one can just add LAPLACE_TRANSFORM TRUE to the *$rem* block.

6.7 Attenuated MP2

MP2(attenuator, basis) approximates MP2 by splitting the Coulomb operator in two pieces and preserving only short-range two-electron interactions, akin to the CASE approximation,[6,28] but without modification of the underlying SCF calculation. While MP2 is a comparatively efficient method for estimating the correlation energy, it converges slowly with basis set size — and, even in the complete basis limit, contains fundamentally inaccurate physics for long-range interactions. Basis set superposition error and the MP2-level treatment of long-range interactions both typically artificially increase correlation energies for non-covalent interactions. Attenuated MP2 improves upon MP2 for inter-

and intramolecular interactions, with significantly better performance for relative and binding energies of non-covalent complexes, frequently outperforming complete basis set estimates of MP2.[45,46]

Attenuated MP2, denoted MP2(attenuator, basis) is implemented in Q-CHEM based on the complementary terf function, below:

$$\mathrm{s(r)} = \mathrm{terfc(r, r_0)} = \frac{1}{2}\left\{\mathrm{erfc}\left[\omega(\mathrm{r} - \mathrm{r_0})\right] + \mathrm{erfc}\left[\omega(\mathrm{r} + \mathrm{r_0})\right]\right\} \tag{6.26}$$

By choosing the terfc short-range operator, we optimally preserve the short-range behavior of the Coulomb operator while smoothly and rapidly switching off around the distance r_0. Since this directly addresses basis set superposition error, parameterization must be done for specific basis sets. This has been performed for the basis sets, aug-cc-pVDZ[45] and aug-cc-pVTZ.[46] Other basis sets are not recommended for general use until further testing has been done.

Energies and gradients are functional with and without the resolution of the identity approximation using correlation keywords ATTMP2 and ATTRIMP2.

Example 6.20 Example of RI-MP2(terfc, aug-cc-pVDZ) energy evaluation

```
$molecule
   0 1
   O    -1.551007   -0.114520    0.000000
   H    -1.934259    0.762503    0.000000
   H    -0.599677    0.040712    0.000000
$end

$rem
   METHOD          attrimp2
   BASIS           aug-cc-pvdz
   AUX_BASIS       rimp2-aug-cc-pvdz
$end
```

Example 6.21 Example of MP2(terfc, aug-cc-pVTZ) geometry optimization

```
$molecule
   0 1
   H    0.0   0.0   0.0
   H    0.0   0.0   0.9
$end

$rem
   JOBTYPE         opt
   METHOD          attmp2
   BASIS           aug-cc-pvtz
$end
```

6.8 Size-Consistent Brillouin-Wigner Perturbation Theory

6.8.1 Introduction

While second-order Møller-Plesset perturbation theory (MP2) offers the simplest and most cost-effective *ab initio* correlation energy, it requires a separate formulation for degenerate states because the MP2 energy diverges in the zero-gap limit. Second-order Brillouin-Wigner perturbation theory (BW2) requires no such reformulation, as the second-order energy given by,

$$E_\mathrm{c}^{(2)} = -\frac{1}{4}\sum_{ijab}\frac{|\langle ij||ab\rangle|^2}{\Delta_{ij}^{ab} + E_\mathrm{c}^{(2)}} \tag{6.27}$$

is finite when the orbital energy gap $\Delta_{ij}^{ab} = \varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j$ is zero due to the presence of the correlation energy in the denominator. However, the BW2 correlation energy is not size-consistent in the sense that it does not satisfy $E(A \cup B) = E(A) + E(B)$ for two distant, noninteracting subsystems A and B, severely limiting its applicability to chemistry.

Recently, Carter-Fenk and Head-Gordon introduced a size-consistent-to-second-order Brillouin-Wigner perturbation theory (BW-s2) that retains this essential property while remaining finite for zero-gap systems.[15] This theory, based on a repartitioning of the zeroth-order Hamiltonian, results in a slightly-modified amplitude equation. Whereas the MP2 amplitudes can be found by solving,

$$\Delta_{ijkl}^{abcd} \cdot t_{kl}^{cd} = -\langle ij||ab\rangle \ , \tag{6.28}$$

the BW-s2 amplitude equation contains a regularizing tensor,

$$\left(\Delta_{ijkl}^{abcd} + R_{ijkl}^{abcd}\right) \cdot t_{kl}^{cd} = -\langle ij||ab\rangle \tag{6.29}$$

In the above equations Δ_{ijkl}^{abcd} is composed of Fock matrix elements and reduces to the familiar Δ_{ij}^{ab} when canonical orbitals are used, and

$$R_{ijkl}^{abcd} = \frac{\alpha}{2}\left(W_{ik}\delta_{jl} + \delta_{ik}W_{jl}\right)\delta_{ac}\delta_{bd} \tag{6.30}$$

where,

$$W_{ij} = \frac{1}{2}\sum_{kab}\left(t_{ik}^{ab}\langle jk||ab\rangle + t_{jk}^{ab}\langle ik||ab\rangle\right) \tag{6.31}$$

This form of $\mathbf{W}$ was chosen because it is size-consistent, leading to a size-consistent BW-s2 correlation energy. Physically, $\mathbf{W}$ represents the correlation energy of a Koopmans' (static orbital) ionization process. Thus, the occupied orbitals in BW-s2 are imbued with correlation such that the occupied/virtual gap increases. After rotating the occupied orbitals into a basis where $\boldsymbol{\Delta} + \boldsymbol{R}$ is diagonal, the BW-s2 working equation looks like that of MP2,

$$E_{\mathrm{c}}^{(2)} = -\frac{1}{4}\sum_{ijab}\frac{|\widetilde{\langle ij||ab\rangle}|^2}{\varepsilon_a + \varepsilon_b - \tilde{\varepsilon}_i - \tilde{\varepsilon}_j} \tag{6.32}$$

but the orbitals and corresponding anti-symmetrized two-electron integrals have been rotated into the new basis.

In equation 6.30 there is an implicit parameter α that controls the regularization strength. Initially, this parameter was set to $\alpha = 1$ to obtain the exact result for the two-electron two-orbital system of minimal-basis H_2 at the dissociation limit, but α was later tuned to achieve much more accurate results for a wide array of chemical problems ranging from thermochemical properties to noncovalent interaction energies and closed-shell transition-metal reaction energies.[16] The optimal α value for the resultant BW-s2(α) approach varies somewhat between chemical problems, but was found to be more flexible than gap-dependent regularizer parameters like κ-, σ-, or σ^2-MP2. A "universal" parameter of $\alpha = 4$ was suggested as a compromise to achieve the best all-around results for a wide array of chemical properties.[16]

6.8.2 BW-s2 Job Control

The BW-s2 approach has been implemented with the resolution-of-the-identity approximation and is invoked with METHOD = RIBWS2. The BW-s2 equation is iterative much like BW2, and keywords that give more precise control over the self-consistent determination of the BW-s2 energy are housed within a separate *$bws2* section. Finally, the underlying algorithm that is used to solve the BW-s2 equation can be used to solve for other forms of the correlation energy such as MP2, BW2, size-extensive xBW2 (which is not size-consistent), or second-order Bethe-Goldstone (BGE2)/independent electron pair approximation (IEPA). While these methods are available for academic purposes, it is advised to only use the BW2, xBW2, and BGE2/IEPA codes for comparison as the first two are not size-consistent and the latter is not orbital invariant. For MP2 calculations, the standard RI-MP2 code should be used.

METHOD

Defines the correlation energy expression to be used.

INPUT SECTION: *$bws2*

TYPE:

STRING

DEFAULT:

BWS2

OPTIONS:

BWS2, BW2, xBW2, BGE2, IEPA, or MP2

RECOMMENDATION:

BW-s2 is the recommended method, but if comparisons with non-size-consistent methods like BW2 or xBW2 or non-orbital-invariant methods like BGE2/IEPA are desired, those options are available.

ALPHA

Controls the extent of BW-s2(α) regularization.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n where $n/1000$ is the value of α.

RECOMMENDATION:

The default is set to the original BW-s2 theory where $\alpha = 1$ (corresponding to $n = 1000$) is exact for a two-electron, two-orbital problem. For general applications $\alpha = 4$ ($n = 4000$) is a broadly successful choice. Other values can be set based on the chemical problem at hand.

ENERGY_CONVERGENCE

Defines the convergence threshold for the correlation energy.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

n = SCF_CONVERGENCE

OPTIONS:

n where the convergence threshold is set to 10^{-n}

RECOMMENDATION:

Use the default unless the cost of the calculation becomes prohibitive, then consider reducing to a slightly lower value.

AMPLITUDE_CONVERGENCE

Defines the convergence threshold for the magnitude of the amplitude equation residual.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n where the convergence threshold is set to 10^{-n}

RECOMMENDATION:

Use the default for small to medium sized molecules. Large molecules may require looser criteria.

MAX_CYCLES

Defines maximum number of iterations to be used in determining the correlation energy.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

n = SCF_MAX_CYCLES

OPTIONS:

$n > 0$

RECOMMENDATION:

Use the default unless convergence is particularly slow, then increase the number of iterations accordingly.

PARAM_GRADIENT

Controls whether or not to compute the derivative of the energy with respect to α.

INPUT SECTION: *$bws2*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Computes the iterative and non-iterative parameter gradient.

FALSE Does not compute the parameter gradient.

RECOMMENDATION:

If required, set to TRUE.

Example 6.22 Example of RI-BW-s2 on an acetylene pyridine complex

```
$molecule
0 1
N   -0.083032490   0.000714589   1.055199987
C   -0.202853764  -1.141725850   0.364933690
H   -0.098485626  -2.055097952   0.937432624
C   -0.446781438  -1.191763671  -1.004512261
H   -0.533649208  -2.145855109  -1.504171553
C   -0.574682087   0.003439534  -1.704309480
H   -0.763683910   0.004480103  -2.768726700
C   -0.453456753   1.197242545  -1.000916473
H   -0.545630798   2.152272643  -1.497795085
C   -0.209311108   1.144507590   0.368367296
H   -0.110167072   2.056697259   0.943573961
C    0.471836022  -0.006058187   5.541718960
H    0.587246066  -0.005483995   6.596732782
C    0.339766259  -0.006607922   4.335471660
H    0.221618143  -0.006345494   3.270966186
$end

$rem
method      ribws2
basis       aug-cc-pvdz
aux_basis   rimp2-aug-cc-pvdz
scf_convergence 8
thresh      14
integral_symmetry   false
$end

! METHOD keyword is not required for BW-s2
! MAX_CYCLES keyword is manually set to the default here
$bws2
alpha       400
max_cycles  50
$end
```

6.9 Maximum Physical Regularisation Perturbation Theory

6.9.1 Introduction

Maximum Physical Regularisaion can be understood both as an extension to size-consistent Brillouin-Wigner theory (BW-s2) as well as to Møller-Plesset perturbation theory (MP2). In either framework, it introduces additional terms into the unperturbed Hamiltonian which allow the user to manipulate ground state properties and energies with more parametric precision. The second order energy in MPR-BWs(2) is given by an equation superficially reminiscient of both MP2 and BW-s2:

$$E_{\mathrm{c}}^{(2)} = \frac{1}{4}\sum_{ijab}\langle ab||ij\rangle t_{ij}^{ab(1)} \tag{6.33}$$

where the first order amplitudes $t_{ij}^{ab(1)}$ are determined by their linear amplitude equation. In strong analogy to BW-s2, MPR-BWs2 contains two matrices W_{ij} and W_{ab} which augment and mould the phyiscal description of the unperturbed ground state. The MPR-BWs2 amplitude equation is given by:

$$\sum_{c}\left((f_{ac}+W_{ac})t_{ij}^{cb(1)}+(f_{cb}+W_{cb})t_{ij}^{ac(1)}\right)-\sum_{k}\left((f_{ik}+W_{ik})t_{kj}^{ab(1)}+(f_{kj}+W_{kj})t_{ik}^{ab(1)}\right)+\langle ij||ab\rangle=0 \tag{6.34}$$

The central distinction between MPR-BWs2, BW-s2 and MP2 lies in the definition of W_{ab} and W_{ij}. The occupied block in MPR-BWs2 has five distinct contributions. The first term (A_0) is identical to BW-s2 and is characterised by the fact that it traces out to the second order energy multiplied by A_0:

$$W_{ij} \leftarrow \frac{A_0}{8}\left(t_{ik}^{ab\,(1)}\langle ab||jk\rangle + t_{jk}^{ab\,(1)}\langle ab||ik\rangle\right) \tag{6.35}$$

Analogously, the second term (A_1) traces out to the opposite-spin part of the second order correlation energy and can best be written in spatial orbitals as:

$$W_{ij} \leftarrow \frac{A_1}{2}\left(t_{i\bar{k}}^{a\bar{b}\,(1)}(ia|\overline{kb}) + t_{j\bar{k}}^{a\bar{b}\,(1)}(ja|\overline{kb})\right) \tag{6.36}$$

The remaining occupied terms contain the pseudo-density matrices ρ_{ij} and ρ_{ab}, which are defined in analogy to MP2, but are notably distinct from the true second order MPR-BWs2 density matrix:

$$\rho_{ij} = -\frac{1}{2}t_{ik}^{ab\,(1)}t_{jk}^{ab\,(1)} \tag{6.37}$$

$$\rho_{ab} = \frac{1}{2}t_{ij}^{ac\,(1)}t_{ij}^{bc\,(1)} \tag{6.38}$$

The A_2 and A_3 terms are constructed as follows:

$$W_{ij} \leftarrow A_2\langle ik||jl\rangle\rho_{kl} + A_3\langle ia||jb\rangle\rho_{ab} \tag{6.39}$$

Finally, the A_4 term is given by a symmetric contraction with the fock matrix:

$$W_{ij} \leftarrow \frac{A_4}{2}\left(f_{ik}\rho_{kj} + f_{jk}\rho_{ki}\right) \tag{6.40}$$

The virtual parameters are labelled according to their occupied counterparts, wherefore the B_0 and B_1 parameters are:

$$W_{ab} \leftarrow \frac{B_0}{8}\left(t_{ij}^{ac\,(1)}\langle bc||ij\rangle + t_{ij}^{bc\,(1)}\langle ac||ij\rangle\right) \tag{6.41}$$

$$W_{ab} \leftarrow \frac{B_1}{2}\left(t_{i\bar{j}}^{a\bar{c}\,(1)}(ib|\overline{jc}) + t_{i\bar{j}}^{b\bar{c}\,(1)}(ia|\overline{jc})\right) \tag{6.42}$$

The B_2 parameter, which would contract ρ_{ab} into W_{ab} is deliberately not implemented, since its evaluation contains a step of order $O(V^4)$, where V is the number of virtual orbitals. The B_3 and B_4 terms are given by:

$$W_{ab} \leftarrow B_3\langle ai||bj\rangle\rho_{ij} \tag{6.43}$$

$$W_{ab} \leftarrow \frac{B_4}{2}\left(f_{ac}\rho_{cb} + f_{bc}\rho_{ca}\right) \tag{6.44}$$

For an interpretation of these parameters and their respective terms, see Ref. 26.

Since terms with quadratic dependence on the amplitudes subsequently turn the amplitude equation into a cubic tensor equation, convergence of the self-consistent construction scheme described in section 6.8 is sometimes difficult or impossible to achieve. For this reason, it can be best to approximate the effect of each parameter non-iteratively. As can be seen from the Hylleraas functional, the second order correlation energy can be rewritten as:

$$E_{\mathrm{c}}^{(2)} = -\sum_{pq} P_{pq}^{(2)}(f_{pq} + W_{pq}) \tag{6.45}$$

where $P_{pq}^{(2)}$ denotes the exact second order density matrix. Replacing $P_{ij}^{(2)}$ with ρ_{ij} and $P_{ab}^{(2)}$ with ρ_{ab} and subtracting the iterative contribution defines the non-iterative approximation:

$$E_{\mathrm{c,\,noniter}}^{(2)} = \sum_{ij}\rho_{ij}W_{ij}^{\mathrm{noniter}} - \sum_{ab}\rho_{ab}W_{ab}^{\mathrm{noniter}} \tag{6.46}$$

6.9.2 MPR-BWs2 Job Control

MPR-BWs2 has been implemented as a specific partition of the RI-BW-s2 program. It is invoked by setting METHOD = RIBWS2 in the *$rem* section and writing METHOD = MPR-BWS2 in the *$bws2* section. Iterative and non-iterative parameters as well as additional variables to control the convergence behaviour of the iterative method are also defined in the *$bws2* section. See 6.8 for variables unrelated to MPR-BWs2

METHOD

Defines the correlation energy expression to be used.

INPUT SECTION: *$bws2*

TYPE:

STRING

DEFAULT:

BWS2

OPTIONS:

MPR-BWS2

RECOMMENDATION:

In order to activate MPR-BWs2, the method keyword has to be set to MPR-BWS2 or MPRBWS2

MPR_A0

Controls the value of the A_0 parameter.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n where $n/1000$ is the value of A_0.

RECOMMENDATION:

Since A_0 is equal to α in BW-s2 theory, its standard value is set to $\alpha = A_0 = 1$ and recreates BW-s2. If both MPR_A0 and ALPHA are defined in the input file, the last read takes priority.

MPR_A1

Controls the value of the A_1 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of A_1.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_A2

Controls the value of the A_2 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of A_2.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_A3

Controls the value of the A_3 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of A_3.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_A4

Controls the value of the A_4 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of A_4.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_B0

Controls the value of the B_0 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of B_0.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_B1

Controls the value of the B_1 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of B_1.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_B3

Controls the value of the B_3 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of B_3.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_B4

Controls the value of the B_4 parameter..

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the value of B_4.

RECOMMENDATION:

Large values lead to convergence issues.

MPR_A0_NONITER

Controls the value of the A_0 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of A_0.

RECOMMENDATION:

Arbitrary values are possible.

MPR_A1_NONITER

Controls the value of the A_1 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of A_1.

RECOMMENDATION:

Arbitrary values are possible.

MPR_A2_NONITER

Controls the value of the A_2 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of A_2.

RECOMMENDATION:

Arbitrary values are possible.

MPR_A3_NONITER

Controls the value of the A_3 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of A_3.

RECOMMENDATION:

Arbitrary values are possible.

MPR_A4_NONITER

Controls the value of the A_4 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of A_4.

RECOMMENDATION:

Arbitrary values are possible.

MPR_B0_NONITER

Controls the value of the B_0 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of B_0.

RECOMMENDATION:

Arbitrary values are possible.

MPR_B1_NONITER

Controls the value of the B_1 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of B_1.

RECOMMENDATION:

Arbitrary values are possible.

MPR_B3_NONITER

Controls the value of the B_3 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of B_3.

RECOMMENDATION:

Arbitrary values are possible.

MPR_B4_NONITER

Controls the value of the B_4 parameter for non-iterative contributions.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n where $n/1000$ is the non-iterative value of B_4.

RECOMMENDATION:

Arbitrary values are possible.

PARAM_GRADIENT

Controls whether or not to compute the derivative of the energy with respect to parameters.

INPUT SECTION: *$bws2*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Computes the iterative and non-iterative parameter gradient.

FALSE Does not compute the parameter gradient.

RECOMMENDATION:

If required, set to TRUE.

Example 6.23 MPR-BWs2 calculation on formaldehyde with $A_0 = 1$ and $B_0 = -1$.

```
$comment
    Standard MPR-BWs2 calculation
$end

$molecule
0 1
C    0.0000      0.0000      0.0000
O    0.0000      0.0000     -1.2195
H    0.9391      0.0000      0.5422
H   -0.9391      0.0000      0.5422
$end

$rem
JOBTYPE             SP
METHOD              RIBWS2              ! activate RI-BWS2 driver
BASIS               DEF2-SVP
AUX_BASIS           RIMP2-DEF2-SVP      ! auxilliary basis is required
SYM_IGNORE          TRUE
SYMMETRY            FALSE
$end

$bws2
METHOD              MPR-BWS2            ! activate MPR-BWs2
MPR_A0              1000                ! A_0 = 1.0
MPR_B0              -1000               ! B_0 = -1.0
MAX_CYCLES          50
$end
```

Example 6.24 MPR-BWs2 calculation on formaldehyde with non-iterative contributions.

```
$comment
    MPR-BWs2 calculation with non-iterative contributions
$end

$molecule
0 1
C   0.0000      0.0000      0.0000
O   0.0000      0.0000     -1.2195
H   0.9391      0.0000      0.5422
H  -0.9391      0.0000      0.5422
$end

$rem
JOBTYPE             SP
METHOD              RIBWS2              ! activate RI-BWS2 driver
BASIS               DEF2-SVP
AUX_BASIS           RIMP2-DEF2-SVP      ! auxilliary basis is required
SYM_IGNORE          TRUE
SYMMETRY            FALSE
$end

$bws2
METHOD              MPR-BWS2            ! activate MPR-BWs2
MPR_A0              500                 ! A_0 = 0.5
MPR_B1_NONITER      -1500               ! non-iterative B_0 = -1.5
MAX_CYCLES          50
$end
```

Example 6.25 MPR-BWs2 calculation on formaldehyde including the parameter gradient.

```
$comment
    MPR-BWs2 calculation with parameter gradient
$end

$molecule
0 1
C    0.0000      0.0000      0.0000
O    0.0000      0.0000     -1.2195
H    0.9391      0.0000      0.5422
H   -0.9391      0.0000      0.5422
$end

$rem
JOBTYPE             SP
METHOD              RIBWS2              ! activate RI-BWS2 driver
BASIS               DEF2-SVP
AUX_BASIS           RIMP2-DEF2-SVP      ! auxilliary basis is required
SYM_IGNORE          TRUE
SYMMETRY            FALSE
$end

$bws2
METHOD              MPR-BWS2            ! activate MPR-BWs2
MPR_A0              1500                ! A_0 = 1.5
MAX_CYCLES          50
PARAM_GRADIENT      TRUE                ! activate parameter gradient
$end
```

6.10 Direct Random Phase Approximation Methods

6.10.1 Introduction

A useful $\mathcal{O}(N^4)$ approach called the direct random phase approximation (dRPA) based on the RI approximation is available. This particular implementation was added by Joonho Lee working with Martin Head-Gordon.[73] RI-dRPA has been applied to the thermochemistry[27] and non-covalent interaction problems[92] and often demonstrated superior performance over RI-MP2. In terms of the computational cost, RI-dRPA should be compared to the scaled-opposite-spin MP2 while theoretically it involves diagrams far beyond second-order and includes infinite-order diagrams similarly to coupled-cluster theory. In fact, one can view dRPA as a reduced coupled-cluster with doubles approach.[109] In a nutshell, we define the dRPA energy as

$$E = E_{\text{HF}} + E_{\text{c}}^{\text{dRPA}} \tag{6.47}$$

where using the plasmon formula we compute[36]

$$E_{\text{c}}^{\text{dRPA}} = \int_{-\infty}^{\infty} \frac{\mathrm{d}\omega}{4\pi} \text{tr}\left[\ln\left(\mathbf{I} + \mathbf{Q}(\omega)\right) - \mathbf{Q}(\omega)\right] \tag{6.48}$$

where

$$\mathbf{Q}(\omega) = 2\mathbf{B}^T\mathbf{D}(\mathbf{D}^2 + \omega^2\mathbf{I})^{-1}\mathbf{B} \tag{6.49}$$

with

$$B_{ia,P} = \sum_Q (ia|Q)(Q|P)^{-1/2} \tag{6.50}$$

$$D_{ia,jb} = \delta_{ij}\delta_{ab}(\epsilon_a - \epsilon_i) \tag{6.51}$$

In this form, the cost of computing the dRPA correlation is quartic-scaling which is comparable to SOS-MP2. To use this method, one must set METHOD = RIDRPA along with AUXBASIS.

6.10.2 dRPA Job Control

CLENSHAW_NGRID

Number of grid points for the Curtis-Clenshaw quadrature.

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

RECOMMENDATION:

Use default.

Example 6.26 Example of RI-DRPA with RHF orbitals applied to a water dimer

```
$molecule
   0  1
   O    -2.766559     0.187083     0.566918
   H    -3.696304     1.179189    -0.642507
   H    -3.395838    -1.509891     0.389284
   O     2.587035     0.275900    -0.746442
   H     3.579141     0.918407     0.633058
   H     0.852266     0.311805    -0.156847
$end

$rem
   METHOD             RIDRPA
   BASIS              cc-pvdz
   AUX_BASIS_CORR     rimp2-cc-pvdz
   UNRESTRICTED       false
   THRESH             14
   INPUT_BOHR         true
   SCF_CONVERGENCE    8
   SCF_ALGORITHM      gdm
   MAXSCF             1000
   SCF_GUESS          sad
   CLENSHAW_NGRID     40 !number of grid points for the Curtis-Clenshaw quadrature
   INTEGRAL_SYMMETRY false
$end
```

6.11 Resolution-of-Identity MP3

6.11.1 Introduction

Joonho Lee working with Martin Head-Gordon developed and added an implementation of RI-MP3 with only cubic-storage requirement.[73] The resulting code can be applied to larger systems than what is possible in CCMAN2 (6.12) due to the improved storage scaling. To use this method, one can set METHOD = RIMP3 along with AUXBASIS. In essence, it applies the RI approximation in Eq. (8.4) to the MP3 correlation energy evaluation. Interested readers are referred to Section 6.3 for further details on the Møller-Plesset perturbation theory.

In addition to RI-MP3 one can incorporate scaled RI-MP2 and RI-MP3 correlation energy contributions and alternative reference molecular orbitals (DFT and OOMP2) as seen in recent works by Bertels *et al.*.[13] and Rettig *et al.*.[104] The use of non-HF orbital references requires a composite job where the first calculation generates the orbital reference and the second first recomputes the SCF energy and Fock with HF on the fixed orbital reference and then proceeds with the correlated calculation.

6.11.2 RI-MP3 Job Control

MP2_SCALING

Scales the RI-MP2 correlation energy contribution.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n corresponding to a scaling factor of $n/10^6$

RECOMMENDATION:

Use default.

MP3_SCALING

Scales the RI-MP3 correlation energy contribution.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n corresponding to a scaling factor of $n/10^6$

RECOMMENDATION:

Use default.

6.11.3 Examples

Example 6.27 RI-MP3 with RHF orbitals applied to a water dimer.

```
$molecule
   0  1
   O   -2.766559    0.187083    0.566918
   H   -3.696304    1.179189   -0.642507
   H   -3.395838   -1.509891    0.389284
   O    2.587035    0.275900   -0.746442
   H    3.579141    0.918407    0.633058
   H    0.852266    0.311805   -0.156847
$end

$rem
   METHOD            RIMP3
   BASIS             cc-pvdz
   UNRESTRICTED      false
   AUX_BASIS_CORR    rimp2-cc-pvdz
   THRESH            14
   INPUT_BOHR        true
   SCF_CONVERGENCE   8
   SCF_ALGORITHM     gdm
   MAXSCF            1000
   INTEGRAL_SYMMETRY false
$end
```

Example 6.28 Scaled RI-MP3 (RI-MP2.5) with BLYP orbitals applied to the water-ammonia dimer.

```
$molecule
   0  1
   O    0.0000000000 -0.0578657100 -1.4797930300
   H    0.0000000000  0.8229338400 -1.8554147400
   H    0.0000000000  0.0794956700 -0.5193425300
   N    0.0000000000  0.0143639400  1.4645462800
   H    0.0000000000 -0.9810485700  1.6534477900
   H   -0.8134835100  0.3987677600  1.9293404900
   H    0.8134835100  0.3987677600  1.9293404900
$end

$rem
   METHOD               blyp
   BASIS                cc-pvdz
   UNRESTRICTED         false
   THRESH               14
   SCF_CONVERGENCE      8
   SCF_ALGORITHM        gdm
   MAXSCF               1000
   GEN_SCFMAN           true
   GEN_SCFMAN_FINAL     true
   INTEGRAL_SYMMETRY    false
$end

@@@

$molecule
   read
$end

$rem
   CORRELATION          rimp3
   BASIS                cc-pvdz
   AUX_BASIS_CORR       rimp2-cc-pvdz
   UNRESTRICTED         false
   THRESH               14
   SCF_CONVERGENCE      8
   MAXSCF               0
   SCF_GUESS            read
   MP2_RESTART_NO_SCF   true
   SKIP_SCFMAN          true
   SCF_ALGORITHM        diis
   INTEGRAL_SYMMETRY    false
   GEN_SCFMAN           false   ! forces HF Fock build and energy evaluation
   GEN_SCFMAN_FINAL     false   ! forces HF Fock build and energy evaluation
   MP2_SCALING          1000000 ! 1.0
   MP3_SCALING          500000  ! 0.5
$end
```

Example 6.29 Scaled RI-MP3 (RI-MP2.8) with κ-OOMP2 orbitals applied to the hydrogen fluoride dimer.

```
$molecule
   0  1
   H    0.0000000000    0.8026798200    1.6952932900
   F    0.0000000000   -0.0459666600    1.3403481800
   H    0.0000000000   -0.1204078700   -0.4908284000
   F    0.0000000000    0.0097694500   -1.4042497800
$end

$rem
   EXCHANGE             hf
   BASIS                cc-pvdz
   UNRESTRICTED         false
   AUX_BASIS_CORR       rimp2-cc-pvdz
   DO_O2                3
   REGULARIZED_O2       2
   REG_VARIABLE         1450
   THRESH               14
   SCF_CONVERGENCE      8
   SCF_ALGORITHM        gdm
   MAXSCF               1000
   GEN_SCFMAN           true
   GEN_SCFMAN_FINAL     true
   INTEGRAL_SYMMETRY    false
$end

@@@

$molecule
   read
$end

$rem
   CORRELATION          rimp3
   BASIS                cc-pvdz
   UNRESTRICTED         false
   AUX_BASIS_CORR       rimp2-cc-pvdz
   THRESH               14
   SCF_CONVERGENCE      8
   MAXSCF               0
   SCF_GUESS            read    ! read in k-OOMP2 orbitals
   MP2_RESTART_NO_SCF   true
   SKIP_SCFMAN          true
   SCF_ALGORITHM        diis
   INTEGRAL_SYMMETRY    false
   GEN_SCFMAN           false   ! forces HF Fock build and energy evaluation
   GEN_SCFMAN_FINAL     false   ! forces HF Fock build and energy evaluation
   MP2_SCALING          1000000 ! 1.0
   MP3_SCALING          800000  ! 0.8
$end
```

6.12 Coupled-Cluster Methods

6.12.1 Introduction

The following sections give short summaries of the various coupled-cluster based methods available in Q-CHEM, most of which are variants of coupled-cluster theory. The basic object-oriented tools necessary to permit the implementation of these methods in Q-CHEM was accomplished by Anna Krylov and David Sherrill, working at Berkeley with Martin Head-Gordon, and then continuing independently at the University of Southern California and Georgia Tech, respectively. While at Berkeley, Krylov and Sherrill also developed the optimized orbital coupled-cluster method, with additional assistance from Ed Byrd. The extension of this code to MP3, MP4, CCSD and QCISD is the work of Steve Gwaltney at Berkeley, while the extensions to QCCD were implemented by Ed Byrd at Berkeley. The original tensor library and CC/EOM suite of methods are handled by the CCMAN module of Q-CHEM. Recently, a new code (termed CCMAN2) has been developed in Krylov group by Evgeny Epifanovsky and others, and a gradual transition from CCMAN to CCMAN2 has begun. During the transition time, both codes will be available for users via the CCMAN2 keyword.

CORRELATION

Specifies the correlation level of theory handled by CCMAN/CCMAN2.

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

CCMP2	Regular MP2 handled by CCMAN/CCMAN2
MP3	CCMAN and CCMAN2
MP4SDQ	CCMAN
MP4	CCMAN
CCD	CCMAN and CCMAN2
CCD(2)	CCMAN
CCSD	CCMAN and CCMAN2
CCSDT	CCMAN2
CC2	CCMAN2
CCSD(T)	CCMAN and CCMAN2
CCSD(2)	CCMAN
CCSD(fT)	CCMAN and CCMAN2
CCSD(dT)	CCMAN
CCSDT	CCMAN2
CCVB-SD	CCMAN2
QCISD	CCMAN and CCMAN2
QCISD(T)	CCMAN and CCMAN2
OD	CCMAN
OD(T)	CCMAN
OD(2)	CCMAN
VOD	CCMAN
VOD(2)	CCMAN
QCCD	CCMAN
QCCD(T)	CCMAN
QCCD(2)	CCMAN
VQCCD	CCMAN
VQCCD(T)	CCMAN
VQCCD(2)	CCMAN

RECOMMENDATION:

Consult the literature for guidance.

Note: All methods implemented in CCMAN2 can be executed in combination with PCM implicit solvation models (at a "zeroth-order" level, as described in Section 11.2.1) and with the EFP method (Section 11.5). Only energies and unrelaxed properties are available, not gradients.

6.12.2 Coupled Cluster Singles and Doubles (CCSD)

The standard approach for treating pair correlations self-consistently are coupled-cluster methods where the cluster operator contains all single and double substitutions,[99] abbreviated as CCSD. CCSD yields results that are only slightly superior to MP2 for structures and frequencies of stable closed-shell molecules. However, it is far superior for reactive species, such as transition structures and radicals, for which the performance of MP2 is quite erratic.

A full textbook presentation of CCSD is beyond the scope of this manual, and several comprehensive references are available. However, it may be useful to briefly summarize the main equations. The CCSD wave function is:

$$|\Psi_{\rm CCSD}\rangle = \exp\left(\hat{T}_1 + \hat{T}_2\right)|\Phi_0\rangle \tag{6.52}$$

where the single and double excitation operators may be defined by their actions on the reference single determinant (which is normally taken as the Hartree-Fock determinant in CCSD):

$$\hat{T}_1|\Phi_0\rangle = \sum_i^{\rm occ}\sum_a^{\rm virt} t_i^a\,|\Phi_i^a\rangle \tag{6.53}$$

$$\hat{T}_2|\Phi_0\rangle = \frac{1}{4}\sum_{ij}^{\rm occ}\sum_{ab}^{\rm virt} t_{ij}^{ab}\,|\Phi_{ij}^{ab}\rangle \tag{6.54}$$

It is not feasible to determine the CCSD energy by variational minimization of $\langle E\rangle_{\rm CCSD}$ with respect to the singles and doubles amplitudes because the expressions terminate at the same level of complexity as full configuration interaction (!). So, instead, the Schrödinger equation is satisfied in the subspace spanned by the reference determinant, all single substitutions, and all double substitutions. Projection with these functions and integration over all space provides sufficient equations to determine the energy, the singles and doubles amplitudes as the solutions of sets of nonlinear equations. These equations may be symbolically written as follows:

$$\begin{aligned}
E_{\rm CCSD} &= \langle\Phi_0|\hat{H}|\Psi_{\rm CCSD}\rangle \\
&= \left\langle \Phi_0\left|\hat{H}\right|\left(1+\hat{T}_1+\frac{1}{2}\hat{T}_1^2+\hat{T}_2\right)\Phi_0\right\rangle_C \qquad (6.55)\\
0 &= \left\langle \Phi_i^a\left|\hat{H}-E_{\rm CCSD}\right|\Psi_{\rm CCSD}\right\rangle \\
&= \left\langle \Phi_i^a\left|\hat{H}\right|\left(1+\hat{T}_1+\frac{1}{2}\hat{T}_1^2+\hat{T}_2+\hat{T}_1\hat{T}_2+\frac{1}{3!}\hat{T}_1^3\right)\Phi_0\right\rangle_C \qquad (6.56)\\
0 &= \left\langle \Phi_{ij}^{ab}\left|\hat{H}-E_{\rm CCSD}\right|\Psi_{\rm CCSD}\right\rangle \\
&= \left\langle \Phi_{ij}^{ab}\left|\hat{H}\right|\left(1+\hat{T}_1+\frac{1}{2}\hat{T}_1^2+\hat{T}_2+\hat{T}_1\hat{T}_2+\frac{1}{3!}\hat{T}_1^3\right.\right. \\
&\qquad \left.\left.+\frac{1}{2}\hat{T}_2^2+\frac{1}{2}\hat{T}_1^2\hat{T}_2+\frac{1}{4!}\hat{T}_1^4\right)\Phi_0\right\rangle_C \qquad (6.57)
\end{aligned}$$

The result is a set of equations which yield an energy that is not necessarily variational (*i.e.*, may not be above the true energy), although it is strictly size-consistent. The equations are also exact for a pair of electrons, and, to the extent that molecules are a collection of interacting electron pairs, this is the basis for expecting that CCSD results will be of useful accuracy.

The computational effort necessary to solve the CCSD equations can be shown to scale with the 6th power of the molecular size, for fixed choice of basis set. Disk storage scales with the 4th power of molecular size, and involves a number of sets of doubles amplitudes, as well as two-electron integrals in the molecular orbital basis. Therefore the improved accuracy relative to MP2 theory comes at a steep computational cost. Given these scalings it is relatively straightforward to estimate the feasibility (or non feasibility) of a CCSD calculation on a larger molecule (or with a larger basis set) given that a smaller trial calculation is first performed. Q-CHEM supports both energies and analytic gradients for CCSD for RHF and UHF references (including frozen core). For ROHF, only energies and unrelaxed properties are available. Available properties include dipole moments, angular momentum projections, $\langle\hat{S}^2\rangle$, static polarizabilities, and g-tensors (see Section 7.10.20 for details).

6.12.3 Coupled Cluster Singles, Doubles and Triples (CCSDT)

The coupled-cluster method with single, double and triple substitutions, abbreviated as CCSDT[93] includes single, double and triple excitation operators in the exponential ansatz. The theory of the method is very similar to that of CCSD – with triple excitations included fully. We only present the basic equations. These can be compared with the CCSD equations presented in the previous section, so as to understand the similarities and differences between CCSD and CCSDT. The CCSDT wave-function defined by

$$|\Psi_{\mathrm{CCSD}}\rangle = \exp\left(\hat{T}_1 + \hat{T}_2 + \hat{T}_3\right)|\Phi_0\rangle \tag{6.58}$$

where, the operators, $\hat{T}_1$ and $\hat{T}_2$ are defined using Eqs. 6.53 and 6.54. The operator, $\hat{T}_3$ is defined by

$$\hat{T}_3|\Phi_0\rangle = \frac{1}{36}\sum_{ijk}^{\mathrm{occ}}\sum_{abc}^{\mathrm{virt}} t_{ijk}^{abc}\left|\Phi_{ijk}^{abc}\right\rangle \tag{6.59}$$

The CCSDT equations are coupled non-linear simultaneous equations of the tensors, $\hat{T}_1$, $\hat{T}_2$ and $\hat{T}_3$. However, the correlation energy depends only on $\hat{T}_1$ and $\hat{T}_2$ amplitudes (The energy equations is same as Eq 6.55). The effect of triples is due to mutual coupling between singles, doubles and triples

$$\begin{aligned}
E_{\mathrm{CCSDT}} &= \langle\Phi_0|\hat{H}|\Psi_{\mathrm{CCSDT}}\rangle \\
&= \left\langle\Phi_0\left|\hat{H}\right|\left(1+\hat{T}_1+\frac{1}{2}\hat{T}_1^2+\hat{T}_2\right)\Phi_0\right\rangle_C && (6.60)\\
0 &= \left\langle\Phi_i^a\left|\hat{H}-E_{\mathrm{CCSDT}}\right|\Psi_{\mathrm{CCSDT}}\right\rangle \\
&= \left\langle\Phi_i^a\left|\hat{H}\right|\left(1+\hat{T}_1+\hat{T}_2+\hat{T}_3+\frac{1}{2}\hat{T}_1^2+\hat{T}_1\hat{T}_2+\frac{1}{3!}\hat{T}_1^3\right)\Phi_0\right\rangle_C && (6.61)\\
0 &= \left\langle\Phi_{ij}^{ab}\left|\hat{H}-E_{\mathrm{CCSDT}}\right|\Psi_{\mathrm{CCSDT}}\right\rangle \\
&= \left\langle\Phi_{ij}^{ab}\left|\hat{H}\right|\left(1+\hat{T}_1+\hat{T}_2+\hat{T}_3+\frac{1}{2}\hat{T}_1^2++\hat{T}_1\hat{T}_2+\hat{T}_1\hat{T}_3+\frac{1}{2}\hat{T}_2^2\right.\right. \\
&\qquad\left.\left.+\frac{1}{3!}\hat{T}_1^3+\frac{1}{2}\hat{T}_1^2\hat{T}_2+\frac{1}{4!}\hat{T}_1^4\right)\Phi_0\right\rangle_C && (6.62)\\
0 &= \left\langle\Phi_{ijk}^{abc}\left|\hat{H}-E_{\mathrm{CCSDT}}\right|\Psi_{\mathrm{CCSDT}}\right\rangle \\
&= \left\langle\Phi_{ijk}^{abc}\left|\hat{H}\right|\left(\hat{T}_2+\hat{T}_3+\hat{T}_1\hat{T}_2+\hat{T}_1\hat{T}_3+\hat{T}_2\hat{T}_3\right.\right. \\
&\qquad\left.\left.+\frac{1}{2}\hat{T}_2^2+\frac{1}{2}\hat{T}_1^2\hat{T}_2+\frac{1}{2}\hat{T}_1^2\hat{T}_3+\frac{1}{2}\hat{T}_1\hat{T}_2^2+\frac{1}{3!}\hat{T}_1^3\hat{T}_2\right)\Phi_0\right\rangle_C && (6.63)
\end{aligned}$$

Currently, the CCSDT functionality is available for computation of correlation energy only.

6.12.4 Second-Order Approximate Coupled Cluster Singles and Doubles (CC2)

The equations for the second-order approximate coupled cluster singles and doubles model (CC2)[17] are similar to the CCSD equations with the doubles amplitude equations approximated as:

$$E_{\mathrm{CC2}} = \langle\Phi_0\left|\hat{H}\exp\left(\hat{T}_1+\hat{T}_2\right)\right|\Phi_0\rangle \tag{6.64}$$

$$0 = \left\langle\Phi_i^a\left|\bar{H}+\left[\bar{H},\hat{T}_2\right]\right|\Phi_0\right\rangle \tag{6.65}$$

$$0 = \left\langle\Phi_{ij}^{ab}\left|\bar{H}+\left[\hat{F},\hat{T}_2\right]\right|\Phi_0\right\rangle \tag{6.66}$$

where the similarity-transformed Hamiltonian with the exponential function of the single excitation cluster operator is given by:

$$\bar{H} = \exp\left(-\hat{T}_1\right) \hat{H} \exp\left(\hat{T}_1\right) \tag{6.67}$$

CC2 energies are available in Q-CHEM, and are requested by setting the keyword METHOD to CC2. Closed and open-shell references (RHF/UHF/ROHF) are available, as well as the frozen core option. The RI approximation (RI-CC2) can be applied by specifying an auxiliary basis set. Furthermore, complex-valued calculations, CAP (Complex Absorbing Potentials) and CBF (Complex Basis Functions), are available for CC2 and RI-CC2 calculations (see Section 7.10.9 for details).

6.12.4.1 CC2 available in libgmbpt

Another implementation of CC2 is available in libgmbpt.[51] A partitioned form of the CC2 equations is employed, which eliminates the need to store double amplitudes. The resolution of the identity (RI) approximation for two-electron integrals can also be invoked to reduce the CPU time needed for calculation and I/O of these integrals.

This implementation can be invoked using the keyword METHOD=CC2 and setting CCMAN2=-1. As of the moment, this implementation is not yet optimized.

An implementation of stochatic resolution of indentity to CC2 (sRI-CC2) is also available in libgmbpt for ground state energy calculations by setting SRI=1 and modest SRI_NTHETA for the number of stochatic orbitals. Look through the samples for details. sRI-CC2 for excited energy (including singlet and triplet) will be soon released after solving some storage bugs.

6.12.5 Size-Consistent Brillouin-Wigner Second-Order Approximate Coupled Cluster Singles and Doubles (BWs-CC2)

Choosing to approximate the Hamiltonian in the CCSD equations using the unperturbed Møller-Plesset Hamiltonian is a popular, but not unique choice. Instead, it is possible to use the unperturbed size-consistent Brillouin-Wigner Hamiltonian[15,16] during evaluation of the Doubles equation. For a definition of this Hamiltonian, see section 6.8. In order to invoke this method, set METHOD = RIBWS2 in the REM section, and EXC_METHOD = CC2 in the *$bws2* section. BWs-CC2 is available for all partitions, recommended is the use of either BW-s2 or MPR-BWs2. Currently, only the electronic ground state is implemented. For general purposes it is recommended to employ MPR-BWs2 with $A_0 = 1$ and $B_0 = -1$ in BWs-CC2 calculations, though the optimal parameter set is dependent on the system and chemical quantity of interest.

6.12.6 Quadratic Configuration Interaction (QCISD)

Quadratic configuration interaction with singles and doubles (QCISD)[97] is a widely used alternative to CCSD, that shares its main desirable properties of being size-consistent, exact for pairs of electrons, as well as being also non variational. Its computational cost also scales in the same way with molecule size and basis set as CCSD, although with slightly smaller constants. While originally proposed independently of CCSD based on correcting configuration interaction equations to be size-consistent, QCISD is probably best viewed as approximation to CCSD. The defining equations are given below (under the assumption of Hartree-Fock orbitals, which should always be used in QCISD). The QCISD equations can clearly be viewed as the CCSD equations with a large number of terms omitted, which are

evidently not very numerically significant:

$$E_{\mathrm{QCISD}} = \left\langle \Phi_0 \left| \hat{H} \right| \left(1 + \hat{T}_2\right) \Phi_0 \right\rangle_C \tag{6.68a}$$

$$0 = \left\langle \Phi_i^a \left| \hat{H} \right| \left(\hat{T}_1 + \hat{T}_2 + \hat{T}_1\hat{T}_2\right) \Phi_0 \right\rangle_C \tag{6.68b}$$

$$0 = \left\langle \Phi_{ij}^{ab} \left| \hat{H} \right| \left(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2}\hat{T}_2^2\right) \Phi_0 \right\rangle_C \tag{6.68c}$$

QCISD energies are available in Q-CHEM, and are requested with the QCISD keyword. As discussed in Section 6.13, the non iterative QCISD(T) correction to the QCISD solution is also available to approximately incorporate the effect of higher substitutions.

6.12.7 Optimized Orbital Coupled Cluster Doubles (OD)

It is possible to greatly simplify the CCSD equations by omitting the single substitutions (*i.e.*, setting the T_1 operator to zero). If the same single determinant reference is used (specifically the Hartree-Fock determinant), then this defines the coupled-cluster doubles (CCD) method, as specified by the following equations:

$$E_{\mathrm{CCD}} = \left\langle \Phi_0 \left| \hat{H} \right| \left(1 + \hat{T}_2\right) \Phi_0 \right\rangle_C \tag{6.69a}$$

$$0 = \left\langle \Phi_{ij}^{ab} \left| \hat{H} \right| \left(1 + \hat{T}_2 + \frac{1}{2}\hat{T}_2^2\right) \Phi_0 \right\rangle_C \tag{6.69b}$$

The CCD method cannot itself usually be recommended because while pair correlations are all correctly included, the neglect of single substitutions causes calculated energies and properties to be significantly less reliable than for CCSD. Single substitutions play a role very similar to orbital optimization, in that they effectively alter the reference determinant to be more appropriate for the description of electron correlation (the Hartree-Fock determinant is optimized in the absence of electron correlation).

This suggests an alternative to CCSD and QCISD that has some additional advantages. This is the optimized orbital CCD method (OO-CCD), which we normally refer to as simply optimized doubles (OD).[110] The OD method is defined by the CCD equations above, plus the additional set of conditions that the cluster energy is minimized with respect to orbital variations. This may be mathematically expressed by

$$\frac{\partial E_{\mathrm{CCD}}}{\partial \theta_i^a} = 0 \tag{6.70}$$

where the rotation angle θ_i^a mixes the ith occupied orbital with the ath virtual (empty) orbital. Thus the orbitals that define the single determinant reference are optimized to minimize the coupled-cluster energy, and are variationally best for this purpose. The resulting orbitals are approximate Brueckner orbitals.

The OD method has the advantage of formal simplicity (orbital variations and single substitutions are essentially redundant variables). In cases where Hartree-Fock theory performs poorly (for example artificial symmetry breaking, or non-convergence), it is also practically advantageous to use the OD method, where the HF orbitals are not required, rather than CCSD or QCISD. Q-CHEM supports both energies and analytical gradients using the OD method. The computational cost for the OD energy is more than twice that of the CCSD or QCISD method, but the total cost of energy plus gradient is roughly similar, although OD remains more expensive. An additional advantage of the OD method is that it can be performed in an active space, as discussed later, in Section 6.14.

6.12.8 Quadratic Coupled Cluster Doubles (QCCD)

The non variational determination of the energy in the CCSD, QCISD, and OD methods discussed in the above subsections is not normally a practical problem. However, there are some cases where these methods perform poorly. One such

example are potential curves for homolytic bond dissociation, using closed shell orbitals, where the calculated energies near dissociation go significantly below the true energies, giving potential curves with unphysical barriers to formation of the molecule from the separated fragments.[124] The Quadratic Coupled Cluster Doubles (QCCD) method[125] recently proposed by Troy Van Voorhis at Berkeley uses a different energy functional to yield improved behavior in problem cases of this type. Specifically, the QCCD energy functional is defined as

$$E_{\mathrm{QCCD}} = \left\langle \Phi_0 \left(1 + \hat{\Lambda}_2 + \frac{1}{2}\hat{\Lambda}_2^2\right) \left|\hat{H}\right| \exp\left(\hat{T}_2\right) \Phi_0 \right\rangle_C \tag{6.71}$$

where the amplitudes of both the $\hat{T}_2$ and $\hat{\Lambda}_2$ operators are determined by minimizing the QCCD energy functional. Additionally, the optimal orbitals are determined by minimizing the QCCD energy functional with respect to orbital rotations mixing occupied and virtual orbitals.

To see why the QCCD energy should be an improvement on the OD energy, we first write the latter in a different way than before. Namely, we can write a CCD energy functional which when minimized with respect to the $\hat{T}_2$ and $\hat{\Lambda}_2$ operators, gives back the same CCD equations defined earlier. This energy functional is

$$E_{\mathrm{CCD}} = \left\langle \Phi_0 \left(1 + \hat{\Lambda}_2\right) \left|\hat{H}\right| \exp\left(\hat{T}_2\right) \Phi_0 \right\rangle_C \tag{6.72}$$

Minimization with respect to the $\hat{\Lambda}_2$ operator gives the equations for the $\hat{T}_2$ operator presented previously, and, if those equations are satisfied then it is clear that we do not require knowledge of the $\hat{\Lambda}_2$ operator itself to evaluate the energy.

Comparing the two energy functionals, Eqs. (6.71) and (6.72), we see that the QCCD functional includes up through quadratic terms of the Maclaurin expansion of $\exp(\hat{\Lambda}_2)$ while the conventional CCD functional includes only linear terms. Thus the bra wave function and the ket wave function in the energy expression are treated more equivalently in QCCD than in CCD. This makes QCCD closer to a true variational treatment[124] where the bra and ket wave functions are treated precisely equivalently, but without the exponential cost of the variational method.

In practice QCCD is a dramatic improvement relative to any of the conventional pair correlation methods for processes involving more than two active electrons (*i.e.*, the breaking of at least a double bond, or, two spatially close single bonds). For example calculations, we refer to the original paper,[125] and the follow-up paper describing the full implementation.[14] We note that these improvements carry a computational price. While QCCD scales formally with the 6th power of molecule size like CCSD, QCISD, and OD, the coefficient is substantially larger. For this reason, QCCD calculations are by default performed as OD calculations until they are partly converged. Q-CHEM also contains some configuration interaction models (CISD and CISDT). The CI methods are inferior to CC due to size-consistency issues, however, these models may be useful for benchmarking and development purposes.

6.12.9 Resolution of the Identity with CC (RI-CC)

The RI approximation (see Section 6.6) can be used in coupled-cluster calculations, which substantially reduces the cost of integral transformation and disk storage requirements. The RI approximations may be used for integrals only such that integrals are generated in conventional MO form and canonical CC/EOM calculations are performed, or in a more complete version when modified CC/EOM equations are used such that the integrals are used in their RI representation. The latter version allows for more substantial savings in storage and in computational speed-up.

The RI for integrals is invoked when AUX_BASIS is specified. All two-electron integrals are used in RI decomposed form in CC when AUX_BASIS is specified.

By default, the integrals will be stored in the RI form and special CC/EOM code will be invoked. Keyword CC_DIRECT_RI allows one to use RI generated integrals in conventional form (by transforming RI integrals back to the standard format) invoking conventional CC procedures.

Note: RI for integrals is available for all CCMAN/CCMAN2 methods. CCMAN requires that the unrestricted reference be used, CCMAN2 does not have this limitation. The RI is also available for jobs that need analytic gradients. Full RI implementation (with integrals used in decomposed form) is only available for CCMAN2. For maximum computational efficiency, combine with FNO (see Sections 6.15.1 and 7.10.13) when appropriate.

6.12.10 Cholesky Decomposition with CC (CD-CC)

Two-electron integrals can be decomposed using Cholesky decomposition[31] giving rise to the same representation as in RI and substantially reducing the cost of integral transformation, disk storage requirements, and improving parallel performance:

$$(\mu\nu|\lambda\sigma) \approx \sum_{P=1}^{M} B_{\mu\nu}^{P} B_{\lambda\sigma}^{P}, \tag{6.73}$$

The rank of Cholesky decomposition, M, is typically 3-10 times larger than the number of basis functions N (Ref. 8); it depends on the decomposition threshold δ and is considerably smaller than the full rank of the matrix, $N(N+1)/2$ (Refs. 8,9,135). Cholesky decomposition removes linear dependencies in product densities $(\mu\nu|$,[8] allowing one to obtain compact approximation to the original matrix with accuracy, in principle, up to machine precision.

Decomposition threshold δ is the only parameter that controls accuracy and the rank of the decomposition. Cholesky decomposition is invoked by specifying CHOLESKY_TOL that defines the accuracy with which decomposition should be performed. For most calculations tolerance of $\delta = 10^{-3}$ gives a good balance between accuracy and compactness of the rank. Tolerance of $\delta = 10^{-2}$ can be used for exploratory calculations and $\delta = 10^{-4}$ for high-accuracy calculations. Similar to RI, Cholesky-decomposed integrals can be transformed back, into the canonical MO form, using CC_DIRECT_RI keyword.

Note: Cholesky decomposition is available for all CCMAN2 methods, including energy, analytic gradients, and properties calculations. For maximum computational efficiency, combine with FNO (see Sections 6.15.1 and 7.10.13) when appropriate.

6.12.11 Job Control Options

There are a large number of options for the coupled-cluster singles and doubles methods. They are documented in Appendix B, and, as the reader will find upon following this link, it is an extensive list indeed. Fortunately, many of them are not necessary for routine jobs. Most of the options for non-routine jobs concern altering the default iterative procedure, which is most often necessary for optimized orbital calculations (OD, QCCD), as well as the active space and EOM methods discussed later in Section 6.14. The more common options relating to convergence control are discussed there, in Section 6.14.6. Below we list the options that one should be aware of for routine calculations.

For memory options and parallel execution, see Section 6.17.

CC_CONVERGENCE

Overall convergence criterion for the coupled-cluster codes. This is designed to ensure at least n significant digits in the calculated energy, and automatically sets the other convergence-related variables (CC_E_CONV, CC_T_CONV, CC_THETA_CONV, CC_THETA_GRAD_CONV) [10^{-n}].

TYPE:

INTEGER

DEFAULT:

6 Energies.

7 Gradients.

OPTIONS:

n Corresponding to 10^{-n} convergence criterion. Amplitude convergence is set automatically to match energy convergence.

RECOMMENDATION:

Use the default

Note: For single point calculations, CC_E_CONV = 6 and CC_T_CONV = 4. Tighter amplitude convergence (CC_T_CONV = 5) is used for gradients and EOM calculations.

CC_DOV_THRESH

Specifies minimum allowed values for the coupled-cluster energy denominators. Smaller values are replaced by this constant during early iterations only, so the final results are unaffected, but initial convergence is improved when the HOMO-LUMO gap is small or when non-conventional references are used.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$abcde$ Integer code is mapped to $ab \times 10^{-de}$, *e.g.*, 2501 corresponds to 0.025, 99001 corresponds to 0.99, etc.

RECOMMENDATION:

Increase to 0.25, 0.5 or 0.75 for non convergent coupled-cluster calculations.

Note: Works only for CCMAN jobs, not enabled in CCMAN2.

CC_SCALE_AMP

If not 0, scales down the step for updating coupled-cluster amplitudes in cases of problematic convergence.

TYPE:

INTEGER

DEFAULT:

0 no scaling

OPTIONS:

$abcd$ Integer code is mapped to $abcd \times 10^{-2}$, *e.g.*, 90 corresponds to 0.9

RECOMMENDATION:

Use 0.9 or 0.8 for non convergent coupled-cluster calculations.

Note: Now available for both CCMAN and CCMAN2.

CC_MAX_ITER

Maximum number of iterations to optimize the coupled-cluster energy.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

n up to n iterations to achieve convergence.

RECOMMENDATION:

None

CC_PRINT

Controls the output from post-MP2 coupled-cluster module of Q-CHEM

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

$0 - 7$ higher values can lead to deforestation...

RECOMMENDATION:

Increase if you need more output and don't like trees

CHOLESKY_TOL

Tolerance of Cholesky decomposition of two-electron integrals

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Corresponds to a tolerance of 10^{-n}

RECOMMENDATION:

2 - qualitative calculations, 3 - appropriate for most cases, 4 - quantitative (error in total energy typically less than 1 μhartree)

CC_DIRECT_RI

Controls use of RI and Cholesky integrals in conventional (undecomposed) form

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE use all integrals in decomposed format

TRUE transform all RI or Cholesky integral back to conventional format

RECOMMENDATION:

By default all integrals are used in decomposed format allowing significant reduction of memory use. If all integrals are transformed back (TRUE option) no memory reduction is achieved and decomposition error is introduced, however, the integral transformation is performed significantly faster and conventional CC/EOM algorithms are used.

6.12.12 Examples

Example 6.30 A series of jobs evaluating the correlation energy (with core orbitals frozen) of the ground state of the NH_2 radical with three methods of coupled-cluster singles and doubles type: CCSD itself, CC2, OD, and QCCD.

```
$molecule
   0  2
   N
   H1  N  1.02805
   H2  N  1.02805  H1  103.34
$end

$rem
   METHOD          ccsd
   BASIS           6-31g*
   N_FROZEN_CORE   fc
$end

@@@

$molecule
   read
$end

$rem
   METHOD          cc2
   BASIS           6-31g*
   N_FROZEN_CORE   fc
$end

@@@

$molecule
   read
$end

$rem
   METHOD          od
   BASIS           6-31g*
   N_FROZEN_CORE   fc
$end

@@@

$molecule
   read
$end

$rem
   METHOD          qccd
   BASIS           6-31g*
   N_FROZEN_CORE   fc
$end
```

Example 6.31 A job for single point CCSDT energy of water with (core orbitals frozen)

```
$molecule
0 1
O
H 1 R1
H 1 R1 2 A

R1=1.0
A=104.0
$end

$rem
   BASIS  =  cc-pVDZ
   METHOD  =  CCSDT
   N_FROZEN_CORE FC
$end
```

Example 6.32 A job evaluating CCSD energy of water using RI-CCSD

```
$molecule
  0 1
  O
  H1 O OH
  H2 O OH H1 HOH

  OH  = 0.947
  HOH = 105.5
$end

$rem
   METHOD          ccsd
   BASIS           aug-cc-pvdz
   AUX_BASIS       rimp2-aug-cc-pvdz
$end
```

Example 6.33 A job evaluating the RI-CC2 energy of water

```
$molecule
  0 1
  O
  H1 O OH
  H2 O OH H1 HOH

  OH  = 0.947
  HOH = 105.5
$end

$rem
   METHOD          cc2
   BASIS           aug-cc-pvdz
   AUX_BASIS       rimp2-aug-cc-pvdz
$end
```

Example 6.34 A job evaluating CCSD energy of water using CD-CCSD (tolerance = 10^{-3})

```
$molecule
  0 1
  O
  H1 O OH
  H2 O OH H1 HOH

  OH  = 0.947
  HOH = 105.5
$end

$rem
   METHOD          ccsd
   BASIS           aug-cc-pvdz
   CHOLESKY_TOL    3
$end
```

Example 6.35 A job evaluating CCSD energy of water using CD-CCSD (tolerance = 10^{-3}) with FNO

```
$molecule
   0 1
   O
   H1 O OH
   H2 O OH H1 HOH

   OH  = 0.947
   HOH = 105.5
$end

$rem
   METHOD          ccsd
   BASIS           aug-cc-pvdz
   CHOLESKY_TOL    3
   CC_FNO_THRESH   9950
$end
```

Example 6.36 A job evaluating the BWs-CC2 energy of formaldehyde with $\alpha = 1$

```
$molecule
0 1
C   0.0000      0.0000      0.0000
O   0.0000      0.0000     -1.2195
H   0.9391      0.0000      0.5422
H  -0.9391      0.0000      0.5422
$end

$rem
JOBTYPE             SP
METHOD              RIBWS2              ! activate RI-BWS2 driver
BASIS               DEF2-SVP
AUX_BASIS           RIMP2-DEF2-SVP      ! auxilliary basis is required
SYM_IGNORE          TRUE
SYMMETRY            FALSE
$end

$bws2
METHOD              BWS2                ! Use BW-s2 partition
EXC_METHOD          CC2                 ! Do a subsequent BWs-CC2 calculation
ALPHA               1000                ! a = 1.0
MAX_CYCLES          50
$end
```

Example 6.37 A job evaluating the BWs-CC2 energy of formaldehyde using the MPR-BWs2 partitioning with $A_0 = 1$ and $B_0 = -1$.

```
$molecule
0 1
C    0.0000      0.0000      0.0000
O    0.0000      0.0000     -1.2195
H    0.9391      0.0000      0.5422
H   -0.9391      0.0000      0.5422
$end

$rem
JOBTYPE             SP
METHOD              RIBWS2              ! activate RI-BWS2 driver
BASIS               DEF2-SVP
AUX_BASIS           RIMP2-DEF2-SVP      ! auxilliary basis is required
SYM_IGNORE          TRUE
SYMMETRY            FALSE
$end

$bws2
METHOD              MPR-BWS2            ! Use MPR-BWs2 partition
EXC_METHOD          CC2                 ! Do a subsequent BWs-CC2 calculation
MPR_A0              1000                ! A_0 = 1.0
MPR_B0              -1000               ! B_0 = -1.0
MAX_CYCLES          50
$end
```

6.13 Non-Iterative Corrections to Coupled Cluster Energies

6.13.1 (T) Triples Corrections

To approach chemical accuracy in reaction energies and related properties, it is necessary to account for electron correlation effects that involve three electrons simultaneously, as represented by triple substitutions relative to the mean field single determinant reference, which arise in MP4. The best standard methods for including triple substitutions are the CCSD(T)[100] and QCISD(T) methods.[97] The accuracy of these methods is well-documented for many cases,[77] and in general is a very significant improvement relative to the starting point (either CCSD or QCISD). The cost of these corrections scales with the 7th power of molecule size (or the 4th power of the number of basis functions, for a fixed molecule size), although no additional disk resources are required relative to the starting coupled-cluster calculation. Q-CHEM supports the evaluation of CCSD(T) and QCISD(T) energies, as well as the corresponding OD(T) correction to the optimized doubles method discussed in the previous subsection. Gradients and properties are not yet available for any of these (T) corrections.

6.13.2 (2) Triples and Quadruples Corrections

While the (T) corrections discussed above have been extraordinarily successful, there is nonetheless still room for further improvements in accuracy, for at least some important classes of problems. They contain judiciously chosen terms from 4th- and 5th-order Møller-Plesset perturbation theory, as well as higher order terms that result from the fact that the converged cluster amplitudes are employed to evaluate the 4th- and 5th-order order terms. The (T) correction therefore depends upon the bare reference orbitals and orbital energies, and in this way its effectiveness still depends on the quality of the reference determinant. Since we are correcting a coupled-cluster solution rather than a single

determinant, this is an aspect of the (T) corrections that can be improved. Deficiencies of the (T) corrections show up computationally in cases where there are near-degeneracies between orbitals, such as stretched bonds, some transition states, open shell radicals, and diradicals.

Prof. Steve Gwaltney, while working at Berkeley with Martin Head-Gordon, has suggested a new class of non iterative correction that offers the prospect of improved accuracy in problem cases of the types identified above.[48] Q-CHEM contains Gwaltney's implementation of this new method, for energies only. The new correction is a true second-order correction to a coupled-cluster starting point, and is therefore denoted as (2). It is available for two of the cluster methods discussed above, as OD(2) and CCSD(2).[48,49] Only energies are available at present.

The basis of the (2) method is to partition not the regular Hamiltonian into perturbed and unperturbed parts, but rather to partition a similarity-transformed Hamiltonian, defined as $\tilde{H} = e^{-\hat{T}}\hat{H}e^{\hat{T}}$. In the truncated space (call it the p-space) within which the cluster problem is solved (*e.g.*, singles and doubles for CCSD), the coupled-cluster wave function is a true eigenvalue of $\tilde{H}$. Therefore we take the zero order Hamiltonian, $\tilde{H}^{(0)}$, to be the full $\tilde{H}$ in the p-space, while in the space of excluded substitutions (the q-space) we take only the one-body part of $\tilde{H}$ (which can be made diagonal). The fluctuation potential describing electron correlations in the q-space is $\tilde{H} - \tilde{H}^{(0)}$, and the (2) correction then follows from second-order perturbation theory.

The new partitioning of terms between the perturbed and unperturbed Hamiltonians inherent in the (2) correction leads to a correction that shows both similarities and differences relative to the existing (T) corrections. There are two types of higher correlations that enter at second-order: not only triple substitutions, but also quadruple substitutions. The quadruples are treated with a factorization ansatz, that is exact in 5th order Møller-Plesset theory,[64] to reduce their computational cost from N^9 to N^6. For large basis sets this can still be larger than the cost of the triples terms, which scale as the 7th power of molecule size, with a factor twice as large as the usual (T) corrections.

These corrections are feasible for molecules containing between four and ten first row atoms, depending on computer resources, and the size of the basis set chosen. There is early evidence that the (2) corrections are superior to the (T) corrections for highly correlated systems.[48] This shows up in improved potential curves, particularly at long range and may also extend to improved energetic and structural properties at equilibrium in problematical cases. It will be some time before sufficient testing on the new (2) corrections has been done to permit a general assessment of the performance of these methods. However, they are clearly very promising, and for this reason they are available in Q-CHEM.

6.13.3 (dT) and (fT) corrections

Alternative inclusion of non-iterative N^7 triples corrections is described in Section 7.10.26. These methods called (dT) and (fT) are of similar accuracy to other triples corrections. CCSD(dT) and CCSD(fT) are equivalent to the CR-CCSD(T)$_L$ and CR-CCSD(T)$_2$ methods of Piecuch and coworkers.[94]

Note: Due to a violation of orbital invariance, the (dT) correction can sometimes lead to spurious results. Therefore, its use is discouraged. Use (fT) instead!

6.13.4 Job Control Options

The evaluation of a non-iterative (T) or (2) correction after a coupled-cluster singles and doubles level calculation (either CCSD, QCISD or OD) is controlled by the correlation keyword, and the specification of any frozen orbitals via N_FROZEN_CORE (and possibly N_FROZEN_VIRTUAL).

For the (2) correction, it is possible to apply the frozen core approximation in the reference coupled cluster calculation, and then correlate all orbitals in the (2) correction. This is controlled by CC_INCL_CORE_CORR, described below.

The default is to include core and core-valence correlation automatically in the CCSD(2) or OD(2) correction, if the reference CCSD or OD calculation was performed with frozen core orbitals. The reason for this choice is that core correlation is economical to include via this method (the main cost increase is only linear in the number of core orbitals), and such effects are important to account for in accurate calculations. This option should be made false if a job with explicitly frozen core orbitals is desired. One good reason for freezing core orbitals in the correction is if the basis set is physically inappropriate for describing core correlation (*e.g.*, standard Pople basis sets, and Dunning cc-pVxZ basis sets are designed to describe valence-only correlation effects). Another good reason is if a direct comparison is desired against another method such as CCSD(T) which is always used in the same orbital window as the CCSD reference.

There are several implementations of non-iterative triples available in Q-CHEM. In the original CCMAN suite, (T), (2), and (dT)/(fT) corrections can be computed. The parallel scaling of this code is very modest (4 cores max). CCMAN2 currently allows only the calculation of (T) correction for CCSD wave fucntions. By default, the CCMAN2 code is used for (T). The CCMAN code CCMAN2 is set to false. There are two versions of (T) in CCMAN2: The default version (native CCMAN2) and a new version using LIBPT. The implementation based on LIBPT is in-core OpenMP parallel. It is significantly faster in most realistic calculations (but it does not use point group symmetry, so it might show slower performance for small jobs with high symmetry). The LIBPT code is enabled by setting USE_LIBPT to true. The CCSD(T) calculation can be restarted using the *$ccsd_pt_restart* data printed during the CCSD(T) calculation. To restart the job simply copy the last printed *$ccsd_pt_restart* to your Q-CHEM input file.

Note: For the best performance of LIBPT (T) code, parallel execution should be requested, see Section 2.2.1.1.

USE_LIBPT

Enable LIBPT for CCSD(T) calculations in CCMAN2.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

LIBPT is now used by default in all real-valued CC/EOM-CC calculations

CC_INCL_CORE_CORR

Whether to include the correlation contribution from frozen core orbitals in non iterative (2) corrections, such as OD(2) and CCSD(2).

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

Use the default unless no core-valence or core correlation is desired (*e.g.*, for comparison with other methods or because the basis used cannot describe core correlation).

6.13.5 Examples

Example 6.38 Two jobs that compare the correlation energy calculated via the standard CCSD(T) method with the new CCSD(2) approximation, both using the frozen core approximation. This requires that CC_INCL_CORE_CORR must be specified as FALSE in the CCSD(2) input.

```
$molecule
   0  2
   O
   H  O  0.97907
$end

$rem
   METHOD          ccsd(t)
   BASIS           cc-pvtz
   N_FROZEN_CORE   fc
$end

@@@

$molecule
   read
$end

$rem
   METHOD              ccsd(2)
   BASIS               cc-pvtz
   N_FROZEN_CORE       fc
   CC_INCL_CORE_CORR   false
$end
```

Example 6.39 Using LIBPT for a standard CCSD(T) calculation

```
$molecule
   0  2
   O
   H  O  0.97907
$end

$rem
   METHOD          ccsd(t)
   BASIS           cc-pvtz
   N_FROZEN_CORE   fc
   USE_LIBPT       true
$end
```

6.14 Coupled Cluster Active Space Methods

6.14.1 Introduction

Electron correlation effects can be qualitatively divided into two classes. The first class is static or non-dynamical correlation: long wavelength low-energy correlations associated with other electron configurations that are nearly as low in energy as the lowest energy configuration. These correlation effects are important for problems such as homolytic bond breaking, and are the hardest to describe because by definition the single configuration Hartree-Fock description is not a good starting point. The second class is dynamical correlation: short wavelength high-energy

correlations associated with atomic-like effects. Dynamical correlation is essential for *quantitative* accuracy, but a reasonable description of static correlation is a prerequisite for a calculation being *qualitatively* correct.

In the methods discussed in the previous several subsections, the objective was to approximate the total correlation energy. However, in some cases, it is useful to model directly the non-dynamical and dynamical correlation energies separately. The reasons for this are pragmatic: with approximate methods, such a separation can give a more balanced treatment of electron correlation along bond-breaking coordinates, or reaction coordinates that involve diradicaloid intermediates. The non-dynamical correlation energy is conveniently defined as the solution of the Schrödinger equation within a small basis set composed of valence bonding, anti-bonding and lone pair orbitals: the so-called full valence active space. Solved exactly, this is the so-called full valence complete active space SCF (CASSCF),[105] or equivalently, the fully optimized reaction space (FORS) method.[107]

Full valence CASSCF and FORS involve computational complexity which increases exponentially with the number of atoms, and is thus unfeasible beyond systems of only a few atoms, unless the active space is further restricted on a case-by-case basis. Q-CHEM includes two relatively economical methods that directly approximate these theories using a truncated coupled-cluster doubles wave function with optimized orbitals.[63] They are active space generalizations of the OD and QCCD methods discussed previously in Sections 6.12.7 and 6.12.8, and are discussed in the following two subsections. By contrast with the exponential growth of computational cost with problem size associated with exact solution of the full valence CASSCF problem, these cluster approximations have only 6th-order growth of computational cost with problem size, while often providing useful accuracy.

The full valence space is a well-defined theoretical chemical model. For these active space coupled-cluster doubles methods, it consists of the union of *valence* levels that are occupied in the single determinant reference, and those that are empty. The occupied levels that are to be replaced can only be the occupied valence and lone pair orbitals, whose number is defined by the sum of the valence electron counts for each atom (*i.e.*, 1 for H, 2 for He, 1 for Li, *etc*..). At the same time, the empty virtual orbitals to which the double substitutions occur are restricted to be empty (usually anti-bonding) valence orbitals. Their number is the difference between the number of valence atomic orbitals, and the number of occupied valence orbitals given above. This definition (the full valence space) is the default when either of the "valence" active space methods are invoked (VOD or VQCCD)

There is also a second useful definition of a valence active space, which we shall call the 1:1 or perfect pairing active space. In this definition, the number of occupied valence orbitals remains the same as above. The number of empty correlating orbitals in the active space is defined as being exactly the same number, so that each occupied orbital may be regarded as being associated 1:1 with a correlating virtual orbital. In the water molecule, for example, this means that the lone pair electrons as well as the bond-orbitals are correlated. Generally the 1:1 active space recovers more correlation for molecules dominated by elements on the right of the periodic table, while the full valence active space recovers more correlation for molecules dominated by atoms to the left of the periodic table.

If you wish to specify either the 1:1 active space as described above, or some other choice of active space based on your particular chemical problem, then you must specify the numbers of active occupied and virtual orbitals. This is done via the standard "window options", documented earlier in this Chapter.

Finally we note that the entire discussion of active spaces here leads only to specific numbers of active occupied and virtual orbitals. The orbitals that are contained within these spaces are optimized by minimizing the trial energy with respect to all the degrees of freedom previously discussed: the substitution amplitudes, and the orbital rotation angles mixing occupied and virtual levels. In addition, there are new orbital degrees of freedom to be optimized to obtain the best active space of the chosen size, in the sense of yielding the lowest coupled-cluster energy. Thus rotation angles mixing active and inactive occupied orbitals must be varied until the energy is stationary. Denoting inactive orbitals by primes and active orbitals without primes, this corresponds to satisfying

$$\frac{\partial E_{\mathrm{CCD}}}{\partial \theta_i^{j'}} = 0 \tag{6.74}$$

Likewise, the rotation angles mixing active and inactive virtual orbitals must also be varied until the coupled-cluster energy is minimized with respect to these degrees of freedom:

$$\frac{\partial E_{\mathrm{CCD}}}{\partial \theta_a^{b'}} = 0 \tag{6.75}$$

6.14.2 VOD and VOD(2) Methods

The VOD method is the active space version of the OD method described earlier in Section 6.12.7. Both energies and gradients are available for VOD, so structure optimization is possible. There are a few important comments to make about the usefulness of VOD. First, it is a method that is capable of accurately treating problems that fundamentally involve 2 active electrons in a given local region of the molecule. It is therefore a good alternative for describing single bond-breaking, or torsion around a double bond, or some classes of diradicals. However it often performs poorly for problems where there is more than one bond being broken in a local region, with the non variational solutions being quite possible. For such problems the newer VQCCD method is substantially more reliable.

Assuming that VOD is a valid zero order description for the electronic structure, then a second-order correction, VOD(2), is available for energies only. VOD(2) is a version of OD(2) generalized to valence active spaces. It per-

mits more accurate calculations of relative energies by accounting for dynamical correlation.

Example 6.40 Calculate the correlation energy of the water molecule with partially stretched bonds, the VOD coupled-cluster active space method. This is a relatively "easy" job to converge, and may be contrasted with the next example, which is not easy to converge. The orbitals are restricted.

```
$molecule
  0  1
  O
  H  1  r
  H  1  r  2  a

  r =   1.5
  a = 104.5
$end

$rem
   METHOD        vod
   BASIS         6-31G
$end
```

Example 6.41 The water molecule with highly stretched bonds, calculated via the VOD coupled-cluster active space method. This is a "difficult" job to converge. The convergence options shown permitted the job to converge after some experimentation (thanks due to Ed Byrd for this!). The difficulty of converging this job should be contrasted with the previous example where the bonds were less stretched.

```
$molecule
   0  1
   O
   H  1  r
   H  1  r  2  a

   r =   3.0
   a = 104.5
$end

$rem
   METHOD               vod
   BASIS                6-31G
   SCF_CONVERGENCE      9
   THRESH               12
   CC_PRECONV_T2Z       50
   CC_PRECONV_T2Z_EACH  50
   CC_DOV_THRESH        7500
   CC_THETA_STEPSIZE    3200
   CC_DIIS_START        75
$end
```

6.14.3 VQCCD

The VQCCD method is the active space version of the QCCD method described earlier in Section 6.12.7. Both energies and gradients are available for VQCCD, so that structure optimization is possible. VQCCD is applicable to a substantially wider range of problems than the VOD method, because the modified energy functional is not vulnerable to non variational collapse. Testing to date suggests that it is capable of describing double bond breaking to similar accuracy as full valence CASSCF, and that potential curves for triple bond-breaking are qualitatively correct, although quantitatively in error by a few tens of kcal/mol. The computational cost scales in the same manner with system size as the VOD method, albeit with a significantly larger prefactor.

6.14.4 CCVB-SD

Working with Prof. Head-Gordon at Berkeley, Dr. D. W. Small and Joonho Lee have developed and implemented a novel single-reference coupled-cluster method with singles and doubles, called CCVB-SD.[72,113] CCVB-SD improves upon a more crude model CCVB (Section 6.19.3) and can be considered a simple modification to restricted CCSD (RCCSD). CCVB-SD inherits good properties from CCVB and RCCSD; it is spin-pure, size-extensive, and capable of breaking multiple bonds as long as only the valence space is correlated. It is a full doubles model and thus scales $\mathcal{O}(N^6)$. However, its energy is invariant under rotations in occupied space and virtual space, which makes it much more black-box than CCVB. Its energy function follows

$$E_{\mathrm{CCVB-SD}} = \left\langle \Phi_0 \left(1 + \hat{\Lambda}\right) \left|\hat{H}\right| \left(\exp\left(\hat{T}\right) - \hat{\mathrm{I}}_{\mathrm{S}} \frac{\hat{Q}^2}{2}\right) \Phi_0 \right\rangle_C \tag{6.76}$$

where $\hat{\mathrm{I}}_{\mathrm{S}}$ is a singlet projection operator and $\hat{Q}$ is a quintet doubles operator. Unlike QCCD, CCVB-SD improves the right eigenfunction while leaving the left eigenfunction unchanged. The quintet term in Eq. (6.76) represents approximate connected quadruples which are responsible for describing strong correlation. The cost of CCVB-SD is only twice as expensive as RCCSD, and it is better suited for strong correlation than QCCD/VQCCD in the sense that the method becomes exact at the dissociation limits of most multiple bond breaking whereas QCCD does not except special cases.

Although CCVB-SD can be used without the active space constraints, we recommend that users use it with the valence active space in general. For benchmarking purposes, using a minimal basis will automatically provide the valence space correctly with frozen cores. Both the energy and nuclear gradients of CCVB-SD are available through CCMAN2.

It should be noted that there is no orbital optimization implemented for CCVB-SD at the moment. This means that using basis sets larger than minimal basis requires choosing right valence orbitals to use. Therefore, we recommend that users run GVB-PP (or CCVB) to obtain orbitals to begin with. Orbital optimization (*i.e.* CCVB-OD) will soon be implemented and running CCVB-OD will be much more black-box than CCVB-SD as it does not require selecting proper valence space orbitals.

Furthermore, CCVB-SD can be applied to only closed-shell molecules at the moment. The extension to open-shell

molecules is under development.

Example 6.42 A CCVB-SD force calculation of benzene in a minimal basis.

```
$comment
CCVB-SD job for benzene computing energy+gradients.
It will also print out natural orbital occupation numbers (NOONs)
$end

$molecule
   0 1
   C    0.000000    0.698200    0.000000
   C    0.000000   -0.698200    0.000000
   C    1.209318    1.396400    0.000000
   C    1.209318   -1.396400    0.000000
   C    2.418636    0.698200    0.000000
   C    2.418636   -0.698200    0.000000
   H   -0.931410    1.235950    0.000000
   H   -0.931410   -1.235950    0.000000
   H    1.209318    2.471900    0.000000
   H    1.209318   -2.471900    0.000000
   H    3.350046    1.235950    0.000000
   H    3.350046   -1.235950    0.000000
$end

$rem
   JOBTYPE           force
   BASIS             sto-3g
   METHOD            ccvbsd
   THRESH            14
   SCF_ALGORITHM     gdm
   SCF_CONVERGENCE   10
   CC_REF_PROP       true
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end
```

6.14.5 Local Pair Models for Valence Correlations Beyond Doubles

Working with Prof. Head-Gordon at Berkeley, John Parkhill has developed implementations for pair models which couple 4 and 6 electrons together quantitatively. Because these truncate the coupled cluster equations at quadruples and hextuples respectively they have been termed the "Perfect Quadruples" and "Perfect Hextuples" models. These can be viewed as local approximations to CASSCF. The PQ and PH models are executed through an extension of Q-CHEM's coupled cluster code, and several options defined for those models will have the same effects although the mechanism may be different (CC_DIIS_START, CC_DIIS_SIZE, CC_DOV_THRESH, CC_CONV, *etc.*).

In the course of implementation, the non-local coupled cluster models were also implemented up to $\hat{T}_6$. Because the algorithms are explicitly sparse their costs relative to the existing implementations of CCSD are much higher (and should never be used in lieu of an existing CCMAN code), but this capability may be useful for development purposes, and when computable, models above CCSDTQ are highly accurate. To use PQ, PH, their dynamically correlated "+SD" versions or this machine generated cluster code set: METHOD = MGC.

MGC_AMODEL

Choice of approximate cluster model.

TYPE:

INTEGER

DEFAULT:

Determines how the CC equations are approximated:

OPTIONS:

0 Local Active-Space Amplitude iterations (pre-calculate GVB orbitals with your method of choice (RPP is good)).

7 Optimize-Orbitals using the VOD 2-step solver.
(Experimental-only use with MGC_AMPS = 2, 24 ,246)

8 Traditional Coupled Cluster up to CCSDTQPH.

9 MR-CC version of the Pair-Models. (Experimental)

RECOMMENDATION:

None

MGC_NLPAIRS

Number of local pairs on an amplitude.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

Must be greater than 1, which corresponds to the PP model. 2 for PQ, and 3 for PH.

RECOMMENDATION:

None

MGC_AMPS

Choice of Amplitude Truncation

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

$2 \leq n \leq 123456$, a sorted list of integers for every amplitude
which will be iterated. Choose 1234 for PQ and 123456 for PH

RECOMMENDATION:

None

MGC_LOCALINTS

Pair filter on an integrals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Enforces a pair filter on the 2-electron integrals, significantly
reducing computational cost. Generally useful for more than 1 pair locality.

RECOMMENDATION:

None

MGC_LOCALINTER

Pair filter on an intermediate.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Any nonzero value enforces the pair constraint on intermediates,
significantly reducing computational cost. Not recommended for ≤ 2 pair locality

RECOMMENDATION:

None

6.14.6 Convergence Strategies and More Advanced Options

These optimized orbital coupled-cluster active space methods enable the use of the full valence space for larger systems than is possible with conventional complete active space codes. However, we should note at the outset that often there are substantial challenges in converging valence active space calculations (and even sometimes optimized orbital coupled cluster calculations without an active space). Active space calculations cannot be regarded as "routine" calculations in the same way as SCF calculations, and often require a considerable amount of computational trial and error to persuade them to converge. These difficulties are largely because of strong coupling between the orbital degrees of freedom and the amplitude degrees of freedom, as well as the fact that the energy surface is often quite flat with respect to the orbital variations defining the active space.

Being aware of this at the outset, and realizing that the program has nothing against you personally is useful information for the uninitiated user of these methods. What the program does have, to assist in the struggle to achieve a converged solution, are accordingly many convergence options, fully documented in Appendix B. In this section, we describe the basic options and the ideas behind using them as a starting point. Experience plays a critical role, however, and so we encourage you to experiment with toy jobs that give rapid feedback in order to become proficient at diagnosing problems.

If the default procedure fails to converge, the first useful option to employ is CC_PRECONV_T2Z, with a value of between 10 and 50. This is useful for jobs in which the MP2 amplitudes are very poor guesses for the converged cluster amplitudes, and therefore initial iterations varying only the amplitudes will be beneficial:

CC_PRECONV_T2Z

Whether to pre-converge the cluster amplitudes before beginning orbital optimization in optimized orbital cluster methods.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

10 If CC_RESTART, CC_RESTART_NO_SCF or CC_MP2NO_GUESS are TRUE

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to n iterations in this pre-convergence procedure.

RECOMMENDATION:

Experiment with this option in cases of convergence failure.

Other options that are useful include those that permit some damping of step sizes, and modify or disable the standard DIIS procedure. The main choices are as follows.

CC_DIIS

Specify the version of Pulay's Direct Inversion of the Iterative Subspace (DIIS) convergence accelerator to be used in the coupled-cluster code.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Activates procedure 2 initially, and procedure 1 when gradients are smaller than DIIS12_SWITCH.

1 Uses error vectors defined as differences between parameter vectors from successive iterations. Most efficient near convergence.

2 Error vectors are defined as gradients scaled by square root of the approximate diagonal Hessian. Most efficient far from convergence.

RECOMMENDATION:

DIIS1 can be more stable. If DIIS problems are encountered in the early stages of a calculation (when gradients are large) try DIIS1.

CC_DIIS_START

Iteration number when DIIS is turned on. Set to a large number to disable DIIS.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n User-defined

RECOMMENDATION:

Occasionally DIIS can cause optimized orbital coupled-cluster calculations to diverge through large orbital changes. If this is seen, DIIS should be disabled.

CC_DOV_THRESH

Specifies minimum allowed values for the coupled-cluster energy denominators. Smaller values are replaced by this constant during early iterations only, so the final results are unaffected, but initial convergence is improved when the guess is poor.

TYPE:

INTEGER

DEFAULT:

2502 Corresponding to 0.25, 2501 corresponds to 0.025

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-de}$

RECOMMENDATION:

Increase to 0.5 or 0.75 for non convergent coupled-cluster calculations.

Note: Works only for CCMAN jobs, not enabled in CCMAN2.

CC_THETA_STEPSIZE

Scale factor for the orbital rotation step size. The optimal rotation steps should be approximately equal to the gradient vector.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 1.0

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-de}$
If the initial step is smaller than 0.5, the program will increase step
when gradients are smaller than the value of THETA_GRAD_THRESH,
up to a limit of 0.5.

RECOMMENDATION:

Try a smaller value in cases of poor convergence and very large orbital gradients. For example, a value of 01001 translates to 0.1

An even stronger—and more-or-less last resort—option permits iteration of the cluster amplitudes without changing the orbitals:

CC_PRECONV_T2Z_EACH

Whether to pre-converge the cluster amplitudes before each change of the orbitals in optimized orbital coupled-cluster methods. The maximum number of iterations in this pre-convergence procedure is given by the value of this parameter.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to n iterations in this pre-convergence procedure.

RECOMMENDATION:

A very slow last resort option for jobs that do not converge.

6.15 Alternative Orbitals for Correlated Calculations

6.15.1 Frozen Natural Orbitals

Large computational savings are possible if the virtual space is truncated using the frozen natural orbital (FNO) approach. For example, using a fraction f of the full virtual space results in a $1/(1-f)^4$-fold speed up for each CCSD iteration (CCSD scales with the forth power of the virtual space size). FNO-based truncation for ground-states CC methods was introduced by Bartlett and coworkers.[116,120,121] Extension of the FNO approach to ionized states within EOM-CC formalism was recently introduced and benchmarked;[65] see Section 7.10.13.

The FNOs are computed as the eigenstates of the virtual-virtual block of the MP2 density matrix [$\mathcal{O}(N^5)$ scaling], and the eigenvalues are the occupation numbers associated with the respective FNOs. By using a user-specified threshold, the FNOs with the smallest occupations are frozen in CC calculations. This could be done in CCSD, CCSD(T), CCSD(2), CCSD(dT), CCSD(fT) as well as CCD, OD, QCCD, VQCCD, and all possible triples corrections for these wave functions.

The truncation can be performed using two different schemes. The first approach is to simply specify the total number of virtual orbitals to retain, *e.g.*, as the percentage of total virtual orbitals, as was done in Refs. 120,121. The second approach is to specify the percentage of total natural occupation (in the virtual space) that needs to be recovered in the truncated space. These two criteria are referred to as the POVO (percentage of virtual orbitals) and OCCT (occupation threshold) cutoffs, respectively.[65]

Since the OCCT criterion is based on the correlation in a specific molecule, it yields more consistent results than POVO. For ionization energy calculations employing 99–99.5% natural occupation threshold should yields errors (relative to the full virtual space values) below 1 kcal/mol.[65] The errors decrease linearly as a function of the total natural occupation recovered, which can be exploited by extrapolating truncated calculations to the full virtual space values. This extrapolation scheme is called the extrapolated FNO (XFNO) procedure.[65] The linear behavior is exhibited by the total energies of the ground and the ionized states as a function of OCCT. Therefore, the XFNO scheme can be employed even when the two states are not calculated on the same level, *e.g.*, in adiabatic energy differences and EOM-IP-CC(2,3) calculations (more on this in Ref. 65).

The FNO truncation often causes slower convergence of the CCSD and EOM procedures. Nevertheless, despite larger number of iterations, the FNO-based truncation of orbital space reduces computational cost considerably, with a negligible decline in accuracy.[65]

Because of the limitation of the implementation, point-group symmetry cannot be used with FNO/OSFNO and will be disabled. Please, adjust your input consistently with CC_SYMMETRY = false.

For open-shell species an open-shell FNO (OSFNO) has been developed for the use in EOM-SF-CC calculations. The benchmarks show negligible errors in singlet–triplet gaps for a variety of molecules. For more details, see Section 7.10.13 and the Ref.[96].

CC_FNO_THRESH

Initialize the FNO truncation and sets the threshold to be used for both cutoffs (OCCT and POVO)

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

range 0000-10000

abcd Corresponding to $ab.cd\%$

RECOMMENDATION:

None

CC_FNO_USEPOP

Selection of the truncation scheme

TYPE:

INTEGER

DEFAULT:

1 OCCT

OPTIONS:

0 POVO

RECOMMENDATION:

None

Example 6.43 CCSD(T) calculation using FNO with POVO = 65%

```
$molecule
   0 1
   O
   H 1 1.0
   H 1 1.0 2 100.
$end

$rem
   METHOD          CCSD(T)
   BASIS           6-311+G(2df,2pd)
   CC_FNO_THRESH   6500   65% of the virtual space
   CC_FNO_USEPOP   0
$end
```

6.15.2 Non-Hartree-Fock Orbitals in Correlated Calculations

In cases of problematic open-shell references, *e.g.*, strongly spin-contaminated doublet radicals, one may choose to use DFT orbitals, which can yield significantly improved results.[10] This can be achieved by first doing DFT calculation and then reading the orbitals and turning the SCF procedure off. Note GEN_SCFMAN must be set to false for the orbitals to

be properly pseudocanonicalized.

Example 6.44 CCSD calculation of triplet methylene using B3LYP orbitals

```
$molecule
   0 3
   C
   H  1  CH
   H  1  CH  2  HCH

   CH  = 1.07
   HCH = 111.0
$end

$rem
   EXCHANGE        b3lyp
   BASIS           cc-pvdz
$end

@@@

$molecule
   read
$end

$rem
   BASIS           cc-pvdz
   METHOD          ccsd
   SCF_GUESS       read
   MAX_SCF_CYCLES  0
   N_FROZEN_CORE   1
$end
```

6.16 Analytic Gradients and Properties for Coupled-Cluster Methods

Analytic gradients are available for CCSD, OO-CCD/VOD, CCD, and QCCD/VQCCD methods for both closed- and open-shell references (UHF and RHF only), including frozen core and/or virtual functionality, as well as RI/Cholesky representations of the electron-repulsion integrals. Analytic gradients are available for CCVB-SD for only closed-shell references (RHF). In addition, gradients for selected GVB models are available.

For the CCSD and OO-CCD wave functions, Q-CHEM can also calculate dipole moments, $\langle r^2 \rangle$ (as well as $\langle x^2 \rangle$, $\langle y^2 \rangle$, and $\langle z^2 \rangle$ moments separately, which is useful for assigning different Rydberg states, *e.g.*, $3p_x$ vs. $3s$, *etc.*), and the $\langle \hat{S}^2 \rangle$ values. Interface of the CCSD and (V)OO-CCD codes with the NBO 5.0 package is also available. This code is closely related to EOM-CCSD properties/gradient calculations (Section 7.10.20). Solvent models available for CCSD are described in Chapter 11.2.

Limitations: Gradients and fully relaxed properties for ROHF and non-HF (*e.g.*, B3LYP) orbitals as well as RI approximation are not yet available.

Note: If gradients or properties are computed with frozen core/virtual, the algorithm will replace frozen orbitals to restricted. This will not affect the energies, but will change the orbital numbering in the CCMAN printout.

CC_REF_PROP

Whether or not the non-relaxed (expectation value) or full response (including orbital relaxation terms) one-particle CCSD properties will be calculated. The properties currently include permanent dipole moment, the second moments ($\langle x^2 \rangle$, $\langle y^2 \rangle$, and $\langle z^2 \rangle$) of the electron density along with $\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$. This option is incompatible with JOBTYPE = FORCE, OPT, or FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations need to be solved (λ-CCSD equations) for properties with the cost approximately the same as CCSD equations. Use the default if you do not need properties. The cost of the properties calculation itself is low. The CCSD one-particle density can be analyzed with NBO package by specifying NBO = TRUE, CC_REF_PROP = TRUE, and JOBTYPE = FORCE.

CC_REF_PROP_TE

Request for calculation of non-relaxed two-particle CCSD properties. The two-particle properties currently include $\langle \hat{S}^2 \rangle$. The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is small compared to the cost of $\langle \hat{S}^2 \rangle$. The variable CC_REF_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are computationally expensive, since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation). Do not request the two-particle properties unless you really need them.

CC_FULLRESPONSE

Fully relaxed properties (including orbital relaxation terms) will be computed. The variable CC_REF_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no orbital response will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Not available for non UHF/RHF references and for the methods that do not have analytic gradients (*e.g.*, QCISD).

Example 6.45 CCSD geometry optimization of HHeF followed up by properties calculations

```
$molecule
   0 1
   H    0.000000    0.000000   -1.886789
   He   0.000000    0.000000   -1.093834
   F    0.000000    0.000000    0.333122
$end

$rem
   JOBTYPE                     OPT
   METHOD                      CCSD
   BASIS                       aug-cc-pVDZ
   GEOM_OPT_TOL_GRADIENT       1
   GEOM_OPT_TOL_DISPLACEMENT   1
   GEOM_OPT_TOL_ENERGY         1
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE           SP
   METHOD            CCSD
   BASIS             aug-cc-pVDZ
   SCF_GUESS         READ
   CC_REF_PROP       1
   CC_FULLRESPONSE   1
$end
```

6.17 Memory Options and Parallelization of Coupled-Cluster Calculations

6.17.1 Introduction

The coupled-cluster suite of methods, which includes ground-state methods mentioned earlier in this Chapter and excited-state methods in the next chapter, has been parallelized to take advantage of distributed memory and multi-core architectures. The code is parallelized at the level of the underlying tensor algebra library.[30]

6.17.2 Serial and Shared Memory Parallel Jobs

Parallelization on multiple CPUs or CPU cores is achieved by breaking down tensor operations into batches and running each batch in a separate thread. Because each thread occupies one CPU core entirely, the maximum number of threads must not exceed the total available number of CPU cores. If multiple computations are performed simultaneously, they together should not run more threads than available cores. For example, an eight-core node can accommodate one eight-thread calculation, two four-thread calculations, and so on.

The number of threads to be used in a calculation is specified as a command line option (*-nt nthreads*). Here, *nthreads* should be given a positive integer value. If this option is not specified, the job will run in the serial mode.

Both CCMAN (old version of the couple-cluster codes) and CCMAN2 (default) have shared-memory parallel capabilities. However, they have different memory requirements as described below.

Setting the memory limit correctly is very important for attaining high performance when running large jobs. To roughly estimate the amount of memory required for a coupled-cluster calculation use the following formula:

$$\text{Memory} = \frac{(\text{Number of basis set functions})^4}{131072} \text{ MB} \tag{6.77}$$

If CCMAN2 is used and the calculation is based on a RHF reference, the amount of memory needed is a half of that given by the formula. If forces or excited states are calculated, the amount should be multiplied by a factor of two. Because the size of data increases steeply with the size of the molecule computed, both CCMAN and CCMAN2 are able to use disk space to supplement physical RAM if so required. The strategies of memory management in CCMAN and CCMAN2 slightly differ, and that should be taken into account when specifying memory-related keywords in the input file.

The MEM_STATIC keyword specifies the amount of memory in megabytes to be made available to routines that run prior to coupled-clusters calculations: Hartree-Fock and electronic repulsion integrals evaluation. A safe recommended value is 500 MB. The value of MEM_STATIC should not exceed 2000 MB even for very large jobs.

The memory limit for coupled-clusters calculations is set by CC_MEMORY. When running CCMAN, CC_MEMORY value is used as the recommended amount of memory, and the calculation can in fact use less or run over the limit. If the job is to run exclusively on a node, CC_MEMORY should be given 50% of all RAM. If the calculation runs out of memory, the amount of CC_MEMORY should be *reduced* forcing CCMAN to use memory-saving algorithms.

CCMAN2 uses a different strategy. It allocates the entire amount of RAM given by CC_MEMORY before the calculation and treats that as a strict memory limit. While that significantly improves the stability of larger jobs, it also requires the user to set the correct value of CC_MEMORY to ensure high performance. The default value is computed automatically based on the job size, but may not always be appropriate for large calculations, especially if the node has more resources available. When running CCMAN2 exclusively on a node, CC_MEMORY should be set to 75–80% of the total available RAM.

Note: When running small jobs, using too large CC_MEMORY in CCMAN2 is not recommended because Q-CHEM will allocate more resources than needed for the calculation, which may affect other jobs that you may wish to run on the same node.

For large disk-based coupled cluster calculations it is recommended to use a new tensor contraction code available in CCMAN2 via *libxm*, which can significantly speed up calculations on Linux nodes. Use the CC_BACKEND variable to switch on *libxm*. The new algorithm represents tensor contractions as multiplications of large matrices, which are performed using efficient *BLAS* routines. Tensor data is stored on disk and is asynchronously prefetched to fast memory before evaluating contractions. The performance of the code is not affected by the amount of RAM after about 128 GB if fast disks (such as SAS array in RAID0) are available on the system.

Note: When using *libxm* CC_BACKEND, sufficient MEM_TOTAL should be specified for integral transofromation (e.g., about 10 GB for a job with 500-700 basis functions).

6.17.3 Distributed Memory Parallel Jobs

CCMAN2 has capabilities to run ground and excited state energy and property calculations on computer clusters and supercomputers using the Cyclops Tensor Framework[115] (CTF) as a computational back-end. To switch on the use of CTF, use the CC_BACKEND keyword. In addition, Q-CHEM should be invoked with the *-np nproc* command line option to specify the number of processors for a distributed calculation as *nproc*. Consult Section 2.2.1.1 for more details about running Q-CHEM in parallel.

Note: This option is not yet available in public release.

6.17.4 Summary of Keywords

MEM_STATIC

Sets the memory for individual Fortran program modules

TYPE:

INTEGER

DEFAULT:

240 corresponding to 240 MB or 12% of MEM_TOTAL

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

For direct and semi-direct MP2 calculations, this must exceed OVN + requirements for AO integral evaluation (32–160 MB). Up to 2000 MB for large coupled-clusters calculations.

CC_MEMORY

Specifies the maximum size, in MB, of the buffers for in-core storage of block-tensors in CCMAN and CCMAN2.

TYPE:

INTEGER

DEFAULT:

50% of MEM_TOTAL. If MEM_TOTAL is not set, use 1.5 GB. A minimum of 192 MB is hard-coded.

OPTIONS:

n Integer number of MB

RECOMMENDATION:

Larger values can give better I/O performance and are recommended for systems with large memory (add to your `.qchemrc` file. When running CCMAN2 exclusively on a node, CC_MEMORY should be set to 75–80% of the total available RAM.)

CC_BACKEND

Used to specify the computational back-end of CCMAN2.

TYPE:

STRING

DEFAULT:

VM Default shared-memory disk-based back-end

OPTIONS:

XM *libxm* shared-memory disk-based back-end

INCORE in-core memory back-end

RECOMMENDATION:

Use XM for large jobs with limited memory or when the performance of the default disk-based back-end is not satisfactory, INCORE for small jobs that fit in main memory.

6.18 Using Single-Precision Arithmetic in Coupled-Cluster Calculations

Memory footprint and execution time of coupled-cluster calculations can be reduced by approximately a factor of 2 by using single-precision arithmetic.[95] The errors due to using single precision are small and comparable with typical convergence thresholds.[95] If loss of accuracy is undesirable, one can follow up a converged single-precision calculation with a small number of "clean-up" iterations in double precision, which recover the full accuracy of a double-precision calculation. Eventually, single-precision execution of coupled-cluster calculations will become the default, however, for now setting the single precision calculation needs to be done manually, as described below.

Single-precision capabilities are only available for the CCMAN2 suite. They work with both variants of the tensor-contraction backend (controlled by CC_BACKEND). Currently enabled features include energies, gradients, and properties calculations for both canonical and RI/CD versions. Single-precision versions of most EOM-CC methods are also available; the respective keywords are described in Section 7.10.14.

Note: When using single-precision, the thresholds need to be adjusted accordingly, as explained below. Using too tight convergence thresholds in single precision results in a non-convergent behavior.

Note: In rare cases, slow performance of the current code in single precision with default CC_BACKEND was noted; this issue is being investigated. Please monitor the performance and report any issues you encounter.

To deploy a single- or a mixed-precision coupled-cluster calculation, use CC_SINGLE_PREC. Its possible values are: 0 (default corresponding to double-precision calculation), 1 (single-precision calculation), and 2 (single-precision calculation followed by a couple of iterations in double precision, to recover full accuracy).

To adjust the convergence thresholds in single-precision calculation, use CC_SP_T_CONV (threshold for T and Λ amplitudes) and CC_SP_E_CONV (threshold for energies). Because too tight convergence criteria can cause non-convergent behavior, these thresholds should not be tighter than 10^{-6} a.u. for energies and 10^{-4} for amplitudes.

Calculations of intermediates, density matrices, and $\langle \hat{S}^2 \rangle$ can also be done in single precision, leading to negligible errors in energies and optimized geometries. This is controlled by the CC_SP_DM variable: 0 corresponds to calculation in double precision, 1 corresponds to calculation in single precision.

Calculation of perturbative triples corrections, (T) and (fT), can be executed in single precision using LIBPT; additional keywords deploying these features are USE_LIBPT and LIBPT_MIXED_PRECISION.

Note: Calculation of numerical derivatives in single precision, such as finite difference evaluation of nuclear gradients and finite-field calculations, require using larger step sizes or field strengths, respectively.

CC_SINGLE_PREC

Precision selection for CCSD calculation. Available in CCMAN2 only.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

2 single-precision calculation followed by double-precision clean-up iterations

RECOMMENDATION:

Do not set too tight convergence thresholds when using single precision

CC_SP_T_CONV

Amplitude convergence threshold in single precision in CCSD calculations.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Set 4 to be consistent with the default threshold in double precision in a pure single-precision run. When used with clean-up version, it should be smaller than double-precision threshold not to introduce extra iterations.

CC_SP_E_CONV

Energy convergence criterion in single precision in CCSD calculations.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Set 6 to be consistent with the default threshold in double precision in a pure single-precision calculation. When used with clean-up version, it should be smaller than double-precision threshold not to introduce extra iterations.

CC_SP_DM

Precision selection for CCSD and EOM-CCSD intermediates, density matrices, gradients, and $\langle \hat{S}^2 \rangle$.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

RECOMMENDATION:

NONE

CC_ERASE_DP_INTEGRALS

Controls storage of requisite objects computed with double precision in a single-precision calculation.

TYPE:

INTEGER

DEFAULT:

0 store

OPTIONS:

1 do not store

RECOMMENDATION:

Do not erase integrals if clean-up in double precision is intended.

LIBPT_MIXED_PRECISION

Deploys single-precision evaluation of (T) and (fT) within LIBPT.

TYPE:

INTEGER

DEFAULT:

0 do not use single precision

OPTIONS:

1 use single precision

RECOMMENDATION:

Use in combination with USE_LIBPT.

Example 6.46 A job evaluating CCSD and CCSD(T) energy using single-precision execution

```
$comment
   Uracil+H2O cc-pvdz
   CCSD(T) energy
$end

$molecule
   0 1
   N      0.034130   -0.986909    0.000000
   N     -1.173397    0.981920    0.000000
   C     -1.218805   -0.408164    0.000000
   C     -0.007302    1.702153    0.000000
   C      1.196200    1.107045    0.000000
   C      1.289085   -0.345905    0.000000
   O      2.310232   -0.996874    0.000000
   O     -2.257041   -1.026495    0.000000
   H      0.049329   -1.997961    0.000000
   H     -2.070598    1.437050    0.000000
   H     -0.125651    2.776484    0.000000
   H      2.111671    1.674079    0.000000
   O      1.747914   -1.338382   -3.040233
   H      2.180817   -1.817552   -2.333676
   H      0.813180   -1.472188   -2.883392
$end

$rem
   METHOD                  ccsd(t)
   BASIS                   cc-pvdz
   USE_LIBPT               true  evaluate triples with libpt code
   CC_SINGLE_PREC          1
   EOM_SINGLE_PREC         1
   CC_SP_DM                1
   CC_SP_T_CONV            4
   CC_SP_E_CONV            6
   CC_ERASE_DP_INTEGRALS   1
   LIBPT_MIXED_PRECISION   1
$end
```

6.19 Simplified Coupled-Cluster Methods Based on a Perfect-Pairing Active Space

6.19.1 Introduction

The methods described below are related to valence bond theory and are handled by the GVBMAN module. The following models are available:

CORRELATION

Specifies the correlation level in GVB models handled by GVBMAN.

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

PP

CCVB

GVB_IP

GVB_SIP

GVB_DIP

OP

NP

2P

RECOMMENDATION:

As a rough guide, use PP for biradicaloids, and CCVB for polyradicaloids involving strong spin correlations. Consult the literature for further guidance.

Molecules where electron correlation is strong are characterized by small energy gaps between the nominally occupied orbitals (that would comprise the Hartree-Fock wave function, for example) and nominally empty orbitals. Examples include so-called diradicaloid molecules,[60] or molecules with partly broken chemical bonds (as in some transition-state structures). Because the energy gap is small, electron configurations other than the reference determinant contribute to the molecular wave function with considerable amplitude, and omitting them leads to a significant error.

Including all possible configurations however, is a vast overkill. It is common to restrict the configurations that one generates to be constructed not from all molecular orbitals, but just from orbitals that are either "core" or "active". In this section, we consider just one type of active space, which is composed of two orbitals to represent each electron pair: one nominally occupied (bonding or lone pair in character) and the other nominally empty, or correlating (it is typically anti-bonding in character). This is usually called the perfect pairing active space, and it clearly is well-suited to represent the bonding/anti-bonding correlations that are associated with bond-breaking.

The quantum chemistry within this (or any other) active space is given by a Complete Active Space SCF (CASSCF) calculation, whose exponential cost growth with molecule size makes it prohibitive for systems with more than about 14 active orbitals. One well-defined coupled cluster (CC) approximation based on CASSCF is to include only double substitutions in the valence space whose orbitals are then optimized. In the framework of conventional CC theory, this defines the valence optimized doubles (VOD) model,[63] which scales as $\mathcal{O}(N^6)$ (see Section 6.14.2). This is still too expensive to be readily applied to large molecules.

The methods described in this section bridge the gap between sophisticated but expensive coupled cluster methods and inexpensive methods such as DFT, HF and MP2 theory that may be (and indeed often are) inadequate for describing molecules that exhibit strong electron correlations such as diradicals. The coupled cluster perfect pairing (PP),[11,18] imperfect pairing[123] (IP) and restricted coupled cluster[126] (RCC) models are local approximations to VOD that include only a linear and quadratic number of double substitution amplitudes respectively. They are close in spirit to generalized valence bond (GVB)-type wave functions,[44] because in fact they are all coupled cluster models for GVB that share the same perfect pairing active space. The most powerful method in the family, the Coupled Cluster Valence Bond (CCVB) method,[111–113] is a valence bond approach that goes well beyond the power of GVB-PP and related methods, as discussed below in Section 6.19.3.

6.19.2 Perfect Pairing (PP)

To be more specific, the coupled cluster PP wave function is written as

$$|\Psi\rangle = \exp\left(\sum_{i=1}^{n_{\text{active}}} t_i \hat{a}_{i*}^{\dagger} \hat{a}_{\bar{i}*}^{\dagger} \hat{a}_{\bar{i}} \hat{a}_i\right) |\Phi\rangle \tag{6.78}$$

where n_{active} is the number of active electrons, and the t_i are the linear number of unknown cluster amplitudes, corresponding to exciting the two electrons in the ith electron pair from their bonding orbital pair to their anti-bonding orbital pair. In addition to t_i, the core and the active orbitals are optimized as well to minimize the PP energy. The algorithm used for this is a slight modification of the GDM method, described for SCF calculations in Section 4.5.7. Despite the simplicity of the PP wave function, with only a linear number of correlation amplitudes, it is still a useful theoretical model chemistry for exploring strongly correlated systems. This is because it is exact for a single electron pair in the PP active space, and it is also exact for a collection of non-interacting electron pairs in this active space. Molecules, after all, are in a sense a collection of interacting electron pairs! In practice, PP on molecules recovers between 60% and 80% of the correlation energy in its active space.

If the calculation is perfect pairing (CORRELATION = PP), it is possible to look for unrestricted solutions in addition to restricted ones. Unrestricted orbitals are the default for molecules with odd numbers of electrons, but can also be specified for molecules with even numbers of electrons. This is accomplished by setting GVB_UNRESTRICTED = TRUE. Given a restricted guess, this will, however usually converge to a restricted solution anyway, so additional REM variables should be specified to ensure an initial guess that has broken spin symmetry. This can be accomplished by using an unrestricted SCF solution as the initial guess, using the techniques described in Chapter 4. Alternatively a restricted set of guess orbitals can be explicitly symmetry broken just before the calculation starts by using GVB_GUESS_MIX, which is described below. There is also the implementation of Unrestricted-in-Active Pairs (UAP),[69] which is the default unrestricted implementation for GVB methods. This method simplifies the process of unrestriction by optimizing only one set of ROHF MO coefficients and a single rotation angle for each occupied-virtual pair. These angles are used to construct a series of 2×2 Givens rotation matrices which are applied to the ROHF coefficients to determine the α spin MO coefficients and their transpose is applied to the ROHF coefficients to determine the β spin MO coefficients. This algorithm is fast and eliminates many of the pathologies of the unrestricted GVB methods near the dissociation limit. To generate a full potential curve we find it is best to start at the desired UHF dissociation solution as a guess for GVB and follow it inwards to the equilibrium bond distance.

GVB_UNRESTRICTED

Controls restricted versus unrestricted PP jobs. Usually handled automatically.

TYPE:

LOGICAL

DEFAULT:

same value as UNRESTRICTED

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set this variable explicitly only to do a UPP job from an RHF or ROHF initial guess. Leave this variable alone and specify UNRESTRICTED = TRUE to access the new unrestricted-in-active-pairs GVB code which can return an RHF or ROHF solution if used with GVB_DO_ROHF

GVB_DO_ROHF

Sets the number of Unrestricted-in-Active Pairs to be kept restricted.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-Defined

RECOMMENDATION:

If n is the same value as GVB_N_PAIRS returns the ROHF solution for GVB, only works with the UNRESTRICTED = TRUE implementation of GVB with GVB_OLD_UPP = 0 (its default value)

GVB_OLD_UPP

Which unrestricted algorithm to use for GVB.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use Unrestricted-in-Active Pairs described in Ref. 69

1 Use Unrestricted Implementation described in Ref. 11

RECOMMENDATION:

Only works for Unrestricted PP and no other GVB model.

GVB_GUESS_MIX

Similar to SCF_GUESS_MIX, it breaks alpha/beta symmetry for UPP by mixing the alpha HOMO and LUMO orbitals according to the user-defined fraction of LUMO to add the HOMO. 100 corresponds to a 1:1 ratio of HOMO and LUMO in the mixed orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined, $0 \leq n \leq 100$

RECOMMENDATION:

25 often works well to break symmetry without overly impeding convergence.

Whilst all of the description in this section refers to PP solved via projection, it is also possible, as described in Section 6.19.3 below, to solve variationally for the PP energy. This variational PP solution is the reference wave function for the CCVB method. In most cases use of spin-pure CCVB is preferable to attempting to improve restricted PP by permitting the orbitals to spin polarize.

6.19.3 Coupled Cluster Valence Bond (CCVB)

Cases where PP needs improvement include molecules with several strongly correlated electron pairs that are all localized in the same region of space, and therefore involve significant inter-pair, as well as intra-pair correlations. For some systems of this type, Coupled Cluster Valence Bond (CCVB) is an appropriate method.[111,112] CCVB is designed to qualitatively treat the breaking of covalent bonds. At the most basic theoretical level, as a molecular system dissociates

into a collection of open-shell fragments, the energy should approach the sum of the ROHF energies of the fragments. CCVB is able to reproduce this for a wide class of problems, while maintaining proper spin symmetry. Along with this, CCVB's main strength, come many of the spatial symmetry breaking issues common to the GVB-CC methods.

Like the other methods discussed in this section, the leading contribution to the CCVB wave function is the perfect pairing wave function, which is shown in Eq. (6.78). One important difference is that CCVB uses the PP wave function as a reference in the same way that other GVBMAN methods use a reference determinant.

The PP wave function is a product of simple, strongly orthogonal singlet geminals. Ignoring normalization, two equivalent ways of displaying these geminals are

$$\begin{aligned}(\phi_i\phi_i + t_i\phi_i^*\phi_i^*)(\alpha\beta - \beta\alpha) \quad &\text{(Natural-orbital form)} \\ \chi_i\chi_i'(\alpha\beta - \beta\alpha) \quad &\text{(Valence-bond form)},\end{aligned} \tag{6.79}$$

where on the left and right we have the spatial part (involving ϕ and χ orbitals) and the spin coupling, respectively. The VB-form orbitals are non-orthogonal within a pair and are generally AO-like. The VB form is used in CCVB and the NO form is used in the other GVBMAN methods. It turns out that occupied UHF orbitals can also be rotated (without affecting the energy) into the VB form (here the spin part would be just $\alpha\beta$), and as such we store the CCVB orbital coefficients in the same way as is done in UHF (even though no one spin is assigned to an orbital in CCVB).

These geminals are uncorrelated in the same way that molecular orbitals are uncorrelated in a HF calculation. Hence, they are able to describe uncoupled, or independent, single-bond-breaking processes, like that found in $C_2H_6 \rightarrow 2$ CH_3, but not coupled multiple-bond-breaking processes, such as the dissociation of N_2. In the latter system the three bonds may be described by three singlet geminals, but this picture must somehow translate into the coupling of two spin-quartet N atoms into an overall singlet, as found at dissociation. To achieve this sort of thing in a GVB context, it is necessary to correlate the geminals. The part of this correlation that is essential to bond breaking is obtained by replacing clusters of singlet geminals with triplet geminals, and re-coupling the triplets to an overall singlet. A triplet geminal is obtained from a singlet by simply modifying the spin component accordingly. We thus obtain the CCVB wave function:

$$\begin{aligned}|\Psi\rangle = |\Phi_0\rangle + \sum_{k<l} t_{kl}|\Phi_{(kl)}\rangle + \sum_{k<l<m<n} \Big[& t_{kl}t_{mn}|\Phi_{(kl)(mn)}\rangle \\ & + t_{km}t_{ln}|\Phi_{(km)(ln)}\rangle + t_{kn}t_{lm}|\Phi_{(kn)(lm)}\rangle \Big] + \cdots .\end{aligned} \tag{6.80}$$

In this expansion, the summations go over the active singlet pairs, and the indices shown in the labellings of the kets correspond to pairs that are being coupled as described just above. We see that this wave function couples clusters composed of even numbers of geminals. In addition, we see that the amplitudes for clusters containing more than 2 geminals are parameterized by the amplitudes for the 2-pair clusters. This approximation is important for computational tractability, but actually is just one in a family of CCVB methods: it is possible to include coupled clusters of odd numbers of pairs, and also to introduce independent parameters for the higher-order amplitudes. At present, only the simplest level is included in Q-CHEM.

Older methods which attempt to describe substantially the same electron correlation effects as CCVB are the IP[123] and RCC[126] wave functions. In general CCVB should be used preferentially. It turns out that CCVB relates to the GVB-IP model. In fact, if we were to expand the CCVB wave function relative to a set of determinants, we would see that for each pair of singlet pairs, CCVB contains only one of the two pertinent GVB-IP doubles amplitudes. Hence, for CCVB the various computational requirements and timings are very similar to those for GVB-IP. The main difference between the two models lies in how the doubles amplitudes are used to parameterize the quadruples, sextuples, etc., and this is what allows CCVB to give correct energies at full bond dissociation.

A CCVB calculation is invoked by setting CORRELATION = CCVB. The number of active singlet geminals must be specified by GVB_N_PAIRS. After this, an initial guess is chosen. There are three main options for this, specified by the following keyword

CCVB_GUESS

Specifies the initial guess for CCVB calculations

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 Standard GVBMAN guess (orbital localization via GVB_LOCAL + Sano procedure).

2 Use orbitals from previous GVBMAN calculation, along with SCF_GUESS = READ.

3 Convert UHF orbitals into pairing VB form.

RECOMMENDATION:

Option 1 is the most useful overall. The success of GVBMAN methods is often dependent on localized orbitals, and this guess shoots for these. Option 2 is useful for comparing results to other GVBMAN methods, or if other GVBMAN methods are able to obtain a desired result more efficiently. Option 3 can be useful for bond-breaking situations when a pertinent UHF solution has been found. It works best for small systems, or if the unrestriction is a local phenomenon within a larger molecule. If the unrestriction is non-local and the system is large, this guess will often produce a solution that is not the global minimum. Any UHF solution has a certain number of pairs that are unrestricted, and this will be output by the program. If GVB_N_PAIRS exceeds this number, the standard GVBMAN initial-guess procedure will be used to obtain a guess for the excess pairs

For potential energy surfaces, restarting from a previously computed CCVB solution is recommended. This is invoked by GVB_RESTART = TRUE. Whenever this is used, or any time orbitals are being read directly into CCVB from another calculation, it is important to also set:

- SCF_GUESS = READ
- MP2_RESTART_NO_SCF = TRUE
- SCF_ALGORITHM = DIIS

This bypasses orthogonalization schemes used elsewhere within Q-CHEM that are likely to jumble the CCVB guess.

In addition to the parent CCVB method as discussed up until now, we have included two related schemes for energy optimization, whose operation is controlled by the following keyword:

CCVB_METHOD

Optionally modifies the basic CCVB method

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Standard CCVB model

3 Independent electron pair approximation (IEPA) to CCVB

4 Variational PP (the CCVB reference energy)

RECOMMENDATION:

Option 1 is generally recommended. Option 4 is useful for preconditioning, and for obtaining localized-orbital solutions, which may be used in subsequent calculations. It is also useful for cases in which the regular GVBMAN PP code becomes variationally unstable. Option 3 is a simple independent-amplitude approximation to CCVB. It avoids the cubic-scaling amplitude equations of CCVB, and also is able to reach the correct dissociation energy for any molecular system (unlike regular CCVB which does so only for cases in which UHF can reach a correct dissociate limit). However the IEPA approximation to CCVB is sometimes variationally unstable, which we have yet to observe in regular CCVB.

Example 6.47 N_2 molecule in the intermediately dissociated region. In this case, SCF_ALGORITHM DIIS is necessary to obtain the symmetry unbroken RHF solution, which itself is necessary to obtain the proper CCVB solution. Note that many keywords general to GVBMAN are also used in CCVB.

```
$molecule
   0 1
   N  0  0  0
   N  0  0  2.0
$end

$rem
   UNRESTRICTED       = false
   BASIS              = 6-31g*
   EXCHANGE           = hf
   CORRELATION        = ccvb
   GVB_N_PAIRS        = 3
   CCVB_METHOD        = 1
   CCVB_GUESS         = 1
   GVB_LOCAL          = 2
   GVB_ORB_MAX_ITER   = 100000
   GVB_ORB_CONV       = 7
   GVB_RESTART        = false
   SCF_CONVERGENCE    = 10
   THRESH             = 14
   SCF_GUESS          = sad
   MP2_RESTART_NO_SCF = false
   SCF_ALGORITHM      = diis
   MAX_SCF_CYCLES     = 2000
   PRINT_ORBITALS     = true
   INTEGRAL_SYMMETRY    = false
   POINT_GROUP_SYMMETRY = false
$end
```

6.19.4 Second-Order Correction to Perfect Pairing: PP(2)

The PP and CCVB models are potential replacements for HF theory as a zero order description of electronic structure and can be used as a starting point for perturbation theory. They neglect all correlations that involve electron configurations with one or more orbitals that are outside the active space. Physically this means that the so-called "dynamic correlations", which correspond to atomic-like correlations involving high angular momentum virtual levels are neglected. Therefore, the GVB models may not be very accurate for describing energy differences that are sensitive to this neglected correlation energy, *e.g.*, atomization energies. It is desirable to correct them for this neglected correlation in a way that is similar to how the HF reference is corrected via MP2 perturbation theory.

For this purpose, the leading (second-order) correction to the PP model, termed PP(2),[12] has been formulated and efficiently implemented for restricted and unrestricted orbitals (energy only). PP(2) improves upon many of the worst failures of MP2 theory (to which it is analogous), such as for open shell radicals. PP(2) also greatly improves relative energies relative to PP itself. PP(2) is implemented using a resolution of the identity (RI) approach to keep the computational cost manageable. This cost scales in the same 5th-order way with molecular size as RI-MP2, but with a pre-factor that is about 5 times larger. It is therefore vastly cheaper than CCSD or CCSD(T) calculations which scale with the 6th and 7th powers of system size respectively. PP(2) calculations are requested with CORRELATION = PP(2). Since the only available algorithm uses auxiliary basis sets, it is essential to also provide a valid value for AUX_BASIS to have a complete input file.

The example below shows a PP(2) input file for the challenging case of the N2 molecule with a stretched bond. For this reason a number of the non-standard options discussed in Sections 6.19.2 and 6.19.5 for orbital convergence are enabled here. First, this case is an unrestricted calculation on a molecule with an even number of electrons, and so it is essential to break the alpha/beta spin symmetry in order to find an unrestricted solution. Second, we have chosen to leave the lone pairs uncorrelated, which is accomplished by specifying GVB_N_PAIRS.

Example 6.48 A non-standard PP(2) calculation. UPP(2) for stretched N2 with only 3 correlating pairs Try Boys localization scheme for initial guess.

```
$molecule
   0 1
   N
   N   1   1.65
$end

$rem
   UNRESTRICTED    true
   CORRELATION     pp(2)
   EXCHANGE        hf
   BASIS           cc-pvdz
   AUX_BASIS       rimp2-cc-pvdz   must use RI with PP(2)
   SCF_GUESS_MIX   10              mix SCF guess 100{\%}
   GVB_GUESS_MIX   25              mix GVB guess 25{\%} also!
   GVB_N_PAIRS     3               correlate only 3 pairs
   GVB_ORB_CONV    6               tighter convergence
   GVB_LOCAL       1               use Boys initial guess
$end
```

6.19.5 Other GVBMAN Methods and Options

In Q-CHEM, the unrestricted and restricted GVB methods are implemented with a resolution of the identity (RI) algorithm that makes them computationally very efficient.[114,127] They can be applied to systems with more than 100 active electrons, and both energies and analytical gradients are available. These methods are requested via the standard CORRELATION keyword. If AUX_BASIS is not specified, the calculation uses four-center two-electron integrals by

default. Much faster auxiliary basis algorithms (see Section 6.6 for an introduction), which are used for the correlation energy (not the reference SCF energy), can be enabled by specifying a valid string for AUX_BASIS. The example below illustrates a simple IP calculation.

Example 6.49 Imperfect pairing with auxiliary basis set for geometry optimization.

```
$molecule
   0 1
   H
   F   1   1.0
$end

$rem
   JOBTYPE       opt
   CORRELATION   gvb_ip
   BASIS         cc-pVDZ
   AUX_BASIS     rimp2-cc-pVDZ
$end
```

If further improvement in the orbitals are needed, the GVB_SIP, GVB_DIP, OP, NP and 2P models are also included.[69] The GVB_SIP model includes all the amplitudes of GVB_IP plus a set of quadratic amplitudes the represent the single ionization of a pair. The GVB_DIP model includes the GVB_SIP models amplitudes and the doubly ionized pairing amplitudes which are analogous to the correlation of the occupied electrons of the ith pair exciting into the virtual orbitals of the jth pair. These two models have the implementation limit of no analytic orbital gradient, meaning that a slow finite differences calculation must be performed to optimize their orbitals, or they must be computed using orbitals from a different method. The 2P model is the same as the GVB_DIP model, except it only allows the amplitudes to couple via integrals that span only two pairs. This allows for a fast implementation of it's analytic orbital gradient and enables the optimization of it's own orbitals. The OP method is like the 2P method except it removes the "direct"-like IP amplitudes and all of the same-spin amplitudes. The NP model is the GVB_IP model with the DIP amplitudes. This model is the one that works best with the symmetry breaking corrections that will be discussed later. All GVB methods except GVB_SIP and GVB_DIP have an analytic nuclear gradient implemented for both regular and RI four-center two-electron integrals.

There are often considerable challenges in converging the orbital optimization associated with these GVB-type calculations. The situation is somewhat analogous to SCF calculations but more severe because there are more orbital degrees of freedom that affect the energy (for instance, mixing occupied active orbitals amongst each other, mixing active virtual orbitals with each other, mixing core and active occupied, mixing active virtual and inactive virtual). Furthermore, the energy changes associated with many of these new orbital degrees of freedom are rather small and delicate. As a consequence, in cases where the correlations are strong, these GVB-type jobs often require many more iterations than the corresponding GDM calculations at the SCF level. This is a reflection of the correlation model itself. To deal with convergence issues, a number of REM values are available to customize the calculations, as listed below.

GVB_ORB_MAX_ITER

Controls the number of orbital iterations allowed in GVB-CC calculations. Some jobs, particularly unrestricted PP jobs can require 500–1000 iterations.

TYPE:

INTEGER

DEFAULT:

256

OPTIONS:

User-defined number of iterations.

RECOMMENDATION:

Default is typically adequate, but some jobs, particularly UPP jobs, can require 500–1000 iterations if converged tightly.

GVB_ORB_CONV

The GVB-CC wave function is considered converged when the root-mean-square orbital gradient and orbital step sizes are less than $10^{-\mathrm{GVB\_ORB\_CONV}}$. Adjust THRESH simultaneously.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n User-defined

RECOMMENDATION:

Use 6 for PP(2) jobs or geometry optimizations. Tighter convergence (*i.e.* 7 or higher) cannot always be reliably achieved.

GVB_ORB_SCALE

Scales the default orbital step size by n/1000.

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

n User-defined, 0–1000

RECOMMENDATION:

Default is usually fine, but for some stretched geometries it can help with convergence to use smaller values.

GVB_AMP_SCALE

Scales the default orbital amplitude iteration step size by n/1000 for IP/RCC. PP amplitude equations are solved analytically, so this parameter does not affect PP.

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

n User-defined, 0–1000

RECOMMENDATION:

Default is usually fine, but in some highly-correlated systems it can help with convergence to use smaller values.

GVB_RESTART

Restart a job from previously-converged GVB-CC orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Useful when trying to converge to the same GVB solution at slightly different geometries, for example.

GVB_REGULARIZE

Coefficient for GVB_IP exchange type amplitude regularization to improve the convergence of the amplitude equations especially for spin-unrestricted amplitudes near dissociation. This is the leading coefficient for an amplitude dampening term $-(c/10000)(e^{t_{ij}^p}-1)/(e^1-1)$

TYPE:

INTEGER

DEFAULT:

0 For restricted

1 For unrestricted

OPTIONS:

c User-defined

RECOMMENDATION:

Should be increased if unrestricted amplitudes do not converge or converge slowly at dissociation. Set this to zero to remove all dynamically-valued amplitude regularization.

GVB_POWER

Coefficient for GVB_IP exchange type amplitude regularization to improve the convergence of the amplitude equations especially for spin-unrestricted amplitudes near dissociation. This is the leading coefficient for an amplitude dampening term included in the energy denominator: -(c/10000)$(e^{t_{ij}^p}-1)/(e^1-1)$

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

p User-defined

RECOMMENDATION:

Should be decreased if unrestricted amplitudes do not converge or converge slowly at dissociation, and should be kept even valued.

GVB_SHIFT

Value for a statically valued energy shift in the energy denominator used to solve the coupled cluster amplitude equations, n/10000.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

Default is fine, can be used in lieu of the dynamically valued amplitude regularization if it does not aid convergence.

Another issue that a user of these methods should be aware of is the fact that there is a multiple minimum challenge associated with GVB calculations. In SCF calculations it is sometimes possible to converge to more than one set of orbitals that satisfy the SCF equations at a given geometry. The same problem can arise in GVB calculations, and based on our experience to date, the problem in fact is more commonly encountered in GVB calculations than in SCF calculations. A user may therefore want to (or have to!) tinker with the initial guess used for the calculations. One way is to set GVB_RESTART = TRUE (see above), to replace the default initial guess (the converged SCF orbitals which are then localized). Another way is to change the localized orbitals that are used in the initial guess, which is controlled by the GVB_LOCAL variable, described below. Sometimes different localization criteria, and thus different initial guesses, lead to different converged solutions. Using the new amplitude regularization keywords enables some control over the solution GVB optimizes.[68] A calculation can be performed with amplitude regularization to find a desired solution, and then the calculation can be rerun with GVB_RESTART = TRUE and the regularization turned off to remove the energy penalty of regularization.

GVB_LOCAL

Sets the localization scheme used in the initial guess wave function.

TYPE:

INTEGER

DEFAULT:

2 Pipek-Mezey orbitals

OPTIONS:

0 No Localization

1 Boys localized orbitals

2 Pipek-Mezey orbitals

RECOMMENDATION:

Different initial guesses can sometimes lead to different solutions. It can be helpful to try both to ensure the global minimum has been found.

GVB_DO_SANO

Sets the scheme used in determining the active virtual orbitals in a Unrestricted-in-Active Pairs GVB calculation.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 No localization or Sano procedure

1 Only localizes the active virtual orbitals

2 Uses the Sano procedure

RECOMMENDATION:

Different initial guesses can sometimes lead to different solutions. Disabling sometimes can aid in finding more non-local solutions for the orbitals.

Other *$rem* variables relevant to GVB calculations are given below. It is possible to explicitly set the number of active electron pairs using the GVB_N_PAIRS variable. The default is to make all valence electrons active. Other reasonable choices are certainly possible. For instance all electron pairs could be active ($n_{\mathrm{active}} = n_{\beta}$). Or alternatively one could make only formal bonding electron pairs active ($n_{\mathrm{active}} = N_{\mathrm{STO-3G}} - n_{\alpha}$). Or in some cases, one might want only the most reactive electron pair to be active (n_{active} =1). Clearly making physically appropriate choices for this variable is essential for obtaining physically appropriate results!

GVB_N_PAIRS

Alternative to CC_REST_OCC and CC_REST_VIR for setting active space size in GVB and valence coupled cluster methods.

TYPE:

INTEGER

DEFAULT:

PP active space (1 occ and 1 virt for each valence electron pair)

OPTIONS:

n user-defined

RECOMMENDATION:

Use the default unless one wants to study a special active space. When using small active spaces, it is important to ensure that the proper orbitals are incorporated in the active space. If not, use the *$reorder_mo* feature to adjust the SCF orbitals appropriately.

GVB_PRINT

Controls the amount of information printed during a GVB-CC job.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

Should never need to go above 0 or 1.

GVB_TRUNC_OCC

Controls how many pairs' occupied orbitals are truncated from the GVB active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

This allows for asymmetric GVB active spaces removing the n lowest energy occupied orbitals from the GVB active space while leaving their paired virtual orbitals in the active space. Only the models including the SIP and DIP amplitudes (*i.e.* NP and 2P) benefit from this all other models this equivalent to just reducing the total number of pairs.

GVB_TRUNC_VIR

Controls how many pairs' virtual orbitals are truncated from the GVB active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

This allows for asymmetric GVB active spaces removing the n highest energy occupied orbitals from the GVB active space while leaving their paired virtual orbitals in the active space. Only the models including the SIP and DIP amplitudes (*i.e.* NP and 2P) benefit from this all other models this equivalent to just reducing the total number of pairs.

GVB_REORDER_PAIRS

Tells the code how many GVB pairs to switch around.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n $0 \leq n \leq 5$

RECOMMENDATION:

This allows for the user to change the order the active pairs are placed in after the orbitals are read in or are guessed using localization and the Sano procedure. Up to 5 sequential pair swaps can be made, but it is best to leave this alone.

GVB_REORDER_1

Tells the code which two pairs to swap first.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 1.

GVB_REORDER_2

Tells the code which two pairs to swap second.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 2.

GVB_REORDER_3

Tells the code which two pairs to swap third.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 3.

GVB_REORDER_4

Tells the code which two pairs to swap fourth.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 4.

GVB_REORDER_5

Tells the code which two pairs to swap fifth.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 5.

It is known that symmetry breaking of the orbitals to favor localized solutions over non-local solutions is an issue with GVB methods in general. A combined coupled-cluster perturbation theory approach to solving symmetry breaking (SB) using perturbation theory level double amplitudes that connect up to three pairs has been examined in the literature,[66,67] and it seems to alleviate the SB problem to a large extent. It works in conjunction with the GVB_IP, NP, and 2P levels of correlation for both restricted and unrestricted wave functions (barring that there is no restricted implementation of the 2P model, but setting GVB_DO_ROHF to the same number as the number of pairs in the system is equivalent).

GVB_SYMFIX

Should GVB use a symmetry breaking fix.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 no symmetry breaking fix

1 symmetry breaking fix with virtual orbitals spanning the active space

2 symmetry breaking fix with virtual orbitals spanning the whole virtual space

RECOMMENDATION:

It is best to stick with type 1 to get a symmetry breaking correction with the best results coming from CORRELATION = NP and GVB_SYMFIX = 1.

GVB_SYMPEN

Sets the pre-factor for the amplitude regularization term for the SB amplitudes.

TYPE:

INTEGER

DEFAULT:

160

OPTIONS:

γ User-defined

RECOMMENDATION:

Sets the pre-factor for the amplitude regularization term for the SB amplitudes: $-(\gamma/1000)(e^{(c*100)*t^2}-1)$.

GVB_SYMSCA

Sets the weight for the amplitude regularization term for the SB amplitudes.

TYPE:

INTEGER

DEFAULT:

125

OPTIONS:

c User-defined

RECOMMENDATION:

Sets the weight for the amplitude regularization term for the SB amplitudes: $-(\gamma/1000)(e^{(c*100)*t^2}-1)$.

We have already mentioned a few issues associated with the GVB calculations: the neglect of dynamic correlation [which can be remedied with PP(2)], the convergence challenges and the multiple minimum issues. Another weakness of these GVB methods is the occasional symmetry-breaking artifacts that are a consequence of the limited number of retained pair correlation amplitudes. For example, benzene in the PP approximation prefers D_{3h} symmetry over D_{6h} by 3 kcal/mol (with a 2° distortion), while in IP, this difference is reduced to 0.5 kcal/mol and less than 1°.[123] Likewise the allyl radical breaks symmetry in the unrestricted PP model,[11] although to a lesser extent than in restricted open shell HF. Another occasional weakness is the limitation to the perfect pairing active space, which is not necessarily appropriate for molecules with expanded valence shells, such as in some transition metal compounds (*e.g.* expansion from $4s3d$ into $4s4p3d$) or possibly hyper-valent molecules (expansion from $3s3p$ into $3s3p3d$). The singlet strongly orthogonal geminal method (see the next section) is capable of dealing with expanded valence shells and could be used for such cases. The perfect pairing active space is satisfactory for most organic and first row inorganic molecules.

To summarize, while these GVB methods are powerful and can yield much insight when used properly, they do have enough pitfalls for not to be considered true "black box" methods.

6.20 Complete Active Space Methods

6.20.1 Introduction & Theory

The complete active space (CAS) methods are a family of methods for dealing with strongly correlated systems. In this method, a subset of a system's orbitals and electrons are denoted as *active* and the full configuration interaction (FCI) problem is solved exactly in this small active space. The remaining occupied orbitals are denoted *inactive* and are treated in a mean-field manner, while the remaining unoccupied orbitals are denoted *virtual*. In CAS-CI, this is the end of the matter. In CASSCF, the orbitals spanning these three spaces (inactive, active, and virtual) are then

optimized to obtain the lowest possible energy. In other words, the CASSCF problem is to find the optimal (by energy) partitioning of the orbital Hilbert space. This allows moderately sized systems to be studied as long as the active space is relatively small, due to combinatorial growth in the number of possible Slater determinants that encompass all possible configurations within the active space. Indeed, the total number of possible Slater determinants for an active space with M *spatial* orbitals, $N_\uparrow$ up spins and $N_\downarrow$ down spins is:

$$N_{\text{total}} = \frac{M!}{N_\uparrow!\,(M-N_\uparrow)!}\frac{M!}{N_\downarrow!\,(M-N_\downarrow)!} \tag{6.81}$$

Modern computing architectures can handle active spaces of approximately 18 electrons in 18 orbitals ($\approx 2 \times 10^9$ determinants), though we do not recommend using such a large active-space for routine calculations.

Nuclear gradients for CASSCF calculation are also available in Q-CHEM. In addition to full CAS calculations, arbitrary order truncated CI (CIS, CISD, CISDT, etc.) may also be carried out in the requested active space and orbitally optimized.

The electronic energy is an exact functional of the 1-RDM and 2-RDM

$$E = \frac{1}{2}\sum_{pqrs}\Gamma_{pqrs}g_{pqrs} + \sum_{pq}D_{pq}h_{pq}, \tag{6.82}$$

Given the 1- and 2-PDMs, the generalized Fock matrices may be generated for this MCSCF. The derivation and further details are neatly described by Helgaker, Jorgensen, and Olsen,[57] but the key results are summarized here. In the following, $m, n, p, q, \ldots$ are general indices, $i, j, k, \ldots$ are inactive indices, $t, u, v, w, \ldots$ are active indices, and $a, b, c, \ldots$ are virtual indices.

The generalized Fock matrix is defined as

$$F_{mn} = \sum_{q}D_{pq}h_{pq} + \sum_{qrs}\Gamma_{mqrs}g_{nqrs} \tag{6.83}$$

where h_{pq} are the 1-electron integrals and g_{nqrs} are the 2-electron integrals and all indices run over all orbital classes (inactive, active, and virtual). This, generally non-symmetric, matrix can be simplified by taking advantage of the fact that the form of the density matrices when some indices are inactive or virtual are much simpler than when the indices are active. When the first index of the generalized Fock matrix is inactive and the second is general:

$$F_{in} = 2({}^{I}F_{ni} + {}^{A}F_{ni}) \tag{6.84}$$

where the *inactive* and *active Fock matrices* are

$${}^{I}F_{mn} = h_{mn} + \sum_{i}(2g_{mnii} - g_{miin}) \tag{6.85}$$

$${}^{A}F_{mn} = \sum_{vw}D_{vw}(g_{mnvw} - g_{mwvn}) \tag{6.86}$$

In other words, the inactive Fock matrix is the Fock matrix formed from using only the inactive density and the active Fock matrix is sum of J and K matrices built from the active space 1-PDM. When the first index is active, and the second index is general, we have

$$F_{tn} = \sum_{u}{}^{I}F_{nu}D_{vu} + Q_{tn} \tag{6.87}$$

where the auxiliary Q matrix is

$$Q_{tm} = \sum_{u,v,w}\Gamma_{tuvw}g_{muvw} \tag{6.88}$$

and finally, if the first index is virtual then $F_{an} = 0$. This formulation of the generalized Fock matrix is quite useful because it only requires density matrices with all indices active and two-electron integrals in the MO basis with three indices active and one general index, greatly reducing the storage and computational cost of the MO transformation. The orbital gradient is then given by

$$\frac{\partial E}{\partial \Delta_{pq}} = 2(F_{pq} - F_{qp}) \tag{6.89}$$

6.20.2 CAS-CI and CASSCF Job Control Options

CAS_METHOD

Indicates whether orbital optimization is requested.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Not running a CAS calculation

1 CAS-CI (no orbital optimization)

2 CASSCF (orbital optimization)

RECOMMENDATION:

Use 2 for best accuracy, but such computations may become infeasible for large active spaces.

CAS_M_S

The number of unpaired electrons desired in the CAS wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N for a wavefunction with N unpaired electrons

RECOMMENDATION:

CAS_N_ELEC

Specifies the number of active electrons.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N include N electrons in the active space

-1 include all electrons in the active space

RECOMMENDATION:

Use the smallest active space possible for the given system.

CAS_N_ORB

Specifies the number of active orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N include N orbitals in the active space

-1 include all orbitals in the active space

RECOMMENDATION:

Use the smallest active space possible for the given system.

CAS_N_ROOTS

Specifies the number of electronic states to determine.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

N solve for N roots of the Hamiltonian

RECOMMENDATION:

CAS_THRESH

Specifies the threshold for matrix elements to be included in the CAS Hamiltonian.

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

CAS_SAVE_NAT_ORBS

Save the CAS natural orbitals in place of the reference orbitals.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE overwrite the reference orbitals with CAS natural orbitals

FALSE do not save the CAS natural orbitals

RECOMMENDATION:

MAX_CASSCF_CYCLES

Maximum number of orbital optimization cycles for CASSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

N set maximum number of optimization cycles to N

RECOMMENDATION:

CAS_USE_RI

Indicates whether the resolution of the identity approximation should be used.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Compute 2-electron integrals analytically

TRUE Use the RI approximation for 2-electron integrals

RECOMMENDATION:

Analytic integrals are more accurate, RI integrals are faster

CAS_DAVIDSON_TOL

Specifies the tolerance for the Davidson solver used in CAS.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

The default should be suitable in most cases

CAS_DAVIDSON_MAXVECTORS

Specifies the maximum number of vectors to augment the Davidson search space in CAS.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

N sets the maximum Davidson subspace size to N+CAS_N_ROOTS

RECOMMENDATION:

The default should be suitable in most cases

CAS_SOLVER

Specifies the solver to be used for the active space.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 CAS-CI/CASSCF

2 ASCI (see Section 6.22)

3 Truncated CI (CIS, CISD, CISDT, etc.)

RECOMMENDATION:

TRUNC_CI_LEVEL

Specifies the order of truncated CI to be used in the active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not carry out truncated CI

1 CIS

2 CISD

3 CISDT

4 CISDTQ

etc.

RECOMMENDATION:

Example 6.50 CASCI(6,14) calculation for the ground state of N_2.

```
$molecule
   0 1
   N   0.0   0.0   0.0
   N   0.0   0.0   1.8
$end

$rem
   EXCHANGE          hf
   BASIS             cc-pvtz
   CAS_METHOD        1    ! 1 for CAS-CI, 2 for CASSCF
   CAS_M_S           0    ! M_s value*2
   ASCI_DIAG         2    ! Arma Sparse=0, Davidson=1, Eigen Sparse=2
   CAS_N_ELEC        6    ! N_elec
   CAS_N_ORB         14   ! N_orb
   CAS_N_ROOTS       1    ! N_roots
   CAS_SOLVER        1    ! 2=ASCI, 1=Olsen, 0=naive
   THRESH            14
   MAX_SCF_CYCLES    400
   SCF_CONVERGENCE   6
   MEM_TOTAL         4000
   MEM_STATIC        1000
   SCF_ALGORITHM     diis_gdm
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 6.51 CASSCF(6,6) calculation for the ground state of N_2.

```
$molecule
   0 1
   N   0.0   0.0   0.0
   N   0.0   0.0   1.8
$end

$rem
   EXCHANGE          hf
   BASIS             cc-pvtz
   CAS_METHOD        2    !1 for CAS-CI, 2 for CASSCF
   CAS_M_S           0    !M_s value*2
   ASCI_DIAG         2    !Arma Sparse=0, Davidson=1, Eigen Sparse=2
   CAS_N_ELEC        6    !N_elec
   CAS_N_ORB         6    !N_orb
   CAS_N_ROOTS       1    !N_roots
   THRESH            14
   MAX_SCF_CYCLES    400
   SCF_CONVERGENCE   6
   MEM_TOTAL         4000
   MEM_STATIC        1000
   SCF_ALGORITHM     diis_gdm
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 6.52 Geometry optimization of the ground state of N_2 at the CASSCF(6,6)/cc-pVTZ level of theory.

```
$molecule
   0 1
   N   0.0   0.0   0.0
   N   0.0   0.0   1.3
$end

$rem
   JOBTYPE              opt
   EXCHANGE             hf
   BASIS                cc-pvtz
   CAS_METHOD           2    !1 for CAS-CI, 2 for CASSCF
   CAS_M_S              0    !M_s value*2
   ASCI_DIAG            2    !Arma Sparse=0, Davidson=1, Eigen Sparse=2
   CAS_N_ELEC           6    !N_elec
   CAS_N_ORB            6    !N_orb
   CAS_N_ROOTS          1    !N_roots
   CAS_SAVE_NAT_ORBS    true !overwrite MOs with CAS natural orbs
   THRESH               14
   MAX_SCF_CYCLES       400
   SCF_CONVERGENCE      6
   MEM_TOTAL            4000
   MEM_STATIC           1000
   SCF_ALGORITHM        diis_gdm
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 6.53 Truncated CI (S, D, T) calculation for the ground state N_2 using the CASCI routines.

```
$molecule
   0 1
   N   0.0   0.0   0.0
   N   0.0   0.0   1.1
$end

$rem
   EXCHANGE          hf
   BASIS             cc-pvdz
   CAS_METHOD        1   !1 for CAS-CI, 2 for CASSCF
   CAS_M_S           0   !M_s value*2
   ASCI_DIAG         2   !Arma Sparse=0, Davidson=1, Eigen Sparse=2
   CAS_N_ELEC        6   !N_elec
   CAS_N_ORB        -1   !N_orb (-1: include all orbitals in active space)
   CAS_N_ROOTS       1   !N_roots
   CAS_SOLVER        3   !2=ASCI, 1=Olsen, 0=naive, 3=truncated CI
   TRUNC_CI_LEVEL    3   !include up to triples excitations
   THRESH            14
   MAX_SCF_CYCLES    400
   SCF_CONVERGENCE   6
   MEM_TOTAL         4000
   MEM_STATIC        1000
   SCF_ALGORITHM     diis_gdm
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end
```

6.21 Incremental Correlation Methods

6.21.1 Introduction

Treating all possible electronic configurations within a wave function via full configuration interaction (FCI) provides the exact solution to the electronic Schrödinger equation. Exponential growth in the number of electronic configurations with system size makes this approach infeasible for all but the smallest systems. The method of increments, however, can be used to systematically capture electron correlation at polynomial cost while maintaining size extensivity. This approach, called incremental FCI (iFCI), utilizes a many-body expansion of the correlation energy, separating it into n-body terms. Higher-orders of n converge toward the FCI solution and often do so without n growing too large to be computationally burdensome. This method is highly parallelizable and features versatile truncation schemes.[101,137–139]

6.21.2 Theory

Electronic energy is retrieved by iFCI using an n-body expansion of the form

$$E = E_{\text{ref}} + E_C = E_{\text{ref}} + \sum_i \epsilon_i + \sum_{j<i} \epsilon_{ij} + \sum_{k<j<i} \epsilon_{ijk} + \cdots \tag{6.90}$$

where each ϵ_X term denotes an increment of correlation energy and i, j, k refer to bodies of the expansion. Incremental correlation energies are defined as

$$\begin{aligned} \epsilon_i &= E_C(i) && (6.91)\\ \epsilon_{ij} &= E_C(ij) - \epsilon_i - \epsilon_j && (6.92)\\ \epsilon_{ijk} &= E_C(ijk) - \epsilon_{ij} - \epsilon_{ik} - \epsilon_{jk} - \epsilon_i - \epsilon_j - \epsilon_k && (6.93)\\ &\cdots \end{aligned}$$

where terms $n > 1$ subtract lower-order increments to avoid double counting. Terms represent n-body additions to the correlation energy from $2n$ electrons in the mean field of the remaining $2(N-n)$ electrons, where each ϵ_X value is computed by solving CAS-CI for $2n$ electrons in $N_v + n$ orbitals. For example, $n = 1$ performs CAS$(2, N_v + 1)$-CI to give the value of $E_C(i) = E(\text{CAS}(2, N_v + 1))$. Proceeding likewise for higher n, CAS$(2n, N_v + n)$-CI produces each $E_C(X)$.

Heat-bath CI (HBCI) is utilized to solve each CAS-CI Hamiltonian, performing selected CI computations according to determinants, j, coupled to the CI wave function in the form $|H_{ij}c_i| > \varepsilon_i$, where ε_i is the energy cutoff and c_i are determinants in the HBCI subspace.

Truncation of incremental terms is performed by considering natural orbital (NO) occupancy cutoffs, $\eta^{(m)}$, where

$$\begin{aligned} \epsilon_i^{(m)} &= E_C(i; \eta^{(m)}) && (6.94)\\ \epsilon_{ij}^{(m)} &= E_C(ij; \eta^{(m)}) - \epsilon_i^{(m)} - \epsilon_j^{(m)} && (6.95)\\ &\cdots \end{aligned}$$

Doing so reduces the size of the virtual space by only including virtual orbitals with sufficiently large NO eigenvalues. Convergence for each iFCI increment is reached when

$$\zeta > |\epsilon^{(m+1)} - \epsilon^m| \tag{6.96}$$

with units of $10^{-\zeta}$ E_h. Further truncation in $n \geq 3$ can be performed by utilizing the ζ parameter and a screening cutoff, $\mathcal{C}_n$, in the form

$$\mathcal{C}_n = 10^{-\zeta} \times \mathcal{S}_n \tag{6.97}$$

where $\mathcal{C}_n$ is in E_h and $\mathcal{S}_n$ is a scalar. This screening is performed by selecting $n-1$ body correlation energy contributions that are above $\mathcal{C}_n$. See Ref. 101 for more details. $\mathcal{S}_n$ is a parameter in the input.

iFCI requires a high-spin perfect pairing (PP) reference, where NOs are localized as local bonding-antibonding pairs, or geminals.

6.21.3 Job Control for iFCI

IFCI_TUPLES

Level of n-body expansion to solve. Note that $n > 2$ can be computationally costly.

TYPE:

INTEGER

DEFAULT:

Must be set.

OPTIONS:

1 $n = 1$

2 $n = 2$

3 $n = 3$

4 $n = 4$

RECOMMENDATION:

Use $n = 2$ for initial system analysis, $n > 2$ for higher accuracy.

IFCI_READ

Restarts iFCI with existing `TUPLES_1E_DATA file`, if it exists.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Start from scratch

1 Restart from previous file

RECOMMENDATION:

Use 0 if no previous run files exist. Use 1 if intending to restart from previous data.

IFCI_TRIPLETS

Set state to solve.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Singlet

1 Triplet

2 Quintet

RECOMMENDATION:

None

IFCI_ZETA

Convergence for each iFCI increment. Note that the format is $\zeta = \text{IFCI\_ZETA}/10$.

TYPE:

INTEGER

DEFAULT:

55

OPTIONS:

45 Loose

55 Moderate

65 Tight

75 Tighter

85 Quite tight

95 Maximum

RECOMMENDATION:

Use 65 and increase to 75 to check convergence.

HBCI_EPS1

Determines dimension of HBCI space.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

N HBCI ε_1 in μ $\mathrm{E_h}$

RECOMMENDATION:

Use default or 500 for tighter convergence.

IFCI_TUPLE_THRESH

Collapse near-degenerate geminals within threshold into one body.

TYPE:

INTEGER

DEFAULT:

2500

OPTIONS:

n in $\mu\mathrm{E_h}$

RECOMMENDATION:

Use default unless there are sets of highly correlating occupied orbitals.

IFCI_TRIPLES_SCREEN

Cutoff ($\mathcal{C}_3$) for determining if a 3-body term is significant.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n where $\mathcal{C}_3 = 10^{-\zeta} \times n$ in $\mathrm{E_h}$

RECOMMENDATION:

Use the default unless looser (higher n) or tighter (lower n) consideration of triads for a given system is desired. Setting to 0 computes all triads (costly).

IFCI_QUAD_SCREEN

Cutoff ($\mathcal{C}_4$) for determining if a 4-body term is significant.

TYPE:

INTEGER

DEFAULT:

IFCI_TRIPLES_SCREEN

OPTIONS:

n where $\mathcal{C}_4 = 10^{-\zeta} \times n$ in $\mathrm{E_h}$

RECOMMENDATION:

Same as IFCI_TRIPLES_SCREEN but note that 4-body terms are significantly more costly.

IFCI_STATE_ADD

Adds additional states to HBCI solver when there is degeneracy amongst states.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Add states within n $\mathrm{mE_h}$

RECOMMENDATION:

Use default unless it is known that degenerate states are present.

IFCI_PRINT

Larger number gives more output.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Minimal output

2 Readable output

3 Extra output

4 Excessive output

5+ Bug testing output

RECOMMENDATION:

2 is recommended, 1-3 is appropriate, larger than 4 is unnecessary (consider yourself warned).

IFCI_NO_THRESH

Equivalent to HBCI ε_1 for increment-specific NO generation step.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n in $\mu\mathrm{E_h}$

RECOMMENDATION:

Set to equal HBCI_EPS1.

IFCI_REF_ITER

Use HF or PP reference density.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 HF

1 PP

RECOMMENDATION:

Use 0.

IFCI_OCC

Specifies the number of active occupied orbitals.

TYPE:

INTEGER

DEFAULT:

Full valence.

OPTIONS:

n Include n orbitals in the active space

-1 Full valence

RECOMMENDATION:

Use full valence active space.

6.21.4 Example

Example 6.6.54 iFCI $n = 2$ calculation for triplet ethylene. PP triplet is used as reference.

View input online

6.22 Adaptive Sampling Configuration Interaction Method

6.22.1 Introduction

Selected CI methods seek to approximate full CI (FCI) energy, possibly in an active space, by including only the most important determinants for a particular problem. There are many flavors of selected CI going back to the 1970s. Q-CHEM includes the adaptive sampling configuration interaction (ASCI) method. In the ASCI method, a trial CI wave function ψ^k is iteratively improved by the inclusion of new determinants that are deemed important. The selection rule is derived from a consistency relationship among the coefficients in a CI expansion of the exact FCI wave function.

6.22.2 Theory

If we have a wave function $|\Psi\rangle = \sum_i C_i|D_i\rangle$ (where $|D_i\rangle$ are Slater determinants with coefficients C_i) as an eigenstate of the Hamiltonian, then

$$C_i = \frac{\sum_{j\neq i} H_{ij}C_j}{(H_{ii} - E)} \,, \tag{6.98}$$

where $H_{ij} = \langle D_i|\mathbf{H}|D_j\rangle$ is the Hamiltonian matrix element between determinants i and j, and E is the energy of the eigenstate $|\Psi\rangle$. This exact relationship can be generalized to a metric to predict the expected weight of a determinant $|D_i\rangle$ in a CI expansion, by how it connects to other determinants in an approximate trial wave function. This metric is also used in Epstein-Nesbet Perturbation theory as coefficients for the determinants in the first order wave function.

In the ASCI method, all determinants $|D_i\rangle$ that are single or double excitations away from the most important determinants (as ranked by magnitude of coefficients) in the trial wave function $|\psi_k\rangle$ are assigned an estimated importance A_i given as

$$A_i = \frac{\sum_{|D_j\rangle\in|\psi_k\rangle} H_{ij}C_j}{(H_{ii} - E_k)} \,, \tag{6.99}$$

where E_k is the energy of the trial wave function $|\psi_k\rangle$. The search and selection is only done in the space spanned by determinants connected to the top c determinants in $|\psi_k\rangle$ because unimportant determinants are unlikely to be the sole

generator for a top ranked determinant, and this pruning of the search space greatly accelerates the algorithm. The top t determinants (as ranked by magnitude of A_i) connected to $|\psi_k\rangle$ are used to form the new wave function $|\psi^{k+1}\rangle$ by exact diagonalization within that Hilbert subspace.

Once several cycles of ASCI has been completed, the wave function will contain all (or very nearly all) of the largest weight determinants in the FCI wave function and the remaining determinants not included should be of small weight. The effect of these many small remaining determinants are estimated by second order Epstein-Nesbet perturbation theory (PT2).[32,91] This final PT2 correction gives extremely accurate results, often within a kcal/mol of the absolute FCI energies even when only a tiny fraction of the Hilbert space is included in the ASCI wave function.[122] An extrapolation of the variational energy against the PT2 correction (to the FCI limit of of zero PT2 correction) can also be carried out to generate more accurate estimates, and predict a metric for error in the final estimate. Indeed, it has been shown that linear or quadratic fits are quite accurate for extrapolation of SCI energies against the PT2 correction.[50,58,82] We observe essentially linear behavior in the case of ASCI.

ASCI may be used as the full-CI solver for a CASSCF calculation, permitting the extension of CASSCF to active spaces of ≈ 50 electrons in ≈ 50 orbitals. The resulting method is termed ASCI-SCF[78]. See section 6.20 for details on CASSCF job control.

6.22.3 ASCI Job Control

Active space specification and convergence details are controlled by the *$rem* variables described in Section 6.20.2. During the course of an ASCI calculation, a file named `wf_data` is created in the scratch directory containing information on the determinants and weights in the ASCI wavefunction.

ASCI_DIAG

Specifies the diagonalization procedure.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Davidson solver

2 Eigen sparse matrix solver

RECOMMENDATION:

Use 2 for best trade-off of speed and memory usage. If memory usage becomes to great, switch to 1.

ASCI_NDETS

Specifies the number of determinants to include in the ASCI wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N for a wavefunction with N determinants

RECOMMENDATION:

Typical ASCI expansions range from 50,000 to 2,000,000 determinants depending on active space size, complexity of problem, and desired accuracy

ASCI_CDETS

Specifies the number of determinants to search over during ASCI wavefunction growth steps.

TYPE:

INTEGER

DEFAULT:

-5

OPTIONS:

$N > 0$ search from the top N determinants

$N < 0$ search from the top determinants whose cumulative weight in the wavefunction corresponds to $1 - 2^N$

RECOMMENDATION:

Using a dynamically determined value ($N < 0$) gives better results.

ASCI_USE_NAT_ORBS

Specifies whether rotation to a natural orbital basis should be carried out between growth steps.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE rotate to a natural orbital basis between growth wavefunction growth steps

FALSE do not rotate to a natural orbital basis

RECOMMENDATION:

Natural orbital rotations significantly improve the compactness and therefore accuracy of the ASCI wavefunction.

ASCI_DAVIDSON_GUESS

Specifies the truncated CI guess used for ASCI's Davidson solver.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

N Order of the truncated CI to solve explicitly ASCI Davidson guess.

RECOMMENDATION:

Accurate excited states and rapid convergence of the ground state benefit from a good zero-order guess for the low energy spectrum. The default is often sufficient.

ASCI_SKIP_PT2

Specifies whether ASCI PT2 correction should be calculated.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE compute ASCI PT2 contribution

TRUE do not compute ASCI PT2 contribution

RECOMMENDATION:

The PT2 correction is essential to obtaining converged ASCI energies.

ASCI_RESTART

Specifies whether to initialize the ASCI wavefunction with the `wf_data` file.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE read CI coefficients from the `wf_data` file

FALSE do not read the CI coefficients from disk

RECOMMENDATION:

ASCI_SPIN_PURIFY

Indicates whether or not the ASCI wavefunction should be augmented with missing determinants to ensure a spin-pure state.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE augment the wavefunction with determinants to ensure a spin eigenstate

FALSE do not augment the wavefunction

RECOMMENDATION:

Example 6.55 CASCI calculation for the lowest triplet ($M_s = 1$) state of N_2. The adaptive sampling method is employed to tackle a large active space (14e, 32o). The PT2 correction is performed on top of the ASCI wavefunction.

```
$molecule
   0 1
   N   0.0   0.0   0.0
   N   0.0   0.0   1.8
$end

$rem
   EXCHANGE          hf
   BASIS             cc-pvtz
   CAS_METHOD        1     !1 for CAS-CI, 2 for CASSCF
   CAS_M_S           2     !M_s value*2 (2: triplet)
   ASCI_DIAG         2     !Arma Sparse=0, Davidson=1, Eigen Sparse=2
   CAS_N_ELEC        14    !N_elec
   CAS_N_ORB         32    !N_orb
   CAS_N_ROOTS       1     !N_roots
   CAS_SOLVER        2     !2=ASCI, 1=Olsen, 0=naive
   ASCI_NDETS        28000 !Number of ASCI Determinants
   THRESH            14
   MAX_SCF_CYCLES    400
   SCF_CONVERGENCE   8
   MEM_TOTAL         4000
   MEM_STATIC        1000
   ROSCF             true
   SCF_ALGORITHM     diis_gdm
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 6.56 ASCI-SCF (using adaptive sapling for CAS) calculation for the ground state of N_2 with a large active space (14e, 32o). The PT2 correction is not performed in this case.

```
$molecule
   0 1
   N   0.0   0.0   0.0
   N   0.0   0.0   1.8
$end

$rem
   EXCHANGE          hf
   BASIS             cc-pvtz
   CAS_METHOD        2  !1 for CAS-CI, 2 for CASSCF
   CAS_M_S           0  !M_s value*2
   ASCI_DIAG         2  !Arma Sparse=0, Davidson=1, Eigen Sparse=2
   CAS_N_ELEC        14 !N_elec
   CAS_N_ORB         32 !N_orb
   CAS_N_ROOTS       1 !N_roots
   CAS_SOLVER        2 !2=ASCI, 1=Olsen, 0=naive
   ASCI_NDETS        28000 !Number of ASCI Determinants
   ASCI_SKIP_PT2     1
   THRESH            14
   MAX_SCF_CYCLES    400
   SCF_CONVERGENCE   6
   MEM_TOTAL         4000
   MEM_STATIC        1000
   SCF_ALGORITHM     diis_gdm
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

6.23 Variational Two-Electron Reduced-Density-Matrix Methods

6.23.1 Introduction

The methods described in this section involve the direct variational optimization of the two-electron reduced-density matrix (2-RDM, ${}^2\mathbf{D}$), subject to necessary ensemble N-representability conditions.[34,35,41,42,88,106] Such conditions place restrictions on the 2-RDM in order to ensure that it is derivable from an ensemble of N-electron density matrices. In the limit that the N-representability of the 2-RDM is exactly enforced, the variational 2-RDM (v2RDM) approach is equivalent to full configuration interaction (CI). Such computations are, in general, computationally infeasible, so the v2RDM optimization is typically carried out under a subset of two- or three-particle conditions. When only partially enforcing N-representability, the v2RDM approach yields a lower bound to the full CI energy.

In Q-CHEM, all v2RDM optimizations are carried out under the following conditions:

- the 2-RDM is positive semidefinite
- the one-electron reduced-density matrix (1-RDM) is positive semidefinite
- the trace of the 2-RDM is equal to the number of pairs of electrons, $N(N-1)$
- each spin block of the 2-RDM properly contracts to the appropriate spin block of the 1-RDM
- the expectation value of $\hat{M}_S$ is $\frac{1}{2}(N_\alpha - N_\beta)$ (the maximal spin projection)

Additionally, an optional spin constraint can be placed on the 2-RDM such that $\langle \hat{S}^2 \rangle = S(S+1)$ (in units of $\hbar^2$), where the S is the spin quantum number. Note that this constraint on the expectation value of $\hat{S}^2$ does not strictly guarantee that the 2-RDM corresponds to an eigenfunction of $\hat{S}^2$. Without additional constraints, a v2RDM optimization would yield poor-quality 2-RDMs with energies far below those of full CI. Reasonable results require, at a minimum, that one enforce the positivity of additional pair-probability density matrices, including the two-hole reduced-density matrix (${}^2\mathbf{Q}$) and the particle-hole reduced-density matrix (${}^2\mathbf{G}$). The positivity of ${}^2\mathbf{D}$, ${}^2\mathbf{Q}$, and ${}^2\mathbf{G}$ constitute the DQG constraints of Garrod and Percus.[42] For many systems, the DQG constraints yield a reasonable description of the electronic structure. However, if high accuracy is desired, it is sometimes necessary to consider constraints on higher-order reduced-density matrices (*e.g.* the three-electron reduced-density matrix [3-RDM]). In Q-CHEM, v2RDM optimizations can be performed under the T1 and T2 partial three-particle conditions,[33,136] which do not explicitly depend upon the 3-RDM; or the full 3-positivity conditions, which include the three-particle reduced-density matrix (${}^3\mathbf{D}$), the three-hole reduced-density matrix (${}^3\mathbf{Q}$), the two-particle-one-hole reduced-density matrix (${}^3\mathbf{E}$), and the one-particle-two-hole reduced-density matrix (${}^3\mathbf{F}$). The full 3-positivity conditions guarantee partial conditions automatically.[86] The positivity conditions imposed in v2RDM computations are controlled through the *$rem* variable RDM_POSITIVITY.

The main utility of the v2RDM approach is in the context of active-space-based descriptions of strong or nondynamical correlation. The most common active-space-based approach for strong correlation is the compete active space self-consistent field (CASSCF) method. By performing a v2RDM computation within an active space and coupling v2RDM to an orbital optimization procedure, one can achieve a v2RDM-driven CASSCF procedure[38,43,85] that provides a lower bound the conventional CI-based CASSCF energy. Because the v2RDM-CASSCF method scales polynomially with respect to the number of active orbitals, v2RDM-CASSCF can handle much larger active spaces (*e.g.*, 50 electrons in 50 orbitals) compared to CI-CASSCF (*e.g.*, 18 electrons in 18 orbitals).

Note that v2RDM-CASSCF only describes electron correlations among the active orbitas. A computationally inexpensive estimate of the remaining correlation effects can be achieved with the multiconfiguration pair-density functional theory (MC-PDFT) approach,[40,84] which evaluates the energy as a functional of the on-top pair density (OTPD). In the MC-PDFT implementation in Q-CHEM, the OTPD is derived from a variationally optimized 2-RDM as described in Ref. 90.

The current v2RDM, v2RDM-CASSCF, and MC-PDFT implementations must make use of the density fitting (DF) approximation to the two-electron integrals. The use of DF integrals is particularly advantageous for v2RDM-CASSCF computations with large active spaces because of the increased efficiency in the orbital optimization/integral transformation step. The v2RDM computation will fail without the *$rem* keyword AUX_BASIS. Analytic gradients are not available when frozen molecular orbitals are requested. Specification of the active space is demonstrated in the examples below. When the formatted checkpoint file is requested, natural orbitals are saved in it.

6.23.2 Theory

The electronic energy is an exact functional of the 1-RDM and 2-RDM

$$E = \frac{1}{2} \sum_{pqrs} {}^2D^{pq}_{rs}(pr|qs) + \sum_{pq} {}^1D^p_q h_{pq}, \tag{6.100}$$

where the 1-RDM (${}^1\mathbf{D}$) and 2-RDM are represented in a given spin-orbital basis indexed by p, q, r, and s. The one-hole RDM (${}^1\mathbf{Q}$), two-hole RDM (${}^2\mathbf{Q}$), particle-hole RDM (${}^2\mathbf{G}$), partial three-particle RDMs ($\mathbf{T1}$ and $\mathbf{T2}$), and full three-particle RDMs (${}^3\mathbf{D}$, ${}^3\mathbf{Q}$, ${}^3\mathbf{E}$, ${}^3\mathbf{F}$) are linear functions of ${}^1\mathbf{D}$ and ${}^2\mathbf{D}$.[38] Minimizing the electronic energy with respect to ${}^2\mathbf{D}$ while enforcing the linear relations among these RDMs, the contraction and spin constraints placed on ${}^2\mathbf{D}$, and the positive semidefinite property of all RDMs constitutes a semidefinite program (SDP). The current v2RDM implementation uses a boundary-point SDP (BPSDP) algorithm to solve the SDP.[83,87,98]

The primal formulation of the SDP is

$$\begin{aligned} \text{minimize} \quad & E_{\text{primal}} = \mathbf{c}^T \cdot \mathbf{x} \\ \text{such that} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ \text{and} \quad & M(\mathbf{x}) \succeq 0. \end{aligned} \tag{6.101}$$

Here, $\mathbf{x}$ represents the primal solution vector, the vector $\mathbf{c}$ contains all information defining the quantum system (the one- and two-electron integrals), and the mapping $M(\mathbf{x})$ maps the primal solution onto the set of positive semidefinite RDMs:

$$M(\mathbf{x}) = \begin{pmatrix} {}^1\mathbf{D} & 0 & 0 & 0 \\ 0 & {}^1\mathbf{Q} & 0 & 0 \\ 0 & 0 & {}^2\mathbf{D} & 0 \\ 0 & 0 & 0 & \ddots \end{pmatrix} \succeq 0. \tag{6.102}$$

Additional RDMs can be included in $M(\mathbf{x})$, depending on the choice of N-representability conditions applied. The action of the constraint matrix, $\mathbf{A}$, on $\mathbf{x}$ is a compact representation of the N-representability conditions. $\mathbf{A}$ maintains the appropriate mappings between each block of $M(\mathbf{x})$ and enforces the appropriate spin and contraction conditions. Alternatively, one could consider the dual formulation of the semidefinite problem, expressed as

$$\begin{aligned} \text{maximize} \quad & E_{\text{dual}} = \mathbf{b}^T \cdot \mathbf{y} \\ \text{such that} \quad & \mathbf{z} = \mathbf{c} - \mathbf{A}^T\mathbf{y} \\ \text{and} \quad & M(\mathbf{z}) \succeq 0 \end{aligned} \tag{6.103}$$

where $\mathbf{y}$ and $\mathbf{z}$ are the dual solutions, and $M(\mathbf{z})$ is constrained to be positive semidefinite.

The BPSDP algorithm involves an iterative two-step procedure:

1. Solve $\mathbf{A}\mathbf{A}^T\mathbf{y} = \mathbf{A}(\mathbf{c} - \mathbf{z}) + \tau\mu(\mathbf{b} - \mathbf{A}\mathbf{x})$ for $\mathbf{y}$ by conjugate gradient methods.
2. Update $\mathbf{x}$ and $\mathbf{z}$ by separating $\mathbf{U} = M(\mu\mathbf{x} + \mathbf{A}^T\mathbf{y} - \mathbf{c})$ into its positive and negative components (by diagonalization). The updated primal and dual solutions $\mathbf{x}$ and $\mathbf{z}$ are given by $M(\mathbf{x}) = \mathbf{U}(+)/\mu$ and $M(\mathbf{z}) = -\mathbf{U}(-)$.

Here, τ is a step-length parameter that lies in the interval [1.0,1.6][87]. The penalty parameter μ controls how strictly the primal or dual constraints are enforced and is updated dynamically according to the protocol outlined in Ref. 87. The frequency with which μ is updated is controlled by the *$rem* keyword RDM_MU_UPDATE_FREQUENCY. The algorithm is considered converged when the primal error $||\mathbf{A}\mathbf{x} - \mathbf{b}||$, the dual error $||\mathbf{A}^T\mathbf{y} - \mathbf{c} + \mathbf{z}||$, and the primal/dual energy gap $|E_{\text{primal}} - E_{\text{dual}}|$ are sufficiently small. The convergence in the primal/dual errors and the primal/dual energy gap are controlled by the *$rem* keywords RDM_EPS_CONVERGENCE and RDM_E_CONVERGENCE, respectively. The BPSDP algorithm scales n^6 for the DQG conditions and n^9 for the T1, T2, and 3POS conditions where n is the number of active orbitals in the v2RDM computation.

In v2RDM-CASSCF, the BPSDP algorithm is carried out to determine the 1- and 2RDM for a subset of active molecular orbitals. These orbitals are optimized with respect to restricted doubly occupied / active and active / external rotations after a chosen number of v2RDM iterations (Steps 1. and 2. above). The frequency of this orbital optimization is controlled by the *$rem* keyword RDM_ORBOPT_FREQUENCY.

Given converged 1- and 2-RDMs from a v2RDM-CASSCF calculation, an estimate of the remaining correlation effects can be obtained through the formalism of MC-PDFT. In MC-PDFT, the total energy for the system is expressed as

$$E_{\text{MC-PDFT}} = \sum_i h_i^i + \sum_{tu} h_u^t \, {}^1D_u^t + E_{\text{H}} + E_{\text{XC}}\left[\rho(\mathbf{r}), \Pi(\mathbf{r}), |\nabla\rho(\mathbf{r})|, |\nabla\Pi(\mathbf{r})|\right] \tag{6.104}$$

where the Hartree energy, E_{H}, is the classical Coulomb repulsion

$$E_{\mathrm{H}} = \frac{1}{2}\sum_{ij}(ii|jj) + \sum_{itu}(tu|ii)^1 D_u^t + \frac{1}{2}\sum_{tuvw}(tu|vw)^1 D_u^t\,{}^1 D_w^v \tag{6.105}$$

and E_{xc} represents an on-top pair density functional. In Eq. 6.105, the labels i and j represent doubly occupied spin orbitals, and the labels t, u, v, and w represent active spin orbitals. The symbols $\rho(\mathbf{r})$ and $\Pi(\mathbf{r})$ represent the density and on-top pair density, respectively, which are defined in terms of the molecular orbitals $\{\phi\}$ as

$$\rho(\mathbf{r}) = \sum_{pq}\phi_p(\mathbf{r})\phi_q(\mathbf{r})^1 D_q^p \tag{6.106}$$

and

$$\Pi(\mathbf{r}) = \sum_{pqrs}\phi_p(\mathbf{r})\phi_q(\mathbf{r})\phi_r(\mathbf{r})\phi_s(\mathbf{r})^2 D_{rs}^{pq} \tag{6.107}$$

and $\nabla\rho(\mathbf{r})$ and $\nabla\Pi(\mathbf{r})$ represent the gradients of these quantities. In Q-Chem, E_{xc} is chosen to be a translated[84] on-top pair density functional, which is essentially the same as a functional of the density and spin density (and their gradients), with the spin-density (and its gradient) re-expressed as a function of the on-top pair density (and its gradient). The specific choice of on-top pair density functional is controlled through the *$rem* variables PDFT_EXCHANGE and PDFT_CORRELATION.

6.23.3 v2RDM Job Control

RDM_POSITIVITY

Indicates positivity conditions enforced in the v2RDM optimization.

TYPE:

STRING

DEFAULT:

DQG

OPTIONS:

DQG,	Two-electron conditions
DQGT1	Two-electron conditions plus the T1 partial three-electron conditions
DQGT2	Two-electron conditions plus the T2 partial three-electron conditions
DQGT1T2	Two-electron conditions plus the T1 and T2 partial three-electron conditions
DQG3POS	Two-electron conditions plus the full three-electron conditions

RECOMMENDATION:

For high-accuracy, use DQG3POS or DQGT2, although such computations become impractical for large active spaces. For large active spaces (*e.g.*, n > 16 for CAS(n, n)), use DQG.

RDM_CONSTRAIN_SPIN

Indicates if the spin-constraints are enforced.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE	Enforce spin-constraints.
FALSE	Do not enforce spin-constraints.

RECOMMENDATION:

Use default.

PDFT_EXCHANGE

Specifies the exchange functional to be used in MC-PDFT calculation.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use PDFT_EXCHANGE = *NAME*, where *NAME* must be one of the LDA or GGA exchange functionals listed in Section 5.3.3. This keyword is only invoked when method is set to RDM(PDFT).

RECOMMENDATION:

In general, consult the literature to guide your selection.

PDFT_CORRELATION

Specifies the correlation functional to be used in MC-PDFT calculation.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

NAME Use PDFT_CORRELATION = *NAME*, where *NAME* is one of the LDA or GGA correlation functionals listed in Section 5.3.4. This keyword is only invoked when method is set to RDM(PDFT).

RECOMMENDATION:

In general, consult the literature to guide your selection.

RDM_E_CONVERGENCE

The threshold for the primal-dual energy gap.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

Increase for gradient computations.

RDM_EPS_CONVERGENCE

The threshold for the error in the primal and dual constraints.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

Increase for gradient computations.

RDM_MAXITER

Maximum number of diagonalization steps in the BPSDP solver.

TYPE:

INTEGER

DEFAULT:

50000

OPTIONS:

$N > 0$

RECOMMENDATION:

Increase for computations that are difficult to converge.

RDM_CG_CONVERGENCE

The minimum threshold for the conjugate gradient solver.

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

Should be at least (RDM_EPS_CONVERGENCE+2).

RDM_CG_MAXITER

Maximum number of iterations for each conjugate gradient computations in the BPSDP algorithm.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

$N > 0$

RECOMMENDATION:

Use default unless problems arise.

RDM_TAU

Step-length parameter used in the BPSDP solver.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

N for a value of 0.1 * N

RECOMMENDATION:

RDM_TAU should range between 10 and 16 for $1.0 \leq \tau \leq 1.6$.

RDM_MU_UPDATE_FREQUENCY

The number of v2RDM iterations after which the penalty parameter μ is updated.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

$N > 0$

RECOMMENDATION:

Change if convergence problems arise.

RDM_TPDM_GUESS

Initial guess for the RDMs

TYPE:

STRING

DEFAULT:

HF_GUESS

OPTIONS:

HF_GUESS	Use RDMs from Hartree-Fock calculations as the initial density for the semidefinite solver
RANDOM_GUESS	Use random numbers as the initial density for the semidefinite solver

RECOMMENDATION:

Use default unless convergence problems arise.

RDM_DIAGONALIZER

The algorithm used to diagonalize matrices inside semidefinite programming.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

0	Use parallel LAPACK function DSYEV
1	Use parallel LAPACK function DSYEVD
10	Use multiple simultaneous calls to serial LAPACK function DSYEV
11	Use multiple simultaneous calls to serial LAPACK function DSYEVD

RECOMMENDATION:

Use default. Under certain circumstances (*e.g.*, low symmetry), algorithm 1 may be faster.

RDM_PRINT

Controls the amount of printing.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Print minimal information.
1	Print information about all iterations.

RECOMMENDATION:

Use 1 to analyze convergence issues.

RDM_ORBOPT_GRADIENT_CONVERGENCE

The threshold for the orbital gradient during orbital optimization.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

N for threshold of 10^{-N}

RECOMMENDATION:

Tighten for gradient computations.

RDM_ORBOPT_ENERGY_CONVERGENCE

The threshold for energy convergence during orbital optimization.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

N for threshold of 10^{-N}

RECOMMENDATION:

Tighten for gradient computations.

RDM_ORBOPT_MAXITER

The maximum number of orbital optimization steps each time the orbital optimization routine is called.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

$N > 0$

RECOMMENDATION:

Use default unless convergence problems arise.

RDM_ORBOPT_FREQUENCY

The number of v2RDM iterations after which the orbital optimization routine is called.

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

$N > 0$

RECOMMENDATION:

Use default unless convergence problems arise.

6.23.4 Examples

Example 6.57 Single-point v2RDM/STO-3G energy computation.

```
$molecule
    0 1
    O  0.0000  0.0000  0.1173
    H  0.0000  0.7572 -0.4692
    H  0.0000 -0.7572 -0.4692
$end

$rem
    BASIS                           sto-3g
    AUX_BASIS                       rimp2-vdz
    METHOD                          rdm
    UNRESTRICTED                    false
    RDM_POSITIVITY                  dqg
    RDM_CONSTRAIN_SPIN              true
    RDM_MU_UPDATE_FREQUENCY         200
    RDM_EPS_CONVERGENCE             4
    RDM_E_CONVERGENCE               4
    RDM_MAXITER                     500000
    RDM_TAU                         10
    RDM_PRINT                       1
$end
```

Example 6.58 Single-point v2RDM/STO-3G energy computation with frozen core orbital.

```
$molecule
    0 1
    O  0.0000  0.0000  0.1173
    H  0.0000  0.7572 -0.4692
    H  0.0000 -0.7572 -0.4692
$end

$rem
    JOBTYPE                         sp
    BASIS                           sto-3g
    AUX_BASIS                       rimp2-vdz
    METHOD                          rdm
    UNRESTRICTED                    false
    RDM_POSITIVITY                  dqg
    RDM_CONSTRAIN_SPIN              true
    RDM_MU_UPDATE_FREQUENCY         200
    RDM_EPS_CONVERGENCE             4
    RDM_E_CONVERGENCE               4
    RDM_MAXITER                     500000
    RDM_TAU                         10
    RDM_PRINT                       1
$end

$rdm_active_space
1 0 0 0 ! frozen orbitals
0 0 0 0 ! restricted orbitals
3 0 2 1 ! active orbitals
$end
```

Example 6.59 Single-point v2RDM-CASSCF/cc-pVDZ energy and gradient computation.

```
$molecule
    0 1
    O  0.0000  0.0000  0.1173
    H  0.0000  0.7572 -0.4692
    H  0.0000 -0.7572 -0.4692
$end

$rem
    JOBTYPE                         force
    BASIS                           cc-pvdz
    AUX_BASIS                       rimp2-cc-pvdz
    METHOD                          rdm
    UNRESTRICTED                    false
    RDM_POSITIVITY                  dqg
    RDM_CONSTRAIN_SPIN              true
    RDM_MU_UPDATE_FREQUENCY         200
    RDM_EPS_CONVERGENCE             4
    RDM_E_CONVERGENCE               4
    RDM_MAXITER                     500000
    RDM_TAU                         10
    RDM_PRINT                       1
    RDM_ORBOPT_ENERGY_CONVERGENCE   7
    RDM_ORBOPT_GRADIENT_CONVERGENCE 4
    RDM_ORBOPT_FREQUENCY            500
    RDM_ORBOPT_MAXITER              5
$end

$rdm_active_space
0 0 0 0 ! frozen orbitals
1 0 0 0 ! restricted orbitals
3 0 2 1 ! active orbitals
$end
```

Example 6.60 Single-point MC-PDFT/cc-pVDZ energy computation with v2RDM-CASSCF optimized RDMs.

```
$molecule
0 1
H
H 1 0.6
$end

$rem
method = rdm(pdft)
pdft_exchange = s
pdft_correlation = vwn1rpa
basis = cc-pvdz
aux_basis = rimp2-cc-pvdz
xc_grid = 000075000302
rdm_optimize_orbitals = false
rdm_e_convergence = 6
rdm_print = 1
$end
```

References and Further Reading

[1] Self-Consistent Field Methods (Chapter 4).

[2] Excited-State Calculations (Chapter 7).

[3] Basis Sets and Effective Core Potentials (Chapter 8).

[4] For a general textbook introduction to electron correlation methods and their respective strengths and weaknesses, see Ref. **?** .

[5] For a tutorial introduction to electron correlation methods based on wavefunctions, see Ref. **?** .

[6] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill. *Chem. Phys. Lett.*, 254:329, 1996. DOI: 10.1016/0009-2614(96)00280-1.

[7] Y. Akinaga, Y. Kawashima, and S. Ten-no. *Chem. Phys. Lett.*, 506:276, 2011. DOI: 10.1016/j.cplett.2011.02.044.

[8] F. Aquilante, T. B. Pedersen, and R. Lindh. *Theor. Chem. Acc.*, 124:1, 2009. DOI: 10.1007/s00214-009-0608-y.

[9] N. H. F. Beebe and J. Linderberg. *Int. J. Quantum Chem.*, 12:683, 1977. DOI: 10.1002/qua.560120408.

[10] G. J. O. Beran, S. R. Gwaltney, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 5:2488, 2003. DOI: 10.1039/b304542k.

[11] G. J. O. Beran, B. Austin, A. Sodt, and M. Head-Gordon. *J. Phys. Chem. A*, 109:9183, 2005. DOI: 10.1021/jp053780c.

[12] G. J. O. Beran, M. Head-Gordon, and S. R. Gwaltney. *J. Chem. Phys.*, 124:114107, 2006. DOI: 10.1063/1.2176603.

[13] L. W. Bertels, J. Lee, and M. Head-Gordon. *J. Phys. Chem. Lett.*, 10:4170, 2019. DOI: 10.1021/acs.jpclett.9b01641.

[14] E. F. C. Byrd, T. Van Voorhis, and M. Head-Gordon. *J. Phys. Chem. B*, 106:8070, 2002. DOI: 10.1021/jp020255u.

[15] K. Carter-Fenk and M. Head-Gordon. *J. Chem. Phys.*, 158:234108, 2023. DOI: 10.1063/5.0150033.

[16] K. Carter-Fenk, J. Shee, and M. Head-Gordon. *J. Chem. Phys.*, 159:171104, 2023. DOI: 10.1063/5.0174923.

[17] O. Christiansen, H. Koch, and P. Jørgensen. *Chem. Phys. Lett.*, 243:409, 1995. DOI: 10.1016/0009-2614(95)00841-Q.

[18] J. Cullen. *Chem. Phys.*, 202:217, 1996. DOI: 10.1016/0301-0104(95)00321-5.

[19] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople. *J. Chem. Phys.*, 94:7221, 1991. DOI: 10.1063/1.460205.

[20] L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov, and J. A. Pople. *J. Chem. Phys.*, 109:7764, 1998. DOI: 10.1063/1.477422.

[21] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople. *J. Chem. Phys.*, 112:7374, 2000. DOI: 10.1063/1.481336.

[22] R. A. DiStasio, Jr. and M. Head-Gordon. *Mol. Phys.*, 105:1073, 2007. DOI: 10.1080/00268970701283781.

[23] R. A. DiStasio, Jr., Y. Jung, and M. Head-Gordon. *J. Chem. Theory Comput.*, 1:862, 2005. DOI: 10.1021/ct050126s.

[24] R. A. DiStasio, Jr., R. P. Steele, and M. Head-Gordon. *Mol. Phys.*, 105:2731, 2007. DOI: 10.1080/00268970701624687.

[25] R. A. DiStasio, Jr., R. P. Steele, Y. M. Rhee, Y. Shao, and M. Head-Gordon. *J. Comput. Chem.*, 28:839, 2007. DOI: 10.1002/jcc.20604.

[26] L. B. Dittmer and M. Head-Gordon. *J. Chem. Phys.*, 162:054109, 2024. DOI: 10.1063/5.0242211.

[27] S. Dohm, A. Hansen, M. Steinmetz, S. Grimme, and M. P. Checinski. *J. Chem. Theory Comput.*, 14:2596, 2018. DOI: 10.1021/acs.jctc.7b01183.

[28] J. P. Dombroski, S. W. Taylor, and P. M. W. Gill. *J. Phys. Chem.*, 100:6272, 1996. DOI: 10.1021/jp952841b.

[29] B. I. Dunlap. *Phys. Chem. Chem. Phys.*, 2:2113, 2000. DOI: 10.1039/b000027m.

[30] E. Epifanovsky, M. Wormit, T. Kuś, A. Landau, D. Zuev, K. Khistyaev, P. Manohar, I. Kaliman, A. Dreuw, and A. I. Krylov. *J. Comput. Chem.*, 34:2293, 2013. DOI: 10.1002/jcc.23377.

[31] E. Epifanovsky, D. Zuev, X. Feng, K. Khistyaev, Y. Shao, and A. I. Krylov. *J. Chem. Phys.*, 139:134105, 2013. DOI: 10.1063/1.4820484.

[32] P. S. Epstein. *Phys. Rev.*, 28:695, 1926. DOI: 10.1103/PhysRev.28.695.

[33] R. M. Erdahl. *Int. J. Quantum Chem.*, 13:697, 1978. DOI: 10.1002/qua.560130603.

[34] R. M. Erdahl. *Rep. Math. Phys.*, 15:147, 1979. DOI: 10.1016/0034-4877(79)90015-6.

[35] R. M. Erdahl, C. Garrod, B. Golli, and M. Rosina. *J. Math. Phys.*, 20:1366, 1979. DOI: 10.1063/1.524243.

[36] H. Eshuis, J. Yarkony, and F. Furche. *J. Chem. Phys.*, 132:234114, 2010. DOI: 10.1063/1.3442749.

[37] M. Feyereisen, G. Fitzgerald, and A. Komornicki. *Chem. Phys. Lett.*, 208:359, 1993. DOI: 10.1016/0009-2614(93)87156-W.

[38] J. Fosso-Tande, T.-S. Nguyen, G. Gidofalvi, and A. E. DePrince III. *J. Chem. Theory Comput.*, 12:2260, 2016. DOI: 10.1021/acs.jctc.6b00190.

[39] M. J. Frisch, M. Head-Gordon, and J. A. Pople. *Chem. Phys. Lett.*, 166:275, 1990. DOI: 10.1016/0009-2614(90)80029-D.

[40] L. Gagliardi, D. G. Truhlar, G. L. Manni, R. K. Carlson, C. E. Hoyer, and J. L. Bao. 50:66, 2017. DOI: 10.1021/acs.accounts.6b00471.

[41] C. Garrod and J. K. Percus. *J. Math. Phys.*, 5:1756, 1964. DOI: 10.1063/1.1704098.

[42] C. Garrod, M. V. Mihailović, and M. Rosina. *J. Math. Phys.*, 16:868, 1975. DOI: 10.1063/1.522634.

[43] G. Gidofalvi and D. A. Mazziotti. *J. Chem. Phys.*, 129:134108, 2008. DOI: 10.1063/1.2983652.

[44] W. A. Goddard III and L. B. Harding. *Annu. Rev. Phys. Chem.*, 29:363, 1978. DOI: 10.1146/annurev.pc.29.100178.002051.

[45] M. Goldey and M. Head-Gordon. *J. Phys. Chem. Lett.*, 3:3592, 2012. DOI: 10.1021/jz301694b.

[46] M. Goldey, A. Dutoi, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 15:15869, 2013. DOI: 10.1039/c3cp51826d.

[47] S. Grimme. *J. Chem. Phys.*, 118:9095, 2003. DOI: 10.1063/1.1569242.

[48] S. R. Gwaltney and M. Head-Gordon. *Chem. Phys. Lett.*, 323:21, 2000. DOI: 10.1016/S0009-2614(00)00423-1.

[49] S. R. Gwaltney and M. Head-Gordon. *J. Chem. Phys.*, 115:5033, 2001. DOI: 10.1063/1.1383589.

[50] D. Hait, N. M. Tubman, D. S. Levine, K. B. Whaley, and M. Head-Gordon. *J. Chem. Theory Comput.*, 15:5370, 2019. DOI: 10.1021/acs.jctc.9b00674.

[51] C. Hattig and F. Weigend. *J. Chem. Phys.*, 113:5154, 2000. DOI: 10.1063/1.1290013.

[52] M. Head-Gordon. *Mol. Phys.*, 96:673, 1999. DOI: 10.1080/00268979909483003.

[53] M. Head-Gordon, J. A. Pople, and M. J. Frisch. *Chem. Phys. Lett.*, 153:503, 1988. DOI: 10.1016/0009-2614(88)85250-3.

[54] M. Head-Gordon, M. S. Lee, and P. E. Maslen. A hierarchy of local electron correlation models based on atomic truncations. In L. R. Pratt and G. Hummer, editors, *Simulation and Theory of Electrostatic Interactions in Solution*, volume 492 of *AIP Conference Proceedings*, pages 301–328. American Institute of Physics, New York, 1999. DOI: 10.1063/1.1301534.

[55] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople. *Ab Initio Molecular Orbital Theory*. Wiley, New York, 1986.

[56] T. Helgaker, J. Gauss, P. Jorgensen, and J. Olsen. *J. Chem. Phys.*, 106:6430, 1997. DOI: 10.1063/1.473634.

[57] T. Helgaker, P. Jørgensen, and J. Olsen. *Molecular Electronic-Structure Theory*. Wiley, 2000.

[58] A. Holmes, C. Umrigar, and S. Sharma. *J. Chem. Phys.*, 147:164111, 2017. DOI: 10.1063/1.4998614.

[59] D. Jayatilaka and T. J. Lee. *Chem. Phys. Lett.*, 199:211, 1992. DOI: 10.1016/0009-2614(92)80108-N.

[60] Y. Jung and M. Head-Gordon. *ChemPhysChem*, 4:522, 2003. DOI: 10.1002/cphc.200200668.

[61] Y. Jung, R. C. Lochan, A. D. Dutoi, and M. Head-Gordon. *J. Chem. Phys.*, 121:9793, 2004. DOI: 10.1063/1.1809602.

[62] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon. *Proc. Natl. Acad. Sci. USA*, 102:6692, 2005. DOI: 0.1073/pnas.0408475102.

[63] A. I. Krylov, C. D. Sherrill, E. F. C. Byrd, and M. Head-Gordon. *J. Chem. Phys.*, 109:10669, 1998. DOI: 10.1063/1.477764.

[64] S. A. Kucharski and R. J. Bartlett. *J. Chem. Phys.*, 108:5243, 1998. DOI: 10.1063/1.475961.

[65] A. Landau, K. Khistyaev, S. Dolgikh, and A. I. Krylov. *J. Chem. Phys.*, 132:014109, 2010. DOI: 10.1063/1.3276630.

[66] K. V. Lawler, G. J. O. Beran, and M. Head-Gordon. *J. Chem. Phys.*, 128:024107, 2008. DOI: 10.1063/1.2817600.

[67] K. V. Lawler, J. A. Parkhill, and M. Head-Gordon. *Mol. Phys.*, 106:2309, 2008. DOI: 10.1080/00268970802443482.

[68] K. V. Lawler, J. A. Parkhill, and M. Head-Gordon. *J. Chem. Phys.*, 130:184113, 2009. DOI: 10.1063/1.3134223.

[69] K. V. Lawler, D. W. Small, and M. Head-Gordon. *J. Phys. Chem. A*, 114:2930, 2010. DOI: 10.1021/jp911009f.

[70] J. Lee and M. Head-Gordon. *J. Chem. Theory Comput.*, 14:5203, 2018. DOI: 10.1021/acs.jctc.8b00731.

[71] J. Lee and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 21:47638, 2019. DOI: 10.1039/C8CP07613H.

[72] J. Lee, D. W. Small, E. Epifanovsky, and M. Head-Gordon. *J. Chem. Theory Comput.*, 13:602, 2017. DOI: 10.1021/acs.jctc.6b01092.

[73] J. Lee, L. Lin, and M. Head-Gordon. *J. Chem. Theory Comput.*, 16:243, 2020. DOI: 10.1021/acs.jctc.9b00820.

[74] M. S. Lee. *Ph. D. Thesis*. PhD thesis, University of California, Berkeley, CA, 2000.

[75] M. S. Lee and M. Head-Gordon. *Int. J. Quantum Chem.*, 76:169, 2000. DOI: 10.1002/(SICI)1097-461X(2000)76:2<169::AID-QUA7>3.0.CO;2-G.

[76] M. S. Lee, P. E. Maslen, and M. Head-Gordon. *J. Chem. Phys.*, 112:3592, 2000. DOI: 10.1063/1.480512.

[77] T. J. Lee and G. E. Scuseria. In S. R. Langhoff, editor, *Quantum Mechanical Calculations with Chemical Accuracy*, volume 13 of *Understanding Chemical Reactivity*, pages 47–108. Kluwer, Dordrecht, 1995. DOI: 10.1007/978-94-011-0193-6_2.

[78] D. S. Levine, D. Hait, N. M. Tubman, S. Lehtola, K. B. Whaley, and M. Head-Gordon. *J. Chem. Theory Comput.*, 16:2340, 2020. DOI: 10.1021/acs.jctc.9b01255.

[79] R. C. Lochan and M. Head-Gordon. *J. Chem. Phys.*, 126:164101, 2007. DOI: 10.1063/1.2718952.

[80] R. C. Lochan, Y. Jung, and M. Head-Gordon. *J. Phys. Chem. A*, 109:7598, 2005. DOI: 10.1021/jp0514426.

[81] R. C. Lochan, Y. Shao, and M. Head-Gordon. *J. Chem. Theory Comput.*, 3:988, 2007. DOI: 10.1021/ct600292h.

[82] P. Loos, A. Scemama, A. Blondel, Y. Garniron, M. Caffarel, and D. Jacquemin. *J. Chem. Theory Comput.*, 14: 4360, 2018. DOI: 10.1021/acs.jctc.8b00406.

[83] J. Malick, J. Povh, F. Rendl, and A. Wiegele. *SIAM J. Optim.*, 20:336, 2009. DOI: 10.1137/070704575.

[84] G. L. Manni, R. K. Carlson, S. Luo, D. Ma, J. Olsen, D. G. Truhlar, and L. Gagliardi. *J. Chem. Theory Comput.*, 9:3669, 2014. DOI: 10.1021/ct500483t.

[85] E. Maradzike, G. Gidofalvi, J. M. Turney, H. F. Schaefer III, and A. E. DePrince III. *J. Chem. Theory Comput.*, 13:4113, 2017. DOI: 10.1021/acs.jctc.7b00366.

[86] D. A. Mazziotti. *Phys. Rev. A*, 74:032501, 2006. DOI: 10.1103/PhysRevA.74.032501.

[87] D. A. Mazziotti. *Phys. Rev. Lett.*, 106:083001, 2011. DOI: 10.1103/PhysRevLett.106.083001.

[88] M. V. Mihailović and M. Rosina. *Nucl. Phys. A*, 237:221, 1975. DOI: 10.1016/0375-9474(75)90420-0.

[89] C. Møller and M. S. Plesset. *Phys. Rev.*, 46:618, 1934. DOI: 10.1103/PhysRev.46.618.

[90] M. Mostafanejad and E. A. DePrince III. *J. Chem. Theory Comput.*, 15:290, 2019. DOI: 10.1021/acs.jctc.8b00988.

[91] R. K. Nesbet. *Proc. Roy. Soc. Ser. A*, 230:312, 1955. DOI: 10.1098/rspa.1955.0134.

[92] B. D. Nguyen, G. P. Chen, M. M. Agee, A. M. Burow, M. P. Tang, and F. Furche. *J. Chem. Theory Comput.*, 16: 2258, 2020. DOI: 10.1021/acs.jctc.9b01176.

[93] J. Noga and R. J. Bartlett. *J. Chem. Phys.*, 86:7041, 1987. DOI: 10.1063/1.452353.

[94] P. Piecuch and M. Włoch. *J. Chem. Phys.*, 123:224105, 2005. DOI: 10.1063/1.2137318.

[95] P. Pokhilko, E. Epifanovsky, and A. I. Krylov. *J. Chem. Theory Comput.*, 14:4088, 2018. DOI: 10.1021/acs.jctc.8b00321.

[96] P. Pokhilko, D. Izmodenov, and A. I. Krylov. *J. Chem. Phys.*, 152:034105, 2020. DOI: 10.1063/1.5138643.

[97] J. A. Pople, M. Head-Gordon, and K. Raghavachari. *J. Chem. Phys.*, 87:5968, 1987. DOI: 10.1063/1.453520.

[98] J. Povh, F. Rendl, and A. Wiegele. *Computing*, 78:277, 2006. DOI: 10.1007/s00607-006-0182-2.

[99] G. D. Purvis and R. J. Bartlett. *J. Chem. Phys.*, 76:1910, 1982. DOI: 10.1063/1.443164.

[100] K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon. *Chem. Phys. Lett.*, 157:479, 1989. DOI: 10.1016/S0009-2614(89)87395-6.

[101] A. E. Rask and P. M. Zimmerman. *J. Phys. Chem. A*, 125:1598, 2021. DOI: 10.1021/acs.jpca.0c07624.

[102] V. A. Rassolov, J. A. Pople, P. C. Redfern, and L. A. Curtiss. *Chem. Phys. Lett.*, 350:573, 2001. DOI: 10.1016/S0009-2614(01)01345-8.

[103] R. M. Razban, D. Stück, and Martin Head-Gordon. *Mol. Phys.*, 115:21029, 2017. DOI: 10.1080/00268976.2017.1284355.

[104] A. Rettig, D. Hait, L. W. Bertels, and M. Head-Gordon. *J. Chem. Theory Comput.*, 16:7473, 2020. DOI: 10.1021/acs.jctc.0c00986.

[105] B. O. Roos. *Adv. Chem. Phys.*, 69:399, 1987. DOI: 10.1002/9780470142943.ch7.

[106] M. Rosina and C. Garrod. *J. Comput. Phys.*, 18:300, 1975. DOI: 10.1016/0021-9991(75)90004-2.

[107] K. Ruedenberg, M. W. Schmidt, M. M. Gilbert, and S. T. Elbert. *Chem. Phys.*, 71:41, 1982. DOI: 10.1016/0301-0104(82)87004-3.

[108] S. Saebo and P. Pulay. *Annu. Rev. Phys. Chem.*, 44:213, 1993. DOI: 10.1146/annurev.pc.44.100193.001241.

[109] G. E. Scuseria, T. M. Henderson, and D. C. Sorensen. *J. Chem. Phys.*, 129:231101, 2008. DOI: 10.1063/1.3043729.

[110] C. D. Sherrill, A. I. Krylov, E. F. C. Byrd, and M. Head-Gordon. *J. Chem. Phys.*, 109:4171, 1998. DOI: 10.1063/1.477023.

[111] D. W. Small and M. Head-Gordon. *J. Chem. Phys.*, 130:084103, 2009. DOI: 10.1063/1.3069296.

[112] D. W. Small and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 13:19285, 2011. DOI: 10.1039/c1cp21832h.

[113] D. W. Small and M. Head-Gordon. *J. Chem. Phys.*, 137:114103, 2012. DOI: 10.1063/1.4751485.

[114] A. Sodt, G. J. O. Beran, Y. Jung, B. Austin, and M. Head-Gordon. *J. Chem. Theory Comput.*, 2:300, 2006. DOI: 10.1021/ct050239b.

[115] E. Solomonik, D. Matthews, J. R. Hammond, J. F. Stanton, and J. Demmel. *J. Parallel Dist. Comp.*, 74:3176, 2014. DOI: 10.1016/j.jpdc.2014.06.002.

[116] C. Sosa, J. Geertsen, G. W. Trucks, and R. J. Bartlett. *Chem. Phys. Lett.*, 159:148, 1989. DOI: 10.1016/0009-2614(89)87399-3.

[117] R. P. Steele, R. A. DiStasio, Jr., Y. Shao, J. Kong, and M. Head-Gordon. *J. Chem. Phys.*, 125:074108, 2006. DOI: 10.1063/1.2234371.

[118] R. P. Steele, R. A. DiStasio, Jr., and M. Head-Gordon. *J. Chem. Theory Comput.*, 5:1560, 2009. DOI: 10.1021/ct900058p.

[119] D. Stück and M. Head-Gordon. *J. Chem. Phys.*, 139:244109, 2013. DOI: 10.1063/1.4851816.

[120] A. G. Taube and R. J. Bartlett. *Collect. Czech. Chem. Commun.*, 70:837, 2005. DOI: 10.1135/cccc20050837.

[121] A. G. Taube and R. J. Bartlett. *J. Chem. Phys.*, 128:164101, 2008. DOI: 10.1063/1.2902285.

[122] N. M. Tubman, J. Lee, T. Y. Takeshita, M. Head-Gordon, and K. B. Whaley. *J. Chem. Phys.*, 145:044112, 2016. DOI: 10.1063/1.4955109.

[123] T. Van Voorhis and M. Head-Gordon. *Chem. Phys. Lett.*, 317:575, 2000. DOI: 10.1016/S0009-2614(99)01413-X.

[124] T. Van Voorhis and M. Head-Gordon. *J. Chem. Phys.*, 113:8873, 2000. DOI: 10.1063/1.1319643.

[125] T. Van Voorhis and M. Head-Gordon. *Chem. Phys. Lett.*, 330:585, 2000. DOI: 10.1016/S0009-2614(00)01137-4.

[126] T. Van Voorhis and M. Head-Gordon. *J. Chem. Phys.*, 115:7814, 2001. DOI: 10.1063/1.1406536.

[127] T. Van Voorhis and M. Head-Gordon. *J. Chem. Phys.*, 117:9190, 2002. DOI: 10.1063/1.1515319.

[128] M. Hernandez Vera and T.-C. Jagau. *J. Chem. Phys.*, 151:111101, 2019. DOI: 10.1063/1.5119695.

[129] M. Hernandez Vera and T.-C. Jagau. *J. Chem. Phys.*, 152:174103, 2020. DOI: 10.1063/5.0004843.

[130] F. Weigend and M. Häser. *Theor. Chem. Acc.*, 97:331, 1997. DOI: 10.1007/s002140050269.

[131] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs. *Chem. Phys. Lett.*, 294:143, 1998. DOI: 10.1016/S0009-2614(98)00862-8.

[132] F. Weigend, A. Kohn, and C. Hättig. *J. Chem. Phys.*, 116:3175, 2002. DOI: 10.1063/1.1445115.

[133] A. F. White, M. Head-Gordon, and C. W. McCurdy. *J. Chem. Phys.*, 142:054103, 2015. DOI: 10.1063/1.4906940.

[134] A. F. White, C. W. McCurdy, and M. Head-Gordon. *J. Chem. Phys.*, 143:074103, 2015. DOI: 10.1063/1.4928529.

[135] S. Wilson. *Comput. Phys. Commun.*, 58:71–81, 1990. DOI: 10.1016/0010-4655(90)90136-O.

[136] Z. Zhao, B. J. Braams, M. Fukuda, M. L. Overton, and J. K. Percus. *J. Chem. Phys.*, 120:2095, 2004. DOI: 10.1063/1.1636721.

[137] P. M. Zimmerman. *J. Chem. Phys.*, 146:104102, 2017. DOI: 10.1063/1.4977727.

[138] P. M. Zimmerman. *J. Chem. Phys.*, 146:224104, 2017. DOI: 10.1063/1.4985566.

[139] P. M. Zimmerman. *J. Phys. Chem. A*, 121:4712, 2017. DOI: 10.1021/acs.jpca.7b03998.

Chapter 7

Open-Shell and Excited-State Methods

7.1 General Excited-State Features

As for ground state calculations, performing an adequate excited-state calculation involves making an appropriate choice of method and basis set. The development of effective approaches to modeling electronic excited states has historically lagged behind advances in treating the ground state. In part this is because of the much greater diversity in the character of the wave functions for excited states, making it more difficult to develop broadly applicable methods without molecule-specific or even state-specific specification of the form of the wave function. Recently, however, a hierarchy of single-reference *ab initio* methods has begun to emerge for the treatment of excited states. Broadly speaking, Q-CHEM contains methods that are capable of giving qualitative agreement, and in many cases quantitative agreement with experiment for lower optically allowed states. The situation is less satisfactory for states that involve two-electron excitations, although even here reasonable results can sometimes be obtained. Moreover, some of the excited state methods can treat open-shell wave functions, *e.g.* diradicals, ionized and electron attachment states and more.[141]

In excited-state calculations, as for ground state calculations, the user must strike a compromise between cost and accuracy. This chapter summarizes Q-CHEM's capabilities in four general classes of excited state methods:

- Single-electron wave function-based methods (Section 7.2). These are excited state treatments of roughly the same level of sophistication as the Hartree-Fock ground state method, in the sense that electron correlation is essentially ignored. Single excitation configuration interaction (CIS) is the workhorse method of this type. The spin-flip variant of CIS extends it to diradicals.
- Time-dependent density functional theory (TDDFT, Section 7.3). TDDFT is a widely used extension of DFT to excited states. For a cost that is only a little larger than that of a CIS calculation, TDDFT typically affords significantly greater accuracy due to a treatment of electron correlation. It, too, has a spin-flip variant that can be used to study di- and tri-radicals as well as bond breaking.
- The Maximum Overlap Method (MOM) for excited ΔSCF states (Section 7.6). This method overcomes some of the deficiencies of TDDFT and, in particular, can be used for modeling charge-transfer and Rydberg transitions as well as core-excited states.
- Restricted open-shell Kohn-Sham (ROKS) method is a spin-purified, orbital optimized approach for excited states (Section 7.8.3). It is very accurate for modeling charge-transfer states and core-excitations.
- The Square Gradient Minimization (SGM) algorithm can be used to converge both ΔSCF and ROKS excited states. Details about using SGM in practice can be found in Section 4.5.15.

- The State-Targeted Energy Projection (STEP) algorithm (Section 7.8.5) is available for ΔSCF and ROKS excited states. It is less expensive than SGM and usually more robust than MOM.
- Wave function-based electron correlation treatments (Sections 7.9, 7.11, 7.12 and 7.10). Roughly speaking, these are excited state analogues of the ground state wave function-based electron correlation methods discussed in Chapter 6. They are more accurate than the methods of Section 7.2, but also significantly more computationally expensive. These methods can also describe certain multi-configurational wave functions, for example, problematic doublet radicals, diradicals, triradicals, and more.

Note: Core electrons are frozen by default in most correlated excited-state calculations (see Section 6.2).

In general, a basis set appropriate for a ground state density functional theory or a Hartree-Fock calculation will be appropriate for describing valence excited states. However, many excited states involve significant contributions from diffuse Rydberg orbitals, and, therefore, it is often advisable to use basis sets that include additional diffuse functions. The 6-31+G* basis set is a reasonable compromise for the low-lying valence excited states of many organic molecules. To describe true Rydberg excited states, Q-CHEM allows the user to add two or more sets of diffuse functions (see Chapter 8). For example the 6-311(2+)G* basis includes two sets of diffuse functions on heavy atoms and is generally adequate for description of both valence and Rydberg excited states.

Q-CHEM supports four main types of excited state calculation:

- **Vertical absorption spectrum**
  This is the calculation of the excited states of the molecule at the ground state geometry, as appropriate for absorption spectroscopy. The methods supported for performing a vertical absorption calculation are: CIS, RPA, XCIS, SF-XCIS, CIS(D), ADC(2)-s, ADC(2)-x, ADC(3), RAS-SF, EOM-CCSD and EOM-OD, each of which will be discussed in turn. The calculation of core-excited states for the simulation of X-ray absorption spectra can be performed with TDDFT as well as EOM-CCSD and ADC within the CVS approximation (Section 7.13). All ADC- and EOM-based methods can be combined with the polarizable continuum model (PCM) to model the absorption spectrum in solution following state-specific nonequilibrium approach. Most EOM methods can be combined with explicit solvent treatments using classical (QM/MM) and polarizable (QM/EFP) embedding.
- **Visualization**
  It is possible to visualize the excited states either by attachment/detachment density analysis (available for CIS, RPA, TDDFT, ADC, EOM-CC) or by plotting the transition density (see *$plots* descriptions in Chapters 3 and 10). Transition densities can be calculated for CIS, EOM-CCSD, and ADC methods. The theoretical basis of the attachment/detachment density analysis is discussed in Section 7.14.2 of this Chapter (more details are given in Section 10.2.12). In addition Dyson orbitals can be calculated and plotted for ionization from the ground and electronically excited states or detachment from electron-attached states for CCSD and EOM-CCSD wave functions. For the RAS-SF method (Section 7.12), one can plot the natural orbitals of a computed electronic state.
- **Excited-state optimization**
  Optimization of the geometry of stationary points on excited state potential energy surfaces is valuable for understanding the geometric relaxation that occurs between the ground and excited state. Analytic first derivatives are available for UCIS, RCIS, TDDFT and EOM-CCSD. Excited state optimizations may also be performed using finite difference methods, however, these can be very time-consuming to perform.
- **Optimization of the crossings between potential energy surfaces**
  Seams between potential energy surfaces can be located and optimized by using analytic gradients within EOM-CCSD, CIS, and TD-DFT formalisms.

- **Properties**
  Properties such as dipole moments, spatial extent of electron densities and $\langle \hat{S}^2 \rangle$ values can be computed for ADC, EOM-CCSD, EOM-MP2, EOM-OD, RAS-SF and CIS wave functions. Static polarizabilities are available for CCSD, EOM-EE-CCSD, and EOM-SF-CCSD methods.

- **Transition properties and state interactions**
  Transition dipole moments and oscillator strengths can be computed with practically all excited-state methods. Matrix elements and cross-sections for two-photon absorption are available for EOM-EE-CCSD and ADC methods. Spin-orbit couplings can be computed for EOM-CCSD, RAS-SF, CIS, and TDDFT wave functions. Dyson orbitals are available for EOM-CC wave functions. Transition properties can be computed between the reference and target states (*e.g.*, HF-CIS) or between different target states (*e.g.*, CIS-CIS).

- **Excited-state vibrational analysis**
  Given an optimized excited state geometry, Q-CHEM can calculate the force constants at the stationary point to predict excited state vibrational frequencies. Stationary points can also be characterized as minima, transition structures or nth-order saddle points. Analytic excited state vibrational analysis can only be performed using the UCIS, RCIS, and TDDFT methods, for which efficient analytical second derivatives are available. EOM-CCSD frequencies are also available using analytic first derivatives and second derivatives obtained from finite difference methods. EOM-OD frequencies are only available through finite difference calculations.

Note: EOM-CC and most of the CI codes are part of CCMAN and CCMAN2. CCMAN is a legacy code which is being phased out. All new developments and performance-enhancing features are implemented in CCMAN2.

METHOD

Specifies the level of theory.

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

CIS	Section 7.2.2
CIS(D)	Section 7.9.2
RI-CIS(D)	Section 7.9.3
SOS-CIS(D)	Section 7.9.4
SOS-CIS(D0)	Section 7.9.5
CISD	Section 7.10.3
CISDT	Section 7.10.3
EOM-OD	Section 7.10.3
EOM-CCSD	Section 7.10.3
EOM-MP2	Section 7.10.15.1
EOM-MP2T	Section 7.10.15.1
EOM-CCSD-S(D)	Section 7.10.15.2
EOM-MP2-S(D)	Section 7.10.15.2
EOM-CCSD(dT)	Section 7.10.26
EOM-CCSD(fT)	Section 7.10.26
EOM-CC(2,3)	Section 7.10.23
ADC(0)	Section 7.11
ADC(1)	Section 7.11
ADC(2)	Section 7.11
ADC(2)-X	Section 7.11
ADC(3)	Section 7.11
SOS-ADC(2)	Section 7.11
SOS-ADC(2)-X	Section 7.11
CVS-ADC(1)	Section 7.11
CVS-ADC(2)	Section 7.11
CVS-ADC(2)-X	Section 7.11
CVS-ADC(3)	Section 7.11
RAS-CI	Section 7.12
RAS-CI-2	Section 7.12

RECOMMENDATION:

Consult the literature for guidance.

7.2 Uncorrelated Wave Function Methods

7.2.1 Introduction

Q-CHEM includes several excited state methods which do not incorporate correlation: CIS, XCIS, and RPA, along with some spin-flip variants thereof.[42,108] These methods are sufficiently inexpensive that calculations on large molecules are possible, and are roughly comparable to the HF treatment of the ground state in terms of performance. They tend to yield qualitative rather than quantitative insight. Excitation energies tend to exhibit errors on the order of an electron

volt, consistent with the neglect of electron correlation effects, which are generally different in the ground state and the excited state.

7.2.2 Configuration Interaction with Single Substitutions (CIS)

7.2.2.1 Theory

The derivation of the CI-singles (CIS) energy and wave function[57,72] begins by selecting the HF single-determinant wave function as reference for the ground state of the system:

$$\Psi_{\mathrm{HF}} = \frac{1}{\sqrt{n!}} \det\left\{\chi_1 \chi_2 \cdots \chi_i \chi_j \cdots \chi_n\right\} \tag{7.1}$$

where n is the number of electrons, and the spin orbitals

$$\chi_i = \sum_{\mu}^{N} c_{\mu i} \phi_\mu \tag{7.2}$$

are expanded in a finite basis of N atomic orbital basis functions. Molecular orbital coefficients $\{c_{\mu i}\}$ are usually found by SCF procedures which solve the Hartree-Fock equations

$$\mathbf{FC} = \boldsymbol{\varepsilon}\mathbf{SC}\ , \tag{7.3}$$

where **S** is the overlap matrix, **C** is the matrix of molecular orbital coefficients, ε is a diagonal matrix of orbital eigenvalues and **F** is the Fock matrix with elements

$$F_{\mu\upsilon} = H_{\mu\upsilon} + \sum_{\lambda\sigma} \sum_{i} c_{\mu i} c_{\upsilon i} \left(\mu\lambda \,||\, \upsilon\sigma\right) \tag{7.4}$$

involving the core Hamiltonian and the anti-symmetrized two-electron integrals

$$(\mu\mu||\lambda\sigma) = \int\int \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_2) \left(\frac{1}{r_{12}}\right) \left[\phi_\lambda(\mathbf{r}_1)\phi_\sigma(\mathbf{r}_2) - \phi_\sigma(\mathbf{r}_1)\phi_\lambda(\mathbf{r}_2)\right] d\mathbf{r}_1\, d\mathbf{r}_2 \tag{7.5}$$

On solving Eq. (7.3), the total energy of the ground state single determinant can be expressed as

$$E_{\mathrm{HF}} = \sum_{\mu\upsilon} P^{\mathrm{HF}}_{\mu\upsilon} H_{\mu\upsilon} + \frac{1}{2} \sum_{\mu\upsilon\lambda\sigma} P^{\mathrm{HF}}_{\mu\upsilon} P^{\mathrm{HF}}_{\lambda\sigma} \left(\mu\lambda \,||\, \upsilon\sigma\right) + V_{\mathrm{nuc}} \tag{7.6}$$

where P^{HF} is the HF density matrix and V_{nuc} is the nuclear repulsion energy.

Equation (7.1) represents only one of many possible determinants made from orbitals of the system; there are in fact $n(N-n)$ possible singly substituted determinants constructed by replacing an orbital occupied in the ground state (i, j, k, ...) with an orbital unoccupied in the ground state (a, b, c, ...). Such wave functions and energies can be written

$$\Psi_i^a = \frac{1}{\sqrt{n!}} \det\left\{\chi_1 \chi_2 \cdots \chi_a \chi_j \cdots \chi_n\right\} \tag{7.7}$$

$$E_{ia} = E_{\mathrm{HF}} + \varepsilon_a - \varepsilon_i - \left(ia \,||\, ia\right) \tag{7.8}$$

where we have introduced the anti-symmetrized two-electron integrals in the molecular orbital basis

$$\left(pq \,||\, rs\right) = \sum_{\mu\upsilon\lambda\sigma} c_{\mu p} c_{\upsilon q} c_{\lambda r} c_{\sigma s} \left(\mu\lambda \,||\, \upsilon\sigma\right) \tag{7.9}$$

These singly excited wave functions and energies could be considered crude approximations to the excited states of the system. However, determinants of the form Eq. (7.7) are deficient in that they:

- do not yield pure spin states
- resemble more closely ionization rather than excitation
- are not appropriate for excitation into degenerate states

These deficiencies can be partially overcome by representing the excited state wave function as a linear combination of *all* possible singly excited determinants,

$$|\Psi_{\text{CIS}}\rangle = \sum_{ia} a_i^a |\Psi_i^a\rangle \tag{7.10}$$

where the coefficients $\{a_i^a\}$ can be obtained by diagonalizing the orbital Hessian matrix $\mathbf{A}$, which is equivalent to the Hamiltonian in the space of single excitations with the ground-state energy removed. Its matrix elements are

$$A_{ia,jb} = \langle \Psi_i^a | \hat{H} - E_{\text{HF}} | \Psi_j^b \rangle = (\varepsilon_a - \varepsilon_j)\delta_{ij}\delta_{ab} - (ja||ib) \; . \tag{7.11}$$

According to Brillouin's, theorem single substitutions do not interact directly with a reference HF determinant, so the resulting eigenvectors from the CIS excited state represent a treatment roughly comparable to that of the HF ground state. The excitation energy is simply the difference between HF ground state energy and CIS excited state energies, and the eigenvectors of $\mathbf{A}$ correspond to the amplitudes of the single-electron promotions.

CIS calculations can be performed in Q-CHEM using restricted (RCIS),[57,72] unrestricted (UCIS), or restricted open-shell[174] (ROCIS) spin orbitals. See Section 7.2.4 for a list of CIS-related job control variables.

Example 7.1 A basic CIS excitation energy calculation on formaldehyde at the HF/6-31G* optimized ground state geometry, which is obtained in the first part of the job. Above the first singlet excited state, the states have Rydberg character, and therefore a basis with two sets of diffuse functions is used.

```
$molecule
   0  1
   C
   O  1  CO
   H  1  CH  2  A
   H  1  CH  2  A  3  D

   CO =   1.2
   CH =   1.0
   A  = 120.0
   D  = 180.0
$end

$rem
   JOBTYPE    =  opt
   EXCHANGE   =  hf
   BASIS      =  6-31G*
$end

@@@

$molecule
   read
$end

$rem
   EXCHANGE       =  hf
   BASIS          =  6-311(2+)G*
   CIS_N_ROOTS    =  15            Do 15 states
   CIS_SINGLETS   =  true          Do do singlets
   CIS_TRIPLETS   =  false         Don't do Triplets
$end
```

7.2.2.2 CIS Analytical Derivatives

While CIS excitation energies are relatively inaccurate, with errors of the order of 1 eV, CIS excited state properties, such as structures and frequencies, are much more useful. This is very similar to the manner in which ground state Hartree-Fock (HF) structures and frequencies are much more accurate than HF relative energies. Generally speaking, for low-lying excited states, it is expected that CIS vibrational frequencies will be systematically 10% higher or so relative to experiment.[80,238,279] If the excited states are of pure valence character, then basis set requirements are generally similar to the ground state. Excited states with partial Rydberg character require the addition of one or preferably two sets of diffuse functions.

Q-CHEM includes efficient analytical first and second derivatives of the CIS energy,[173,176] to yield analytical gradients, excited state vibrational frequencies, force constants, polarizabilities, and infrared intensities. Analytical gradients can be evaluated for any job where the CIS excitation energy calculation itself is feasible, so that efficient excited-state geometry optimizations and vibrational frequency calculations are possible at the CIS level. In such cases, it is necessary to specify on which Born-Oppenheimer potential energy surface the optimization should proceed, and care must be taken to ensure that the optimization remains on the excited state of interest, as state crossings may occur. (A state-tracking algorithm, as discussed in Section 9.8.5, can aid with this.[274])

Sometimes it is precisely the crossings between Born-Oppenheimer potential energy surfaces (*i.e.*, conical intersections) that are of interest, as these intersections provide pathways for nonadiabatic transitions between electronic states.[110,170] A feature of Q-CHEM that is not otherwise widely available in an analytic implementation (for both CIS and TDDFT) of the nonadiabatic couplings that define the topology around conical intersections.[68,198,272,273] Due to the analytic implementation, these couplings can be evaluated at a cost that is not significantly greater than the cost of a CIS or TDDFT analytic gradient calculation,[272] and the availability of these couplings allows for much more efficient optimization of minimum-energy crossing points along seams of conical intersection, as compared to when only analytic gradients are available.[272] These features, including a brief overview of the theory of conical intersections, can be found in Section 9.8.1.

For CIS vibrational frequencies, a semi-direct algorithm similar to that used for ground-state Hartree-Fock frequencies is available, whose computer time scales as approximately $\mathcal{O}(N^3)$ for large molecules.[175] The main complication associated with analytical CIS frequency calculations is ensuring that Q-CHEM has sufficient memory to perform the calculations. Default settings are adequate for many purposes but if a large calculation fails due to a memory limitation, then the following additional information may be useful.

The memory requirements for CIS (and HF) analytic frequencies primarily come from dynamic memory, defined as

$$\text{dynamic memory} = \text{MEM\_TOTAL} - \text{MEM\_STATIC} \quad .$$

This quantity must be large enough to contain several arrays whose size is $3N_{\text{atoms}}N_{\text{basis}}^2$. Meanwhile the value of the *$rem* variable MEM_STATIC, which obviously reduces the available dynamic memory, must be sufficiently large to permit integral evaluation, else the job may fail. For most purposes, setting MEM_STATIC to about 80 MB is sufficient, and by default MEM_TOTAL is set to a larger value that what is available on most computers, so that the user need not guess or experiment about an appropriate value of MEM_TOTAL for low-memory jobs. However, a memory allocation error will occur if the calculation demands more memory than available.

Note: Unlike Q-CHEM's MP2 frequency code, the analytic CIS second derivative code currently does not support frozen core or virtual orbitals. These approximations do not lead to large savings at the CIS level, as all computationally-expensive steps are performed in the atomic orbital basis.

Example 7.2 This example illustrates a CIS geometry optimization followed by a vibrational frequency analysis on the lowest singlet excited state of formaldehyde. This $n \rightarrow \pi^*$ excited state is non-planar, unlike the ground state. The optimization converges to a non-planar structure with zero forces, and all frequencies real.

```
$molecule
   0 1
   C
   O  1  CO
   H  1  CH  2  A
   H  1  CH  2  A  3  D

   CO  =    1.2
   CH  =    1.0
   A   =  120.0
   D   =  150.0
$end

$rem
   JOBTYPE           =  opt
   EXCHANGE          =  hf
   BASIS             =  6-31+G*
   CIS_STATE_DERIV   =  1         Optimize state 1
   CIS_N_ROOTS       =  3         Do 3 states
   CIS_SINGLETS      =  true      Do do singlets
   CIS_TRIPLETS      =  false     Don't do Triplets
$end

@@@

$molecule
  read
$end

$rem
   JOBTYPE           =  freq
   EXCHANGE          =  hf
   BASIS             =  6-31+G*
   CIS_STATE_DERIV   =  1         Focus on state 1
   CIS_N_ROOTS       =  3         Do 3 states
   CIS_SINGLETS      =  true      Do do singlets
   CIS_TRIPLETS      =  false     Don't do Triplets
$end
```

7.2.2.3 Random Phase Approximation

The Random Phase Approximation (RPA),[26,93] also known as time-dependent Hartree-Fock (TD-HF) theory, is an alternative to CIS for uncorrelated calculations of excited states. An RPA calculation is requested using the RPA *$rem* variable that is described in Section 7.2.4.1.

RPA offers some advantages for computing oscillator strengths, *e.g.*, exact satisfaction of the Thomas-Reike-Kuhn sum rule,[181] and is roughly comparable in accuracy to CIS for singlet excitation energies, but is inferior for triplet states. RPA energies are non-variational, and in moving around on excited-state potential energy surfaces, this method can occasionally encounter singularities (arising from triplet instabilities in the underlying reference state) that prevent numerical solution of the RPA equations.[53,106,107] In Q-CHEM, these instabilities are generally accompanied by an error

message to the effect that an imaginary root has been detected in the RPA equations. This is mathematically impossible in CIS calculations, which decouple the excitation energy problem from the stability problem.[106]

7.2.2.4 DFT/CIS

DFT/CIS[84] is a semi-empirical method that uses Kohn-Sham orbitals instead of Hartree-Fock orbitals with a CIS-like formalism. This leads to added correlation energy from the Kohn-Sham orbitals and, perhaps more importantly, orbital energy differences $\varepsilon_a - \varepsilon_i$ that better resemble excitation energies, and bound virtual orbitals that better approximate localized excited states as compared to Hartree-Fock virtual orbitals, which are typically unbound.[107] Along with certain semi-empirical corrections, DFT/CIS significantly improves conventional Hartree–Fock-based CIS. Two parameterizations of the DFT/CIS Hamiltonian are available in Q-CHEM: Grimme's original one,[84] which is designed for use with the B3LYP functional, and a new parameterization designed for CAM-B3LYP.[163] The latter (CAM-B3LYP/CIS) implementation has specific parameterizations aimed at accurately reproducing X-ray spectra, whereas B3LYP/CIS is suitable only for valence excitations. The nature of the parameterization limits use of DFT/CIS to the aforementioned two functionals, and primarily to two basis sets. The CAM-B3LYP/CIS method is parameterized for use with def2-TZVPD while B3LYP/CIS is parameterized for use with TZVP.

The DFT/CIS method is controlled by the DFTCIS and DFTCIS_PARAMS job control variables that are described in Section 7.2.4.1.

Example 7.3 A DFT/CIS (B3LYP/CIS) calculation of the first five singlet excited states of water.

```
$molecule
0 1
O
H   1 0.960652
H   1 0.960652 2 103.913458
$end

$rem
JOBTYPE         = sp
METHOD          = b3lyp
UNRESTRICTED    = false
BASIS           = TZVP
DFTCIS          = true
DFTCIS_PARAMS   = 1 !0=CIS, 1=B3LYP/CIS, 2=CAM-B3LYP/CIS
CIS_N_ROOTS     = 5
CIS_TRIPLETS    = false
SCF_ALGORITHM   = diis
$end
```

The core/valence separation (CVS) approximation is implemented for use with the DFT/CIS method, which can therefore be used to compute core-level excitation spectra at elemental K-, L-, or M-edges.[162,163] L-edge spectra require spin-orbit couplings, which are discussed in Section 7.3.8.1. Example 7.19 in that section illustrates a DFT/CIS of an L-edge spectrum, including spin-orbit couplings. The Q-CHEM output file from this example can be read by the PYSETSOC program[161] and used to construct and diagonalize the Breit-Pauli Hamiltonian, which affords the coupled excited states and oscillator strengths.[162]

7.2.3 CIS Methods with Extended Excitation Manifolds

A variety of CIS-like methods have been proposed that add a limited number of double excitations to the CIS excitation space,[108] in order to overcome certain deficiencies of CIS without incurring the prohibitive $\mathcal{O}(N^6)$ cost of a method

such as CISD or CCSD that includes the full complement of double excitations. These "extended" CIS methods are discussed in this section, starting with the original excited CIS (XCIS) procedure of Maurice and Head-Gordon.[175]

7.2.3.1 Extended CIS (XCIS)

The motivation for XCIS stems from the fact that ROCIS and UCIS are less effective for radicals than CIS is for closed shell molecules. Using the attachment/detachment density analysis procedure,[100,107] the failing of ROCIS and UCIS methodologies for the nitromethyl radical was traced to the neglect of a particular class of double substitution which involves the simultaneous promotion of an α spin electron from the singly occupied orbital and the promotion of a β spin electron into the singly occupied orbital. The spin-adapted configurations

$$\left|\tilde{\Psi}_i^a(1)\right\rangle = \frac{1}{\sqrt{6}}\left(|\Psi_{\bar{i}}^{\bar{a}}\rangle - |\Psi_i^a\rangle\right) + \frac{2}{\sqrt{6}}|\Psi_{p\bar{i}}^{a\bar{p}}\rangle \tag{7.12}$$

are of crucial importance. (Here, $a, b, c, \ldots$ are virtual orbitals; $i, j, k, \ldots$ are occupied orbitals; and $p, q, r, \ldots$ are singly-occupied orbitals.) It is quite likely that similar excitations are also very significant in other radicals of interest.

The XCIS proposal, a more satisfactory generalization of CIS to open shell molecules, is to simultaneously include a restricted class of double substitutions similar to those in Eq. (7.12). To illustrate this, consider the resulting orbital spaces of an ROHF calculation: doubly occupied (d), singly occupied (s) and virtual (v). From this starting point we can distinguish three types of single excitations of the same multiplicity as the ground state: $d \rightarrow s$, $s \rightarrow v$ and $d \rightarrow v$. Thus, the spin-adapted ROCIS wave function is

$$|\Psi_{\mathrm{ROCIS}}\rangle = \frac{1}{\sqrt{2}}\sum_{ia}^{dv} a_i^a\left(|\Psi_i^a\rangle + |\Psi_{\bar{i}}^{\bar{a}}\rangle\right) + \sum_{pa}^{sv} a_p^a|\Psi_p^a\rangle + \sum_{ip}^{ds} a_{\bar{i}}^{\bar{p}}|\Psi_{\bar{i}}^{\bar{p}}\rangle \tag{7.13}$$

The extension of CIS theory to incorporate higher excitations maintains the ROHF as the ground state reference and adds terms to the ROCIS wave function similar to that of Eq. (7.13), as well as those where the double excitation occurs through different orbitals in the α and β space:

$$\begin{aligned}|\Psi_{\mathrm{XCIS}}\rangle = {} & \tfrac{1}{\sqrt{2}}\sum_{ia}^{dv} a_i^a\left(|\Psi_i^a\rangle + |\Psi_{\bar{i}}^{\bar{a}}\rangle\right) + \sum_{pa}^{sv} a_p^a|\Psi_p^a\rangle + \sum_{ip}^{ds} a_{\bar{i}}^{\bar{p}}|\Psi_{\bar{i}}^{\bar{p}}\rangle \\ & + \sum_{iap}^{dvs}\tilde{a}_i^a(p)|\tilde{\Psi}_i^a(p)\rangle + \sum_{ia,p\neq q}^{dv,ss} a_{p\bar{i}}^{a\bar{q}}|\Psi_{p\bar{i}}^{a\bar{q}}\rangle\end{aligned} \tag{7.14}$$

XCIS is defined only from a restricted open shell Hartree-Fock ground state reference, as it would be difficult to uniquely define singly occupied orbitals in a UHF wave function. In addition, β unoccupied orbitals, through which the spin-flip double excitation proceeds, may not match the half-occupied α orbitals in either character or even symmetry.

For molecules with closed shell ground states, both the HF ground and CIS excited states emerge from diagonalization of the Hamiltonian in the space of the HF reference and singly excited substituted configuration state functions. The XCIS case is different because the restricted class of double excitations included could mix with the ground state and lower its energy. This mixing is avoided to maintain the size consistency of the ground state energy.

With the inclusion of the restricted set of doubles excitations in the excited states, but not in the ground state, it could be expected that some fraction of the correlation energy be recovered, resulting in anomalously low excited state energies. However, the fraction of the total number of doubles excitations included in the XCIS wave function is very small and those introduced cannot account for the pair correlation of any pair of electrons. Thus, the XCIS procedure can be considered one that neglects electron correlation.

The computational cost of XCIS is approximately four times greater than CIS and ROCIS, and its accuracy for open shell molecules is generally comparable to that of the CIS method for closed shell molecules. In general, it achieves

qualitative agreement with experiment. XCIS is available for doublet and quartet excited states beginning from a doublet ROHF treatment of the ground state, for excitation energies only.

Example 7.4 An XCIS calculation of excited states of an unsaturated radical, the phenyl radical, for which double substitutions make considerable contributions to low-lying excited states.

```
$comment
   C6H5 phenyl radical C2v symmetry MP2(full)/6-31G* = -230.7777459
$end

$molecule
   0  2
   c1
   x1   c1   1.0
   c2   c1   rc2   x1   90.0
   x2   c2   1.0   c1   90.0   x1    0.0
   c3   c1   rc3   x1   90.0   c2    tc3
   c4   c1   rc3   x1   90.0   c2   -tc3
   c5   c3   rc5   c1    ac5   x1  -90.0
   c6   c4   rc5   c1    ac5   x1   90.0
   h1   c2   rh1   x2   90.0   c1  180.0
   h2   c3   rh2   c1    ah2   x1   90.0
   h3   c4   rh2   c1    ah2   x1  -90.0
   h4   c5   rh4   c3    ah4   c1  180.0
   h5   c6   rh4   c4    ah4   c1  180.0

   rh1   =    1.08574
   rh2   =    1.08534
   rc2   =    2.67299
   rc3   =    1.35450
   rh4   =    1.08722
   rc5   =    1.37290
   tc3   =   62.85
   ah2   =  122.16
   ah4   =  119.52
   ac5   =  116.45
$end

$rem
   BASIS             =  6-31+G*
   EXCHANGE          =  hf
   MEM_STATIC        =  80
   INTSBUFFERSIZE    =  15000000
   SCF_CONVERGENCE   =  8
   CIS_N_ROOTS       =  5
   XCIS              =  true
$end
```

7.2.3.2 Spin-Flip Extended CIS (SF-XCIS)

Spin-flip extended CIS (SF-XCIS)[39] is a spin-complete extension of the spin-flip single excitation configuration interaction (SF-CIS) method.[138] The method includes all configurations in which no more than one virtual level of the high spin triplet reference becomes occupied and no more than one doubly occupied level becomes vacant.

SF-XCIS is defined only from a restricted open shell Hartree-Fock triplet ground state reference. The final SF-XCIS wave functions correspond to spin-pure $M_S = 0$ (singlet or triplet) states. The fully balanced treatment of the half-occupied reference orbitals makes it very suitable for applications with two strongly correlated electrons, such as single bond dissociation, systems with important diradical character or the study of excited states with significant double excitation character.

The computational cost of SF-XCIS scales in the same way with molecule size as CIS itself, with a pre-factor 13 times larger.

Example 7.5 A SF-XCIS calculation of ground and excited states of trimethylenemethane (TMM) diradical, for which double substitutions make considerable contributions to low-lying excited states.

```
$molecule
   0 3
   C
   C 1 CC1
   C 1 CC2   2 A2
   C 1 CC2   2 A2    3 180.0
   H 2 C2H   1 C2CH  3 0.0
   H 2 C2H   1 C2CH  4 0.0
   H 3 C3Hu  1 C3CHu 2 0.0
   H 3 C3Hd  1 C3CHd 4 0.0
   H 4 C3Hu  1 C3CHu 2 0.0
   H 4 C3Hd  1 C3CHd 3 0.0

   CC1    = 1.35
   CC2    = 1.47
   C2H    = 1.083
   C3Hu   = 1.08
   C3Hd   = 1.08
   C2CH   = 121.2
   C3CHu  = 120.3
   C3CHd  = 121.3
   A2     = 121.0
$end

$rem
  UNRESTRICTED    = false   SF-XCIS runs from ROHF triplet reference
  EXCHANGE        = HF
  BASIS           = 6-31G*
  SCF_CONVERGENCE = 10
  SCF_ALGORITHM   = DM
  MAX_SCF_CYCLES  = 100
  SPIN_FLIP_XCIS  = true    Do SF-XCIS
  CIS_N_ROOTS     = 3
  CIS_SINGLETS    = true    Do singlets
  CIS_TRIPLETS    = true    Do triplets
$end
```

7.2.3.3 Spin-Adapted Spin-Flip CIS

Spin-Adapted Spin-Flip CIS (SA-SF-CIS)[274] is a spin-complete extension of the spin-flip single excitation configuration interaction (SF-CIS) method.[138] Unlike SF-XCIS, SA-SF-CIS only includes the minimal set of electronic configurations needed to remove the spin contamination in the conventional SF-CIS method. The target SA-SF-CIS states have spin eigenvalues one less than the reference ROHF state, *i.e.*, if singlet states ($S = 0$) are targeted then the reference state should be a triplet ($S = 1$), or if doublet states ($S = 1/2$) are targeted then the reference state should be a quartet ($S = 3/2$). The SA-SF-CIS approach uses a tensor equation-of-motion formalism,[274] such that the dimension of the CI vectors in SA-SF-CIS remains exactly the same as that in conventional SF-CIS. A DFT correction to SA-SF-CIS (*i.e.*, SA-SF-TDDFT) can be added.[274] As with other SF-TDDFT methods,[275] the BH&HLYP functional has become

something of a *de facto* standard choice.[108,110]

Example 7.6 An SA-SF-DFT calculation of singlet ground and excited states of ethylene.

```
$molecule
0 3
 C
 C    1    B1
 H    1    B2    2    A1
 H    1    B3    2    A2    3    D1
 H    2    B4    1    A3    3    D2
 H    2    B5    1    A4    3    D3

   B1       1.32808942
   B2       1.08687297
   B3       1.08687297
   B4       1.08687297
   B5       1.08687297
   A1     121.62604150
   A2     121.62604150
   A3     121.62604150
   A4     121.62604150
   D1     180.00000000
   D2       0.00000000
   D3     180.00000000
$end

$rem
   EXCHANGE        bhhlyp
   BASIS           cc-pvtz
   BASIS2          sto-3g
   UNRESTRICTED    false
   CIS_N_ROOTS     5
   SASF_CIS        1
   CIS_TRIPLETS    false
$end
```

7.2.4 Job Control Options

Basic job control variables to run CIS and related methods are discussed in Section 7.2.4.1 and more advanced options are listed in Section 7.2.4.2.

7.2.4.1 Basic Job Control

CIS-type jobs are requested by setting the *$rem* variable EXCHANGE = HF and CORRELATION = NONE, as in a ground-state Hartree-Fock calculation, but then also specifying a number of excited-state roots using the *$rem* keyword CIS_N_ROOTS.

Note: For RHF case, n singlets and n triplets will be computed, unless specified otherwise by using CIS_TRIPLETS and CIS_SINGLETS.

CIS_N_ROOTS

Sets the number of CI-Singles (CIS) excited state roots to find.

TYPE:

INTEGER

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

n $n > 0$ Looks for n CIS excited states.

RECOMMENDATION:

None

CIS_SINGLETS

Solve for singlet excited states in RCIS calculations (ignored for UCIS).

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for singlet states.

FALSE Do not solve for singlet states.

RECOMMENDATION:

None

CIS_TRIPLETS

Solve for triplet excited states in RCIS calculations (ignored for UCIS).

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for triplet states.

FALSE Do not solve for triplet states.

RECOMMENDATION:

None

CIS_STATE_DERIV

Sets CIS state for excited state optimizations and vibrational analysis.

TYPE:

INTEGER

DEFAULT:

0 Does not select any of the excited states.

OPTIONS:

n Select the nth state.

RECOMMENDATION:

Check to see that the states do not change order during an optimization, due to state crossings.

RPA

Do an RPA calculation in addition to a CIS or TDDFT/TDA calculation.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an RPA calculation.

TRUE Do an RPA calculation.

2 Do an RPA calculation without running CIS or TDDFT/TDA first.

RECOMMENDATION:

RPA = 2 is not available for restricted open-shell wavefunctions.

SPIN_FLIP

Selects whether to perform a standard excited state calculation, or a spin-flip calculation. Spin multiplicity should be set to 3 for systems with an even number of electrons, and 4 for systems with an odd number of electrons.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

SPIN_FLIP_XCIS

Controls whether to do a SF-XCIS calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an SF-XCIS calculation.

TRUE Do an SF-XCIS calculation (requires ROHF triplet ground state).

RECOMMENDATION:

None

SFX_AMP_OCC_A

Defines a custom amplitude guess vector in SF-XCIS method.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n builds a guess amplitude with an α-hole in the nth orbital (requires SFX_AMP_VIR_B).

RECOMMENDATION:

Only use when default guess is not satisfactory.

SFX_AMP_VIR_B

Defines a user-specified amplitude guess vector in SF-XCIS method.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n builds a guess amplitude with a β-particle in the nth orbital (requires SFX_AMP_OCC_A).

RECOMMENDATION:

Only use when default guess is not satisfactory.

XCIS

Controls whether to do an XCIS calculation in addition to a CIS calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an XCIS calculation.

TRUE Do an XCIS calculation (requires ROHF ground state).

RECOMMENDATION:

None

SASF_CIS

Controls whether to do an SA-SF-CIS/DFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an SA-SF-CIS/DFT calculation.

TRUE Do an SA-SF-CIS/DFT calculation.

RECOMMENDATION:

The SA-SF method requires a restricted open-shell ground-state calculation.

DFTCIS

Controls whether to do a DFT/CIS calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do a DFT/CIS calculation.

TRUE Do a DFT/CIS calculation.

RECOMMENDATION:

None

DFTCIS_PARAMS

Selects what variant of DFT/CIS

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do CIS

1 Do B3LYP/CIS

2 Do CAM-B3LYP/CIS

RECOMMENDATION:

Requires DFTCIS to be set to 1 or 2.

7.2.4.2 Additional CIS Customization

Additional variables for fine-tuning CIS and related calculations are provided below.

N_FROZEN_CORE

Controls the number of frozen core orbitals.

TYPE:

INTEGER/STRING

DEFAULT:

0 No frozen core orbitals.

OPTIONS:

FC Frozen core approximation.

n Freeze n core orbitals.

RECOMMENDATION:

There is no computational advantage to using frozen core for CIS, and analytical derivatives are only available when no orbitals are frozen. It is helpful when calculating CIS(D) corrections (see Section 7.9).

N_FROZEN_VIRTUAL

Controls the number of frozen virtual orbitals.

TYPE:

INTEGER

DEFAULT:

0 No frozen virtual orbitals.

OPTIONS:

n Freeze n virtual orbitals.

RECOMMENDATION:

There is no computational advantage to using frozen virtuals for CIS, and analytical derivatives are only available when no orbitals are frozen.

MAX_CIS_CYCLES

Maximum number of CIS iterative cycles allowed.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User-defined number of cycles.

RECOMMENDATION:

Default is usually sufficient.

MAX_CIS_SUBSPACE

Maximum number of subspace vectors allowed in the CIS iterations

TYPE:

INTEGER

DEFAULT:

As many as required to converge all roots

OPTIONS:

n User-defined number of subspace vectors

RECOMMENDATION:

The default is usually appropriate, unless a large number of states are requested for a large molecule. The total memory required to store the subspace vectors is bounded above by $2nOV$, where O and V represent the number of occupied and virtual orbitals, respectively. n can be reduced to save memory, at the cost of a larger number of CIS iterations. Convergence may be impaired if n is not much larger than CIS_N_ROOTS.

CIS_CONVERGENCE

CIS is considered converged when error is less than $10^{-\mathrm{CIS\_CONVERGENCE}}$

TYPE:

INTEGER

DEFAULT:

6 CIS convergence threshold 10^{-6}

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

Also controls convergence of the CPSCF equations.

CIS_DYNAMIC_MEM

Controls whether to use static or dynamic memory in CIS and TDDFT calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Partly use static memory

TRUE Fully use dynamic memory

RECOMMENDATION:

The default control requires static memory (MEM_STATIC) sufficient to hold an array whose size grows by $2 \times OV \times N_{\text{roots}}$ at each CIS iteration, where N_{roots} is the number of unconverged roots ($\leq$ CIS_N_ROOTS). For a large calculation, one has to specify a large value for MEM_STATIC, which is not recommended (see Chapter 2). Therefore, it is recommended to use dynamic memory for large calculations.

CIS_RELAXED_DENSITY

Use the relaxed CIS density for attachment/detachment density analysis as well as for for the general excited-state analysis of Section 10.2.12.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use the relaxed CIS density in analysis.

TRUE Use the relaxed CIS density in analysis.

RECOMMENDATION:

None

CIS_GUESS_DISK

Read the CIS guess from disk (previous calculation).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Create a new guess.

TRUE Read the guess from disk.

RECOMMENDATION:

Requires a guess from previous calculation.

CIS_GUESS_DISK_TYPE

Determines the type of guesses to be read from disk

TYPE:

INTEGER

DEFAULT:

Nil

OPTIONS:

0 Read triplets only

1 Read triplets and singlets

2 Read singlets only

RECOMMENDATION:

Must be specified if CIS_GUESS_DISK is TRUE.

STS_MOM

Control calculation of the transition moments between excited states in CIS and TDDFT calculations (including spin-flip variants).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate state-to-state transition moments.

TRUE Do calculate state-to-state transition moments.

RECOMMENDATION:

When set to true requests the state-to-state dipole transition moments for all pairs of excited states and for each excited state with the ground state. This is not available for restricted open-shell wavefunctions.

CIS_MOMENTS

Controls calculation of excited-state (CIS or TDDFT) multipole moments.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate excited-state moments.

TRUE Calculate moments for each excited state.

RECOMMENDATION:

Set to TRUE if excited-state moments are desired. (This is a trivial additional calculation.) The MULTIPOLE_ORDER controls how many multipole moments are printed. This option is not available for spin-flip methods.

7.3 Time-Dependent Density Functional Theory (TDDFT)

7.3.1 Brief Introduction

Excited states may be obtained from density functional theory via linear response,[45,63,106] which for historical reasons is known as "time-dependent" (TD-)DFT.[106] This should not be confused with the explicitly time-dependent methods that are discussed in Section 7.4, however linear-response DFT is nearly universally called TDDFT and we shall use

that nomenclature as well. This approach calculates poles in the response of the ground state density to a time-varying applied electric field. These poles are Bohr frequencies, or in other words the excitation energies. Operationally, this involves solution of an eigenvalue equation

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^\dagger & \mathbf{A}^\dagger \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} = \omega \begin{pmatrix} -\mathbf{1} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \tag{7.15}$$

where the elements of the matrix $\mathbf{A}$ similar to those used at the CIS level, Eq. (7.11), but with an exchange-correlation correction.[106] Elements of $\mathbf{B}$ are similar. Equation (7.15) is solved iteratively for the lowest few excitation energies, ω. Alternatively, one can make a Tamm-Dancoff approximation (TDA) in which the "de-excitation" amplitudes $\mathbf{Y}$ are neglected.[114] In that case, the $\mathbf{B}$ matrix is not required and Eq. (7.15) reduces to a CIS-like equation $\mathbf{Ax} = \omega\mathbf{x}$.

TDDFT is popular because its computational cost is roughly similar to that of the simple CIS method, but a description of differential electron correlation effects is implicit in the method. It is advisable to only employ TDDFT for low-lying valence excited states that are below the first ionization potential of the molecule,[45] or more conservatively, below the first Rydberg state, and in such cases the valence excitation energies are often remarkably improved relative to CIS, with an accuracy of ~ 0.3 eV for many functionals.[106,148] The calculation of the nuclear gradients of full TDDFT and within the TDA is implemented.[154]

Standard TDDFT also does not yield a good description of static correlation effects (see Section 6.14), because it is based on a single reference configuration of Kohn-Sham orbitals. A variant called spin-flip (SF) TDDFT has been developed to address this issue.[108,228] SF-TDDFT is different from standard TDDFT in two ways:

- The reference is a high-spin triplet (quartet) for a system with an even (odd) number of electrons;
- One electron is spin-flipped from an alpha Kohn-Sham orbital to a beta orbital during the excitation.

SF-TDDFT can describe the ground state as well as a few low-lying excited states, and has been applied to bond-breaking processes, and di- and tri-radicals with degenerate or near-degenerate frontier orbitals. A SF-TDDFT method with a non-collinear exchange-correlation potential, originally developed by Ziegler and co-workers,[227,261] has also been implemented.[18] This non-collinear version sometimes improves upon collinear SF-TDDFT for excitation energies but contains a factor of spin density ($\rho_\alpha - \rho_\beta$) in the denominator that sometimes causes stability problems. Best results are obtained using functionals with $\approx 50\%$ Hartree-Fock exchange,[18,108,228] behavior that was later explained on theoretical grounds.[117] Becke's half-and-half functional BH&HLYP has become something of a standard approach when using SF-TDDFT.[108] A spin-adapted version of SF-TDDFT has also been developed.[274]

7.3.2 Charge-Transfer Metrics

Standard density functionals do not yield a potential with the correct long-range Coulomb tail, owing to the self-interaction problem. Therefore, excitation energies corresponding to states that sample this tail (*e.g.*, diffuse Rydberg states and some charge transfer excited states) are not given accurately.[46,106,146,247] The extent to which a particular excited state is characterized by charge transfer can be assessed using an a spatial overlap metric, Λ, defined as[107,109,201]

$$\Lambda = \frac{\sum_{ia}(x_{ia} + y_{ia})^2 O_{ia}}{\sum_{jb}(x_{jb} + y_{jb})^2} \tag{7.16}$$

where O_{ia} is the spatial overlap of occupied MO ψ_i with virtual MO ψ_a:

$$O_{ia} = \int |\psi_i(\mathbf{r})| \cdot |\psi_a(\mathbf{r})| \, d\mathbf{r} \; . \tag{7.17}$$

The absolute value signs are necessary since the occupied and virtual MOs are orthogonal, $\langle\psi_i|\psi_a\rangle$, although Q-CHEM includes the option to use squares of the MOs instead:

$$\tilde{O}_{ia} = \int |\psi_i(\mathbf{r})|^2 |\psi_a(\mathbf{r})|^2 \, d\mathbf{r} \; . \tag{7.18}$$

In that case, $\tilde{O}_{ia}$ is used in place of O_{ia} in Eq. (7.16). For the original version of the metric (using O_{ia}, and where ψ_i and ψ_a are canonical MOs), Tozer and coworkers find that $0.45 \leq \Lambda \leq 0.89$ for localized valence excitations whereas Rydberg excitations lie in the range $0.08 \leq \Lambda \leq 0.27$.[201] Furthermore, they suggest functional-specific cutoffs for when a particular excitation may have too much charge-transfer character for TDDFT results to be trusted. Note that while Q-CHEM implements the definition in Eq. (7.16) that was proposed in Ref. 201, the normalizing denominator in this expression is inconsistent for full TDDFT calculations (*i.e.*, those not invoking the TDA),[109] for which the normalization condition on $\mathbf{x}$ and $\mathbf{y}$ is $\sum_{ia}(x_{ia}^2 - y_{ia}^2) = 1$ rather than $\sum_{ia}(x_{ia}^2 + y_{ia}^2) = 1$.[106]

Empirically, the Λ metric provides very good correlations with TDDFT errors,[201] yet its numerical value is difficult to interpret in terms of a physical charge separation. For that reason, a similar metric

$$\Delta r = \frac{\sum_{ia}(x_{ia} + y_{ia})^2 \|\mathbf{R}_{ia}\|}{\sum_{jb}(x_{jb} + y_{jb})^2} \tag{7.19}$$

was proposed,[88,89,222] where

$$\mathbf{R}_{ia} = \langle\psi_i|\hat{\mathbf{r}}|\psi_i\rangle - \langle\psi_a|\hat{\mathbf{r}}|\psi_i\rangle \tag{7.20}$$

is the vector displacement between the centroids of orbitals ψ_i and ψ_a, and $\|\mathbf{R}_{ia}\|$ in Eq. (7.21) is the distance between their centroids. This makes Δr an incoherent average of electron displacements ($\psi_i \rightarrow \psi_a$), which has consequences for states where a coherent superposition of displacements is qualitatively important.[109] Because Δr vanishes for any centrosymmetric molecule, it was later augmented by a quantity

$$\Delta\sigma = \frac{\sum_{ia}(x_{ia} + y_{ia})^2 \sigma_{ia}}{\sum_{jb}(x_{jb} + y_{jb})^2} \tag{7.21}$$

where

$$\sigma_{ia} = \left|\langle\psi_i|\hat{r}^2|\psi_i\rangle - \langle\psi_a|\hat{r}^2|\psi_a\rangle\right|^{1/2}, \tag{7.22}$$

where $\langle\psi_r|\hat{r}^2|\psi_r\rangle$ is the second moment of orbital ψ_r. The quantity σ_{ia} provides a measure of the change in orbital size between donor (ψ_i) and acceptor (ψ_a) MOs, and a metric

$$\Gamma = \Delta r + \Delta\sigma \tag{7.23}$$

was proposed to measure charge-transfer character for TDDFT excitations.[89] It was suggested that a "trust radius" based on Γ, which has dimensions of length, could be used to replace Tozer's critical values of Λ as a detector of problematic charge-transfer transitions.[89]

A major shortcoming, however, is that none of these metrics (Λ, Δr, $\Delta\sigma$, or Γ) is invariant to unity transformation of the MOs.[109] As such, their numerical values can depend quite strongly on which orbitals $\{\psi_i\}$ and $\{\psi_a\}$ are used. In their original formulations, it was assumed that these would be canonical MOs, although it was subsequently noticed that values of Γ for Rydberg transitions were somewhat erratic but more stable (especially with regard to changes in basis set) when natural transition orbitals (NTOs, Section 7.14.3) were used instead.[89] An explanation was provided later.[109] When the transition in question is dominated by a single occupied/virtual pair of NTOs (as is often the case, although counterexamples are also easy to find),[107,109] then Δr and Γ approximate expectation values with respect to an excitonic wave function, and any proper expectation value must be invariant to unitary transformations of the occupied MOs and, separately, the virtual MOs.[109] For this reason, if these metrics are going to be used then it is recommended that they be evaluated in the NTO basis rather than the canonical MO basis; Q-CHEM can provide both values, for each of Λ, Δr, $\Delta\sigma$, and Γ.

Although the NTO representation affords better stability with respect to changes in basis set (especially, with regard to addition of diffuse functions),[109] it remains the case that each of these metrics is formulated as an incoherent, amplitude-weighted average of $\psi_i \rightarrow \psi_a$ excitations. In a proper expectation value, the excitation amplitudes have a coherent phase relationship, as in the CIS wave function in Eq. (7.10). It is possible to find examples where the Γ metric provides a very misleading measure of charge separation, because several individual $\psi_i \rightarrow \psi_a$ excitations

are associated with large values of $\|\mathbf{R}_{ia}\|$, yet they interfere destructively such that the net result hardly displaces any charge at all.[109]

Thus, while the metrics Λ, Δr, and Γ (in both the canonical and NTO representations) are included in order to help users make contact with existing literature, it is recommended that charge-transfer analysis be based on proper expectation values with respect to transition density matrices. These can be computed using the LIBWFA module that is described in Section 10.2.12. As documented in Table 10.1 of that section, LIBWFA can compute expectation values such as

$$d_{\text{e-h}} = \|\langle \mathbf{r}_{\text{elec}} - \mathbf{r}_{\text{hole}} \rangle\| \ , \tag{7.24}$$

or in other words the expectation value of the distance between the centroid of the excited electron and that of the hole. This constitutes a proper measure of electron–hole separation.[109] Exciton size can be measured via

$$d_{\text{exc}} = \langle \|\mathbf{r}_{\text{elec}} - \mathbf{r}_{\text{hole}}\|^2 \rangle^{1/2} \ , \tag{7.25}$$

which is also available from LIBWFA. Other metrics that were recommended in Ref. 109 can also be computed using LIBWFA, including

$$d_{\text{CD1}} = d_{\text{e-h}} + |\sigma_{\text{hole}} - \sigma_{\text{elec}}| \ , \tag{7.26}$$

where σ_{elec} and σ_{hole} represent the root-mean-square size of the excited electron and the hole, respectively.

7.3.3 TDDFT within a Reduced Single-Excitation Space

Much of chemistry and biology occurs in solution or on surfaces. The molecular environment can have a large effect on electronic structure and may change chemical behavior. Q-CHEM is able to compute excited states within a local region of a system through performing the TDDFT (or CIS) calculation with a reduced single excitation subspace,[19] in which some of the amplitudes $\mathbf{x}$ in Eq. (7.15) are excluded. (This is implemented within the TDA, so $\mathbf{y} \equiv \mathbf{0}$.) This allows the excited states of a solute molecule to be studied with a large number of solvent molecules reducing the rapid rise in computational cost. The success of this approach relies on there being only weak mixing between the electronic excitations of interest and those omitted from the single excitation space. For systems in which there are strong hydrogen bonds between solute and solvent, it is advisable to include excitations associated with the neighboring solvent molecule(s) within the reduced excitation space.

The reduced single excitation space is constructed from excitations between a subset of occupied and virtual orbitals. These can be selected from an analysis based on Mulliken populations and molecular orbital coefficients. For this approach the atoms that constitute the solvent needs to be defined. Alternatively, the orbitals can be defined directly. Truncated excitation space within TDDFT/TDA is deployed by activating the TRNSS and TRTYPE keywords. The atoms or orbitals are specified within a *$solute* block. These approach is implemented within the TDA and has been used to study the excited states of formamide in solution,[23] CO on the Pt(111) surface,[20] and the tryptophan chromophore within proteins.[218]

Restricting excitation space by using TRNSS and *$solute* can be used to deploy core-valence separation[47] within TDDFT in calculations of core-excited states, see Section 7.13.2.

7.3.4 Electron-Affinity (EA-) TDDFT

X-ray absorption spectra (XAS) pose a significant challenge for TDDFT. The errors in peak positions for the K-edge XAS of small gas-phase molecules are routinely in excess of 10 eV, and in condensed phase the errors in peak positions and intensities cause severe distortions in K-edge spectra. These errors can be traced to two sources:

1. A lack of orbital relaxation effects that is poorly compensated by linear-response theory with only single excitations atop ground state reference orbitals, and

2. a form of excited-state self-interaction error in the TDDFT potential called particle–hole self-interaction error.

Both can be rectified by choosing a more suitable set of reference orbitals for the core-excitation problem (*e.g.*, a core-ionized reference determinant) as done with the static-exchange approximation in Section 7.7.2. The generalization of STEX to include density functional theory correlation is accomplished through two separate linear responses. First, from an initial set of core-ionized molecular orbitals, the missing core electron is reattached (response #1), then a second linear response is done to excite the core electron. The second linear response occurs atop a non-stationary initial density, which is problematic due to the adiabatic approximation. Fortunately, an exact first-order correction to errors associated with the adiabatic approximation can be derived and is applied automatically in all EA-TDDFT calculations. This correction happens to take a form reminiscent of excited-state self-interaction error, so the EA-TDDFT equations are naturally excited-state self-interaction free. Because of this, EA-TDDFT can achieve statistical errors of just 0.5 eV root-mean-squared deviation with standard density functionals.[34] Inclusion of orbital relaxation effects at the level of the reference orbitals in EA-TDDFT also offers critical improvements to the spectra of solvated molecules relative to TDDFT.[32] EA-TDDFT is only defined for restricted open-shell orbitals and produces spin-pure spectra.

The full EA-TDDFT equations take a familiar form,

$$A_{ia,ib} = E^{+}\delta_{ab} + F^{+}_{ab} + (ia|ib) + (1 - C_{\mathrm{HF}})(ia|f^{+}_{\mathrm{xc}}|ib) \tag{7.27a}$$

$$B_{ia,ib} = (ia|ib) + (1 - C_{\mathrm{HF}})(ia|f^{+}_{\mathrm{xc}}|ib) \tag{7.27b}$$

where E^{+} is the self-consistently optimized energy of the core-ionized determinant, F^{+}_{ab} are elements of the virtual-virtual block of the core-ion Fock matrix, and all integrals are computed using the core-ion orbitals. While the full EA-TDDFT equations can be solved at low cost (by requesting **EA_RPA** in the *$nocis* input section), it is seldom necessary to solve the whole RPA-like expression for K-edge XAS. Instead, for K-edge transitions the Tamm-Dancoff approximation can be used by setting **EA_TDA** in *$nocis* without sacrificing any accuracy.[32] Note only one occupied orbital index is considered in the EA-TDDFT equations, meaning that the core-valence separation approximation is invoked implicitly. Additionally, because EA-TDDFT is based on linear-response theory, there is no construction of non-orthogonal matrix elements between the ground state and the singly-excited determinants as is done in STEX or 1C-NOCIS. This means that EA-CIS (EA-TDDFT with the Hartree-Fock functional) will produce slightly different results than STEX, but the differences generally do not exceed 0.1 eV.[34]

The presence of E^{+} in the above equation means that core-ionized reference orbitals must be optimized for each core orbital from which excitations will be generated. In the NOCIS and STEX implementations, this has been done exclusively with the Maximum Overlap Method (MOM) from section 7.6. However, Q-CHEM now has several excellent algorithms for converging non-*Aufbau* solutions to the SCF problem such as State-Targeted Energy Projection (STEP)[33] and Square Gradient Minimization (SGM);[91] see Sections 7.8.5 and 7.8.4, respectively. All of these approaches are now available in the EA-TDDFT code by setting either **DSCF_ALGORITHM** or **REF_SCF_ALGORITHM** within the *$nocis* input section. A hybrid algorithm consisting of running STEP in the initial cycles of an optimization to guide the solution towards a set of well-conditioned guess orbitals for a subsequent MOM optimization has been used with great success when other algorithms fail to converge. This can be invoked by setting **DSCF_ALGORITHM = STEP_MOM** and setting the point at which STEP turns off and MOM turns on with the keyword **STEP_MOM_START**.

There are two points at which localization can be used to improve the results or make orbital selection more convenient within EA-TDDFT. First, when two or more reference orbitals are selected a Boys localization is automatically performed on the core orbitals of interest to improve convergence of the core-ion SCF to a proper minimum. This localization routine may be controlled manually (*e.g.*, localizing several orbitals but only computing excitations out of one of them, or turning off localization altogether) with the keyword **LOCALIZE_ORBITALS**. In large systems with many core orbitals of similar energy (*e.g.*, water clusters where the response out of a particular O(1s) orbital is desired), Subsystem Projected AO Decomposition (SPADE) orbitals[51] can be used to localize orbitals onto a particular subset of atoms before **ORB_OFFSET** is considered. For instance, in the water cluster example given above, if the first water

molecule in the *$molecule* section is of interest then one would set "**SUBSYSTEM_ATOMS** 1 2 3" within the *$nocis* section and the variable **ORB_OFFSET** would be set to 0 for the O(1s) orbital on this subset of atoms.

7.3.5 Job Control for TDDFT

7.3.5.1 Basic Settings

Input for time-dependent density functional theory calculations follows very closely the input already described for the uncorrelated excited state methods described in the previous section (in particular, see Section 7.2.4). There are several points to be aware of:

- The exchange and correlation functionals are specified exactly as for a ground state DFT calculation, through EXCHANGE and CORRELATION. To active TDDFT, set CIS_N_ROOTS to a value ≥ 1, specifying the number of excited states to compute.
- If RPA is set to TRUE, a "full" TDDFT calculation will be performed, however the default value is RPA = FALSE, which invokes the TDA,[114] in which the de-excitation amplitudes $\mathbf{y}$ in Eq. (7.15) are neglected, which is usually a good approximation for excitation energies, although oscillator strengths within the TDA no longer formally satisfy the Thomas-Reiche-Kuhn sum rule.[45] For RPA = TRUE, a TDA calculation is performed first and used as the initial guess for the full TDDFT calculation. The TDA calculation can be skipped altogether using RPA = 2. RPA is not implemented for restricted open-shell calculations, only TDA.
- If SPIN_FLIP is set to TRUE when performing a TDDFT calculation, a SF-TDDFT calculation will also be performed. At present, SF-TDDFT is only implemented within the TDA so RPA must be set to FALSE. Remember to set the spin multiplicity to 3 for systems with an even-number of electrons (*e.g.*, diradicals), and 4 for odd-number electron systems (*e.g.*, triradicals).
- If MGGA_GINV is set to 1 when performing a TDDFT calculation, gauge invariance correction will be added to meta-GGA functionals.[12]

Some basic job control variables (to be added to the *$rem* section) are described below. There are additional options for X-ray spectroscopy; see Section 7.13.2.

CIS_SINGLETS
Controls whether to compute singlet excited states.
TYPE:
LOGICAL
DEFAULT:
TRUE
OPTIONS:
FALSE Do not compute singlet excitations.
TRUE Compute singlet excitations.
RECOMMENDATION:
This option makes sense only for a singlet ground state, since the use of an open-shell ground state does not afford spin-pure excited states.

CIS_TRIPLETS

Controls whether to compute triplet excited states.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not compute triplet excitations.

TRUE Compute triplet excitations.

RECOMMENDATION:

This option makes sense only for a singlet ground state, since the use of an open-shell ground state does not afford spin-pure excited states.

MGGA_GINV

Controls whether to add gauge invariance correction to meta-GGA functionals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No correction.

1 Add gauge invariance correction to meta-GGA functionals.

RECOMMENDATION:

Not recommended when TDA is used because the TDA violates gauge invariance.

WANG_ZIEGLER_KERNEL

Controls whether to use the Wang-Ziegler non-collinear exchange-correlation kernel in a SF-TDDFT calculation. Set NEW_DFT = TRUE if using a Q-CHEM version older than 5.0.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use non-collinear kernel.

TRUE Use non-collinear kernel.

RECOMMENDATION:

None

CIS_GUESS_TYPE

Controls how to generate the initial guess excitation vectors in CIS/TDA/RPA calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Generate N (no. of roots requested) occupied $\rightarrow$ virtual single orbital transitions according to their orbital energy difference order (from low to high). This is the common scenario.

1 Generate $N - 1$ occupied $\rightarrow$ virtual single orbital transitions according to their orbital energy difference order (from low to high), and generate another guess excitation vector consist of all the remaining single orbital transitions in the occupied $\rightarrow$ virtual transition space with equal weights.

2 Generate N occupied/virtual single orbital transitions according to their orbital energy difference order (from low to high), and generate one more guess excitation vector consist of all the remaining single orbital transitions in the occupied $\rightarrow$ virtual transition space with equal weights.

RECOMMENDATION:

The default setting should work for most of the cases. However, when the number of roots is small, in some CIS/TDA/RPA calculations, low energy excited states could be missing. The options CIS_GUESS_TYPE = 1 or 2 may remedy this root missing issue by sampling more vectors in the transition space. Setting CIS_GUESS_TYPE = 1 or 2 may take more cycles to converge in the Davidson iteration, but the results are expected to be more reliable. Currently, CIS_GUESS_TYPE = 1 or 2 are not supported in SF-XCIS calculations. Setting TRNSS = TRUE also disables the setting of CIS_GUESS_TYPE.

7.3.5.2 Options for Charge-Separation Metrics

The charge-separation metrics Λ, Δr, $\Delta\sigma$, and Γ can be computed (for each TDDFT excited state), and in both the canonical and NTO representations, by setting CIS_CT_METRICS = TRUE. Some additional options are described below. For LIBWFA job control (*e.g.*, to compute the recommended metrics $d_{\text{e-h}}$, d_{exc}, and d_{CD1}), see Section 10.2.12.

CIS_CT_METRICS

Controls whether to compute the charge-separation metrics Λ, Δr, $\Delta\sigma$, and Γ (Section 7.3.2).

TYPE:

INTEGER

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute these metrics.

TRUE Compute all four metrics (for each excited state), in both the canonical and NTO representations.

RECOMMENDATION:

Request if desired. (There is some overhead associated with computing Λ, but it should be small.)

SPATIAL_OVERLAP_METRIC

Controls whether O_{ia} or $\tilde{O}_{ia}$ is used to evaluate Λ in Eq. (7.16).

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Compute Λ using O_{ia}.

2 Compute Λ using $\tilde{O}_{ia}$.

RECOMMENDATION:

Option 1 represents the original metric suggested by Tozer and co-workers in Ref. 201.

SPATIAL_OVERLAP_GRID

Controls the grid that is used to evaluate O_{ia} or $\tilde{O}_{ia}$ in Eq. (7.17).

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use a EML grid with $N_r = 300$ and $N_\Omega = 302$.

2 Use a EML grid with $N_r = 400$ and $N_\Omega = 434$.

RECOMMENDATION:

None.

SPATIAL_OVERLAP_PRINT

Controls whether to print the spatial overlaps O_{ia} or $\tilde{O}_{ia}$.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not print the O_{ia}.

1 Print the frontier overlaps only (5 occupied and 5 virtual orbitals.

2 Print all of the O_{ia}.

RECOMMENDATION:

These may be useful for *a posteriori* analysis of the spatial proximity of various MOs; however, option 2 will engender significant printing for large molecules. Whether it is O_{ia} [Eq. (7.17)] or $\tilde{O}_{ia}$ [Eq. (7.18)] that is printed depends upon the setting of SPATIAL_OVERLAP_ANAL.

7.3.5.3 Options for a Truncated Subspace

The following options can be used to implement the reduced excitation space that was described in Section 7.3.3.

CISTR_PRINT

Controls level of output.

TYPE:

LOGICAL

DEFAULT:

FALSE Minimal output.

OPTIONS:

TRUE Increase output level.

RECOMMENDATION:

None

CUTOCC

Specifies occupied orbital cutoff.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

0-200 Use a cutoff of CUTOCC/100

RECOMMENDATION:

None

CUTVIR

Specifies virtual orbital cutoff.

TYPE:

INTEGER

DEFAULT:

0 No truncation

OPTIONS:

0-100 CUTOFF = CUTVIR/100

RECOMMENDATION:

None

N_SOL

Specifies number of atoms or orbitals in the *$solute* or *$alist* section.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

Reads from either the *$solute* or *$alist* input section.

TRNSS

Controls whether reduced single excitation space is used.

TYPE:

LOGICAL

DEFAULT:

FALSE Use full excitation space.

OPTIONS:

TRUE Use reduced excitation space.

RECOMMENDATION:

None

TRTYPE

Controls how reduced subspace is specified.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Select orbitals localized on a set of atoms.

2 Specify a set of orbitals.

3 Specify a set of occupied orbitals, include excitations to all virtual orbitals.

RECOMMENDATION:

None

7.3.5.4 Job Control for EA-TDDFT

Like NOCIS/STEX/1C-NOCIS, the options for the EA-TDDFT/EA-CIS method are controlled via the *$nocis* section after NOCIS is set to TRUE in *$rem*.

NOCIS

Requests a NOCIS/STEX/1C-NOCIS/EA-TDDFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not run these methods.

TRUE Run one of these methods, options controlled in *$nocis*.

RECOMMENDATION:

None

The options below are set within the *$nocis* section.

NUM_REF

Sets the number of reference orbitals in a NOCIS/STEX/1C-NOCIS/EA-TDDFT calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Positive integer

RECOMMENDATION:

Set according to the number of consecutive orbitals of interest for the calculation. For example, for the oxygen K-edge in CO_2, the number of references would be 2 (two oxygen 1s orbitals), whereas for the carbon K-edge it would be 1 (one carbon 1s).

ORB_OFFSET

Determines the starting orbital for a NOCIS/STEX/1C-NOCIS/EA-TDDFT calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Positive integer

RECOMMENDATION:

Set according to the first orbital of interest in the system in question. For example, this would be set to 0 for the oxygen K-edge in CO_2 because the two O(1s) orbitals lie below the C(1s), so for the carbon K-edge this would be set to 2.

LOCALIZE_ORBITALS

Choose which core orbitals to localize with the Boys objective function.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

-1 for NUM_REF = 1 and equal to **NUM_REF** otherwise.

OPTIONS:

n Integer, normally positive.

RECOMMENDATION:

Set to -1 to skip Boys localization entirely, or set to a value greater than **NUM_REF** to include more orbitals in the preliminary localization, otherwise use the default.

SUBSYSTEM_ATOMS

Choose which atoms to consider for an EA-TDDFT calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

List of integers delimited by spaces: $i\ j\ k\ldots$

RECOMMENDATION:

Use only if excitations out of a particular subset of atoms is of interest and the definition of **ORB_OFFSET** and **NUM_REF** is nontrivial. This is an expert option, and should only be used in specific situations such as isolating the oxygen K-edge of all water molecules at the air/water interface.

DSCF_ALGORITHM

Sets the ΔSCF algorithm to be used in the optimization of the core-ionized reference determinants.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

MOM

OPTIONS:

MOM, IMOM, STEP, or STEP_MOM

RECOMMENDATION:

Use MOM unless convergence issues arise. IMOM occasionally provides improved convergence, but the combination of STEP and MOM used by the **STEP_MOM** option is particularly robust. In rare cases, STEP may be necessary, but this is not recommended as the number of SCF cycles required is quite large. If **STEP_MOM** is requested, the option **STEP_MOM_START** must also be set.

STEP_MOM_START

Sets the SCF cycle on which the STEP algorithm stops and the MOM algorithm starts.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n Positive integer.

RECOMMENDATION:

For best results, set such that STEP reaches a convergence threshold of at least 10^{-4} a.u. before switching to MOM, but if MOM continues to collapse or oscillate, then wait until tighter convergence is achieved to switch algorithms.

REF_SCF_ALGORITHM

Sets the SCF algorithm for the core-ion reference calculations.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

DIIS

OPTIONS:

DIIS, GDM, GDM_LS, SGM, or SGM_LS

RECOMMENDATION:

Use DIIS for MOM, IMOM, STEP, or STEP_MOM calculations, but if these algorithms are not providing satisfactory convergence to the desired core-ion state, this can be set to SGM or SGM_LS.

REF_SCF_CONVERGENCE

Sets SCF convergence threshold for the core-ion reference calculations.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

Same as SCF_CONVERGENCE in *$rem*

OPTIONS:

n Positive integer

RECOMMENDATION:

None

REF_SCF_GUESS

Initial guess for core-ion reference calculations.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

Koopman

OPTIONS:

Read (the Koopmans guess is automatically used if no option is specified)

RECOMMENDATION:

The default core-ion guess is generated by removing an electron from the core orbital using the ground-state MO coefficients (Koopmans guess). Alternatively, when convergence issues arise it may be useful to run an EA-TDDFT job with a different functional (*i.e.*, LDA or GGA), then read the converged core-ion orbitals into a subsequent EA-TDDFT job with the functional of interest (*i.e.*, a range-separated hybrid).

EA_TDA

Invokes EA-TDDFT within the Tamm-Dancoff approximation.

INPUT SECTION: *$nocis*

TYPE:

NONE

DEFAULT:

NONE

OPTIONS:

The presence of this keyword will activate **EA_TDA**.

RECOMMENDATION:

This is the most cost-effective form of the EA-TDDFT equations and has almost no effect on results for K-edge XAS.

EA_RPA

Solves the full EA-TDDFT equations.

INPUT SECTION: *$nocis*

TYPE:

NONE

DEFAULT:

NONE

OPTIONS:

The presence of this keyword will activate **EA_RPA**.

RECOMMENDATION:

No recommendation.

SINGLETS

Compute only singlets.

INPUT SECTION: *$nocis*

TYPE:

NONE

DEFAULT:

NONE

OPTIONS:

The presence of this keyword without the **Triplets** keyword will compute singlets without computing triplets.

RECOMMENDATION:

No recommendation.

TRIPLETS

Compute only triplets.

INPUT SECTION: *$nocis*

TYPE:

NONE

DEFAULT:

NONE

OPTIONS:

The presence of this keyword without the **Singlets** keyword will compute triplets without computing singlets.

RECOMMENDATION:

No recommendation.

N_ROOTS

Sets the number of roots to print.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

All

OPTIONS:

n where $n < m_{\mathrm{roots}}$

RECOMMENDATION:

No recommendation. Beware that unlike CIS/TDDFT jobs this keyword only prints fewer roots, but the entirety of the Hamiltonian is still diagonalized via direct diagonalization.

PRINT_NTOS

Prints natural transition orbitals to cube files for each root that is printed via the **N_ROOTS** keyword.

INPUT SECTION: *$nocis*

TYPE:

NONE

DEFAULT:

NONE

OPTIONS:

The presence of this keyword automatically generates all NTOs (default), or **N_ROOTS** NTOs.

RECOMMENDATION:

No recommendation, but one should be sure to also include a *$plots* section with appropriate details such that the cube files can be generated.

Example 7.7 EA-TDDFT with Tamm-Dancoff approximation for the K-edge of NH_3. This job truncates the number of printed (singlet) roots to 10. The core-ion SCF will use the SGM algorithm.

```
$rem
method          rCAM-B3LYP
rel_x2c         true
basis           aug-cc-pCVDZ
nocis           true
scf_convergence 8
thresh          14
integral_symmetry    false
point_group_symmetry false
$end

$nocis
n_roots         10
orb_offset      0
num_ref         1
singlets
ea_tda
ref_scf_algorithm sgm
$end

$molecule
0 1
N   0.0000   0.0000   0.1163
H   0.0000   0.9399  -0.2713
H   0.8140  -0.4700  -0.2713
H  -0.8140  -0.4700  -0.2713
$end
```

Example 7.8 Full EA-TDDFT for the K-edge of NH_3. The core-ion SCF will use the default MOM algorithm.

```
$rem
method          rCAM-B3LYP
rel_x2c         true
scf_convergence 8
basis           aug-cc-pCVDZ
nocis           true
thresh          14
integral_symmetry    false
point_group_symmetry false
$end

$nocis
n_roots         10
orb_offset      0
num_ref         1
singlets
ea_rpa
$end

$molecule
0 1
N   0.0000   0.0000   0.1163
H   0.0000   0.9399  -0.2713
H   0.8140  -0.4700  -0.2713
H  -0.8140  -0.4700  -0.2713
$end
```

7.3.6 TDDFT + PCM for Excitation and Emission Energies in Solution

As described in Section 11.2.3, polarizable continuum models (PCMs) are a simple means of including solvation effects in quantum chemistry calculations at the level of the dielectric continuum. In this section, the available options in Q-CHEM for PCM solvation of TDDFT excited states are discussed. For further details on the associated keywords, see Sections 11.2.3.3 and 11.2.4. For background information on PCM solvation models, see Ref. 105.

TDDFT can be combined with PCM in two different formalisms: *linear response* (LR)- and *state-specific* (SS)-PCM. The LR-PCM approach[28,105] employs the transition density between the ground and excited state and describes the coupling between the excitation process on the solute and the solvent environment. (For more details, see Section 11.2.3.3.) LR-PCM entails only a minor additional computational cost. However, the use of the transition density limits its applicability to optically-allowed transitions, whereas the LR-PCM contribution vanishes for charge-transfer states or singlet to triplet excitations. In Q-CHEM, TDDFT/LR-PCM is available for excitation energies, as well as excited-state energy gradients and Hessians,[157] to allow for geometry optimization and vibrational frequency calculations as described in Section 7.3.7.

The state-specific PCM formalism[29,30,105] is difference-density-based and takes into account the instantaneous response of the solvent to the change in the solute wave function. This contribution is not included with LR-PCM and is nonzero even for vanishing transition densities in optically-forbidden transitions. Q-CHEM includes SS-PCM both as a perturbative correction[182,271] (ptSS-PCM) as well as in two fully iterative implementations. Generally, the SS-PCM implementation for TDDFT[77] follows closely the ADC(n)-family of methods described in Section 7.11.10. The ptSS-PCM approach,[182,271] which is alternatively called corrected linear-response (cLR-PCM),[30] evaluates a state-specific correction based on the TDDFT excited state density for each calculated state. This first-order correction excludes the response of the excited state density to the state-specific reaction field, as to include this response requires an iterative procedure. The ptSS-PCM approach can be combined with a perturbative approximation to LR-PCM, ptLR-PCM. For more details, see Section 11.2.3.3.

To go beyond a first-order correction, one of the iterative implementations of SS-PCM has to be used. Here, a state-specific Schrödinger equation is solved self-consistently via iterative optimization of the reaction field for a specific state. In Q-CHEM, iterative SS-PCM is implemented[77] either via *external iteration* (EI-SS-PCM, sometimes called the Improta-Barone-Scalmani-Frisch approach),[120] or else via *internal iteration* (II-SS-PCM, otherwise known as the *vertical excitation model* or VEM).[168] Whereas ptSS-PCM adds only negligible computational cost, the iterative procedures increase it by a factor of the required number of iterations.

Excited-state solvation with TDDFT is available in two regimes: *equilibrium* and *nonequilibrium* solvation. Nonequilibrium solvation applies to fast processes (*e.g.*, during vertical excitation or emission), in which the slow nuclear degrees of freedom of the solvent remain equilibrated with respect to the initial state. Hence, the solute has to be modeled for an out-of-equilibrium state of the (implicit) solvent, with only the fast electronic polarization of the solvent relaxed using the final state's charge density. This fast solvent response is governed by the infinite-frequency or "optical" dielectric constant (ε_∞), often taken to be the square of the solvent's index of refraction: $\varepsilon_\infty = n^2_{\text{refr}}$.[105] Equilibrium solvation, on the other hand, is appropriate for excited states with a sufficiently long lifetime to allow for complete relaxation of the solvent. Here, the zero-frequency or static dielectric constant (ε_0) is the relevant one, as it includes all polarization mechanisms including both the fast electronic degrees of freedom and the slow vibrational and orientational degrees of freedom.

7.3.6.1 Linear-Response PCM

LR-PCM for TDDFT is activated by setting TDDFT_LR_PCM = TRUE in the *$pcm* section. The default is a nonequilibrium calculation employing the optical dielectric constant ε_∞ for the excited state, whereas the static dielectric constant is employed for the ground-state solvation of the used MOs. This corresponds to the "full linear response theory" of

Ref. 157. The value of ε_∞ is specified using the keyword **OpticalDielectric** in the *$solvent* input section, or can be chosen from a list of preset values as described in Section 11.2.4.3; an equilibrium calculation in the excited state could be requested by setting **OpticalDielectric** equal to the value of the solvent's static dielectric constant.

A perturbative approximation to full LR-PCM is also available, along with state-specific corrections that are somewhat more theoretically rigorous,[105] and have also been implemented for TDDFT.[182,271] This approach is described in further detail below in Sections 7.3.6.2 and 11.2.3.3. If the C-PCM model is used, then analytical excited-state energy gradients are available for TDDFT + LR-PCM and are automatically used if LR-PCM is activated for TDDFT geometry optimization. As excited-state geometry optimization presupposes equilibration of the solute structure in the excited state, it is probably appropriate to employ the static dielectric constant for geometry optimization.

The LR-PCM approach considers the solvent response to the electronic transition density and provides only for optically-allowed transitions a non-vanishing contribution. Optically-forbidden transitions should be described with a state-specific PCM. This includes triplet excited states accessed from a singlet ground state and also charge-transfer states with a vanishing transition density

Example 7.9 TDDFT/LR-C-PCM low-lying vertical excitation energy

```
$molecule
   0 1
   C     0.0    0.0    0.0
   O     0.0    0.0    1.21
$end

$rem
   EXCHANGE          B3lyp
   CIS_N_ROOTS       10
   CIS_SINGLETS      true
   CIS_TRIPLETS      true
   RPA               TRUE
   BASIS             6-31+G*
   XC_GRID           1
   SOLVENT_METHOD    pcm
$end

$pcm
   Theory   CPCM
   Method   SWIG
   Solver   Inversion
   Radii    Bondi
$end

$solvent
   Dielectric         78.39
   OpticalDielectric  1.777849
$end
```

TDDFT_LR_PCM

Controls LR-PCM for TDDFT, *i.e.*, whether or not to add the LR-PCM contributions to the TDDFT eigenvalue problem.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not do LR-PCM (0th-order solvent correction only).

TRUE Perform full LR-PCM.

RECOMMENDATION:

Assuming that PCM solvation is turned on for the ground state (SOLVENT_METHOD = PCM in the *$rem* section), then disabling LR-PCM by setting TDDFT_LR_PCM = FALSE will afford a "0th-order" solvation correction, in which solvent-polarized MOs and energy levels are used in what is otherwise equivalent to a gas-phase TDDFT calculation. This is the first step in more sophisticated "nonequilibrium" TDDFT + PCM methods, which are discussed in Section 11.2.3.3. The LR-PCM correction to the excitation energies has some peculiar properties, such as the fact that it vanishes for optically-forbidden states,[105] and the state-specific approaches that are discussed in Section 11.2.3.3 are likely preferable.

7.3.6.2 State-Specific PCM

The state-specific PCM jobs are controlled with the StateSpecific keyword in the *$pcm* block, which can be employed to activate either a perturbative or one of the iterative SS-PCMs.

StateSpecific

Specifies which the state-specific PCM will be used.

INPUT SECTION: *$pcm*

TYPE:

Various

DEFAULT:

NONE

OPTIONS:

Perturb Perform ptSS and ptLR for vertical excitations.

External Perform self-consistent EI-SS-PCM to the excited state (for emission).

Internal Perform self-consistent II-SS-PCM to the excited state (for emission or excitation).

RECOMMENDATION:

Use for vertical excitation energies ptSS-PCM or in very polar cases the nonequilibrium II-SS-PCM, and equilibrium EI-SS-PCM for emission energies of long-lived excited states.

1. Perturbative State-Specific (ptSS) PCM

Perturbative state-specific PCM (ptSS-PCM)[182,271] is activated by setting **StateSpecific = Perturb** in the *$pcm* block. By default, the ptSS correction is evaluated in the nonequilibrium regime governed by the optical dielectric constant (**OpticalDielectric** in the *$solvent* block). To force an *equilibrium* ptSS-PCM calculation, **OpticalDielectric** must be set to the static value, ε_0. To separate the fast and slow contributions to the reaction field, the charge-separation procedure is chosen with the **ChargeSeparation** keyword. (**Marcus** is the default, for more information, see Section 11.2.3.3 and Ref. 271.) Alternatively, setting **NonEquilibrium = TRUE** results in a default nonequilibrium ptSS-PCM calculation with **Marcus** charge separation and deactivated LR-PCM. The ptSS-PCM output provides for each calculated

state both a ptSS and a ptLR-correction. In the nonequilibrium case, the corrected energy corresponds to the vertical excitation energy. Results are provided both with the unrelaxed density and the relaxed density, if the latter has been recommended. Results based on the relaxed density are recommended.[271]

NonEquilibrium

Activate nonequilibrium ptSS-PCM.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

False

OPTIONS:

TRUE Activate nonequilibrium ptSS-PCM.

FALSE Deactivate nonequilibrium ptSS-PCM.

RECOMMENDATION:

NonEquilibrium activates a default ptSS-PCM calculation with Marcus charge separation and no additional LR-PCM.

ChargeSeparation

Partition fast and slow charges in solvent equilibrium state

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Marcus Do slow/fast charge separation with the Marcus partition.

Pekar Do slow/fast charge separation with the Pekar partition.

RECOMMENDATION:

Charge separation is used in conjunction with the **StateSpecific** keyword in *$pcm*.

Example 7.10 PCM solvation effects on the vertical excitation energies of planar DMABN using the ptSS and ptLR methods.

```
$molecule
   0  1
   C      0.000046   -0.000398    1.904953
   C      1.210027    0.000379    1.186051
   C      1.214640   -0.000065   -0.194515
   C      0.000164   -0.000616   -0.933832
   C     -1.214349   -0.001557   -0.194687
   C     -1.209753   -0.001846    1.185775
   H      2.151949    0.001377    1.722018
   H      2.164371    0.000481   -0.709640
   H     -2.164082   -0.002008   -0.709781
   H     -2.151763   -0.002287    1.721615
   C     -0.000227    0.001061    3.325302
   N     -0.000475    0.002405    4.484321
   N      0.000053   -0.000156   -2.297372
   C     -1.258656    0.001284   -3.036994
   H     -1.041042    0.001615   -4.102376
   H     -1.860897   -0.885647   -2.811117
   H     -1.859247    0.889133   -2.810237
   C      1.258563   -0.000660   -3.037285
   H      1.860651    0.886208   -2.810755
   H      1.859362   -0.888604   -2.811461
   H      1.040664   -0.000097   -4.102609
$end

$rem
   EXCHANGE              LRC-wPBEPBE
   OMEGA                 260
   BASIS                 6-31G*
   CIS_N_ROOTS           10
   RPA                   2
   CIS_SINGLETS          1
   CIS_TRIPLETS          0
   CIS_RELAXED_DENSITY   TRUE
   SOLVENT_METHOD        PCM
$end

$pcm
   NonEquilibrium
   Theory                IEFPCM
   StateSpecific         Perturb
$end

$solvent
   Dielectric            35.688000       ! Acetonitrile
   OpticalDielectric     1.806874
$end
```

2. Iterative State-Specific PCM

The control options for iterative state-specific PCM, as implemented in Ref. 77, follow the same setup as for the ADC(n)-family of methods described in Section 7.11.10. By setting the keyword **EqSolv** to the intended maximum number of iterations, self-consistent SS-PCM is activated for the reference state indicated in **EqState**. Iterations are performed until either the maximum number of iterations is reached or the convergence criteria are satisfied. These criteria are based on the excited-state energy and the surface charges and are set using **Eqs_Conv**. During the iterative optimization of the reaction field, states may change their ordering (root-flipping). Automatic state-following

(**EqState_Follow = true**) based on a combination of the electrostatic potential at the surface, the energy changes, and excited state dipole moments is available. It performs well for separated states or states of substantially different character (*i.e.* charge-transfer and locally excited states) but can break down for similar states close to their crossing points.

Note that the excited state energy with iterative state-specific PCMs is only physically meaningful for the reference state (marked in the output as "EqS-reference"). For all other states, only energies relative to the self-consistent reference state can be interpreted. The iterative equilibrium SS-PCM output lists the emission energy to the ground state (only EI-SS-PCM), as well as transition energies to the other excited states (both EI- and II-SS-PCM) in zeroth- and first-order (with nonequilibrium ptSS correction). For nonequilibrium II-SS-PCM, only the vertical excitation energy to the self-consistently optimized excited state is physically meaningful. For the nonequilibrium II-SS-PCM and ptSS-PCM calculations, again the **ChargeSeparation** keyword is employed.

EqSolv

Main switch of the self-consistent SS-PCM procedure.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	No self-consistent SS-PCM.
1	Single SS-PCM calculation (SCF + TDDFT) with the solvent field found on disk.
$n > 1$	Do a maximum of n automatic solvent-field iterations.

RECOMMENDATION:

We recommend to use 15 steps max. Typical convergence is 3–5 steps. In difficult cases 6–12. If the solvent-field iteration do not converge in 15 steps, something is wrong. For EI-SS-PCM jobs make sure a solvent field was written to disk by a previous job. II-SS-PCM will automatically begin by setting up the solvent field.

EqState

Specifies the state for which the solvent field is to be optimized.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1	energetically lowest excited state
2	2nd lowest excited state
...	

RECOMMENDATION:

Given that only one class of excited states is calculated, the state will be selected according to its energetic position shown in the "Exited-State Summary" of the output file.

Eqs_Conv

Controls the convergence of the solvent-field iterations by setting the convergence criteria (a mixture of SCF energy (EI-SS-PCM) or excited state energy (II-SS-PCM) and charge-vector). SCF energy criterion computes as $10^{-\text{value}}\ E_h$.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

SCF_CONVERGENCE$-4 = 4$

OPTIONS:

3 May be sufficient for emission energies

4 Assured converged total energies (2.7 meV)

5 Very tight

RECOMMENDATION:

Use the default.

EqState_Follow

Controls the automatic state-following based on the electrostatic potential at the surface, excited state energy changes, and excited state dipole moments.

INPUT SECTION: *$pcm*

TYPE:

NONE

DEFAULT:

FALSE

OPTIONS:

TRUE Activate automatic state-following

FALSE Deactivate automatic state-following.

RECOMMENDATION:

State-Following works well for separated states of different character and can become problematic for nearly degenerate states. It can be advisable to deactivate state-following if similar states are close.

2. (A) External-Iteration SS-PCM

EI-SS-PCM, otherwise known as Improta-Barone-Scalmani-Frisch PCM,[120] is activated by setting **StateSpecific** = **External** in the *$pcm* section. EI-SS-PCM iterates over both the SCF and TDDFT calculation by returning the excited-state reaction field into a frozen-Reaction-Field (fRF) SCF. Prior to this, the reaction field has to be set up in a separate job by setting **StateSpecific** = External without adding the keyword **EqSolv**. In the following iterative job, the reference state (**EqState**) can then be chosen freely among the states calculated prior, and iterations can be restarted in the same way (*e.g.*, to manually correct the automatic state-following). The EI-SS-PCM convergence is judged based on the change in the surface charge vector and the fRF-SCF energy, in order to converge all excited states to the same level. Additionally, the self-ptSS term indicates convergence though it is not used as an actual convergence criterion.

EI-SS-PCM is always performed as equilibrium solvation employing the full dielectric response of the solvent based on the static dielectric constant (**Dielectric**). Non-equilibrium emission and transition energies are calculated in first order at the ptSS-PCM level. EI-SS-PCM provides the correct solution of the state-specific Schrödinger equation without contamination from the ground state MOs, and hence, should be used whenever the excited state is to be treated on

equal footing with the ground state.

Example 7.7.11 TDDFT with External Iteration SS-PCM for nitrate in acetonitrile.

View input online

2. (B) Internal-Iteration SS-PCM

II-SS-PCM, otherwise known as the vertical excitation model (VEM),[168] is activated by setting the keyword **StateSpecific = Internal**. II-SS-PCM retains the same ground state wave-function in the iterations but exchanges the ground and excited-state reaction field during the TDDFT calculation. This leads to slight contamination of the excited state by the ground state MOs. II-SS-PCM is available both for equilibrium (adiabatic excitation model, AEM) and nonequilibrium (VEM), and with either the full matrix (f) or only the diagonal elements (d) of the excited state reaction field, as described in Ref. 168. To choose the exact model, the keyword **InternalIteration** is set to **VEM(d)**, **VEM(f)**, **AEM(d)**, or **AEM(f)**. In contrast to EI-SS-PCM, the II-SS-PCM approach automatically begins with a setup calculation to produce the initial excited state reaction field for the iterations, which then cannot be restarted. To still allow besides the automatic also for manual state following, a specific sequence of states can be listed by appending the chosen method in **InternalIteration** with -state. The sequence is terminated by a zero, after which state-following is re-engaged. For II-SS-PCM, convergence is judged on the reference state energy and the surface charge vector.

A nonequilibrium II-SS-PCM or VEM calculation corresponds to an iterative version of nonequilibrium ptSS-PCM for vertical excitation energies. Here, also the response of the density to the fast excited state reaction field is included. Hence, the VEM result for the converged reference state (indicated by "EqS-reference" in the output) corresponds to the vertical excitation energy, while the other states have no physical meaning.

Equilibrium II-SS-PCM or AEM, on the other hand, provides the excited reference state energy and transition energies to other excited states. However, emission energies from the reference state to the ground state are adiabatic because the ground state orbitals are not affected by the excited state reaction field. To obtain vertical emission energies, a further fRF ground state calculation has to be performed by setting **RF_ptSS_Save = true** and loading the reaction field in a consecutive job by setting **RF_ptSS_Read = true**. See Section 7.8.6 for further information.

InternalIteration
Controls the used internal iteration SS-PCM model. Additionally, a sequence of state to be followed can be appended after adding -state.

INPUT SECTION: *$pcm*

TYPE:
NONE

DEFAULT:
NONE

OPTIONS:

VEM(f)	Activate nonequilibrium II-SS-PCM (VEM) with the full matrix contribution.
VEM(d)	Activate nonequilibrium II-SS-PCM (VEM) with only diagonal matrix elements.
AEM(f)	Activate equilibrium II-SS-PCM (AEM) with the full matrix contribution.
AEM(d)	Activate equilibrium II-SS-PCM (AEM) with only diagonal matrix elements.
...-state x y z ... 0	Specify sequence of states to be followed (terminate with 0).

RECOMMENDATION:
The nonequilibrium versions (VEM) provide vertical excitation energies, while in equilibrium (AEM) the initial state for emission energies is prepared. By attaching the **-state** ending an explicit sequence of states to be followed can be provided. After terminating with a zero, state-following will be re-engaged.

Example 7.7.12 TDDFT with Internal Iteration SS-PCM and the additional ground state calculation for vertical emission energies of nitrate in acetonitrile.

View input online

7.3.7 Analytic Excited-State Hessian in TDDFT

To carry out vibrational frequency analysis of an excited state with TDDFT,[155,156] an optimization of the excited-state geometry is always necessary. Like the vibrational frequency analysis of the ground state, the frequency analysis of the excited state should be also performed at a stationary point on the excited state potential surface. The *$rem* variable CIS_STATE_DERIV should be set to the excited state for which an optimization and frequency analysis is needed, in addition to the *$rem* keywords used for an excitation energy calculation.

Compared to the numerical differentiation method, the analytic calculation of geometrical second derivatives of the excitation energy needs much less time but much more memory. The computational cost is mainly consumed by the steps to solve both the CPSCF equations for the derivatives of molecular orbital coefficients C^x and the CP-TDDFT equations for the derivatives of the transition vectors, as well as to build the Hessian matrix. The memory usages for these steps scale as $\mathcal{O}(3mN^2)$, where N is the number of basis functions and m is the number of atoms. For large systems, it is thus essential to solve all the coupled-perturbed equations in segments. In this case, the *$rem* variable CPSCF_NSEG is always needed.

In the calculation of the analytic TDDFT excited-state Hessian, one has to evaluate a large number of energy-functional derivatives: the first-order to fourth-order functional derivatives with respect to the density variables as well as their derivatives with respect to the nuclear coordinates. Therefore, a very fine integration grid for DFT calculation should be adapted to guarantee the accuracy of the results.

Analytic TDDFT/C-PCM Hessian has been implemented in Q-CHEM. Normal mode analysis for a system in solution can be performed with the frequency calculation by TDDFT/C-PCM method. The *$rem* and *$pcm* variables for the excited state calculation with TDDFT/C-PCM included in the vertical excitation energy example above are needed.

When the properties of large systems are calculated, you must pay attention to the memory limit.

Example 7.13 B3LYP/6-31G* optimization in gas phase, followed by a frequency analysis for the first excited state of the peroxy radical.

```
$molecule
   0 2
   C  1.004123  -0.180454   0.000000
   O -0.246002   0.596152   0.000000
   O -1.312366  -0.230256   0.000000
   H  1.810765   0.567203   0.000000
   H  1.036648  -0.805445  -0.904798
   H  1.036648  -0.805445   0.904798
$end

$rem
   JOBTYPE           opt
   EXCHANGE          b3lyp
   CIS_STATE_DERIV   1
   BASIS             6-31G*
   CIS_N_ROOTS       10
   CIS_SINGLETS      true
   CIS_TRIPLETS      false
   XC_GRID           000075000302
   RPA               0
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE           freq
   EXCHANGE          b3lyp
   CIS_STATE_DERIV   1
   BASIS             6-31G*
   CIS_N_ROOTS       10
   CIS_SINGLETS      true
   CIS_TRIPLETS      false
   XC_GRID           000075000302
   RPA               0
$end
```

Example 7.14 Geometry optimization for a low-lying excited state of cyclopentadienone in methanol, using TDDFT/C-PCM

```
$molecule
   0 1
  C   -0.0000000    0.6920860    1.4656691
  C   -0.0000000   -0.6920860    1.4656691
  C   -0.0000000   -1.1528931    0.1065000
  C    0.0000000   -0.0000000   -0.7957576
  C    0.0000000    1.1528931    0.1065000
  O    0.0000000   -0.0000000   -2.0301721
  H   -0.0000000    1.3254423    2.3427356
  H   -0.0000000   -1.3254423    2.3427356
  H   -0.0000000   -2.1834532   -0.2231979
  H    0.0000000    2.1834532   -0.2231979
$end

$rem
   JOBTYPE             opt
   EXCHANGE            b3lyp
   CIS_N_ROOTS         5
   CIS_SINGLETS        true
   CIS_TRIPLETS        true
   CIS_STATE_DERIV     1   Lowest TDDFT state
   BASIS               6-311G*
   XC_GRID             3
   SOLVENT_METHOD      pcm
   THRESH              12
$end

$pcm
   Theory      CPCM
   Method      SWIG
   Solver      Inversion
   Radii       Bondi
$end

$solvent
   Dielectric 32.613
$end
```

Example 7.15 Hessian calculation, using the optimized geometry from the previous example.

```
$molecule
   0 1
   C     0.0000000000    0.6940558365    1.4635362645
   C     0.0000000000   -0.6940558367    1.4635362652
   C     0.0000000000   -1.1539902580    0.1063088532
   C     0.0000000000    0.0000000000   -0.7890068343
   C     0.0000000000    1.1539902569    0.1063088524
   O     0.0000000000    0.0000000000   -2.0333287471
   H     0.0000000000    1.3287019844    2.3394236351
   H     0.0000000000   -1.3287019837    2.3394236361
   H     0.0000000000   -2.1861329696   -0.2193590119
   H     0.0000000000    2.1861329678   -0.2193590131
$end

$rem
   JOBTYPE           freq
   EXCHANGE          b3lyp
   CIS_N_ROOTS       5
   CIS_SINGLETS      true
   CIS_TRIPLETS      true
   CIS_STATE_DERIV   1   Lowest TDDFT state
   BASIS             6-311G*
   XC_GRID           3
   SOLVENT_METHOD    pcm
   MEM_STATIC        4000
   MEM_TOTAL         24000
   CPSCF_NSEG        3
   THRESH            12
$end

$pcm
   Theory      CPCM
   Method      SWIG
   Solver      Inversion
   Radii       Bondi
$end

$solvent
   Dielectric  32.613
$end
```

7.3.8 Spin-Orbit Coupling

The following sections describe two different ways to include spin-orbit couplings (SOCs) into TDDFT calculations.

7.3.8.1 SOCs Between TDDFT States

There are several levels at which SOCs can be incorporated into TDDFT:[133]

(i) one-electron part of the Breit-Pauli Hamiltonian,

(ii) one-electron SOC using scaled nuclear charges, and

(iii) full SOC using the mean-field treatment of the two-electron part.[133,211]

Options (i) and (ii) are are available for both TDA and RPA variants (including TDHF and CIS states), for restricted Kohn-Sham references only. Option (iii) is available for both restricted and unrestricted variants, but only within TDA. Calculations of SOC for SF-TDDFT are also possible within TDA using option (iii). A Python code is available,[161] which harvests SOCs along with TDDFT energies and oscillator strengths from the Q-CHEM output and uses them to construct and diagonalize the Breit-Pauli Hamiltonian, to obtain the coupled states.[162] It can be used to compute L- and M-edge spectra for TDDFT, CIS, and DFT/CIS methods.

The implementation of one-electron SOC, options (i) and (ii), is based on the following. The SOCs are computed by evaluating matrix elements of the one-electron part of the Breit-Pauli Hamiltonian:

$$\hat{H}_{\text{SO}} = -\frac{\alpha_0^2}{2} \sum_{i,A} \frac{Z_A}{r_{iA}^3} \left(\mathbf{r}_{iA} \times \mathbf{p}_i\right) \cdot \mathbf{s}_i \tag{7.28}$$

where i denotes electrons, A denotes nuclei, $\alpha_0 = 1/137.037$ is the fine structure constant, and Z_A is the bare positive charge on nucleus A. In the second quantization representation, the spin-orbit Hamiltonian in different directions can be expressed as

$$\hat{H}_{\text{SO},x} = -\frac{\alpha_0^2}{2} \sum_{pq} \tilde{L}_{x,pq} \frac{\hbar}{2} \left(\hat{a}_p^\dagger \hat{a}_{\bar{q}} + \hat{a}_{\bar{p}}^\dagger \hat{a}_q\right) \tag{7.29a}$$

$$\hat{H}_{\text{SO},y} = -\frac{\alpha_0^2}{2} \sum_{pq} \tilde{L}_{y,pq} \frac{\hbar}{2\text{i}} \left(\hat{a}_p^\dagger \hat{a}_{\bar{q}} - \hat{a}_{\bar{p}}^\dagger \hat{a}_q\right) \tag{7.29b}$$

$$\hat{H}_{\text{SO},z} = -\frac{\alpha_0^2}{2} \sum_{pq} \tilde{L}_{z,pq} \frac{\hbar}{2} \left(\hat{a}_p^\dagger \hat{a}_q - \hat{a}_{\bar{p}}^\dagger \hat{a}_{\bar{q}}\right) \tag{7.29c}$$

where $\hat{\tilde{L}}_\alpha = \hat{L}_\alpha r^{-3}$ for $\alpha \in \{x, y, z\}$ and $\tilde{L}_{\alpha,pq}$ are the matrix elements of this operator. The single-reference *ab initio* excited states (within the TDA) are given by

$$\left|\Phi_{\text{singlet}}^I\right\rangle = \sum_{i,a} s_i^{Ia} \left(\hat{a}_a^\dagger \hat{a}_i + \hat{a}_{\bar{a}}^\dagger \hat{a}_{\bar{i}}\right) |\Phi_{\text{HF}}\rangle \tag{7.30a}$$

$$\left|\Phi_{\text{triplet}}^{I,M_s=0}\right\rangle = \sum_{i,a} t_i^{Ia} \left(\hat{a}_a^\dagger \hat{a}_i - \hat{a}_{\bar{a}}^\dagger \hat{a}_{\bar{i}}\right) |\Phi_{\text{HF}}\rangle \tag{7.30b}$$

$$\left|\Phi_{\text{triplet}}^{I,M_s=1}\right\rangle = \sqrt{2} \sum_{i,a} t_i^{Ia}\ \hat{a}_a^\dagger \hat{a}_{\bar{i}} |\Phi_{\text{HF}}\rangle \tag{7.30c}$$

$$\left|\Phi_{\text{triplet}}^{I,M_s=-1}\right\rangle = \sqrt{2} \sum_{i,a} t_i^{Ia}\ \hat{a}_{\bar{a}}^\dagger \hat{a}_i |\Phi_{\text{HF}}\rangle \tag{7.30d}$$

where s_i^{Ia} and t_i^{Ia} are singlet and triplet excitation coefficients of the Ith singlet or triplet state respectively, with the normalization

$$\sum_{ia} (s_i^{Ia})^2 = \sum_{ia} (t_i^{Ia})^2 = \frac{1}{2}\,. \tag{7.31}$$

The quantity $|\Phi_{\text{HF}}\rangle$ refers to the Hartree-Fock ground state. Thus the SOC constant from the singlet state to different triplet manifolds are

$$\left\langle \Phi_{\text{singlet}}^I \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s=0} \right\rangle = \frac{\alpha_0^2 \hbar}{2} \left(\sum_{i,a,b} \tilde{L}_{z,ab}\ s_i^{Ia}\ t_i^{Jb} - \sum_{i,j,a} \tilde{L}_{z,ij}\ s_i^{Ia}\ t_j^{Ja} \right) \tag{7.32}$$

or

$$\begin{aligned} \left\langle \Phi_{\text{singlet}}^I \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s=\pm 1} \right\rangle = {} & \mp \frac{\alpha_0^2 \hbar}{2\sqrt{2}} \left(\sum_{i,a,b} \tilde{L}_{x,ab}\ s_i^{Ia}\ t_i^{Jb} - \sum_{i,j,a} \tilde{L}_{x,ij}\ s_i^{Ia}\ t_j^{Ja} \right) \\ & + \frac{\alpha_0^2 \hbar}{2\sqrt{2}\text{i}} \left(\sum_{i,a,b} \tilde{L}_{y,ab}\ s_i^{Ia}\ t_i^{Jb} - \sum_{i,j,a} \tilde{L}_{y,ij}\ s_i^{Ia}\ t_j^{Ja} \right)\,. \end{aligned} \tag{7.33}$$

The SOC constant between different triplet manifolds can be obtained as

$$\begin{aligned}\left\langle \Phi_{\text{triplet}}^{I,M_s=0} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s=\pm 1} \right\rangle = \mp \frac{\alpha_0^2 \hbar}{2\sqrt{2}} \left(\sum_{i,a,b} \tilde{L}_{x,ab}\, t_i^{Ia}\, t_i^{Jb} + \sum_{i,j,a} \tilde{L}_{x,ij}\, t_i^{Ia}\, t_j^{Ja} \right) \\ \frac{\alpha_0^2 \hbar}{2\sqrt{2}\mathrm{i}} \left(\sum_{i,a,b} \tilde{L}_{y,ab}\, t_i^{Ia}\, t_i^{Jb} + \sum_{i,j,a} \tilde{L}_{y,ij}\, t_i^{Ia}\, t_j^{Ja} \right)\end{aligned} \tag{7.34}$$

or

$$\left\langle \Phi_{\text{triplet}}^{I,M_s=\pm 1} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s=\pm 1} \right\rangle = \pm \frac{\alpha_0^2 \hbar}{2} \left(\sum_{i,a,b} \tilde{L}_{z,ab}\, t_i^{Ia}\, t_i^{Jb} + \sum_{i,j,a} \tilde{L}_{z,ij}\, t_i^{Ia}\, t_j^{Ja} \right) . \tag{7.35}$$

Note that

$$\left\langle \Phi_{\text{triplet}}^{I,M_s=0} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s=0} \right\rangle = 0 = \left\langle \Phi_{\text{triplet}}^{I,M_s=\pm 1} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s=\mp 1} \right\rangle . \tag{7.36}$$

The total (root-mean-square) spin-orbit coupling is

$$\left\langle \Phi_{\text{singlet}}^{I} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J} \right\rangle = \left(\sum_{M_s=0,\pm 1} \left\| \left\langle \Phi_{\text{singlet}}^{I} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s} \right\rangle \right\|^2 \right)^{1/2} \tag{7.37a}$$

$$\left\langle \Phi_{\text{triplet}}^{I} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J} \right\rangle = \left(\sum_{M_s=0,\pm 1} \left\| \left\langle \Phi_{\text{triplet}}^{I,M_s} \middle| \hat{H}_{\text{SO}} \middle| \Phi_{\text{triplet}}^{J,M_s} \right\rangle \right\|^2 \right)^{1/2} . \tag{7.37b}$$

For RPA states, the SOC constant can simply be obtained by replacing $s_i^{Ia}\, t_j^{Jb}$ with $X_{i,\text{triplet}}^{Ia} X_{j,\text{triplet}}^{Jb} + Y_{i,\text{singlet}}^{Ia} Y_{j,\text{triplet}}^{Jb}$ and $t_i^{Ia}\, t_j^{Jb}$ with $X_{i,\text{triplet}}^{Ia} X_{j,\text{triplet}}^{Jb} + Y_{i,\text{triplet}}^{Ia} Y_{j,\text{triplet}}^{Jb}$.

A TDDFT + SOC calculation is activated by *$rem* variable CALC_SOC:

CALC_SOC

Controls whether to calculate the SOC constants for EOM-CC, RAS-CI, CIS, TDDFT/TDA and TDDFT/RPA.

TYPE:

INTEGER/LOGICAL

DEFAULT:

0

OPTIONS:

0 Do not perform the SOC calculation.

1 Perform the SOC calculation using the one-electron part of the Breit-Pauli Hamiltonian

2 Perform the full SOC calculation using a mean-field treatment of the two-electron contribution

4 Perform the SOC calculation using scaled nuclear charges

RECOMMENDATION:

The final two options are described in Section 7.10.20.4.

Examples 7.16, 7.17, and 7.18 illustrate calculations of SOCs for (SF)-TDDFT states using the above features. These calculations can also be carried out for CIS states by modifying METHOD appropriately.

The LIBWFA analysis of the spinless one-particle transition density matrices (as described in Ref. 209) is implemented for the TDDFT and SF-TDDFT calculations. To activate this analysis, use the following: CALC_SOC = 2, STATE_ANALYSIS = TRUE, MOLDEN_FORMAT = TRUE, and NTO_PAIRS = N. Consult Section 10.2 for details of the LIBWFA package.

Note: This analysis differs from the NTO analysis of the regular transition density matrix between the singlet and triplet states.

Example 7.16 Calculation of one-electron SOCs for using TDDFT/TDA.

```
$comment
   This sample input calculates the spin-orbit coupling constants for water
   between its ground state and its TDDFT/TDA excited triplets as well as the
   coupling between its TDDFT/TDA singlets and triplets.  Results are given in
   cm-1.
$end

$molecule
   0 1
   H       0.000000    -0.115747     1.133769
   H       0.000000     1.109931    -0.113383
   O       0.000000     0.005817    -0.020386
$end

$rem
   EXCHANGE             b3lyp
   BASIS                6-31G
   CIS_N_ROOTS          4
   CIS_CONVERGENCE      8
   MAX_SCF_CYCLES       600
   MAX_CIS_CYCLES       50
   SCF_ALGORITHM        diis
   MEM_STATIC           300
   MEM_TOTAL            2000
   CIS_SINGLETS         true
   CIS_TRIPLETS         true
   CALC_SOC             true
   MAX_CIS_CYCLES       300
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 7.17 Calculation of full SOCs for water molecule including mean-field treatment of the two-electron part of the Breit-Pauli Hamilton and UHF/TDDFT/TDA.

```
$comment
Calculation of full SOCs for water molecule inlcuding mean-field treatment
of the two-electron part of the Breit-Pauli Hamiltonian and Wigner-Eckart theorem.
UHF/TDDFT/B3LYP/6-31G within the TDA.
$end

$molecule
0 1
H       0.000000    -0.115747     1.133769
H       0.000000     1.109931    -0.113383
O       0.000000     0.005817    -0.020386
$end

$rem
jobtype                 sp
unrestricted            true
method                  b3lyp
basis                   6-31G
cis_n_roots             4
cis_convergence         8
cis_singlets            true
cis_triplets            true
calc_soc                2
$end
```

Example 7.18 Calculation of SOCs for methylene using non-collinear SF-TDDFT/PBE0.

```
$comment
Calculation of SOCs for methylene using non-collinear SF-TDDFT/PBE0,
with tight convergence.
$end

$molecule
0 3
H1
C  H1 1.0775
H2 C  1.0775 H1 133.29
$end

$rem
method = pbe0
basis = cc-pvtz
scf_convergence = 12
cis_convergence = 12
thresh = 14
cis_n_roots = 2
calc_soc = 2                    Compute full SOC with mean-field treatment of 2el part
wang_ziegler_kernel =  true     Important for 1,1 diradicals
spin_flip = true
$end
```

Example 7.19 DFT/CIS calculation of the sulfur $L_{2,3}$-edge in H_2S. Input options are set to compute SOCs and state-to-state transition moments that can be read by the PYSETSOC code,[161] to construct and diagonalize the Breit-Pauli Hamiltonian. See Section 7.2.2.4 for additional details about the DFT/CIS method.

```
$molecule
0 1
S
H   1 1.343015
H   1 1.343015 2 92.538291
$end


$rem
 method          cam-b3lyp
 unrestricted    false
 cis_n_roots     200
 calc_soc        2     ! spin-orbit mean-field
 sts_mom         true  ! state-to-state moments, for osc. strengths
 dftcis          1
 dftcis_params   2
 basis           def2-tzvpd
 trnss           true
 trtype          3     ! use alist
 n_sol           3
$end

$alist
2 3 4 5
$end
```

7.3.8.2 TDDFT+SOC for Spin-Adiabats

For a system exhibiting strong SOC, it can be extremely helpful to calculate the electronic eigenfunctions (called "spin-adiabats") of the entire (Coulomb + SOC) Hamiltonian. Computation of such spin-adiabatic states has been implemented within the CIS or TDDFT/TDA framework, with the SOC treated by the one-electron Breit-Pauli Hamiltonian at the post-SCF level.

Within a CIS or TDDFT/TDA-SOC, the wavefunction is represented as

$$|\Psi\rangle = \sum_{\epsilon} \sum_{ia} t_{ia}^{(\epsilon)} \left|\Phi_i^{a(\epsilon)}\right\rangle \tag{7.38}$$

where ϵ is a label of spin, and

$$\left|\Phi_i^{a(\epsilon)}\right\rangle \equiv \begin{cases} \frac{1}{\sqrt{2}}\left(|\Psi_{i_\alpha}^{a_\alpha}\rangle + |\Psi_{i_\beta}^{a_\beta}\rangle\right) & \text{if } \epsilon = \text{singlet} \\ \frac{1}{\sqrt{2}}\left(|\Psi_{i_\alpha}^{a_\alpha}\rangle - |\Psi_{i_\beta}^{a_\beta}\rangle\right) & \text{if } \epsilon = \text{triplet}, m_s = 0 \\ |\Psi_{i_\beta}^{a_\alpha}\rangle & \text{if } \epsilon = \text{triplet}, m_s = +1 \\ |\Psi_{i_\alpha}^{a_\beta}\rangle & \text{if } \epsilon = \text{triplet}, m_s = -1 \end{cases} \tag{7.39}$$

The spin-adiabatic wavefunctions in Eq. (7.38) are obtained as the eigenstates of following Hamiltonian having $(4N_O N_V)^2$ total matrix elements:

$$H = A + V_{\text{SO}} + V_m \tag{7.40}$$

In Eq. (7.40), A is the usual CIS-like TDDFT/TDA linear response matrix (including the one-electron and two-electron terms):

$$\begin{aligned}\langle \Phi_{\boldsymbol{i}}^{\boldsymbol{a}} | A | \Phi_{\boldsymbol{j}}^{\boldsymbol{b}} \rangle = F_{\boldsymbol{ab}}\delta_{\boldsymbol{ij}} - F_{\boldsymbol{ji}}\delta_{\boldsymbol{ab}} + \langle \boldsymbol{aj}|\boldsymbol{ib}\rangle - c_{\text{HF}}\langle \boldsymbol{aj}|\boldsymbol{bi}\rangle \\ + \Omega_{\boldsymbol{aibj}} + E_{gs}\delta_{\boldsymbol{ab}}\delta_{\boldsymbol{ij}}\end{aligned} \tag{7.41}$$

where the indices $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{i}, \boldsymbol{j}$ are the spin-orbitals, F is the Fock matrix, E_{gs} is the SCF energy, and Ω is the DFT exchange-correlation functional. More details can be found in Sec. 7.3.1 and Ref. [16].

The V_{SO} in Eq. (7.40) is the one-electron SOC Hamiltonian appearing in Eq. (7.28), and the $V_m = g\mu_B \mathbf{B}\cdot\mathbf{S}$ is the spin-Zeeman term. Here $g = 2$ is assumed. (The coupling between orbital motion and magnetic field is not yet included.) In matrix form, $V_{\text{SO}} + V_m$ reads

$$\langle \Psi_{\boldsymbol{i}}^{\boldsymbol{a}} | V_{\text{SO}} + V_m | \Psi_{\boldsymbol{j}}^{\boldsymbol{b}} \rangle = V_{\boldsymbol{ab}}\delta_{\boldsymbol{ij}} - V_{\boldsymbol{ji}}\delta_{\boldsymbol{ab}} \tag{7.42}$$

The matrix elements of V are given by

$$V_{p_\alpha q_\alpha} = \mu_B B_z \delta_{pq} - \frac{\hbar\alpha_0^2}{4}\tilde{L}_{pq}^z \tag{7.43}$$

$$V_{p_\beta q_\beta} = -\mu_B B_z \delta_{pq} + \frac{\hbar\alpha_0^2}{4}\tilde{L}_{pq}^z \tag{7.44}$$

$$V_{p_\alpha q_\beta} = \left(\mu_B B_x \delta_{pq} - \frac{\hbar\alpha_0^2}{4}\tilde{L}_{pq}^x\right) - i\left(\mu_B B_y \delta_{pq} - \frac{\hbar\alpha_0^2}{4}\tilde{L}_{pq}^y\right) \tag{7.45}$$

$$V_{p_\beta q_\alpha} = \left(\mu_B B_x \delta_{pq} - \frac{\hbar\alpha_0^2}{4}\tilde{L}_{pq}^x\right) + i\left(\mu_B B_y \delta_{pq} - \frac{\hbar\alpha_0^2}{4}\tilde{L}_{pq}^y\right) \tag{7.46}$$

where α_0 is the fine-structure constant and $\tilde{\mathbf{L}} = \mathbf{L}/r^3$.

Note: Currently, this CIS or TDDFT/TDA-SOC implementation supports only restricted SCF calculations (which would produce excited state singlets and triplets in the absence of SOC). Also, the ground state is not coupled to the excited states through SOC (or otherwise).

The code is triggered by setting *$rem* variable CIS_SOC = N, where N is the number of roots desired:

CIS_SOC

Controls the roots of performing TDDFT/TDA-SOC calculation.

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform the calculation.

N Solve the N lowest spin-adiabatic states of TDDFT/TDA-SOC.

RECOMMENDATION:

Less than or equal to $4\times$CIS_N_ROOTS. TDDFT/TDA-SOC first performs a standard TDDFT/TDA calculation so as to generate an initial guess before rerunning the diagonalization to generate the spin adiabats. Therefore, it is a good idea to perform a stand-alone normal TDDFT/TDA calculation and generate the excited spin-diabats and check the desired range of energies, before generating the excited spin-adiabats.

To include a post-SCF magnetic field, the *$bfield_postscf* section must be initialized:

```
$bfield_postscf
B_x B_y B_z
$end
```

Here, the B_x, B_y, B_z are the spatial components of the external magnetic field in Tesla. Their format must be float

(i.e., "0" should be "0.0").

Derivative coupling calculations are possible between spin-adiabats, but note that the value of the real and imaginary components may vary between different calculations – because the wavefunctions are complex-valued and may acquire a different global phase (gauge) for each calculation. The norm of derivative coupling should not change.

Examples 7.20, 7.21 and 7.22 illustrate the energy, gradient and derivative coupling calculations using TDDFT/TDA-SOC. In Example 7.22, a post-SCF external magnetic field is also included.

Example 7.20 TDDFT/TDA-SOC calculation of formaldehyde using the ωB97X functional for the lowest 7 spin-adiabats.

```
$rem
jobtype                 sp
exchange                wb97x
basis                   6-31g
cis_convergence         8
cis_n_roots             2
symmetry                false
sym_ignore              true
cis_soc                 7
$end

$molecule
0   1
C 0 0 0
H 0 0 1.101224
H 1.00171657 0 -0.45744749
O -1.00461935 0 -0.645642
$end
```

Example 7.21 Nuclear gradient of the 4th state in the TDDFT/TDA-SOC calculation of formaldehyde with the ωB97X functional.

```
$rem
jobtype                 force
exchange                wb97x
basis                   6-31g
cis_convergence         8
cis_n_roots             2
symmetry                false
sym_ignore              true
cis_soc                 4
cis_state_deriv         4
new_dft                 1 ! This is necessary
$end

$molecule
0   1
C 0 0 0
H 0 0 1.101224
H 1.00171657 0 -0.45744749
O -1.00461935 0 -0.645642
$end
```

Example 7.22 Derivative coupling between the 4th and 7th TDDFT/TDA-SOC states for a calculation of benzaldehyde with the ωB97X functional, in the presence of 10 Tesla external magnetic field along the z-direction.

```
$rem
jobtype                 sp
exchange                wb97x
basis                   6-31g
scf_convergence         10
cis_convergence         10
cis_n_roots             2
symmetry                false
sym_ignore              true
cis_soc                 7
calc_nac                true
cis_der_numstate        2
$end

$derivative_coupling
...
4 7
$end

$bfield_postscf
0.0 0.0 10.0
$end

$molecule
0   1
C -0.0827555811 -0.0021927111 0.0742168160
C 0.0163712008 -0.0019362341 1.3503524350
H 0.9426483579 -0.0021873514 1.9554892885
C -1.2462538108 -0.0012178895 2.1035447415
H -1.1928615664 -0.0009432042 3.1942205420
C -2.4029093145 -0.0008942470 1.4089112839
H -3.3571329386 -0.0003915665 1.9718859042
C -2.4418611557 -0.0012099958 0.0672877891
H -3.3674530317 -0.0009676703 -0.5174127708
C -1.1678601986 -0.0019316068 -0.6798528411
H -1.2448466539 -0.0022668669 -1.7738119236
C 1.1607172056 -0.0028394281 -0.8113481262
H 1.2563061018 -0.0031882755 -1.8250763954
O 2.4950317353 -0.0032860998 -0.1735593327
$end
```

7.3.9 CIS-1D and TDDFT-1D

Propagating nonadiabatic systems forward in time remains challenging particularly at crossings between the ground state (S_0) and the first excited state (S_1). For S_1/S_0 conical intersections, conventional electronic structure methods (such as HF/CIS and DFT/TDDFT) recover the wrong dimensionality of the crossing seam. The TDDFT-1D method address this limitation by combining DFT and configuration interaction to achieve smooth crossings of S_0 and S_1 states[244] while keeping the cost of the calculation remains low (comaprable to a standard TDDFT calculation).

The idea behind the CIS-1D and the TDDFT-1D method is to diagonalize a configuration interaction Hamiltonian whose basis includes the reference state, all singly excited configurations, and one unique doubly excited configuration. This unique double (the 1D in CIS-1D/TDDFT-1D) in the configuration interaction Hamiltonian gives rise to the CIS-

1D method if one starts from a Hartree-Fock reference, and the TDDFT-1D method if one starts from a Kohn-Sham reference.

The doubly excited configuration is formed by exciting a pair of electrons from the HOMO (h) to the LUMO (l). To find the unique double, the cannonical molecular orbitals are rotated such that the energy of the doubly excited configuration (E_d) is minimized[8].

From these optimized orbitals, the following configuration interaction Hamiltonian is constructed, as explained previosly, in the basis of the reference state, all singles $|\psi_i^a\rangle$, and one unique double $|\psi_{h\bar{h}}^{l\bar{l}}\rangle$:

$$\boldsymbol{H} = \begin{pmatrix} \epsilon_0 & 0 & \left\langle \psi_0 | H | \psi_{h\bar{h}}^{l\bar{l}} \right\rangle \\ 0 & \left\langle \psi_i^a | H | \psi_j^b \right\rangle & \left\langle \psi_i^a | H | \psi_{h\bar{h}}^{l\bar{l}} \right\rangle \\ \left\langle \psi_0 | H | \psi_{h\bar{h}}^{l\bar{l}} \right\rangle & \left\langle \psi_{h\bar{h}}^{l\bar{l}} | H | \psi_i^a \right\rangle & \left\langle \psi_{h\bar{h}}^{l\bar{l}} | H | \psi_{h\bar{h}}^{l\bar{l}} \right\rangle \end{pmatrix}. \tag{7.47}$$

Diagonalizing the above Hamiltonian yields the CIS-1D or TDDFT-1D states. The excitation energies computed with TDDFT-1D are generally at the same level of accuracy as TDDFT with the Tamm-Dancoff approximation (TDA) at geometries far from crossing points. However, TDDFT-1D's key advantage appears near ground and excited state crossing points, where it produces smooth potential energy surfaces.

To facilitate nonadiabatic dynamics simulations, analytical gradients and derivative couplings[9,245] are available for the TDDFT-1D method. Further, minimum energy crossing points (MECPs) can be located for TDDFT-1D states.

7.3.9.1 Job Control and examples for TDDFT-1D

The rem variables for TDDFT-1D calculations are similar to those for CIS calculations. The following rem variables are used to control the TDDFT-1D calculation:

CIS1D_N_ROOTS

Sets the number of CIS-1D and TDDFT-1D states to calculate. The lowest CIS-1D/TDDFT-1D state is the ground state, which is followed by excited states.

TYPE:

INTEGER

DEFAULT:

0 Do not calculate any CIS-1D or TDDFT-1D states.

OPTIONS:

n $n > 0$, Calculate the lowest n CIS-1D or TDDFT-1D states.

RECOMMENDATION:

None

CIS1D_ED_CONVERGENCE

The first step in TDDFT-1D is to find the optimized orbitals for the double by minimizing the energy of the doubly excited configuration, E_d. This variable controls the convergence criterion for the minimization of E_d. The orbitals are optimized when the error is lower than $10^{-\text{CIS1D\_ED\_CONVERGENCE}}$.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n Convergence achieved when the error is lower than 10^{-n}.

RECOMMENDATION:

The convergence criterion for the roots of the CIS-1D and TDDFT-1D calculations is set by the CIS_CONVERGENCE rem variable, which is set to 9 by default for a CIS-1D/TDDFT-1D calculation. If CIS_CONVERGENCE is increased, then CIS1D_ED_CONVERGENCE should also be increased.

CIS1D_SCALE_GD

The coupling element between the ground and doubly excited configuration in the Hamiltonian in Eq. (7.47), ($\langle\psi_0|H|\psi^{l\bar{l}}_{h\bar{h}}\rangle$) are scaled by this factor.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n $0 \leq n \leq 100$, The coupling element is scaled by $n/100$.

RECOMMENDATION:

Since the KS reference is not a true wave function, there is no rigorous way to determine the coupling elements in the configuration interaction Hamiltonian. It sometimes becomes necessary to scale the coupling elements to obtain accurate potential energy surfaces, and the scaling factor has to be determined with some benchmarking[8].

CIS1D_SCALE_SD

The coupling element between the singles and the double in the Hamiltonian in Eq. (7.47), ($\langle\psi^a_i|H|\psi^{l\bar{l}}_{h\bar{h}}\rangle$) are scaled by this factor.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n $0 \leq n \leq 100$, The coupling element is scaled by $n/100$.

RECOMMENDATION:

Same as CIS1D_SCALE_GD.

CIS1D_STATE_DERIV

Selects the CIS-1D/TDDFT-1D state for which gradients are calculated. This is useful for jobs such as geometry optimizations.

TYPE:

INTEGER

DEFAULT:

-1 Does not select any state

OPTIONS:

n $n \geq 0$, Selects the nth CIS-1D/TDDFT-1D state.

RECOMMENDATION:

None

CIS1D_DER_NUMSTATE

Determines the number of states for which derivative couplings are to be calculated. The states are specified in the *$derivative_coupling* section

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not calculate derivative couplings.

n Calculate $n(n-1)/2$ pairs of derivative couplings.

RECOMMENDATION:

None

Example 7.23 TDDFT-1D calculation of water using the B3LYP functional for the lowest 5 states.

```
$molecule
0 1
H 0.96 0 0
O 0.000000000000000 0.000000000000000 0.000000000000000
H -0.240364803892264 0.929421734762983 0
$end

$rem
METHOD    b3lyp
BASIS     cc-pvdz
SYM_IGNORE  true
cis1d_n_roots 5
$end
```

Example 7.24 TDDFT-1D calculation of the gradient of the 1st excited state of water in the presence of external charges

```
$molecule
0 1
H 0.96 0 0
O 0.000000000000000 0.000000000000000 0.000000000000000
H -0.240364803892264 0.929421734762983 0
$end

$rem
METHOD    b3lyp
BASIS     cc-pvdz
SYM_IGNORE  true
cis1d_n_roots 4
cis1d_state_deriv 1
jobtype force
$end

$external_charges
0.0 0.0 -1.0 -1.0
0.0 0.0 1.2 1.0
0.0 0.0 -2.0 -1.0
0.0 0.0 2.2 1.0
$end
```

Example 7.25 Minimum energy crossing point (MECP) between the ground and the 1st excited state of ethylene using the ωB97X functional.

```
$rem
geom_opt_driver optimize
scf_convergence 10
thresh_diis_switch 9
cis_convergence 8
jobtype opt
mecp_opt      true
mecp_methods  branching_plane
MECP_PROJ_HESS true
mecp_state1  [0,0]
mecp_state2  [0,1]
unrestricted false
calc_nac 1
scf_algorithm diis_gdm
cis1d_n_roots 5
exchange wb97x
basis 6-31G*
symmetry_ignore true
$end

$molecule
0 1
C          0.0188526101    0.0181382820   -0.3533168752
C          1.3813178426    0.0038321438   -0.0594695222
H          2.0062266305   -0.8967843506    0.0891927721
H         -0.3788613100   -0.7769905091   -1.0012547284
H         -0.0243170648   -0.6123230262    0.6288689086
H          1.9213240937    0.9262667800    0.1943673040
$end
```

7.3.10 Examples

Example 7.26 This example shows two jobs which request variants of time-dependent density functional theory calculations. The first job, using the default value of RPA = FALSE, performs TDDFT in the Tamm-Dancoff approximation (TDA). The second job, with RPA = TRUE performs a both TDA and full TDDFT calculations.

```
$comment
   methyl peroxy radical
   TDDFT/TDA and full TDDFT with 6-31+G*
$end

$molecule
   0  2
   C  1.00412  -0.18045    0.00000
   O -0.24600   0.59615    0.00000
   O -1.31237  -0.23026    0.00000
   H  1.81077   0.56720    0.00000
   H  1.03665  -0.80545   -0.90480
   H  1.03665  -0.80545    0.90480
$end

$rem
   EXCHANGE          b
   CORRELATION       lyp
   CIS_N_ROOTS       5
   BASIS             6-31+G*
   SCF_CONVERGENCE   7
$end

@@@

$molecule
   read
$end

$rem
   EXCHANGE          b
   CORRELATION       lyp
   CIS_N_ROOTS       5
   CIS_MAX_CYCLES    40
   RPA               true
   BASIS             6-31+G*
   SCF_CONVERGENCE   7
$end
```

Example 7.27 This example shows a calculation of the excited states of a formamide-water complex within a reduced excitation space of the orbitals located on formamide.

```
$comment
   formamide-water TDDFT/TDA in reduced excitation space
$end

$molecule
   0 1
   H  1.13  0.49 -0.75
   C  0.31  0.50 -0.03
   N -0.28 -0.71  0.08
   H -1.09 -0.75  0.67
   H  0.23 -1.62 -0.22
   O -0.21  1.51  0.47
   O -2.69  1.94 -0.59
   H -2.59  2.08 -1.53
   H -1.83  1.63 -0.30
$end

$rem
   EXCHANGE          b3lyp
   CIS_N_ROOTS       10
   BASIS             6-31++G**
   TRNSS             TRUE
   TRTYPE            1
   CUTOCC            60
   CUTVIR            40
   CISTR_PRINT       TRUE
   N_SOL             6
$end

$solute
1
2
3
4
5
6
$end
```

Example 7.28 SF-TDDFT SP calculation of the 6 lowest states of the TMM diradical using recommended 50-50 functional.

```
$molecule
   0 3
   C
   C  1  CC1
   C  1  CC2   2  A2
   C  1  CC2   2  A2     3  180.0
   H  2  C2H   1  C2CH   3    0.0
   H  2  C2H   1  C2CH   4    0.0
   H  3  C3Hu  1  C3CHu  2    0.0
   H  3  C3Hd  1  C3CHd  4    0.0
   H  4  C3Hu  1  C3CHu  2    0.0
   H  4  C3Hd  1  C3CHd  3    0.0

   CC1    = 1.35
   CC2    = 1.47
   C2H    = 1.083
   C3Hu   = 1.08
   C3Hd   = 1.08
   C2CH   = 121.2
   C3CHu  = 120.3
   C3CHd  = 121.3
   A2    = 121.0
$end

$rem
   EXCHANGE          gen
   BASIS             6-31G*
   SCF_GUESS         core
   SCF_CONVERGENCE   10
   MAX_SCF_CYCLES    100
   SPIN_FLIP         1
   CIS_N_ROOTS       6
   CIS_CONVERGENCE   10
   MAX_CIS_CYCLES    100
$end

$xc_functional
   X  HF    0.50
   X  S     0.08
   X  B     0.42
   C  VWN   0.19
   C  LYP   0.81
$end
```

Example 7.29 SF-TDDFT with non-collinear exchange-correlation functional for low-lying states of CH_2.

```
$comment
  Non-collinear SF-DFT calculation for CH2 at 3B1 state geometry from
  EOM-CCSD(fT) calculation
$end

$molecule
   0 3
   C
   H  1 rCH
   H  1 rCH  2 HCH

   rCH = 1.0775
   HCH = 133.29
$end

$rem
   EXCHANGE             PBE0
   BASIS                cc-pVTZ
   SPIN_FLIP            1
   WANG_ZIEGLER_KERNEL  TRUE
   SCF_CONVERGENCE      10
   CIS_N_ROOTS          6
   CIS_CONVERGENCE      10
$end
```

Example 7.30 SF-TDDFT calculation with collinear B5050LYP for p-benzyne with wave-function analysis (natural orbitals and NTOs) performed by LIBWFA.

```
$comment
Para-benzyne diradical
Equilibrium singlet state geom from:
J. Chem. Phys. 136, 204103 (2012)
Enu = 187.2138176166 hartree
$end

$molecule
   0 3
   H    2.145810  -1.225292   0.000000
   C    1.201382  -0.709285   0.000000
   C    1.201382   0.709285   0.000000
   H    2.145810   1.225292   0.000000
   C    0.000000   1.335664   0.000000
   C   -1.201382   0.709285   0.000000
   H   -2.145810   1.225291   0.000000
   C   -1.201382  -0.709285   0.000000
   H   -2.145810  -1.225291   0.000000
   C    0.000000  -1.335664   0.000000
$end

$rem
   METHOD          = b5050lyp
   BASIS           = 6-31G*
   CIS_N_ROOTS     = 4
   SPIN_FLIP       = true
   NEW_DFT         = true
   STATE_ANALYSIS  = true
   WFA_REF_STATE   = 1
   MOLDEN_FORMAT   = true
   NTO_PAIRS       = 4
$end
```

7.4 Real-Time SCF Methods

7.4.1 Introduction & Theory

Although the theory discussed in Section 7.3 is known universally as "time-dependent" DFT (TDDFT), in truth it is the frequency-domain transformation of linear-response (LR) DFT,[45,78] and is sometimes given the additional designation of LR-TDDFT in order to distinguish it from the "real time" (RT) version of TDDFT that is described in this section. The phrase "real-time time-dependent DFT" (RT-TDDFT) is sufficiently awkward that the theory described here is also known as *time-dependent Kohn-Sham* (TDKS) theory,[111,276,277] terminology that is actually more consistent with the original language used by Gross and co-workers developing a theory based on the time-dependent Kohn-Sham equation.[85–87] The TDKS approach is explicitly time-dependent, and amounts to propagation of time-dependent Kohn-Sham MOs following a perturbation of the ground-state density.

LR-TDDFT calculations are often the most efficient way to predict resonant electronic response frequencies and intensities when only a small number of low-lying excited states are desired. To obtain broadband spectra (in the x-ray regime, say), hundreds of excited states may be required, however. In such cases, the real-time approach may be preferable because it can be used to obtain the entire absorption spectrum (at all excitation energies) via Fourier transform

of the time-dependent dipole moment function, without the need to compute the spectrum state-by-state. This is the theoretical basis of *real-time* electronic structure methods in general.[153,215] A perturbation creates a superposition of all (symmetry-allowed) excitations, and the Fourier components of the ensuing time evolution encode all of the excitation energies. This theory is described in somewhat more detail in the next section, following which the TDKS job control variables are described in Section 7.4.2. Calculation of broadband absorption spectra using the TDKS approach is discussed in Section 7.4.3.

Starting with v. 5.3, Q-CHEM's TDKS module has been substantially rewritten, including support for advanced propagators,[276] complex absorbing potentials,[277] and Padé approximants to accelerate convergence of the Fourier-transformed dipole moment function.[111] Users of the TDKS/RT-TDDFT code are asked to cite Refs. 277 and 111.

Following a perturbation to the ground-state MOs $\phi_k(\mathbf{r},0)$ at $t=0$, these MOs evolve in time according to the time-dependent Schrödinger equation. For an SCF level of theory, this is a one-electron equation

$$\mathrm{i}\hbar\frac{d\phi_k(\mathbf{r},t)}{dt} = \hat{F}(t)\phi_k(\mathbf{r},t)\ . \tag{7.48}$$

This time evolution can equivalently be expressed in terms of the Liouville-von Neumann equation for the time evolution of the density $\rho(\mathbf{r},t)$:

$$\mathrm{i}\hbar\frac{d\rho(\mathbf{r},t)}{dt} = \left[\hat{F}(t),\ \rho(\mathbf{r},t)\right]\ . \tag{7.49}$$

In addition to obtaining broadband spectra, real-time SCF methods can be used to simulate attosecond dynamics of electrons, perhaps in the presence of strong fields. Note that the dynamics that is simulated by integrating either Eq. (7.48) or Eq. (7.49) is *electron* dynamics, the fundamental timescale of which is attoseconds, as can be estimated by the magnitude of the atomic unit of time ($\hbar/E_h \approx 2.4\times10^{-17}$ s). The finite integration time step Δt must be small compared to this value, and the default in Q-CHEM is set to $\Delta t = 0.02$ a.u. $= 4.8\times10^{-4}$ fs. The maximum timescale that can therefore reasonably be simulated is likely only picoseconds, and at present this time propagation is available only within the clamped-nuclei approximation, *i.e.*, it is not possible to simulate the couple electron–nuclear dynamics.

Because the Fock operator $\hat{F}$ depends on its own (time-evolving) eigenfunctions $\phi_i(\mathbf{r},t)$, the operator $\hat{F}(t)$ that governs the time evolution in Eq. (7.48) or Eq. (7.49) is time-dependent, which complicates the integration of these equations.[276] The simplest possible algorithm to integrate these equations (over a finite time step Δt is the *modified midpoint unitary transformation* (MMUT) procedure,[152] which approximates the operator $\hat{F}(t+\Delta t/2)$. When the MMUT algorithm is used, the cost of a single electron dynamics time step is comparable to the cost of a single SCF cycle of a ground-state SCF calculation, *i.e.*, it requires a single construction and diagonalization of the Fock matrix. The memory footprint is about twice that of the ground state, because the time-dependent MOs are complex-valued, but this is usually considerably smaller than the memory footprint for linear-response (LR-)TDDFT, especially of the number of roots requested in the LR-TDDFT calculation is large (as required for broadband spectra), or if the density of states is high (as in models of semiconductors).[184]

As compared to the first-order MMUT algorithm, higher-order predictor/corrector algorithms to integrate the dynamics are also available.[276] These algorithms enable the use of larger time steps Δt at a cost of a few Fock builds per time step. Perhaps more importantly, the predictor/corrector algorithms iterate the Fock operators $\hat{F}(t)$ and $\hat{F}(t+\Delta t)$ to self-consistency over each time step, which guarantees stable time propagation (assuming that the self-consistent procedure converges). Stable dynamics is *not* guaranteed by the MMUT algorithm, and total energy conservation turns out to be a necessary but not sufficient criterion to ensure that the trajectory has been integrated accurately. Using MMUT, examples can be found where energy is conserved yet spectra are still shifted (with respect to benchmarks results obtained using very small time steps) due to the use of a too-large value of Δt that is undetected and undiagnosed by non-self-consistent MMUT algorithm.[276]

Propagation of the electron dynamics requires only ground-state computational machinery (albeit with complex-valued orbitals), and thus is available at any SCF level of theory, including Hartree-Fock theory or DFT. The name "TDKS"

(in contrast to the cumbersome "RT-TDDFT") emphasizes that Eq. (7.48) is the Kohn-Sham analogue of the time-dependent Schrödinger equation. The cost per time step for a TDKS calculation should be no larger than a few times the cost of a ground-state SCF cycle. Q-CHEM's implementation exploits shared-memory parallelism and the use of at least 8 (but possibly more) processor cores is highly recommended, since the number of required time steps (and thus the number of Fock builds) is likely to be quite large. (The use of multiple cores is requested using the `-nt` flag.)

A TDKS calculation is requested by setting TDKS = TRUE in the *$rem* input section, and other job control options are discussed in Section 7.4.2. Section 7.4.3 discusses how to compute broadband spectra from TDKS simulations of electron dynamics. Section 7.4.4 discusses how to compute high-harmonic generation (HHG) spectra from TDKS simulations of electron dynamics.

TDKS

Job control keyword to turn on TDKS calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a TDKS calculation following a ground-state SCF calculation

FALSE Do not perform a TDKS calculation

RECOMMENDATION:

None.

7.4.2 Job Control

Only two TDKS job-control variables belong in the *$rem* section: TDKS = TRUE to request such a calculation, and (optionally) TDKS_RESTART = TRUE to resume a previous time propagation. All other job control relies on a *$tdks* input section that is described below. The propagator algorithms are described in detail in Ref. 276 and the use of TDKS to obtain broadband spectra is described in Refs. 276 and 111.

7.4.2.1 Basics: Propagators and Time Step

Assuming that TDKS_RESTART = FALSE, a standard ground-state SCF calculation is performed first in order to obtain the density at $t = 0$. For this calculation, the value of SCF_CONVERGENCE needs to be set reasonably tightly because in the subsequent TDKS calculation the density will be perturbed by an external field (to generate the time-dependent superposition state) so any convergence error in the initial density needs to be smaller than the perturbation-induced fluctuations that one is trying to integrate in the TDKS time-propagation steps. The use of incremental Fock builds (INCFOCK = TRUE) is recommended to reduce the cost of the numerous Fock builds, which are closely spaced in time. The input file for a basic TDKS propagation is illustrated in the following example. The basic job is relatively simple

but sophisticated jobs require additional input files, and generate additional output.

Example 7.31 Basic TDKS job.

```
$molecule
   0 1
   O 0.000000 0.000000 0.000000
   H 0.758602 0.000000 0.504284
   H 0.758602 0.000000 -0.504284
$end

$rem
   METHOD            pbe0
   BASIS             6-31G*
   TDKS              true
   SCF_CONVERGENCE   7
$end

$tdks
   DT             0.05
   MAXITER        10 ! for production, want a much larger value
   PROPAGATOR     MMUT
   FIELD_VECTOR   1 1 1
   FIELD_TYPE     delta
   FIELD_AMP      0.0001
$end
```

Numerous time steps are required for most practical applications of the TDKS approach and therefore a restart capability is provided. A long dynamics simulation can therefore be executed in segments (*e.g.*, to sidestep wall-time limits on shared computing resources), by setting TDKS_RESTART = TRUE in the *$rem* section. Results from the previous job are stored in the scratch directory and the next job is started by reading the data from the same directory, picking up where the previous time propagation left off. As such, the `-save` flag is required for the subsequent Q-CHEM job, and the scratch directory should be given the same name for both jobs. By default, the intermediate results are automatically stored every 1,000 steps. If the job is stopped accidentally before reaching the requested number of time steps, it can be easily be restarted from the last saved step.

TDKS_RESTART

Restart the calculation by continuing the previous job

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE The TDKS calculation continues from the previous calculation.

FALSE The TDKS calculation starts from the beginning.

RECOMMENDATION:

None.

Example 7.32 TDKS calculation illustrating restart capability.

```
$molecule
   0 1
   H    0.000000   0.000000   0.000000
   H    0.000000   0.000000   0.750000
$end

$rem
   METHOD            pbe
   BASIS             6-31G
   SCF_CONVERGENCE   9
   TDKS_RESTART      0
   TDKS              1
   INTEGRAL_SYMMETRY false
$end

$tdks
   dt                0.05
   maxiter           25
   propagator        MMUT
   field_vector      1 1 1
   field_type        delta
   field_amp         0.0001
$end

@@@

$molecule
   read
$end

$rem
   METHOD            pbe
   BASIS             6-31G
   SCF_CONVERGENCE   9
   TDKS_RESTART      1
   TDKS              1
   SCF_GUESS         read
   INTEGRAL_SYMMETRY false
$end

$tdks
   dt                0.05
   maxiter           25
   propagator        MMUT
   field_vector      1 1 1
   field_type        delta
   field_amp         0.0001
$end
```

The remaining job control parameters belong in the *$tdks* input section.

dt

The value for the time step Δt, in atomic units.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0.02

OPTIONS:

$\Delta t > 0$

RECOMMENDATION:

$\Delta t = 0.1$ a.u. for general-purpose calculation of broadband spectra at excitation energies $\hbar\omega < 30$ eV. For higher energies, $\Delta t = \pi/(10\Delta E)$ (in atomic units) is a reliable choice.

MaxIter

The max number of steps

INPUT SECTION: *$tdks*

TYPE:

INTEGER

DEFAULT:

15000

OPTIONS:

> 0

RECOMMENDATION:

The total propagation length is $\Delta t \times$ **MaxIter**.

Propagator

Time propagation algorithm

INPUT SECTION: *$tdks*

TYPE:

STRING

DEFAULT:

MMUT

OPTIONS:

EULER	Euler method
MMUT	Modified mid-point unitary transformation method
LFLPPC	Linear Fock, linear density predictor/corrector method
EPPC	Exponential density predictor/corrector method

RECOMMENDATION:

Use MMUT, LFLPPC, or EPPC. (The Euler method is not recommended.) The two predictor/corrector methods provide stable dynamics using larger time steps as compared to MMUT, and furthermore provide on-the-fly consistency checks on the stability of the dynamics, but these algorithms are more expensive than MMUT on a per-step basis.

PC_Fock_Thresh

Fock matrix threshold for consistency checking in predictor/corrector algorithms.

INPUT SECTION: *$tdks*

TYPE:

INTEGER

DEFAULT:

7 (for 10^{-7})

OPTIONS:

> 0

RECOMMENDATION:

None.

PC_Den_Thresh

Density matrix threshold for consistency checking in predictor/corrector algorithms.

INPUT SECTION: *$tdks*

TYPE:

INTEGER

DEFAULT:

7 (for 10^{-7})

OPTIONS:

> 0

RECOMMENDATION:

None.

PC_Max_Iter

Maximum number of self-consistent iterations (per time step) for predictor-corrector methods

INPUT SECTION: *$tdks*

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

> 0

RECOMMENDATION:

None.

7.4.2.2 Perturbing Field

The TDKS approach is based on applying a perturbing electric field $\boldsymbol{\mathcal{E}}$ to a (previously field-free) ground state $|\Psi_0\rangle$. The perturbation generates a superposition of all symmetry-allowed excited states $|\Psi_n\rangle$, *i.e.*, those having a non-vanishing matrix element $\langle\Psi_n|\boldsymbol{\mathcal{E}}|\Psi_0\rangle$. Several choices for the perturbing field are available, and this choice is specified by means of the **Field_Type** keyword in the *$tdks* input section. (Additional job control parameters are required for some field types, as described below.)

Field_Type

The external applied field

INPUT SECTION: *$tdks*

TYPE:

STRING

DEFAULT:

DELTA

OPTIONS:

DELTA	δ-function kick
CW	continuous-wave field
IMPULSE	impulse field (Gaussian envelope)
IMPULSE2	impulse field (cosine-squared envelope)
STATIC	static field
NONE	no field

RECOMMENDATION:

None.

These choices for **Field_Type** can be summarized as follows. Note that $\boldsymbol{\mathcal{E}} = (\mathcal{E}_x, \mathcal{E}_y, \mathcal{E}_z)$ is a vector quantity whose magnitude is controlled by **Field_Amp** and whose direction is controlled by **Field_Vector**.

- **Delta** simulates a Dirac δ-function kick, with the field $\boldsymbol{\mathcal{E}}$ turned on (at a constant amplitude) only during the first two time steps. The amplitude is controlled by the **Field_Amp** keyword that is documented below. In order to normalize across different choices of Δt, **Field_Amp** actually specifies the *integrated* field intensity $\boldsymbol{\mathcal{E}}\Delta t$, meaning that the electric field intensity itself is (**Field_Amp**)/(Δt).[111] That way, **Field_Amp** controls the total amount of energy that is put into this system by the δ-function impulse, which is the only way that results from simulations with two different time steps can be compared side-by-side.

- **CW** simulates a continuous-wave electric field of the form $\boldsymbol{\mathcal{E}}(t) = \mathbf{A}_0 \sin(\omega t)$, whose amplitude $\mathbf{A}_0$ and frequency ω are set using the keywords **Field_Amp** and **Field_Frequency**, respectively, in the *$tdks* section.

- The **Impulse** field has a Gaussian envelope,

$$\boldsymbol{\mathcal{E}}(t) = \mathbf{A}_0 \exp\left(\frac{(t - t_{\text{peak}})^2}{2\tau^2}\right) \sin(\omega t) \tag{7.50}$$

  with $\mathbf{A}_0$ and frequency ω set using **Field_Amp** and **Field_Frequency**, respectively. The field parameters τ and t_{peak} are set using **Field_Tau** and **Field_Peak**, respectively, in the *$tdks* section. The direction $\mathbf{A}_0$ is set using **Field_Vector**.

- The **Impulse2** field has a cosine-squared envelope, with the field $\boldsymbol{\mathcal{E}}$ turned on initially and turned off at $t = 2t_{\text{peak}}$. For $0 \leq t \leq 2t_{\text{peak}}$,

$$\boldsymbol{\mathcal{E}}(t) = \mathbf{A}_0 \cos^2\left[\frac{\pi}{2t_{\text{peak}}}(t_{\text{peak}} - t)\right] \cos[\omega(t - t_{\text{peak}})] \tag{7.51}$$

  with $\mathbf{A}_0$ and frequency ω set using **Field_Amp** and **Field_Frequency**, respectively. The field parameter t_{peak} is set using **Field_Peak** in the *$tdks* section. The direction $\mathbf{A}_0$ is set using **Field_Vector**.

Field_Vector

The direction of the external applied field vector

INPUT SECTION: *$tdks*

TYPE:

VECTOR

DEFAULT:

1.0 1.0 1.0

OPTIONS:

NONE

RECOMMENDATION:

NONE

Field_Amp

The amplitude of the external field (in a.u.)

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0.0001

OPTIONS:

NONE

RECOMMENDATION:

Values of 10^{-3}–10^{-4} a.u. correspond to weak fields and are appropriate for simulating absorption spectra (*i.e.*, within the linear-response regime but without the root-by-root calculation that is required for LR-TDDFT. Larger field amplitudes correspond to strong-field dynamics, as for example in second harmonic generation.

Field_Frequency

The frequency ω of the external field, in eV units.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0.001

OPTIONS:

NONE

RECOMMENDATION:

The use of energy units is for convenience when using the TDKS to generate broadband spectra.

Field_Peak

The peak position t_{peak} (in a.u. of time) for the Gaussian or cosine-squared impulse field.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0.0

OPTIONS:

NONE

RECOMMENDATION:

NONE

Field_Tau

The value of τ (in a.u. of time) for the Gaussian impulse field.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0.7

OPTIONS:

NONE

RECOMMENDATION:

NONE

7.4.2.3 Complex Absorbing Potential

Simulations of broadband spectra at high energies (*e.g.*, for X-ray absorption spectroscopy at the K-edge, corresponding to 1s $\rightarrow$ virtual excitations), the requisite electron dynamics often corresponds to fluctuations that take the density out to the edge of the region of space that is spanned by the Gaussian basis set. The edge of the basis set imposes an artificial potential wall, and a time-dependent wave packet can reflect off of this wall and then interfere with its own outgoing wave, in an artificial manner that simply reflects the finite-basis approximation.[277] In practice, this can introduce artificial oscillations into broadband spectra obtained from TDKS calculations. The same is true in TDKS simulations of strong-field electron dynamics, such as high harmonic generation.[277] These unwanted artifacts can be removed by the introduction of a complex absorbing potential (CAP) that annihilates the outgoing wave that encounters it.

For TDKS calculations, the CAP that is available is constructed from a sum of spherical, overlapping atom-centered potentials.[277] Within each of these, the potential is zero within a cutoff radius r_0 around $\mathbf{R}_k$, the position of atom k. Outside of that radius, the potential rises quadratically with curvature η. This is implemented using a set of atom-centered CAP functions

$$f_k^{\text{CAP}}(\mathbf{r}) = \begin{cases} 0, & \|\mathbf{r}-\mathbf{R}_k\| < r_0 \\ \eta\big(\|\mathbf{r}-\mathbf{R}_k\| - r_0\big)^2, & \|\mathbf{r}-\mathbf{R}_k\| \geq r_0 \end{cases} . \tag{7.52}$$

At any point $\mathbf{r}$ in space, the value of the CAP is taken to be $\mathrm{i}\upsilon_{\text{CAP}}(\mathbf{r})$, where $\upsilon_{\text{CAP}}(\mathbf{r})$ is the minimum of the various atom-centered potentials and a cutoff $E_{\text{max}} = 10\ E_h$, the latter of which is used to avoid numerical overflow problems. All together,

$$\upsilon_{\text{CAP}}(\mathbf{r}) = \min\left\{E_{\text{max}}, f_1^{\text{CAP}}(\mathbf{r}), f_2^{\text{CAP}}(\mathbf{r}), \ldots,\right\} . \tag{7.53}$$

The value of r_0 is user-specifiable and should probably be tested for specific applications. (For simulating strong-field ionization dynamics, a value $r_0 = 3.5 \times R_{\text{vdW}}$ has been used,[135] where R_{vdW} is a representative atomic van der Waals

radius.) Note that placing r_0 beyond the extent of the Gaussian basis functions themselves will have no effect. For a Gaussian basis function with exponent ζ, the full width at half maximum of that function is[104]

$$\mathrm{FWHM}(\zeta) = 2\left(\frac{\ln 2}{\zeta}\right)^{1/2}, \tag{7.54}$$

and half that value is therefore a measure of the radial extent of the basis function in question. This can be used to estimate the spatial extent of the basis, taking ζ to be the smallest (most diffuse) exponent.[104] Note that when basis function information is requested using PRINT_GENERAL_BASIS = TRUE, the exponents ζ are printed in atomic units of a_0^{-2}.

The following variables in the *$tdks* input section control the use of a CAP in TDKS calculations.

Do_CAP

Activate a complex absorbing potential for TDKS calculations

INPUT SECTION: *$tdks*

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

TRUE Use a CAP

FALSE Do not use a CAP

RECOMMENDATION:

None.

CAP_R0

Cutoff radius for the CAP (in a.u.)

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0

OPTIONS:

> 0

RECOMMENDATION:

A value greater than twice the largest atomic van der Waals radius is recommended and should be tested as needed.

CAP_Eta

Specifies the curvature of the CAP

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

1.0

OPTIONS:

> 0

RECOMMENDATION:

Values of 4.0–5.0 are typical but testing is recommended.

7.4.3 Calculation of Absorption Spectra

7.4.3.1 Theory

The absorption cross-section $\sigma_{\kappa\kappa}(\omega)$ for light polarized in the κ direction ($\kappa \in \{x, y, z\}$) can be obtained from the imaginary part ($\Im$) of the frequency-dependent polarizability, $\alpha_{\kappa\kappa}(\omega)$:[111,276]

$$\sigma_{\kappa\kappa}(\omega) = \left(\frac{4\pi\omega}{c}\right) \Im\big[\alpha_{\kappa\kappa}(\omega)\big] \ . \tag{7.55}$$

Within the electric dipole approximation, a rotationally-averaged absorption spectrum $A(\omega)$ is then simply

$$A(\omega) = \tfrac{1}{3}\big[\sigma_{xx}(\omega) + \sigma_{yy}(\omega) + \sigma_{zz}(\omega)\big] \ . \tag{7.56}$$

Components $\alpha_{\kappa\lambda}(\omega)$ of the frequency-dependent polarizability tensor $\boldsymbol{\alpha}(\omega)$ are obtained from the Fourier transform ($\mathcal{F}$) of the time-dependent dipole moment component $\mu_\kappa(t)$, for a perturbing field $\mathcal{E}_\lambda$ in the λ direction:[276]

$$\alpha_{\kappa\lambda}(\omega) = \frac{\mathcal{F}\big[\mu_\kappa(t)\big]}{\mathcal{F}\big[\mathcal{E}_\lambda(t)\big]} \ . \tag{7.57}$$

To compute the spectrum in Eq. (7.56), three separate perturbations in the x, y, and z directions are required, else some excitations may be missing if their transition moment is strictly perpendicular to the applied field, causing the matrix element $\langle\Psi_n|\mathcal{E}_\lambda|\Psi_0\rangle$ to vanish. However, these three perturbations $\mathcal{E}_x$, $\mathcal{E}_y$, and $\mathcal{E}_z$ can be applied all at once in a single calculation, in order to generate a superposition consisting of all possible excitations out of the ground state.

7.4.3.2 Fourier Transform Scripts

Two different scripts are provided to obtain the spectrum after the TDKS simulation is completed:

- `$QC/bin/tools/tdks_fft.py`
- `$QC/bin/tools/tdks_pade.py`

The first of these uses the Fourier transform method in Eq. (7.57) directly while the second makes use of Padé approximants to obtain comparable spectra with shorter propagation times.[111] The scripts can be run as follows:

```
$QC/bin/tools/tdks_fft.py output spectrum.txt
$QC/bin/tools/tdks_pade.py output spectrum.txt
```

The file `spectrum.txt` produced by the processing script will contain two columns: frequency (eV) and strength

(arbitrary units). These data can be visualized as an (x, y) plot to view the spectrum.

Example 7.33 TDKS job using a CW field and a CAP.

```
$molecule
   0 1
   H    0.000000   0.000000   0.000000
   H    0.000000   0.000000   0.750000
$end

$rem
   BASIS                6-31G
   METHOD               lrc-wpbe
   TDKS                 true
   LRC_DFT              true
   OMEGA                300
   SCF_CONVERGENCE      9
   INTEGRAL_SYMMETRY    false
$end

$tdks
   DT                   0.10
   MAXITER              5
   PROPAGATOR           MMUT
   FIELD_VECTOR         1 1 1
   FIELD_TYPE           cw
   FIELD_FREQUENCY      1.55
   FIELD_AMP            0.0001
   DO_CAP               true
   CAP_TYPE             atom_centered_spherical
   CAP_R0               18.5  ! units are bohr
   CAP_ETA              0.1   ! units are hartree/bohr^2
$end
```

7.4.3.3 Dipole Filtering

An important but not-well-recognized aspect of TDKS broadband spectrum calculations is that they may be contaminated by continuum artifacts corresponding to transitions to high-energy virtual MOs.[111,270] This can be especially problematic for core-level spectra, where one X-ray edge may exhibit artifacts arising from a lower-energy edge. For example, N(1s) $\rightarrow$ continuum transitions can appear at energies comparable to the oxygen K-edge consisting of O(1s) $\rightarrow$ valence transitions.[111] This is unfortunate, since X-ray spectroscopy is a potentially useful application of the real-time approach, whereas a state-by-state approach (as in LR-TDDFT or any other eigenvalue-based method) may require the calculation of hundreds of individual states in order to obtain a converged spectrum. The TDKS approach is almost always (significantly) more expensive than LR-TDDFT,[111] but the former lacks the large memory bottleneck that is incurred by the latter.

Fortunately, artificial states appearing in the TDKS spectrum can be eliminated by means of a dipole-filtering technique,[111,270] although this does require some advance knowledge of what transitions to expect. The key is to recognize that the dipole moment is a linear function of the density matrix,

$$\mu_\lambda(t) = \mathrm{tr}\big[\mathbf{P}(t)\,\mathbf{D}_\lambda(t)\big]\ . \tag{7.58}$$

Here, $\lambda \in \{x, y, z\}$ represents a Cartesian component of the dipole moment and $\mathbf{D}_\lambda$ is the matrix representation of the dipole moment operator $\hat{\mu}_\lambda$, expressed in the MO basis. The sum over occupied MOs k allows us to decompose μ_λ

into a separate contribution from each. To do this, define a matrix

$$\mathbf{D}_{\lambda,k} = \begin{pmatrix} 0 & \cdots & (\mathbf{D}_\lambda)_{1,k} & \cdots & 0 \\ \vdots & & \vdots & & \vdots \\ (\mathbf{D}_\lambda)_{k,1} & \cdots & (\mathbf{D}_\lambda)_{k,k} & \cdots (\mathbf{D}_\lambda)_{k,n_{\text{occ}}} & \\ \vdots & & \vdots & & \vdots \\ 0 & \cdots & (\mathbf{D}_\lambda)_{n_{\text{occ}},k} & \cdots & 0 \end{pmatrix} \tag{7.59}$$

in which all but the kth row and column of $\mathbf{D}_\lambda$ have been set to zero. Then

$$\mu_\lambda(t) = \sum_k^{\text{occ}} \mu_{\lambda,k}(t) \tag{7.60}$$

where

$$\mu_{\lambda,k}(t) = \text{tr}\big[\mathbf{P}(t)\ \mathbf{D}_{\lambda,k}(t)\big]\ . \tag{7.61}$$

Taking the Fourier transform of the time series $\mu_{\lambda,k}(t)$ provides the spectral contribution of the kth MO, and the sum of all such contributions affords the full TDKS spectrum. This need not be done on an orbital-by-orbital basis; instead, contributions from groups of MOs could be summed together, representing various partitions of the summation in Eq. (7.60). An oxygen K-edge spectrum that is free of contamination from N(1s) $\rightarrow$ continuum transitions could be obtained by only allowing the sum over k to include the O(1s) orbital(s).

The aforementioned technique is a way of decomposing the full TDKS spectrum into contributions from individual occupied MOs or groups thereof. Arbitrary groupings can be computed, although only when those groups of MOs are non-disjoint will the Fourier transforms sum to the full TDKS spectrum. In this way, the effects of different occupied MOs can be deleted from the spectrum, both to look for (and remove) artifacts, but also to assign spectral features by determining the occupied MOs from which they arise.

Dipole filtering is accomplished by adding the keyword **active_groups** to the *$tdks* input section. An example of the format is

```
active_groups  k1 k2 k3 | k4 k5 | k6 k7 k8
```

This requests a decomposition into three groups of occupied MOs, with indices $\{k_1, k_2, k_3\}$ for the first group, $\{k_4, k_5\}$ for the second group, and $\{k_6, k_7, k_8\}$ for the third. (Group indices are separated by the "|" symbol.) For unrestricted calculations, the α- and β-spin MO indices for the active groups must be listed separately, using keywords **active_alpha_groups** and **active_beta_groups** that both have the same format as **active_groups**.

active_groups

Specify groups of active MO indices for dipole filtering.

INPUT SECTION: *$tdks*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

$k_1\ k_2 \cdots | k_3\ k_4 \cdots$ Define groups $\{k_1, k_2, \ldots\}$ and $\{k_3, k_4, \ldots\}$.

RECOMMENDATION:

Use if desired. In addition to the full dipole moment function $\boldsymbol{\mu}(t)$, separate dipole moment functions for each group will be written to the scratch directory. For unrestricted SCF calculations, use **active_alpha_groups** and **active_beta_groups** instead.

The following example demonstrates how to decompose the absorption spectrum for TiO_2 into contributions from Ti(1s), Ti(2s), O(1s), and Ti(2p) orbitals. In order to setup this job, one must have performed a prior SCF calculation to ascertain which indices correspond to the MOs of interest.

Example 7.34 Example of dipole filtering for TDKS. Note that the propagation time (**maxiter**) would need to be much longer for a real simulation.

```
$comment
TDKS with dipole filtering
Decompose the spectrum into contributions from active groups of occupied MOs,
   1    |    2    |  3 4  |  5 6 7
 Ti(1s)    Ti(2s)   O(1s)   Ti(2p)
$end

$molecule
0 1
Ti  0.0000000 0.000000  0.6131572
O  -1.3586178 0.000000 -0.3066726
O   1.3586178 0.000000 -0.3066726
$end

$rem
exchange src1-r1
basis    def2-tzvpd
tdks     true
thresh   12
scf_convergence 8
sym_ignore      true
$end

$tdks
active_groups  1 | 2 | 3 4 | 5 6 7
dt 0.02
maxiter 10
Propagator MMUT
field_vector 1 1 1
field_type delta
field_amp 0.0001
$end
```

The relevant output from a TDKS calculation with dipole filtering is placed into the scratch directory, in a subdirectory named `$QCSCRATCH/tdks-dipole`. Upon completion of the calculation that director is copied into the same directory that contains the input and output files, and is renamed as `outFileName.tdks-dipole` where `outFileName` is the Q-CHEM output file. For restricted SCF calculations, the `tdks-dipole` directory contains a file named `full.dat` consisting of the full dipole moment function, arranged as components $\mu_x(t)$, $\mu_y(t)$, and $\mu_z(t)$. In the same directory are files named `group.n.dat` that contain the same information but restricted to the nth group of MOs. For unrestricted SCF calculations, `full.dat` is replaced by `full-alpha.dat` and `full-beta.dat`, while `group.n.dat` is replaced by `group-alpha.n.dat` and `group-beta.n.dat`. The data in these `.dat` files can be individually Fourier transformed to obtain spectral contributions from each group of MOs. If there are no omissions and no redundancies in the list of MOs, then the group spectra should sum to the full spectrum. For example, `full-alpha.dat` and `full-beta.dat` together provide the full spectrum for an unrestricted TDKS calculation.

7.4.4 Calculation of High-Harmonic Generation (HHG) Spectra

The high-harmonic generation (HHG) spectrum $H(\omega)$ in the dipole acceleration form is calculated by[13,52,254]

$$H(\omega) = \sum_{\kappa=x,y,z} \frac{1}{2\pi} \left| \int \frac{d^2\mu_\kappa(t)}{dt^2} w(t) e^{-i\omega t} dt \right|^2 \tag{7.62}$$

where $w(t)$ is some kind of window function to improve spectrum quality, and $\mu_\kappa(t)$ is the time-dependent dipole moment component. For light polarized in the κ direction ($\kappa \in \{x, y, z\}$), we have $\mu_\lambda(t) = 0$ ($\lambda \in \{x, y, z\}, \lambda \neq \kappa$), and Eq. (7.62) becomes

$$H(\omega) = \frac{1}{2\pi} \left| \int \frac{d^2\mu_\kappa(t)}{dt^2} w(t) e^{-i\omega t} dt \right|^2 \tag{7.63}$$

The incorporation of a complex absorbing potential (CAP) is frequently preferred to mitigate artifacts arising from the finite-basis approximation, see Section 7.4.2.3.

A script is provided to obtain the spectrum after the TDKS simulation is completed:

- `$QC/bin/tools/tdks_hhg.py`

This uses Eq. (7.63) with $w(t)$ taken to be the Hamming window function.[277] For $\kappa = z$, the script can be run as follows:

```
python3 $QC/bin/tools/tdks_hhg.py z output spectrum.txt
```

The file `spectrum.txt` produced by the processing script will contain two columns: harmonic order and logarithmic strength (arbitrary units). The harmonic order is ω divided by FIELD_FREQUENCY, and the logarithmic strength is $\log[H(\omega)]$. These data can be visualized as an (x, y) plot to view the spectrum. Users of Q-CHEM's TDKS code for HHG spectra are asked to cite Ref. 277.

7.4.5 Real-Time Extension of TAO-DFT (RT-TAO)

In order to investigate the time-dependent dynamics of large, strongly correlated systems starting from the ground state, and with a focus on minimizing computational complexity, Hung-Yi Tsai and Prof. Jeng-Da Chai have extended thermally-assisted-occupation density functional theory (TAO-DFT)[48] to real-time TAO-DFT (RT-TAO).[253] The RT-TAO method is initiated with a TAO-DFT calculation (refer to Section 5.12.3) and is followed by time propagation (refer to Section 7.4). The computational cost associated with RT-TAO is comparable to that of time-dependent Kohn-Sham, and it becomes equivalent to the cost of time-dependent Kohn-Sham in the absence of significant static correlation effects.

To perform an RT-TAO calculation, users are instructed to set TAO_DFT = TRUE and TDKS = TRUE in the *$rem* input section. Additional job control options are detailed in Section 5.12.3 and Section 7.4.2.

RT-TAO, serving as a real-time extension of TAO-DFT, is governed by a fictitious temperature parameter denoted as θ. Setting TAO_DFT_THETA = 0 brings the method back to the ordinary time-dependent Kohn-Sham.[253] In the adiabatic approximation, alongside the exchange-correlation (XC) functional, an additional functional $E_\theta[\rho]$ is required for RT-TAO.

Two illustrative examples of RT-TAO calculations are provided:

Example 7.35 RT-TAO-ALDA, spin-restricted calculation on stretched H_2

```
$molecule
   0 1
   H 0.000000 0.000000 1.000000
   H 0.000000 0.000000 -1.000000
$end

$rem
   BASIS               6-31G*
   EXCHANGE            gen
   TDKS                true
   SCF_CONVERGENCE     7
   TAO_DFT             true
   TAO_DFT_THETA       40   ! theta = 40 mhartree
   TAO_DFT_THETA_NDP   3    ! default
$end

$xc_functional
   X   S             1.0
   C   PW92          1.0
   X   ETheta_LDA    1.0
$end

$tdks
   DT             0.05
   MAXITER        10 ! for production, want a much larger value
   PROPAGATOR     MMUT
   FIELD_VECTOR   1 1 1
   FIELD_TYPE     delta
   FIELD_AMP      0.0001
$end
```

Example 7.36 RT-TAO-ALDA, spin-unrestricted calculation on stretched H_2

```
$molecule
   0 1
   H 0.000000 0.000000 1.000000
   H 0.000000 0.000000 -1.000000
$end

$rem
   UNRESTRICTED        true
   BASIS               6-31G*
   EXCHANGE            gen
   TDKS                true
   SCF_CONVERGENCE     7
   TAO_DFT             true
   TAO_DFT_THETA       40    ! theta = 40 mhartree
   TAO_DFT_THETA_NDP   3     ! default
   SCF_GUESS           gwh
   SCF_GUESS_MIX       3     ! mix in 30% LUMO in alpha to break symmetry
$end

$xc_functional
   X   S             1.0
   C   PW92          1.0
   X   ETheta_LDA    1.0
$end

$tdks
   DT             0.05
   MAXITER        10 ! for production, want a much larger value
   PROPAGATOR     MMUT
   FIELD_VECTOR   1 1 1
   FIELD_TYPE     delta
   FIELD_AMP      0.0001
$end
```

7.5 Non-Orthogonal Configuration Interaction (NOCI)

7.5.1 Introduction

Systems such as transition metals, open-shell species, and molecules with highly-stretched bonds often exhibit multiple, near-degenerate solutions to the SCF equations. Multiple solutions can be located using SCF metadynamics (Section 4.9.2), but given the approximate nature of the SCF calculation in the first place, there is in such cases no clear reason to choose one of these solutions over another. These SCF solutions are not subject to any non-crossing rule, and often do cross (*i.e.*, switch energetic order) as the geometry is changed. Consequently, the lowest energy state may switch abruptly with consequent discontinuities in the energy gradients. It is therefore desirable to have a method that treats all of these near-degenerate SCF solutions on an equal footing and might yield a smoother, qualitatively correct potential energy surface. This can be achieved by using multiple SCF solutions (obtained, *via e.g.* SCF metadynamics) as a basis for a configuration interaction (CI) calculation. Since the various SCF solutions are not orthogonal to one another — meaning that one solution cannot be constructed as a single determinant composed of orbitals from another solution — this CI is a bit more complicated and is denoted as a non-orthogonal CI (NOCI).[246]

NOCI can be viewed as an alternative to CASSCF within an "active space" consisting of the SCF states of interest, and has the advantage that the SCF states, and thus the NOCI wave functions, are size-consistent. In common with

CASSCF, it is able to describe complicated phenomena such as avoided crossings (where states mix instead of passing through each other) as well as conical intersections (whereby via symmetry or else accidental reasons, there is no coupling between the states, and they pass cleanly through each other at a degeneracy).

Another use for a NOCI calculation is that of symmetry restoration. At some geometries, the SCF states break spatial or spin symmetry to achieve a lower energy single determinant than if these symmetries were conserved. As these symmetries still exist within the proper electronic Hamiltonian, its exact eigenfunctions should preserve them. In the case of spin, this manifests as spin contamination, while for spatial symmetries it usually manifests as artefactual localization. To recover a (yet lower energy) wave function retaining the correct symmetries, one can include these broken-symmetry states (with all relevant symmetry permutations) in a NOCI calculation; the resultant eigenfunction will have the true symmetries restored, as a linear combination of the broken-symmetry states.

A common example occurs in the case of a spin-contaminated UHF reference state. Performing a NOCI calculation in a basis consisting of this state, plus a second state in which all α and β orbitals have been switched, often reduces spin contamination in the same way as the half-projected Hartree-Fock method,[178] although there is no guarantee that the resulting wave function is an eigenfunction of $\hat{S}^2$. Another example consists of using a UHF wave function with $M_S = 0$, along with its spin-exchanged version (wherein all $\alpha \leftrightarrow \beta$ orbitals are switched), resulting in two new NOCI eigenfunctions, one with even S (a mixture of $S = 0, 2, \ldots$), and one with odd S (mixing $S = 1, 3, \ldots$). These may be used to approximate singlet and triplet wave functions.

NOCI can be enabled by specifying CORRELATION = NOCI, and will automatically use all of the states located with SCF metadynamics. Two spin-exchanged versions of a UHF wave function can be requested simply by *not* turning on metadynamics. For more customization, a *$noci* input section can be included having, *e.g.*, the following format:

```
$noci
   1 2 -2 4
   2
$end
```

In this particular case, the first line specifies that states 1, 2, and 4 are to be included in the NOCI calculation, along with state "-2", which indicates the spin-exchanged version of state 2. The second (optional) line indicates which eigenvalue is to be returned to Q-CHEM, with the convention that 0 indicates the lowest state so the *$noci* input section above is requesting the third state.

Analytic gradients are not available for NOCI but geometry optimizations (if requested) will be performed automatically using finite-difference gradients.

7.5.2 Job Control

Q-CHEM 5.2 features a new package LIBNOCI dedicated to running NOCI calculations, and accessed using USE_LIBNOCI = TRUE. The LIBNOCI implementation introduces flexible job control, whereby the method used to generate multiple states for the NOCI basis can be defined by the user. Initially, a set of reference determinants must be chosen, either using a single SCF calculation or by reading sets or orbital coefficients from file as requested by the *$rem* variable NOCI_REFGEN. From these reference states, multiple non-orthogonal states are then created either using SCF metadynamics, or as excitations from the reference determinants, as requested by the *$rem* variable NOCI_DETGEN. When generating multiple using excitations within an active space, the active orbitals are controlled using the *$active_orbitals*

input section. For example:

```
$active_orbitals
    1 4 11 14
$end
```

The β orbitals are offset by the total number of α molecular orbitals (*e.g.* the above example selects α orbitals 1 and 4, along with β orbitals 1 and 4, with a total of 10 α MOs including occupied and virtual orbitals). The resulting multiple determinants are then individually relaxed at the SCF level, unless this is explicitly skipped using SKIP_SCFMAN = TRUE. Finally, NOCI is run using the full set of multiple determinants identified.

The LIBNOCI package also incorporates compatibility with the new SCF metadynamics implementation (see Section 4.9.3), as well as the holomorphic Hartree–Fock approach which allows multiple Hartree–Fock solutions to be continued across all geometries (see Section. 4.9.4).

USE_LIBNOCI

Turn on the use of LIBNOCI for running NOCI calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

False Do not use LIBNOCI (uses original Q-CHEM implementation).

True Use the LIBNOCI implementation.

RECOMMENDATION:

The *$rem* variables detailed below are only available in LIBNOCI.

NOCI_REFGEN

Control how the initial reference determinants are created.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Generate initial reference determinant from a single SCF calculation.

1 Read (multiple) initial reference determinants from a previous calculation.

RECOMMENDATION:

The specific reference determinants to be read from a previous calculation can be indicated using SCF_READMINIMA.

NOCI_DETGEN

Control how the multiple determinants for NOCI are created.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use only the initial reference determinants.

1 Generate CIS excitations from each reference determinant.

2 Generate all FCI excitations from each reference determinant.

3 Generate n multiple determinants using SCF metadynamics, where n is specified using SCF_SAVEMINIMA = n.

4 Generate all CAS excitations from each reference determinant, where the active orbitals are specified using the *$active_orbitals* input section.

RECOMMENDATION:

By default, these multiple determinants are optimized at the SCF level before running NOCI. This behavior can be turned off using by specifying SKIP_SCFMAN = TRUE.

NOCI_NEIGVAL

The number of NOCI eigenvalues to be printed.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Positive integer

RECOMMENDATION:

Increase this to print progressively higher NOCI energies.

NOCI_PRINT

Specify the debug print level of NOCI.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Positive integer

RECOMMENDATION:

Increase this for additional debug information.

Example 7.37 NOCI for H_2 run in the LIBNOCI implementation, with multiple determinants generated using SCF metadynamics.

```
$molecule
   0 1
   H     0.0000000   0.0000000   0.0000000
   H     0.0000000   0.0000000   4.0000000
$end

$rem
   EXCHANGE                   hf
   CORRELATION                noci
   UNRESTRICTED               true
   BASIS                      sto-3g
   SCF_CONVERGENCE            10
   MAX_SCF_CYCLES             1000
   MOM_START                  1
   USE_LIBNOCI                true
   SCF_SAVEMINIMA             4
   SCF_MINFIND_RANDOMMIXING   30000
   SCF_MINFIND_MIXMETHOD      1
   NOCI_REFGEN                0
   NOCI_DETGEN                3
   NOCI_NEIGVAL               4
$end
```

7.6 Maximum Overlap Method (MOM) for ΔSCF Excited States

The Maximum Overlap Method (MOM) is a useful alternative to CIS and TDDFT for obtaining low-cost excited states.[79] It works by modifying the orbital selection step in the SCF procedure. By choosing orbitals that most resemble those from the previous cycle, rather than those with the lowest eigenvalues, non-*aufbau*, excited-state SCF determinants can be determined, in what has sometimes been called *excited-state Kohn-Sham theory*.[94,106] This represents a form of "ΔSCF" approach to computing excitation energies,[106] which has advantages over TDDFT is certain cases. For example, TDDFT exhibits systemic problems with the description of charge-transfer and Rydberg excitations,[106] both of which can be modeled using the ΔSCF approach. The use of MOM also allows the user to easily target very high energy states, such as those involving excitation of core electrons,[24] which can be difficult to capture using other excited state methods. Other ΔSCF approaches are described in Section 7.8.

In order to calculate an excited state using MOM, the user must correctly identify the orbitals involved in the transition. For example, in a $\pi \rightarrow \pi^*$ transition, the π and π^* orbitals must be identified and this usually requires a preliminary calculation. The user then manipulates the orbital occupancies using the *$occupied* section, removing an electron from the π and placing it in the π^*. The MOM is then invoked to preserve this orbital occupancy. The success of the MOM relies on the quality of the initial guess for the calculation. If the virtual orbitals are of poor quality then the calculation may 'fall down' to a lower energy state of the same symmetry. Often the virtual orbitals of the corresponding cation are more appropriate for using as initial guess orbitals for the excited state.

Because the MOM states are single determinants, all of Q-CHEM's existing single determinant properties and derivatives are available. This allows, for example, analytic harmonic frequencies to be computed on excited states. The orbitals from a Hartree-Fock MOM calculation can also be used in an MP2 calculation. For all excited state calculations, it is important to add diffuse functions to the basis set. This is particularly true if Rydberg transitions are being sought. For DFT based methods, it is also advisable to increase the size of the quadrature grid so that the more diffuse densities are accurately integrated.

Due to its single-determinant description, the MOM algorithm for unrestricted ΔSCF calculations of open-shell singlets is often badly spin contaminated. Spin contamination can be alleviated by resorting to a restricted open-shell Kohn-Sham (ROKS) formalism (see Section 7.8.3), in which the objective function $E_S = 2E_M - E_T$ is optimized, where E_S is the spin-pure singlet energy, E_M is the energy of the "mixed" (or spin-contaminated) open-shell singlet state, and E_T is the triplet energy. MOM can be used in conjunction with ROKS to evaluate a spin-pure energy for arbitrary open-shell singlet electron configurations (formerly only S_1 was accessible within the ROKS formulation).

The MOM-based ΔSCF method can be combined with a PCM based solvation description for both equilibrium and nonequilibrium effects (for more details see Section 7.8.6).

The following *$rem* is used to invoke the MOM:

MOM_START

Determines when MOM is switched on to preserve orbital occupancies.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 (FALSE)	MOM is not used
n	MOM begins on cycle n.

RECOMMENDATION:

For calculations on excited states, an initial calculation without MOM is usually required to get satisfactory starting orbitals. These orbitals should be read in setting SCF_GUESS = TRUE and MOM_START = 1.

MOM_METHOD

Determines the target orbitals with which to maximize the overlap on each SCF cycle.

TYPE:

INTEGER

DEFAULT:

MOM

OPTIONS:

MOM	Maximize overlap with the orbitals from the previous SCF cycle.
IMOM	Maximize overlap with the initial guess orbitals.

RECOMMENDATION:

If appropriate guess orbitals can be obtained, then IMOM can provide more reliable convergence to the desired solution.[10]

Example 7.38 Calculation of the lowest singlet state of CO.

```
$comment
   CO spin-purified calculation
   Step 1: prepare MOs
$end

$molecule
   0 1
   C
   O  C  1.05
$end

$rem
   METHOD            B3LYP
   BASIS             6-31G*
$end

@@@

$comment
  Step 2: spin purification (OPSING=TRUE)
$end

$molecule
   read
$end

$rem
   METHOD            B3LYP
   BASIS             6-31G*
   SCF_GUESS         read
   MOM_START         1
   UNRESTRICTED      true
   OPSING            true
$end

$occupied
   1 2 3 4 5 6 7
   1 2 3 4 5 6 8
$end
```

Example 7.39 Input for obtaining the ^{2}A′ excited state of formamide corresponding to the $\pi \rightarrow \pi^*$ transition. The ^{1}A′ ground state is obtained if MOM is not used in the second calculation. Note the use of diffuse functions and a larger quadrature grid to accurately model the larger excited state.

```
$molecule
   1  2
   C
   H  1  1.091480
   O  1  1.214713  2  123.10
   N  1  1.359042  2  111.98  3  -180.00
   H  4  0.996369  1  121.06  2    -0.00
   H  4  0.998965  1  119.25  2  -180.00
$end

$rem
   METHOD          B3LYP
   BASIS           6-311(2+,2+)G(d,p)
   XC_GRID         000100000194
$end

@@@

$molecule
   0  1
   C
   H  1  1.091480
   O  1  1.214713  2  123.10
   N  1  1.359042  2  111.98  3  -180.00
   H  4  0.996369  1  121.06  2    -0.00
   H  4  0.998965  1  119.25  2  -180.00
$end

$rem
   METHOD          B3LYP
   BASIS           6-311(2+,2+)G(d,p)
   XC_GRID         000100000194
   MOM_START       1
   SCF_GUESS       read
   UNRESTRICTED    true
$end

$occupied
   1:12
   1:11 13
$end
```

Additionally, it is possible to perform a CIS/TDDFT calculation on top of the MOM excitation. This capability can be useful when modeling pump-probe spectra. In order to run MOM followed by CIS/TDDFT, the *$rem* variable

CIS_N_ROOTS must be specified. Truncated subspaces may also be used, see Section 7.3.3.

Example 7.40 MOM valence excitation followed by core-state TDDFT using a restricted subspace

```
$molecule
   0 1
   O    0.0000    0.0000    0.1168
   H    0.0000    0.7629   -0.4672
   H    0.0000   -0.7629   -0.4672
$end

$rem
   METHOD       B3LYP
   BASIS        aug-cc-pvdz
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
   read
$end

$rem
   METHOD         B3LYP
   BASIS          aug-cc-pvdz
   SCF_GUESS      read
   MOM_START      1
   UNRESTRICTED   true
   CIS_N_ROOTS    5
   TRNSS          true  ! use truncated subspace for TDDFT
   TRTYPE         3     ! specify occupied orbitals
   CUTVIR         15    ! truncate high energy virtual orbitals
   N_SOL          1     ! number core orbitals, specified in $solute section
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end

$solute
   1
$end

$occupied
   1 2 3 4 5
   1 2 3 4 6
$end
```

If the MOM excitation corresponds to a core hole, a reduced subspace must be used to avoid de-excitations to the core hole. The *$rem* variable CORE_IONIZE allows only the hole to be specified so that not all occupied orbitals need to be entered in the *$solute* section.

CORE_IONIZE

Indicates how orbitals are specified for reduced excitation spaces.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 all valence orbitals are listed in *$solute* section

2 only hole(s) are specified all other occupations same as ground state

RECOMMENDATION:

For MOM + TDDFT this specifies the input form of the *$solute* section. If set to 1 all occupied orbitals must be specified, 2 only the empty orbitals to ignore must be specified.

Example 7.41 O(1s) core excited state using MOM followed by excitations among valence orbitals. Note that a reduced excitation subspace must be used to avoid "excitations" into the empty core hole

```
$molecule
   0 1
   O   0.0000   0.0000   0.1168
   H   0.0000   0.7629  -0.4672
   H   0.0000  -0.7629  -0.4672
$end

$rem
   METHOD       B3LYP
   BASIS        aug-cc-pvdz
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
   read
$end

$rem
   METHOD            B3LYP
   BASIS             aug-cc-pvdz
   SCF_GUESS         read
   MOM_START         1
   UNRESTRICTED      true
   CIS_N_ROOTS       5
   TRNSS             true  ! use truncated subspace for TDDFT
   TRTYPE            3     ! specify occupied orbitals
   N_SOL             1     ! number core holes, specified in $solute section
   CORE_IONIZE       2     ! hole orbital specified
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

$solute
   6
$end

$occupied
   1 2 3 4 5
   2 3 4 5 6
$end
```

7.7 Non-Orthogonal CIS and Static Exchange (STEX)

7.7.1 Non-Orthogonal CIS (NOCIS)

The motivation for the NOCIS method[195–197] is the desire to improve on CIS while still maintaining a reasonably low computational scaling. It does so by including orbital relaxation, which CIS neglects altogether, and the non-orthogonal interaction between multiple core-hole references, such as the O $1s$ orbitals in O_2.

A brief overview of the NOCIS algorithm is as follows: following a ground-state orbital optimization, an SCF calculation is performed using the maximum overlap method (MOM, see Section 4.5.14) to compute an ionization from each localized core orbital of interest. This introduces orbital relaxation, and also renders the excited states non-orthogonal to the ground state. The Hamiltonian, overlap, and total spin squared matrices are constructed using the Slater-Condon rules for matrix elements between determinants which share a common set of orbitals and NOCI for the remaining matrix elements.[246] Finally, the generalized eigenvalue problem is solved.

A key feature in open-shell NOCIS is a separate optimization of any open-shell references, which are states in which a core-electron is excited to a singly-occupied ground-state orbital. These separate optimizations render these states non-orthogonal to the other excited states.

NOCIS is spin-pure, size consistent, and maintains spatial symmetry. Like CIS, NOCIS produces excited states with the same m_s as the reference but potentially with larger total spin. For example, performing NOCIS on a molecule with a singlet ground state will produce both singlet and triplet excited states.

7.7.2 Static Exchange

In the simplest form of STEX,[119] one computes excitation energies by diagonalizing the virtual space of a Fock matrix for the ionized system. When using STEX on multiple core-ionized states,[4] the underlying assumptions are that the coupling between non-orthogonal determinants is negligible, and that all relevant excited determinants can be formed by single electron-attachment to the core-ionized state. In this way, STEX can be considered to be an approximation to NOCIS.

The STEX algorithm is very similar to NOCIS. The ground-state calculation includes the Boys localization of the reference orbitals before the MOM. However, the open-shell references are formed from the core-ionized reference, instead of optimizing them separately, rendering these states orthogonal to the rest of the core-excitations from that particular atom. After the matrix build, the orthogonal matrix blocks are projected against the ground state (contrasted with NOCIS, where the whole matrix is projected against the ground state), the eigenvalue problem is solved. Because the basis of excited determinants is not orthogonal to the ground state, NOCI is required to compute the oscillator strengths.

Like NOCIS, STEX is spin-pure and size-consistent. However, due to the de-coupling of the references, STEX calculations break the spatial symmetry of the final states.

7.7.3 One-Center NOCIS (1C-NOCIS)

7.7.3.1 One-electron M_S = 1/2 doublet and two-electron M_S = 1 triplet references

There is also another approximate method, one-center NOCIS (1C-NOCIS),[197] which is an intermediate between NOCIS and STEX. The open-shell determinants are separately optimized as in NOCIS, but the coupling between non-orthogonal determinants with core holes on different centers is ignored, and NOCI is used to compute the remaining

matrix elements between non-orthogonal determinants. 1C-NOCIS constructs the orthogonal Slater-Condon components of the matrices, and then performs NOCI to obtain the relevant non-orthogonal components. The diagonal blocks are then projected against the ground state. For closed-shell singlet NOCIS, 1C-NOCIS is the same as STEX, since there are no open-shell ground-state orbitals.

There are two main advantages of 1C-NOCIS. First, it is substantially cheaper to evaluate than NOCIS and so enables the treatment of larger molecules. Second, and in contrast to STEX, it allows the open-shell states to relax separately, which may have a substantial impact on accuracy.

7.7.3.2 Two-electron M_S = 0 singlet and triplet references

Inspired by the 1C-NOCIS model, a theory for core excited states out of an M_S = 0 two-electron open-shell (2eOS) reference (either singlet or triplet), such as those in ultra-fast UV-pump XUV- or X-ray-probe experiments, was subsequently developed.[7] For a 2eOS state describable by a single NTO pair, where an electron in occupied orbital o is promoted into a target virtual orbital t, the excited-state wave function takes the form

$$\left|^{1,\,3}\Psi_i\right\rangle = (2)^{1/2}\left(\left|\Phi_o^t\right\rangle \pm \left|\Phi_{\bar{o}}^{\bar{t}}\right\rangle\right) \tag{7.64}$$

where the plus corresponds to the singlet and the minus to the triplet. Subsequently promoting a core electron c into an arbitrary particle level y, while remaining within the M_s = 0 manifold, results in eight possible configurations

$$\left|\Phi_{oc}^{to}\right\rangle,\ \left|\Phi_{\bar{o}\bar{c}}^{\bar{t}\bar{o}}\right\rangle,\ \left|\Phi_{oc}^{ty}\right\rangle,\ \left|\Phi_{\bar{o}\bar{c}}^{\bar{t}\bar{y}}\right\rangle,\ \left|\Phi_{\bar{o}c}^{\bar{t}y}\right\rangle,\ \left|\Phi_{o\bar{c}}^{t\bar{y}}\right\rangle,\ \left|\Phi_{\bar{o}c}^{\bar{y}t}\right\rangle,\ \left|\Phi_{o\bar{c}}^{y\bar{t}}\right\rangle \tag{7.65}$$

Diagonalizing the S^2 operator within the span of these configurations results in several possible configuration state fuctions (CSFs). A special one is the 2eOS singlet or triplet CSFs associated with the re-pairing of the core electron c with the electron in the singly-occupied molecular orbital (SOMO) that was formerly fully-occupied in the closed-shell state, o.

$$\left|^{1,\,3}\Phi_o^t\right\rangle = (2)^{-1/2}\left(\left|\Phi_c^t\right\rangle \pm \left|\Phi_{\bar{c}}^{\bar{t}}\right\rangle\right) \tag{7.66}$$

When the core electron is promoted to an arbitrary virtual orbital (with respect to the closed-shell configuration), the states can be described by the remaining 4eOS singlet, triplet, and quintet CSFs.

$$\left|^{1_A}\Phi_{oc}^{ty}\right\rangle = (12)^{-1/2}\left(2\left|\Phi_{oc}^{ty}\right\rangle + 2\left|\Phi_{\bar{o}\bar{c}}^{\bar{t}\bar{y}}\right\rangle + \left|\Phi_{\bar{o}c}^{\bar{t}y}\right\rangle + \left|\Phi_{o\bar{c}}^{t\bar{y}}\right\rangle - \left|\Phi_{\bar{o}c}^{\bar{y}t}\right\rangle - \left|\Phi_{o\bar{c}}^{y\bar{t}}\right\rangle\right) \tag{7.67a}$$

$$\left|^{1_B}\Phi_{oc}^{ty}\right\rangle = (2)^{-1}\left(\left|\Phi_{\bar{o}c}^{\bar{t}y}\right\rangle + \left|\Phi_{o\bar{c}}^{t\bar{y}}\right\rangle + \left|\Phi_{\bar{o}c}^{\bar{y}t}\right\rangle + \left|\Phi_{o\bar{c}}^{y\bar{t}}\right\rangle\right) \tag{7.67b}$$

$$\left|^{3_C}\Phi_{oc}^{ty}\right\rangle = (2)^{-1/2}\left(\left|\Phi_{oc}^{ty}\right\rangle - \left|\Phi_{\bar{o}\bar{c}}^{\bar{t}\bar{y}}\right\rangle\right) \tag{7.67c}$$

$$\left|^{3_D}\Phi_{oc}^{ty}\right\rangle = (2)^{-1/2}\left(\left|\Phi_{\bar{o}c}^{\bar{t}y}\right\rangle - \left|\Phi_{o\bar{c}}^{t\bar{y}}\right\rangle\right) \tag{7.67d}$$

$$\left|^{3_E}\Phi_{oc}^{ty}\right\rangle = (2)^{-1/2}\left(\left|\Phi_{\bar{o}c}^{\bar{y}t}\right\rangle - \left|\Phi_{o\bar{c}}^{y\bar{t}}\right\rangle\right) \tag{7.67e}$$

$$\left|^{5_F}\Phi_{oc}^{ty}\right\rangle = (6)^{-1/2}\left(\left|\Phi_{oc}^{ty}\right\rangle + \left|\Phi_{\bar{o}\bar{c}}^{\bar{t}\bar{y}}\right\rangle - \left|\Phi_{\bar{o}c}^{\bar{t}y}\right\rangle - \left|\Phi_{o\bar{c}}^{t\bar{y}}\right\rangle + \left|\Phi_{\bar{o}c}^{\bar{y}t}\right\rangle + \left|\Phi_{o\bar{c}}^{y\bar{t}}\right\rangle\right) \tag{7.67f}$$

Note that the special 2eOS CSF associated with repairing of the core electron with the particle SOMO t are included in the 4eOS CSFs presented above as the cases when $y = t$. As per the original 1C-NOCIS and STEX methods, the variant for M_S = 0 2eOS references diagonalizes the Hamiltonian in the subspan of all the aforementioned CSFs. In deviation with the original doublet and triplet 1C-NOCIS implementation, however, the 2eOS variant offers three possible sets of reference orbitals:

- The 1eOS, M_S = 1/2 doublet core ion orbitals, with the virtual space rotated into the NTO basis of a standard closed-shell STEX calculation
- The 2eOS, M_S = 0 ROKS-optimized orbitals for the $\left|^1\Phi_o^t\right\rangle$ configuration

- The 3eOS, M_S = 3/2 quartet core ion ROHF-optimized orbitals for the $\left|{}^4\Phi^t_{oc}\right\rangle$ configuration

Crucially, all of these references incorporate relaxation due to the presence of the core hole and hope to provide a well-defined particle state t. Since the STEX and 1C-NOCIS models are, in essence, a truncated CI formalism for open-shell systems, the final excited states constructed are sensitive to the underlying choice of orbitals.[132] By virtue of being occupied in all the CSFs employed to describe the pump-probe states, the quality of the normally-empty t orbital is essential. For the moment, we recommend using the $\left|{}^1\Phi^t_o\right\rangle$ orbitals when targeting singlet states and $\left|{}^4\Phi^t_{oc}\right\rangle$ orbitals when targeting triplet states.

The valence-excited state orbitals are used as a guess for the $\left|{}^1\Phi^t_o\right\rangle$ ROKS or $\left|{}^4\Phi^t_{oc}\right\rangle$ ROHF optimization. Alternatively, the code can use the 1eOS, M_S = 1/2 doublet core ion orbitals as a guess. Since HF provides virtual orbitals of poor quality, the doublet core ion calculation can be performed with DFT to provide a better guess for the subsequent calculation. The DFT method is specified with the ENV_METHOD rem variable.

The 1C-NOCIS 2eOS implementation is more flexible than the original 1C-NOCIS implementation for doublets and triplets. It allows to target arbitrary M_S = 0 singlet or triplet states obtained via CIS or ROKS and allows for control of the underlying calculations of the initial state and the SCF optimizations for the reference orbitals. Correspondingly, there is a larger number of $ rem variables available exclusively for 1C-NOCIS calculations for 2eOS references.

7.7.4 Job Control

NOCIS, 1C-NOCIS, and STEX run as subroutines in GEN_SCFMAN. Thus, for all calculations, you must set GEN_SCFMAN = TRUE. It is also highly recommended that you run an open-shell ground state calculation prior to running your NOCIS/STEX/1C-NOCIS calculations, which are all restricted.

NOCIS

Requests a NOCIS/STEX/1C-NOCIS/EA-TDDFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not run these methods.

TRUE Run one of these methods, options controlled in *$nocis*.

RECOMMENDATION:

None

Options below this line are set within the *$nocis* section.

STEX

Run a STEX calculation

INPUT SECTION: *$nocis*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

The presence of this keyword will activate STEX.

RECOMMENDATION:

None

ONE_CENTER

Run a 1C-NOCIS calculation

INPUT SECTION: *$nocis*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

False Run a NOCIS calculation.

True Run a 1C-NOCIS calculation.

RECOMMENDATION:

None

ORB_OFFSET

Determines the starting orbital for a NOCIS/STEX/1C-NOCIS/EA-TDDFT calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Positive integer

RECOMMENDATION:

Set according to the first orbital of interest in the system in question. For example, this would be set to 0 for the O K-edge in CO_2 because the two O 1s orbitals lie below the C 1s, so for the C K-edge this would be set to 2.

NUM_REF

Sets the number of reference orbitals in a NOCIS/STEX/1C-NOCIS/EA-TDDFT calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Positive integer

RECOMMENDATION:

Set according to the number of consecutive orbitals of interest for the calculation. For example, for the oxygen K-edge in CO_2, the number of references would be 2 (two O 1s orbitals), whereas for the carbon K-edge it would be 1 (one C 1s).

Options below this line are exclusive to 1C-NOCIS calculations for 2eOS references.

OS_IS

Triggers a 1C-NOCIS calculation with an M_S = 0 open-shell initial state (IS).

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

ROKS

OPTIONS:

CIS	Employ a CIS excited-state wave function transformed into the NTO basis as the IS
ROKS	Employ a ROKS- or ROHF-optimized wave function (singlets or triplets) as the IS
CIS_ROKS	Use CIS NTOs as a guess for an RO calculation on the IS.

RECOMMENDATION:

Use CIS when targeting a well-defined initial state. Else use ROSCF. Note that, if the CIS_ROKS option is used, the previous excited state job can employ TDDFT instead of CIS to provide a better guess.

OS_IS_CIS_STATE

When using CIS or CIS_ROKS for OS_IS, specifies which CIS state to target.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Positive integer

RECOMMENDATION:

Set according to the desired CIS state.

OS_IS_OCC_VALENCE

When using ROKS for OS_IS, specifies which occupied orbital to excite from.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

HOMO

OPTIONS:

n Positive integer

RECOMMENDATION:

Set according to the desired ROKS state. This is a 0-indexed variable (the lowest orbital has index 0).

OS_IS_VIR_VALENCE

When using ROKS for OS_IS, specifies which virtual orbital to excite into.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

LUMO

OPTIONS:

n Positive integer

RECOMMENDATION:

Set according to the desired ROKS state. This is a 0-indexed variable (the lowest orbital has index 0).

OS_IS_SCF_ALGORITHM

When using ROKS for OS_IS, specifies the SCF algorithm for convergence of the IS.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

GDM when OS_IS_CIS_STATE = 1 or IS_OCC_VALENCE = HOMO and IS_VIR_VALENCE = LUMO

SGM otherwise

OPTIONS:

DIIS

GDM

GDM_LS

SGM

SGM_LS

RECOMMENDATION:

Use defaults.

OS_IS_DSCF_ALGORITHM

When using ROKS for OS_IS, specifies the SCF algorithm for convergence of the IS.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

MOM

IMOM

STEP

STEP_MOM

RECOMMENDATION:

When using DIIS, MOM or IMOM may be necessary to target higher valence excited states.

OS_IS_SCF_CONVERGENCE

When using ROKS or CIS_ROKS for OS_IS, specifies the SCF convergence for the IS ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

7 for SGM- and GDM-based solvers

8 otherwise

OPTIONS:

n Positive integer

RECOMMENDATION:

Modify as needed for convergence, but thresholds below 5 for descent-based methods (GDM, SGM), and 7 for others may provide unsufficiently-converged orbitals.

OS_IS_SGM_GRADIENT

When using SGM for OS_IS_SCF_ALGORITHM, specifies the DELTA_GRADIENT_SCALE for the IS ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

75 when using SGM as the SCF solver for the initial valence state

NONE otherwise

OPTIONS:

n Positive integer below 100

RECOMMENDATION:

See DELTA_GRADIENT_SCALE $rem variable. If ROKS_IS_SS_MIXING is set to FALSE, try the default. Otherwise, a value of around 10 might be required. If its converging to the wrong state, try decreasing.

OS_IS_SS_MIXING

Controls whether to allow mixing between the singly-occupied orbitals during the ROKS procedure.

INPUT SECTION: *$nocis*

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

FALSE Supress coupling.

TRUE Allow coupling.

RECOMMENDATION:

See ROKS_SS_MIXING $rem variable. Supressing coupling is useful when the resuling states are difficult to converge and / or overlap significantly with the ground state, and seems to be harmless otherwise. Certain $\pi \longrightarrow \pi^*$ excited states are particularly susceptible to this (Example 7.49). Only works with descent-based solvers (GDM or SGM).

OS_FS_REFERENCE_ORB

Choice of reference orbitals for the final states (FS)

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

ROKS_2EOS for singlet valence excited states

QUARTET_ION for triplet valence excited states

OPTIONS:

OC_NOCIS Uses ROSCF doublet core ion orbitals, rotated into the closed-shell 1C-NOCIS NTO basis

QUARTET_ION Uses the ROSCF quartet core ion orbitals

ROKS_2EOS Uses the $\left|{}^1\Phi_o^t\right\rangle$ orbitals optimized via ROKS.

RECOMMENDATION:

Use defaults.

OS_FS_NS_REFERENCE

Specifies a single common reference to use for all core orbitals when NUM_REF > 1.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Non-negative integer

RECOMMENDATION:

Experimental feature. When simulating, say, the $M_{2,\,3}$-edge of an iron compound, the 3s core hole reference may be useful - while it doesn't optimize each individual 3p core hole individually, it partially captures core hole orbital realxation and provides an unbiased reference for the set of three 3p orbitals. Note no spin-orbit coupling is implemented at the moment for these calculations.

OS_FS_D_ION_SCF_ALGORITHM

Specifies the SCF algorithm for convergence of the doublet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

DIIS

OPTIONS:

DIIS

SGM

SGM_LS

RECOMMENDATION:

Use default. If convergence is not possible, try SGM.

OS_FS_D_ION_DSCF_ALGORITHM

Specifies the SCF algorithm for convergence of the doublet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

MOM when OS_FS_D_ION_SCF_ALGORITHM = DIIS

NONE otherwise

OPTIONS:

MOM

IMOM

STEP

STEP_MOM

RECOMMENDATION:

None.

OS_FS_D_ION_SCF_CONVERGENCE

Specifies the SCF convergence for the doublet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

ijk... Positive integer. i, j, k, ... relate to reference 1, 2, 3, ...

RECOMMENDATION:

Setting this $rem variable to 0 skips the doublet core ion calculations and instead uses the valence excited state orbitals as a guess for either the quartet core ion ROHF calculation or $\left|{}^1\Phi_o^t\right\rangle$ ROKS calculations. When that fails to converge the reference orbitals, attempt to use doublet core ion orbitals as a guess. Thresholds below 5 for SGM-Based solvers, or 7 for DIIS / MOM may provide insufficiently converged orbitals. This may not be as critical (and perhaps necessar) if just using them as a guess for subsequent calculations. Note DFT orbitals may serve as better guess - specify the desired functional via the ENV_METHOD $rem variable. When using closed-shell 1C-NOCIS reference orbitals for the 1C-NOCIS 2eOS calculation, proper convergence and not DFT must be used.

OS_FS_D_ION_SGM_GRADIENT

When using SGM for OS_FS_D_ION_SCF_ALGORITHM, specifies the DELTA_GRADIENT_SCALE for the doublet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Positive integer

RECOMMENDATION:

Anecdotically, a value of 1 - 10 seems to work well for core ions. If the ROSCF calculation is not converging, try increasing.

OS_FS_Q_ION_SCF_ALGORITHM

Specifies the SCF algorithm for convergence of the quartet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

DIIS

OPTIONS:

DIIS

SGM

SGM_LS

RECOMMENDATION:

Use DIIS when possible. For problematic cases, use SGM.

OS_FS_Q_ION_DSCF_ALGORITHM

Specifies the SCF algorithm for convergence of the quartet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

MOM when OS_FS_D_ION_SCF_ALGORITHM = DIIS

NONE otherwise

OPTIONS:

MOM

IMOM

STEP

STEP_MOM

RECOMMENDATION:

For efficiency, attempt using MOM, IMOM, STEP, or STEP_MOM first (requires DIIS for OS_FS_Q_ION_SCF_ALGORITHM). If the SCF procedure fails to converge or converges to the wrong state, do not set this rem variable and attempt using SGM in OS_FS_Q_ION_SCF_ALGORITHM.

OS_FS_Q_ION_SCF_CONVERGENCE

Specifies the SCF convergence for the quartet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

6 (for each reference) for SGM-based solvers

8 (for each reference) otherwise.

OPTIONS:

$ijk...$ Positive integer. i, j, k, ... relate to reference 1, 2, 3, ...

RECOMMENDATION:

Modify as needed for convergence, but thresholds below 5 for SGM-based methods, and 7 for others may provide unsufficiently-converged orbitals.

OS_FS_Q_ION_SGM_GRADIENT

When using SGM for OS_FS_Q_ION_SCF_ALGORITHM, specifies the DELTA_GRADIENT_SCALE for the quartet core ion ROSCF calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Positive integer

RECOMMENDATION:

Anecdotically, a value of 1 - 10 seems to work well for core ions. If the ROSCF calculation is not converging, try increasing.

OS_FS_2EOS_SCF_ALGORITHM

Specifies the SCF algorithm for convergence of the $\left|{}^{1}\Phi_{o}^{t}\right\rangle$ ROKS calculation.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

SGM

OPTIONS:

DIIS

SGM

SGM_LS

RECOMMENDATION:

Use DIIS for very simple (small) molecules and low-lying valence excited states. Else, use SGM.

OS_FS_2EOS_DSCF_ALGORITHM

Specifies the SCF algorithm for convergence of the $\left|{}^1\Phi_o^t\right\rangle$ ROKS calculation.

INPUT SECTION: *$nocis*

TYPE:

STRING

DEFAULT:

MOM when OS_FS_D_ION_SCF_ALGORITHM = DIIS

NONE otherwise

OPTIONS:

MOM

IMOM

STEP

STEP_MOM

RECOMMENDATION:

Use defaults.

OS_FS_2EOS_SCF_CONVERGENCE

Specifies the SCF convergence for the $\left|{}^1\Phi_o^t\right\rangle$ ROKS calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

6 (for each reference) for SGM-based solvers

8 (for each reference) otherwise.

OPTIONS:

$ijk...$ Positive integer. i, j, k, ... relate to reference 1, 2, 3, ...

RECOMMENDATION:

Modify as needed for convergence, but thresholds below 5 for SGM-based methods, and 7 for others may provide unsufficiently-converged orbitals.

OS_FS_2EOS_SGM_GRADIENT

When using SGM for OS_FS_2EOS_SCF_ALGORITHM, specifies the DELTA_GRADIENT_SCALE for the $\left|{}^1\Phi_o^t\right\rangle$ core- excited ROKS calculation.

INPUT SECTION: *$nocis*

TYPE:

INTEGER

DEFAULT:

75 when ROKS_FS_2EOS_SS_MIXING is set to FALSE

5 otherwise

OPTIONS:

n Positive integer below 100

RECOMMENDATION:

See DELTA_GRADIENT_SCALE $rem variable. If ROKS_FS_2EOS_SS_MIXING is set to FALSE, try the default. Otherwise, a value of around 1 - 10 might be required. If its converging to the wrong state, try decreasing.

OS_FS_2EOS_SS_MIXING

Controls whether to allow mixing between the singly-occupied orbitals during the ROKS procedure.

INPUT SECTION: *$nocis*

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

FALSE Supress coupling.

TRUE Allow coupling.

RECOMMENDATION:

See ROKS_SS_MIXING $rem variable. Supressing coupling is useful when the resuling states are difficult to converge and / or overlap significantly with the ground state, and seems to be harmless otherwise. 1s$\longrightarrow$ 3s excited states, for example, are particularly susceptible to this phenomena. Only works with descent-based solvers (GDM or SGM).

Example 7.42 NOCIS for the nitrogen K-edge of N_2

```
$molecule
   0 1
   N     0.000000     0.000000     0.564990
   N     0.000000     0.000000    -0.564990
$end

$rem
   METHOD           hf
   BASIS            sto-3g
   UNRESTRICTED     false
   GEN_SCFMAN       true
   NOCIS            true
   THRESH           14
   MAX_SCF_CYCLES   500
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

! the default behavior is NOCIS
$nocis
   ORB_OFFSET       0
   NUM_REF          2
$end
```

Example 7.43 STEX for the carbon K-edge of CN

```
$molecule
   0 2
   C  0.0000000   0.0000000  -0.6258140
   N  0.0000000   0.0000000   0.5364120
$end

$rem
   METHOD            hf
   BASIS             sto-3g
   SCF_GUESS         core
   SCF_ALGORITHM     diis_gdm
   MAX_SCF_CYCLES    5000
   THRESH            14
   SCF_CONVERGENCE   10
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
read
$end

$rem
   METHOD            hf
   BASIS             sto-3g
   SCF_GUESS         read
   UNRESTRICTED      false
   SCF_ALGORITHM     diis_gdm
   GEN_SCFMAN        true
   NOCIS             true
   MAX_SCF_CYCLES    5000
   THRESH            14
   SCF_CONVERGENCE   10
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end

$nocis
   STEX
   ORB_OFFSET        1
   NUM_REF           1
$end
```

Example 7.44 1C-NOCIS for the oxygen K-edge of O_2

```
$molecule
   0 3
   O    0.0000000   0.0000000    0.6021380
   O    0.0000000   0.0000000   -0.6021380
$end

$rem
   METHOD           hf
   BASIS            sto-3g
   GEN_SCFMAN       true
   THRESH           14
   MAX_SCF_CYCLES   500
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
read
$end

$rem
   UNRESTRICTED     false
   SCF_GUESS        read
   METHOD           hf
   BASIS            sto-3g
   GEN_SCFMAN       true
   NOCIS            true
   THRESH           14
   MAX_SCF_CYCLES   500
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

$nocis
   ONE_CENTER
   ORB_OFFSET       0
   NUM_REF          2
$end
```

Example 7.45 1C-NOCIS 2eOS for the oxygen K-edge of the first singlet excited state of water. A CIS calculation on is used to provide a guess for the ROKS optimization of the target valence state.

```
$comment
  CIS calculation on water to generate the target initial state.
$end

$molecule
  0   1
  O   0.0000      0.0000      0.1173
  H   0.0000      0.7572     -0.4692
  H   0.0000     -0.7572     -0.4692
$end

$rem
  METHOD               HF
  UNRESTRICTED         FALSE
  BASIS                DEF2-SVP
  SYMMETRY             FALSE
  SYM_IGNORE           TRUE
  GEN_SCFMAN           TRUE
  SCF_ALGORITHM        DIIS_GDM
  CIS_N_ROOTS          3
  CIS_SINGLETS         TRUE
  CIS_TRIPLETS         FALSE
  REL_X2C              TRUE
  THRESH               14
  MEM_TOTAL            1000
  MEM_STATIC           100
  NTO_PAIRS            3
$end

@@@

$comment
  1C-NOCIS 2eOS singlet calculation.
$end

$molecule
  read
$end

$rem
  METHOD               HF
  UNRESTRICTED         FALSE
  BASIS                DEF2-SVP
  SYMMETRY             FALSE
  SYM_IGNORE           TRUE
  GEN_SCFMAN           TRUE
  SCF_GUESS            READ
  MAX_SCF_CYCLES       0
  REL_X2C              TRUE
  NOCIS                1
  THRESH               14
  MEM_TOTAL            1000
  MEM_STATIC           100
$end

$nocis
  SINGLETS
  ONE_CENTER
  OS_IS                CIS ! Use CIS NTOs for the valence excited state (IS)
$end
```

Example 7.46 1C-NOCIS 2eOS for the nitrogen K-edge of the first singlet excited state of pyrazine. The canonical orbitals are used as a guess for the ROKS optimization of the target valence state.

```
$comment
  1C-NOCIS 2eOS singlet calculation on pyrazine for the lowest excited state.
$end

$molecule
  0   1
  N   0.0000      0.0000      1.3814
  N   0.0000      0.0000     -1.3814
  C   0.0000      1.1192      0.6914
  C   0.0000     -1.1192      0.6914
  C   0.0000     -1.1192     -0.6914
  C   0.0000      1.1192     -0.6914
  H   0.0000      2.0412      1.2411
  H   0.0000     -2.0412      1.2411
  H   0.0000     -2.0412     -1.2411
  H   0.0000      2.0412     -1.2411
$end

$rem
  METHOD               HF
  UNRESTRICTED         FALSE
  BASIS                STO-3G
  SYMMETRY             FALSE
  SYM_IGNORE           TRUE
  GEN_SCFMAN           TRUE
  REL_X2C              TRUE
  NOCIS                1
  THRESH               14
  MEM_TOTAL            1000
  MEM_STATIC           100
$end

$nocis
  SINGLETS
  ONE_CENTER
  LOCALIZE_ORBITALS             2         ! Localize the two nitrogen 1s orbitals
  NUM_REF                       2         ! Two nitrogen atoms
  OS_IS                         ROKS      ! Use ROKS for the valence excited state (IS)
$end
```

Example 7.47 1C-NOCIS 2eOS for the oxygen K-edge of the triplet ground state of molecular oxygen. The default reference orbitals for triplets are the quartet core ion orbtials. Note it begins from a closed-shell job.

```
$comment
  1C-NOCIS 2eOS triplet calculation. Note we begin from a closed-shell reference.
$end

$molecule
  0   1
  O   0.0000      0.0000      0.60375
  O   0.0000      0.0000     -0.60375
$end

$rem
  METHOD                HF
  UNRESTRICTED          FALSE
  BASIS                 STO-3G
  SYMMETRY              FALSE
  SYM_IGNORE            TRUE
  GEN_SCFMAN            TRUE
  SCF_ALGORITHM         DIIS_GDM
  REL_X2C               TRUE
  NOCIS                 1
  THRESH                14
  MEM_TOTAL             1000
  MEM_STATIC            100
$end

$nocis
  TRIPLETS                          ! Activate a triplet 2eOS reference
  ONE_CENTER                        ! One-center approximation
  LOCALIZE_ORBITALS             2   ! Localize 2 orbitals
  NUM_REF                       2   ! Two oxygen atoms
  OS_IS                         ROKS ! Use ROHF to converge the valence excited state (IS)
  OS_FS_Q_ION_SCF_CONVERGENCE   88  ! Convergence of the quartet core ion orbitals
$end
```

7.8 Restricted Open-Shell and ΔSCF Methods

7.8.1 Introduction

Standard TDDFT is prone to catastrophic failure in regimes where there is a substantial difference in density between ground and excited states, such as charge-transfer, Rydberg, or core excitations.[63,106] This can be greatly ameliorated via inclusion of orbital relaxation beyond linear response, via explicit optimization of excited-state orbitals in a manner analogous to a ground-state SCF calculation, and several methods for doing so are described in this section. Several of these methods recognize that a single-determinant description of any singlet excited state cannot be spin pure, and the minimal description of an open-shell singlet state requires two determinants. This can be handled in a computationally tractable manner using restricted open-shell Kohn-Sham (ROKS) calculation, as described in Section 7.8.3. For open-shell singlet states (*i.e.*, biradicals), a simplified version of this approach is approximate spin purification as described in Section 7.8.2.

Excited-state orbital optimization is generally more challenging as compared to the ground-state SCF problem because excited-state solutions of the SCF equations are generally not local minima in the orbital rotation space, but are instead

typically saddle points. Traditional orbital optimizers like DIIS or GDM often fail to locate these excited-state, non-aufbau solutions to the SCF equations, and instead collapse to lower-energy solutions (usually the ground state). This problem of "variational collapse" restricts the utility of excited-state orbital-optimization methods. The maximum overlap method (MOM, Section 7.6) can prevent this in many cases, but is constrained by the convergence issues stemming from the underlying SCF algorithm. Q-CHEM includes two alternative procedures that tend to be more robust as compared to MOM and can be used to find excited-state solutions to the SCF equations: squared-gradient minimization (Section 7.8.4),[91] and state-targeted energy projection (Section 7.8.5).[33] Both methods can be used on their own to produce a single-determinant "ΔSCF" estimate of an excitation energy,[106] using ground-state machinery, or else combined with the ROKS procedure in order to avoid the spin contamination problems associated with the ΔSCF approach.

7.8.2 Approximate Spin Purification

Singlet biradicals are an important special case, whose description within a single-determinant SCF formalism is typical characterized by significant spin contamination, $\langle \hat{S}^2 \rangle \approx 1$ (in atomic units), indicating an approximately equal mixture of singlet ($\langle \hat{S}^2 \rangle = 0$) and triplet ($\langle \hat{S}^2 \rangle = 2$) wave functions. This is a result of the fact that a proper description of two electrons in two half-filled orbitals requires a minimum of two determinants. A simple means to correct for this is to use the approximate spin-purification formula[33,56,106,278]

$$E_{\text{OpSing}} \approx 2E_{\text{BS}} - E_{\text{trip}} \ , \tag{7.68}$$

which expresses the open-shell singlet energy (E_{OpSing}) in terms of the energy of the broken-symmetry solution (E_{BS}, meaning the state with $\langle \hat{S}^2 \rangle \approx 1$) and the triplet energy (E_{trip}). Evaluation of Eq. (7.68) requires two separate SCF calculations but these can be performed together automatically by specifying OPSING = TRUE in the *$rem* section; see Example 7.38. Analytic gradients have been implemented for this approximate spin-purification approach, which again requires two separate SCF gradient computations. For the specific case of open-shell singlets, the ROKS method that is discussed below can be understood as a fully self-consistent or orbital-optimized version of the formula in Eq. (7.68),[106] using a consistent set of orbitals for both terms in the E_{OpSing} formula, whereas the method that is requested using OPSING = TRUE performs two independent SCF calculations with different orbitals in each.

OPSING

Controls whether approximate spin purification will be performed according to Eq. (7.68).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Spin purification is not performed.

TRUE Spin purification will be performed.

RECOMMENDATION:

Set to TRUE if spin purification calculation is desired. Make sure that UNRESTRICTED is set to TRUE Gradients are available.

7.8.3 Restricted Open-Shell Kohn-Sham Method (ROKS)

Singly-excited states of closed-shell molecules cannot be described via a single non-*aufbau* filled Slater determinant as both the up and down spins are equally likely to be excited, leading to at least two configurations with equal weights. Triplet energies can nonetheless be found from a single determinant by switching from the $M_S = 0$ subspace of the ground state to $M_S = \pm 1$ (*i.e.*, by having both unpaired electrons have spins pointing in the same direction instead of

having one up and one down spin). This tactic however does not work on singlet excited states, with non-*aufbau* filled configurations where only the up (or down) spin is excited being intermediate between singlet and triplet (and thus spin contaminated). This mixed state is not unlike spin-symmetry broken, unrestricted ground state solutions. An actual singlet energy can be obtained via approximate spin-purification post SCF, by removing the triplet contribution to the energy. The triplet energy thus has to be separately estimated with a second orbital optimization.

The *restricted open-shell Kohn-Sham* (ROKS) method offers an alternative route to singlet excited states of this nature. The mixed non-*aufbau* configuration (with either the up or down spin being excited) is exactly halfway between a singlet and triplet when restricted open-shell orbitals are used, and has an energy E_{mix}. The triplet energy E_{T} is also computable from a single determinant within the the $M_S = \pm 1$ subspaces. Consequently, ROKS optimizes a set of spin-restricted orbitals $\{\phi_{\text{ROKS}}\}$ such that the spin-purified singlet energy $E_{\text{S}} = 2E_{\text{mix}}[\{\phi_{\text{ROKS}}\}] - E_{\text{T}}[\{\phi_{\text{ROKS}}\}]$ is stationary. This therefore needs only one orbital optimization, in contrast to the two sets needed for the ΔSCF approach mentioned in the preceding paragraph. The structure of the ROKS Fock matrix however is more complex, by virtue of the two-determinant nature of the equations.[113,134] It is also important to note that this excited-state method is distinct from ROKS theory for open-shell ground states, which is a single-determinant method corresponding to the high-spin state with multiple unpaired spins.

The implementation of ROKS excited states in Q-CHEM largely follows the theoretical framework established by Filatov and Shaik;[71] see Ref. 134 for the case of the lowest excited singlet (S_1 state) with a DIIS-based approach. An example is provided below. ROKS for higher excited states is possible using either the squared-gradient approach (Section 7.8.4), the maximum overlap method (Section 7.6), or else state-targeted energy projection (Section 7.8.5).

ROKS has been found to be significantly more accurate than TDDFT for describing charge-transfer states,[92] and preliminary evidence shows the same to hold for Rydberg states. ROKS is also extremely accurate for core excitation energies, with the SCAN functional yielding errors below 0.5 eV for both K- and L-edge excitations of small molecules.[90] Examples of using ROKS/SGM to compute core-excited states are provided in Section 7.8.4. Analytic nuclear gradients (in the excited state) are also available, enabling geometry optimization and molecular dynamics calculations as well, along with finite-difference frequency calculations. Users of the ROKS code are requested to cite Ref. 134, and in addition Ref. 91 if the SGM implementation is employed, as well as Ref. 92 for charge-transfer states and Ref. 90 for application to core excitations.

The chief limitation of ROKS is that it can only describe states with one broken electron pair. It is consequently applicable only to certain excited states of closed-shell systems: all singlet single excitations well-described by a single natural transition orbital (NTO) pair, or higher singlets where only one electron pair is broken in total (like the $^1\text{B}_{3g}$ doubly excited state of tetrazine). Fortunately, most charge-transfer and core-excitations do not require more than one broken electron pair, and so this limitation is not a major problem in practice.

Furthermore, ROKS often suffers from an un-physical mixing between the open-shell orbitals when they belong to the same spatial symmetry, which often manifests in $\pi \longrightarrow \pi^*$ and $1s \longrightarrow 3s$ excitations.[76,134] This is associated with the inherent ambiguity in defining an effective Fock opereator for a two-determinant wave function.[76,113] Convergence is often difficult for the states where this mixing manifests, and the overlap with the ground state may be significant. Two protocols can be used to alleviate this issue. The first is a level-shift technique that splits the energy between the two singly-occupied orbitals thereby supressing their mixing.[134] It is only available with DIIS and for the ROKS implementation on the old SCF engine (GEN_SCFMAN = FALSE). The second is a total supression of the mixing between the sinlgy-occupied orbitals. While this may disregard the correct variational condition,[113] it produces excited states with much smaller overlaps with the ground state and provides smoother convergence. See 7.49 for a demonstration.

To perform an ROKS excited state calculation, simply set the keyword ROKS = TRUE and ensure that UNRESTRICTED = FALSE. It is recommended to perform a preliminary ground-state calculation on the system first, and then use the ground-state orbitals to construct the initial guess using SCF_GUESS = READ.

ROKS

Controls whether ROKS calculation will be performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE ROKS is not performed.

TRUE ROKS will be performed.

RECOMMENDATION:

Set to TRUE if ROKS calculation is desired. Make sure that UNRESTRICTED is not set to TRUE.

ROKS_LEVEL_SHIFT

Introduce a level shift of N/100 hartree to aid DIIS convergence.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No shift

N level shift of N/100 hartree.

RECOMMENDATION:

Use in cases of problematic DIIS convergence. Only available for the ROKS implementation on the old SCF engine (GEN_SCFMAN = FALSE).

ROKS_SS_MIXING

Allow coupling between the two singly-occupied molecular orbitals.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Supress coupling.

1 Allow coupling.

RECOMMENDATION:

Supress coupling when the resuling states are difficult to converge and / or overlap significantly with the ground state. Only for GDM and SGM-based solvers.

Example 7.48 RO-PBE0/6-311+G* excited state gradient of formaldehyde, using the ground state orbitals as an initial guess. This used the DIIS based implementation of Ref 134.

```
$comment
   ROKS excited state gradient of formaldehyde
   Use orbitals from ground state for initial guess
$end

$rem
   EXCHANGE          pbe0
   BASIS             6-311+G*
   SCF_CONVERGENCE   9
point_group_symmetry False
$end

$molecule
   0 1
   H      -0.940372    0.000000    1.268098
   H       0.940372    0.000000    1.268098
   C       0.000000    0.000000    0.682557
   O       0.000000    0.000000   -0.518752
$end

@@@

$molecule
   read
$end

$rem
   ROKS              true
   UNRESTRICTED      false
   EXCHANGE          pbe0
   BASIS             6-311+G*
   JOBTYPE           force
   SCF_CONVERGENCE   9
point_group_symmetry False
   SCF_GUESS         read
$end
```

Example 7.49 ROKS calculation on the lowest $\pi \longrightarrow \pi^*$ excited state of acetone using ωB97X-D / STO-3G. To observe the mixing of the open-shells, as well as the effect of mixing supression, try running this example with ROKS_SS_MIXING set to TRUE/FALSE.

```
$comment
  Plain old SCF on the ground state.
$end

$molecule
  0   1
  C   0.00000000      0.00000000      0.18807702
  C   0.00000000      2.42007545     -1.31764698
  C   0.00000000     -2.42007545     -1.31764698
  O   0.00000000      0.00000000      2.48269094
  H   0.00000000      4.03690733     -0.05185132
  H   0.00000000     -4.03690733     -0.05185132
  H   1.66061256      2.48420530     -2.53995285
  H  -1.66061256      2.48420530     -2.53995285
  H   1.66061256     -2.48420530     -2.53995285
  H  -1.66061256     -2.48420530     -2.53995285
$end

$rem
  METHOD               WB97X-D
  UNRESTRICTED         FALSE
  BASIS                STO-3G
```

7.8.4 Squared-Gradient Minimization

The squared-gradient minimization (SGM) algorithm[91] sidesteps the challenge of optimizing a saddle point in the space of orbital rotation variables $\vec{\theta}$, by instead minimizing the square of the energy gradient with respect to those variables. Ground-state SCF methods seek to minimize the energy E with respect to $\vec{\theta}$ and therefore the gradient $\hat{\boldsymbol{\nabla}}_{\vec{\theta}}E$ must be zero at convergence. It is therefore possible to obtain the same result by minimizing $\Delta(\vec{\theta}) = \|\hat{\boldsymbol{\nabla}}_{\vec{\theta}}E\|^2$ to zero. However, all stationary points of E are minima of $\Delta(\vec{\theta})$, not just the ground state. It is therefore possible to optimize excited-state orbitals by starting from a reasonable guess (such as a non-*aufbau* configuration corresponding to the excitation) and minimizing $\Delta(\vec{\theta})$. This avoids all the pitfalls of attempting to optimize unstable stationary points in E and thus averts variational collapse.

The SGM algorithm in Q-CHEM can be used to optimize orbitals for two different excited state approaches: ΔSCF and ROKS. The former simply attempts to minimize the energy of a single Slater determinant, which is often sufficient for many challenging excitations (including many double excitations).[10,33,91] However, many excitations (including all single excitations from a closed shell ground state) break electron pairs, leading to states that cannot be described with a single determinant. It is possible to spin-purify the energy of a spin-contaminated, non-*aufbau* determinant *a posteriori*, but this requires at least two separate orbital optimizations. An alternative is ROKS (as described in Section 7.8.3, which requires optimization of only a single set of orbitals, for which the spin-purified energy is stationary. Analytic nuclear gradients are available for both ΔSCF and ROKS, permitting geometry optimizations and *ab initio* molecular dynamics. Analytic frequencies are available for ΔSCF, including functionals that contain VV10 nonlocal correlation.

There are some slight differences between use of SGM for different orbital classes due to ease of implementation. The ΔSCF procedure with restricted closed-shell (R) and unrestricted (U) orbitals can be run with SCF_ALGORITHM = SGM_LS or SCF_ALGORITHM = SGM_QLS, with initial orbital occupation specified by the *$occupied* block (as described in Section 7.6 and in Examples 7.50 and 7.52 below). A ΔSCF calculation with restricted open-shell (RO) orbitals or an ROKS calculation can be performed via SCF_ALGORITHM = SGM or SCF_ALGORITHM = SGM_LS, and a re-ordering of orbitals to ensure that the unpaired ones lie at the frontier. (See Examples 7.51 and 7.53 below.) The gradient of $\Delta(\vec{\theta})$ is computed analytically (except in the case of functionals that contain VV10 nonlocal correlation), for R-, U- and RO-ΔSCF, at a cost equal to a single Fock build. However, the gradient $\Delta(\vec{\theta})$ in the ROKS case, and for functionals containing VV10, is computed with a finite-difference approach [see Eq. (4.47)]. In those cases, the cost is equal to that of two Fock builds. Cumulatively, a single SGM iteration costs twice as much as a single GDM iteration when the analytic $\Delta(\vec{\theta})$ gradient is available, and three times as much if the finite difference construction must be used, although this does not affect the asymptotic scaling of the calculation with respect to system size.

Excited-state orbital optimization sometimes requires more iterations than what is typical for ground-state SCF calculations, so MAX_SCF_CYCLES should be set to a large value (perhaps 200), rather than the default value of 50. A loose convergence threshold of SCF_CONVERGENCE = 4 is also permissible if only energies are desired, as long as it is explicitly confirmed that the variation in energy over several iterations is much less than the desired accuracy after job completion. (A variation greater than 10^{-3} E_h or 0.03 eV would be quite problematic, for example.) Further reduction of SCF_CONVERGENCE likely compromises properties such as dipole moments or nuclear gradients, and is not recommended.

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

SGM

SGM_LS

SGM_QLS

RECOMMENDATION:

SGM should be used for RO-ΔSCF or ROKS calculations only. SGM_LS is recommended for R- or U-ΔSCF, though it can also be used for RO-ΔSCF or ROKS. SGM_QLS is a slower but more robust option for R- and U-ΔSCF calculations.

DELTA_GRADIENT_SCALE

Scales the gradient of Δ by N/100, which can be useful for cases with troublesome convergence by reducing step size.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

N

RECOMMENDATION:

Use default. For problematic cases, N =50, 25, 10 or even $N = 1$ could be useful.

ROKS

Controls whether ROKS calculation will be performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE ROKS is not performed.

TRUE ROKS will be performed.

RECOMMENDATION:

Set to TRUE if ROKS calculation is desired. UNRESTRICTED = FALSE should also be ensured.

Example 7.50 Restricted ΔSCF double excitation from $2s$ to $2p$ of Be atom, using SGM_QLS and the ground state orbitals as an initial guess.

```
$comment
   Calculates Delta-SCF excitation energy for the 2s^2 -> 2p^2 excitation of Be
   using SCAN and SGM_QLS scf convergence
$end

$molecule
   0 1
   Be
$end

$rem
   METHOD            scan
   BASIS             aug-cc-pVTZ
   THRESH            14
   SCF_CONVERGENCE   8
   SCF_ALGORITHM     diis
   XC_GRID           000099000590
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
   read
$end

$rem
   METHOD            scan
   BASIS             aug-cc-pVTZ
   THRESH            14
   SCF_ALGORITHM     sgm_qls
   SCF_GUESS         read
   XC_GRID           000099000590
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end

$occupied
   1 3
   1 3
$end
```

Example 7.51 ROKS single excitation from HOMO to LUMO for an $NH_3 \cdots F_2$ model complex, which describes electron transfer from NH_3 to F_2. Ground state orbitals are used as an initial guess.

```
$molecule
   0 1
   N  0.0000   0.0000    0.0000
   H  0.0000  -0.9377   -0.3816
   H  0.8121   0.4689   -0.3816
   H -0.8121   0.4689   -0.3816
   F  0.0000   0.0000    6.0000
   F  0.0000   0.0000    7.4120
$end

$rem
   METHOD            pbe0
   BASIS             cc-pVDZ
   SCF_CONVERGENCE   8
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY False
$end

@@@

$comment
   The reorder section is superfluous here since the excitation is HOMO to LUMO
   and thus the unpaired electron orbitals are already at the frontier.
$end

$molecule
   read
$end

$rem
   METHOD            pbe0
   BASIS             cc-pVDZ
   SCF_ALGORITHM     sgm
   ROKS              true
   SCF_GUESS         read
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

$reorder_mo
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
$end
```

Example 7.52 Unrestricted ΔSCF single excitation from HOMO−1 to LUMO+1 for HCHO using SGM_LS SCF convergence algorithm and the ground state orbitals as an initial guess.

```
$comment
   Calculates Delta-SCF excitation energy for the HOMO-1 -> LUMO+1
   excitation of HCHO using SCAN and SGM_LS convergence
$end

$molecule
   0 1
   O1     0.0000   0.0000   1.2050
   C2     0.0000   0.0000   0.0000
   H3     0.0000   0.9429  -0.5876
   H4     0.0000  -0.9429  -0.5876
$end

$rem
   METHOD            scan
   BASIS             aug-cc-pVTZ
   THRESH            14
   SCF_CONVERGENCE   8
   SCF_ALGORITHM     diis
   XC_GRID           000099000590
   GEN_SCFMAN        true
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
   read
$end

$rem
   METHOD            scan
   BASIS             aug-cc-pVTZ
   THRESH            14
   SCF_ALGORITHM     sgm_ls
   SCF_GUESS         read
   XC_GRID           000099000590
   GEN_SCFMAN        true
   UNRESTRICTED      true
   SCF_CONVERGENCE   7
   MAX_SCF_CYCLES    500
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

$occupied
   1 2 3 4 5 6 8 10
   1 2 3 4 5 6 7 8
$end
```

Example 7.7.53 Combined RO-ΔSCF core-ionization and $1s \rightarrow \text{LUMO} + 1$ ROKS core-excited state.

View input online

7.8.5 State-Targeted Energy Projection

The state-targeted energy projection (STEP) method[33] supplies a robust and cost-effective alternative to the MOM and IMOM procedures that were described in Section 7.6. STEP applies a level shift via a simple modification of the Fock matrix,

$$\mathbf{F}' = \mathbf{F} + \eta\mathbf{SQS} \,, \tag{7.69}$$

where $\mathbf{Q}$ is the matrix representation of the projector onto the user-defined virtual space, and η is a parameter. The level shift supplied by $\eta\mathbf{SQS}$ elevates the energy of virtual orbital ψ_a from ε_a to $\varepsilon_a + \eta$ for each unoccupied orbital that is contained in $\mathbf{Q}$. The parameter η is chosen to provide the smallest level shift that retains the desired electron configuration and is defined as follows:

$$\eta = |\varepsilon_{\text{HOMO}} - \varepsilon_{\text{LUMO}}| + \epsilon' \tag{7.70}$$

The HOMO/LUMO gap in Eq. (7.70) pertains to the HOMO and LUMO of the user-defined configuration; meaning that like the MOM procedure, STEP requires a set of initial-guess molecular orbitals (usually from a ground state calculation). The small empirical parameter ϵ' controls the magnitude of the gap between the occupied and unoccupied orbitals and is settable by the *$rem* variable STEP_EPSILON. Application of the STEP level shift constrains the solutions of the SCF equations to prevent variational collapse by forcing an *aufbau* occupation of the desired occupied space at every SCF cycle.

The implementation of STEP in Q-CHEM takes advantage of the fact that faster convergence is generally achieved by allowing a dynamic level shift parameter η that changes each cycle depending on the new HOMO/LUMO gap, which differs from the static η parameter reported in Ref. 33. In the most extreme of cases, if the desired *aufbau* configuration is trivially satisfied without application of a level shift projection, STEP will set $\eta = 0$, which allows for unconstrained occupied/virtual rotations in optimizing the Fock matrix and thus for rapid convergence to the desired state. The parameter ϵ' is nonetheless held constant as to allow control over the relative magnitude of the level shift in cases where one is necessary throughout the optimization.

In more difficult cases, the dynamic level-shift approach (while more efficient) can lead to variational collapse. If the dynamic level-shift is insufficient, reverting to the implementation that was originally reported in Ref. 33 can increase the robustness of STEP appreciably. In this approach, the level-shift parameter in Eq. (7.70) is always active at every SCF cycle such that η is never zero.

STEP can be applied directly as a ΔSCF procedure, wherein spin contamination of the excited state is often introduced, or within a restricted open-shell framework (Section 7.8.3) in order to directly converge spin-pure excited states. The STEP algorithm is available for restricted, unrestricted, and restricted open-shell orbitals in Q-CHEM.

Job control for ΔSCF (R- or U-STEP) and RO-STEP calculations: After STEP is activated in the *$rem* section, the remainder of the options for STEP are handled through the *$step* input section.

STEP

Activates the STEP procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not apply the STEP level-shift algorithm.

TRUE Apply the STEP level-shift algorithm.

RECOMMENDATION:

None

Epsilon

Scales the size of the occupied/virtual gap imposed by the level-shift by N/100 Hartree.

INPUT SECTION: *$step*

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

N

RECOMMENDATION:

Use the default unless convergence issues arise, in which case a larger value can be used until the desired state is found. Be aware that increasing the occupied/virtual gap in level-shift algorithms slows convergence so it may be advisable to increase SCF_MAX_CYCLES if large shifts are required.

Print

Controls the print level for STEP algorithm information.

INPUT SECTION: *$step*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Do not print any information about STEP between SCF cycles.

1 Print the level-shift applied at each SCF cycle (R- and U-STEP).

2 Print the level-shift for both mixed and triplet states at each SCF cycle (RO-STEP).

RECOMMENDATION:

Use the default. Level shifts of 0 indicate that an *aufbau* criterion is sufficient to determine orbital occupation, and shifts > 0 imply non-*aufbau* selection of the occupied space.

Always_Active

Toggles the original implementation of STEP where the level-shift is static (applied every cycle).

INPUT SECTION: *$step*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

alpha Apply a constant level-shift to the alpha spin orbitals.

beta Apply a constant level-shift to the beta spin orbitals.

both Apply a constant level-shift to both alpha and beta spin orbitals.

RECOMMENDATION:

Use in cases where the dynamic level-shift does not achieve satisfactory results. In the case of U-STEP, the constraint need only be applied to orbitals that must maintain a non-Aufbau configuration (*i.e.*an alpha-electron promotion requires only the alpha constraint, but two-electron promotions will require both constraints). For RO-STEP this keyword is set to *both* by default and cannot be turned off. In R-STEP it is only sensible to use *both* if the desired doubly-excited configuration cannot be found with the dynamic level-shift parameter.

ROKS

Controls whether ROKS calculation will be performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE ROKS is not performed.

TRUE ROKS will be performed.

RECOMMENDATION:

Set to TRUE if ROKS calculation is desired. UNRESTRICTED = FALSE should also be ensured.

Example 7.54 Lowest energy $\pi \rightarrow \pi^*$ transition in acetylene using a ΔSCF approach with STEP.

```
$comment
   Ground state calculation for reference orbitals
$end

$molecule
   0 1
   C    0.0000000000   -0.0000000177   -0.6043240964
   C    0.0000000000    0.0000000000    0.6043240820
   H    0.0000000000    0.0000000654   -1.6654864149
   H    0.0000000000    0.0000000198    1.6654865011
$end

$rem
   METHOD            b3lyp
   BASIS             def2-tzvpd
   SCF_CONVERGENCE   7
$end

@@@

$comment
   Actual U-STEP calculation
$end

$molecule
   read
$end

$rem
   METHOD            b3lyp
   BASIS             def2-tzvpd
   SCF_GUESS         read
   STEP              true
   UNRESTRICTED      true
   SCF_CONVERGENCE   7
$end

! default level-shift is 0.1 Hartree
! beta orbital promotion means only betas need constrained
$step
Epsilon 10
Always_Active   beta
$end

$occupied
   1:7
   1:6 8
$end
```

Example 7.55 A spin-pure lowest energy $\pi \rightarrow \pi^*$ transition in acetylene using ROKS with STEP.

```
$comment
   Generates the ground-state reference orbitals
$end

$molecule
   0 1
   C    0.0000000000   -0.0000000177   -0.6043240964
   C    0.0000000000    0.0000000000    0.6043240820
   H    0.0000000000    0.0000000654   -1.6654864149
   H    0.0000000000    0.0000000198    1.6654865011
$end

$rem
   METHOD            b3lyp
   BASIS             def2-tzvpd
   SCF_CONVERGENCE   7
$end

@@@

$comment
   Actual RO-STEP calculation
$end

$molecule
   read
$end

$rem
   METHOD            b3lyp
   BASIS             def2-tzvpd
   SCF_ALGORITHM     gdm ! usually recommended with ROKS, but not necessary
   SCF_GUESS         read
   ROKS              true
   STEP              true
   SCF_CONVERGENCE   7
$end

! prints all level-shift information
$step
Epsilon 10
Print   2
$end

$reorder_mo
   1 2 3 4 5 6 7
   1 2 3 4 5 6 7
$end
```

7.8.6 Non-equilibrium Solvation for ROKS and ΔSCF Methods

Both ΔSCF calculations (*e.g.*, based on MOM) and ROKS calculations can be performed with continuum solvation effects modeled by a polarizable continuum model (PCM). (The solvation model itself is described in Section 11.2.3.) As for the ground state, the self-consistent PCM treatment along the SCF procedure provides *equilibrium* solvation of the calculated excited state (set SOLVENT_METHOD = PCM and include the *$pcm* block). During equilibrium

solvation, both slow (nuclear) and fast (electronic) solvent degrees of freedom are relaxed (governed by **Dielectric** in the *$solvent* block), as is appropriate for long-lived states. However, fast vertical excitation or emission occurs on a shorter timescale, requiring the relaxation of only the fast electronic solvent polarization (governed by **Dielectric_Infi**, see Section 11.2.3.3). In Q-CHEM, such *nonequilibrium* solvation effects can be included to first order with a perturbative state-specific (ptSS) correction.[182,271]

To begin with, a nonequilibrium ΔSCF or ROKS calculation requires equilibrium solvation of the initial state (ground state for absorption and excited state for emission). By activating **RF_ptSS_Save= true** in the *$pcm* block, the equilibrated reaction field is then stored on disk. Setting **RF_ptSS_Read = true** in an arbitrary later job for the final state reads the reaction field again. This automatically triggers a frozen reaction-field (fRF) SCF with the stored reaction field of the initial state. The resulting SCF energy corresponds to a zeroth order calculation of the final state in the reaction field of the initial state. To obtain the first-order nonequilibrium result, the fast solvent polarization is relaxed for the final state by adding a perturbative ptSS correction, which is printed after the converged fRF-SCF calculation.

RF_ptSS_Save

Save the current reaction field to disk.

INPUT SECTION: *$pcm*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Save the reaction field in the current job.

FALSE Do not save the reaction field in the current job.

RECOMMENDATION:

The reaction field is saved for the reference state, which is for ΔSCF or ROKS just the SCF result, but for TDDFT + II-SS-PCM the chosen reference state (see Section 7.3.6.2). Activating RF_ptSS_SAVE for several subsequent jobs overwrites the reaction field.

RF_ptSS_Read

Read the reaction field from disk and perform a fRF-SCF + ptSS-PCM calculation with it.

INPUT SECTION: *$pcm*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Read the reaction field in the current job.

FALSE Do not read the reaction field in the current job.

RECOMMENDATION:

The SCF result corresponds to zeroth order solvation of the final state. For first-order nonequilibrium solvation add the ptSS correction printed after the fRF-SCF.

Example 7.56 Vertical excitation energy of formaldehyde in acetonitrile with ROKS + PCM and a nonequilibrium ptSS-PCM correction.

```
$rem
   method               pbe0
   basis                def2-SVP
   scf_convergence      8
   point_group_symmetry false
   solvent_method       pcm
$end

$pcm
   theory              iefpcm
   rf_ptss_save        true            ! Save the final ground state reaction field
$end

$solvent
   Dielectric          35.688000       ! Acetonitrile
   Dielectric_Infi     1.806874
$end

$molecule
   0 1
   H      -0.940372    0.000000    1.268098
   H       0.940372    0.000000    1.268098
   C       0.000000    0.000000    0.682557
   O       0.000000    0.000000   -0.518752
$end

@@@

$rem
   method               pbe0
   basis                def2-SVP
   scf_guess            read
   roks                 true
   unrestricted         false
   scf_convergence      8
   point_group_symmetry false
   solvent_method       pcm
$end

$pcm
   theory              iefpcm
   rf_ptss_read        true            ! Read the prior reaction field and start fRF-SCF
$end

$solvent
   Dielectric          35.688000       ! Acetonitrile
   Dielectric_Infi     1.806874
$end

$molecule
   0 1
   read
$end
```

7.9 Correlated Excited State Methods: The CIS(D) Family

7.9.1 Introduction

CIS(D) is a simple, size-consistent doubles correction to CIS which has a computational cost scaling as the fifth power of the basis set for each excited state.[99,101] In this sense, CIS(D) can be considered as an excited state analog of the ground state MP2 method. CIS(D) yields useful improvements in the accuracy of excitation energies relative to CIS, and yet can be applied to relatively large molecules using Q-CHEM's efficient integrals transformation package. In addition, as in the case of MP2 method, the efficiency can be significantly improved through the use of the auxiliary basis expansions (Section 6.6).[216]

7.9.2 CIS(D) Theory

The CIS(D) excited state procedure is a second-order perturbative approximation to the computationally expensive CCSD, based on a single excitation configuration interaction (CIS) reference. The coupled-cluster wave function, truncated at single and double excitations, is the exponential of the single and double substitution operators acting on the Hartree-Fock determinant:

$$|\Psi\rangle = \exp\left(T_1 + T_2\right)|\Psi_0\rangle \tag{7.71}$$

Determination of the singles and doubles amplitudes requires solving the two equations

$$\langle\Psi_i^a|\, H - E \left|\left(1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1T_2 + \frac{1}{3!}T_1^3\right)\Psi_0\right\rangle = 0 \tag{7.72}$$

and

$$\left\langle\Psi_{ij}^{ab}\right|\, H - E \left|\left(1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1T_2 + \frac{1}{3!}T_1^3 + \frac{1}{2}T_2^2 + \frac{1}{2}T_1^2T_2 + \frac{1}{4!}T_1^4\right)\Psi_0\right\rangle = 0 \tag{7.73}$$

which lead to the CCSD excited state equations. These can be written

$$\langle\Psi_i^a|\, H - E \left|\left(U_1 + U_2 + T_1U_1 + T_1U_2 + U_1T_2 + \frac{1}{2}T_1^2U_1\right)\Psi_0\right\rangle = \omega b_i^a \tag{7.74}$$

and

$$\begin{aligned}\langle\Psi_i^a|\, H - E \,\big|\big(&U_1 + U_2 + T_1U_1 + T_1U_2 + U_1T_2 + \tfrac{1}{2}T_1^2U_1 \;+ T_2U_2 \\ &+\tfrac{1}{2}T_1^2U_2 + T_1T_2U_1 + \tfrac{1}{3!}T_1^3U_1\big|\,\Psi_0\rangle = \omega b_{ij}^{ab}\end{aligned} \tag{7.75}$$

This is an eigenvalue equation $\mathbf{Ab} = \omega\mathbf{b}$ for the transition amplitudes ($\mathbf{b}$ vectors), which are also contained in the U operators.

The second-order approximation to the CCSD eigenvalue equation yields a second-order contribution to the excitation energy which can be written in the form

$$\omega^{(2)} = \mathbf{b}^{(0)^\mathrm{t}}\mathbf{A}^{(1)}\mathbf{b}^{(1)} + \mathbf{b}^{(0)^\mathrm{t}}\mathbf{A}^{(2)}\mathbf{b}^{(0)} \tag{7.76}$$

or in the alternative form

$$\omega^{(2)} = \omega^{\mathrm{CIS(D)}} = E^{\mathrm{CIS(D)}} - E^{\mathrm{MP2}} \tag{7.77}$$

where

$$E^{\mathrm{CIS(D)}} = \left\langle\Psi^{\mathrm{CIS}}\right|\, V \left|U_2\Psi^{\mathrm{HF}}\right\rangle + \left\langle\Psi^{\mathrm{CIS}}\right|\, V \left|T_2U_1\Psi^{\mathrm{HF}}\right\rangle \tag{7.78}$$

and

$$E^{\mathrm{MP2}} = \left\langle\Psi^{\mathrm{HF}}\right|\, V \left|T_2\Psi^{\mathrm{HF}}\right\rangle \tag{7.79}$$

The output of a CIS(D) calculation contains useful information beyond the CIS(D) corrected excitation energies themselves. The stability of the CIS(D) energies is tested by evaluating a diagnostic, termed the "theta diagnostic".[199] The

theta diagnostic calculates a mixing angle that measures the extent to which electron correlation causes each pair of calculated CIS states to couple. Clearly the most extreme case would be a mixing angle of $45°$, which would indicate breakdown of the validity of the initial CIS states and any subsequent corrections. On the other hand, small mixing angles on the order of only a degree or so are an indication that the calculated results are reliable. The code can report the largest mixing angle for each state to all others that have been calculated.

7.9.3 Resolution of the Identity CIS(D) Methods

Because of algorithmic similarity with MP2 calculation, the "resolution of the identity" approximation can also be used in CIS(D). In fact, RI-CIS(D) is orders of magnitudes more efficient than previously explained CIS(D) algorithms for effectively all molecules with more than a few atoms. Like in MP2, this is achieved by reducing the prefactor of the computational load. In fact, the overall cost still scales with the fifth power of the system size.

Presently in Q-CHEM, RI approximation is supported for closed-shell restricted CIS(D) and open-shell unrestricted UCIS(D) energy calculations. The theta diagnostic is not implemented for RI-CIS(D).

7.9.4 SOS-CIS(D) Model

As in MP2 case, the accuracy of CIS(D) calculations can be improved by semi-empirically scaling the opposite-spin components of CIS(D) expression:

$$E^{\mathrm{SOS-CIS(D)}} = c_U \left\langle \Psi^{\mathrm{CIS}} \right| V \left| U_2^{\mathrm{OS}} \Psi^{\mathrm{HF}} \right\rangle + c_T \left\langle \Psi^{\mathrm{CIS}} \right| V \left| T_2^{\mathrm{OS}} U_1 \Psi^{\mathrm{HF}} \right\rangle \tag{7.80}$$

with the corresponding ground state energy

$$E^{\mathrm{SOS-MP2}} = c_T \left\langle \Psi^{\mathrm{HF}} \right| V \left| T_2^{\mathrm{OS}} \Psi^{\mathrm{HF}} \right\rangle \tag{7.81}$$

More importantly, this SOS-CIS(D) energy can be evaluated with the 4th power of the molecular size by adopting Laplace transform technique.[216] Accordingly, SOS-CIS(D) can be applied to the calculations of excitation energies for relatively large molecules.

7.9.5 SOS-CIS(D_0) Model

CIS(D) and its cousins explained in the above are all based on a second-order non-degenerate perturbative correction scheme on the CIS energy ("diagonalize-and-then-perturb" scheme). Therefore, they may fail when multiple excited states come close in terms of their energies. In this case, the system can be handled by applying quasi-degenerate perturbative correction scheme ("perturb-and-then-diagonalize" scheme). The working expression can be obtained by slightly modifying CIS(D) expression shown in Section 7.9.2.[102]

First, starting from Eq. (7.76), one can be explicitly write the CIS(D) energy as[43,102]

$$\omega^{\mathrm{CIS}} + \omega^{(2)} = \mathbf{b^{(0)}}^{\mathbf{t}} \mathbf{A}_{\mathrm{SS}}^{(0)} \mathbf{b^{(0)}} + \mathbf{b^{(0)}}^{\mathbf{t}} \mathbf{A}_{\mathrm{SS}}^{(2)} \mathbf{b^{(0)}} - \mathbf{b^{(0)}}^{\mathbf{t}} \mathbf{A}_{\mathrm{SD}}^{(1)} \left(\mathbf{D}_{\mathrm{DD}}^{(0)} - \omega^{\mathrm{CIS}} \right)^{-1} \mathbf{A}_{\mathrm{DS}}^{(1)} \mathbf{b^{(0)}} \tag{7.82}$$

To avoid the failures of the perturbation theory near degeneracies, the entire single and double blocks of the response matrix should be diagonalized. Because such a diagonalization is a non-trivial non-linear problem, an additional approximation from the binomial expansion of the $\left(\mathbf{D}_{\mathrm{DD}}^{(0)} - \omega^{\mathrm{CIS}} \right)^{-1}$ is further applied:[102]

$$\left(\mathbf{D}_{\mathrm{DD}}^{(0)} - \omega^{\mathrm{CIS}} \right)^{-1} = \left(\mathbf{D}_{\mathrm{DD}}^{(0)} \right)^{-1} \left(1 + \omega \left(\mathbf{D}_{\mathrm{DD}}^{(0)} \right)^{-1} + \omega^2 \left(\mathbf{D}_{\mathrm{DD}}^{(0)} \right)^{-2} + \ldots \right) \tag{7.83}$$

The CIS(D_0) energy ω is defined as the eigen-solution of the response matrix with the zero-th order expansion of this equation. Namely,

$$\left(\mathbf{A}_{\mathrm{SS}}^{(0)} + \mathbf{A}_{\mathrm{SS}}^{(2)} - \mathbf{A}_{\mathrm{SD}}^{(1)}(\mathbf{D}_{\mathrm{DD}}^{(0)})^{-1}\mathbf{A}_{\mathrm{DS}}^{(1)}\right)\mathbf{b} = \omega\mathbf{b} \tag{7.84}$$

Similar to SOS-CIS(D), SOS-CIS(D_0) theory is defined by taking the opposite-spin portions of this equation and then scaling them with two semi-empirical parameters:[43]

$$\left(\mathbf{A}_{\mathrm{SS}}^{(0)} + c_T\mathbf{A}_{\mathrm{SS}}^{\mathrm{OS}(2)} - c_U\mathbf{A}_{\mathrm{SD}}^{\mathrm{OS}(1)}(\mathbf{D}_{\mathrm{DD}}^{(0)})^{-1}\mathbf{A}_{\mathrm{DS}}^{\mathrm{OS}(1)}\right)\mathbf{b} = \omega\mathbf{b} \tag{7.85}$$

Using the Laplace transform and the auxiliary basis expansion techniques, this can also be handled with a 4th-order scaling computational effort. In Q-CHEM, an efficient 4th-order scaling analytical gradient of SOS-CIS(D_0) is also available. This can be used to perform excited state geometry optimizations on the electronically excited state surfaces.

7.9.6 CIS(D) Job Control and Examples

The legacy CIS(D) algorithm in Q-CHEM is handled by the CCMAN/CCMAN2 modules of Q-CHEM's and shares many of the *$rem* options. RI-CIS(D), SOS-CIS(D), and SOS-CIS(D_0) do not depend on the coupled-cluster routines. Users who will not use this legacy CIS(D) method may skip to Section 7.9.7.

As with all post-HF calculations, it is important to ensure there are sufficient resources available for the necessary integral calculations and transformations. For CIS(D), these resources are controlled using the *$rem* variables CC_MEMORY, MEM_STATIC and MEM_TOTAL (see Section 6.12.11).

To request a CIS(D) calculation the METHOD *$rem* should be set to CIS(D) and the number of excited states to calculate should be specified by EE_STATES (or EE_SINGLETS and EE_TRIPLETS when appropriate). Alternatively, CIS(D) will be performed when EXCHANGE = HF, CORRELATION = CI and EOM_CORR = CIS(D). The SF-CIS(D) is invoked by using SF_STATES.

EE_STATES

Sets the number of excited state roots to find. For closed-shell reference, defaults into EE_SINGLETS. For open-shell references, specifies all low-lying states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EE_SINGLETS

Sets the number of singlet excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EE_TRIPLETS

Sets the number of triplet excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

SF_STATES

Sets the number of spin-flip target states roots to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any spin-flip states.

OPTIONS:

$[i, j, k \ldots]$ Find i SF states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

Note: It is a symmetry of a *transition* rather than that of a *target state* that is specified in excited state calculations. The symmetry of the target state is a product of the symmetry of the reference state and the transition. For closed-shell molecules, the former is fully symmetric and the symmetry of the target state is the same as that of transition, however, for open-shell references this is not so.

CC_STATE_TO_OPT

Specifies which state to optimize.

TYPE:

INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[i,j] optimize the jth state of the ith irrep.

RECOMMENDATION:

None

Note: Since there are no analytic gradients for CIS(D), the symmetry should be turned off for geometry optimization and frequency calculations, and CC_STATE_TO_OPT should be specified assuming C_1 symmetry, *i.e.*, as [1,N] where N is the number of state to optimize (the states are numbered from 1).

Example 7.57 CIS(D) excitation energy calculation for ozone at the experimental ground state geometry C_{2v}

```
$molecule
   0 1
   O
   O  1  RE
   O  2  RE  1  A

   RE = 1.272
   A  = 116.8
$end

$rem
   METHOD           CIS(D)
   BASIS            6-31G*
   N_FROZEN_CORE    3            use frozen core
   EE_SINGLETS      [2,2,2,2]    find 2 lowest singlets in each irrep.
   EE_TRIPLETS      [2,2,2,2]    find 2 lowest triplets in each irrep.
   CCMAN2           false
$end
```

Example 7.58 CIS(D) geometry optimization for the lowest triplet state of water. The symmetry is automatically turned off for finite difference calculations

```
$molecule
   0 1
   o
   h  1  r
   h  1  r  2  a

   r  0.95
   a  104.0
$end

$rem
   JOBTYPE            opt
   BASIS              3-21g
   METHOD             cis(d)
   EE_TRIPLETS        1  calculate one lowest triplet
   CC_STATE_TO_OPT    [1,1] optimize the lowest state (1st state in 1st irrep)
$end
```

Example 7.59 CIS(D) excitation energy and transition property calculation (between all states) for ozone at the experimental ground state geometry C_{2v}

```
$molecule
   0 1
   O
   O  1  RE
   O  2  RE  1  A

   RE = 1.272
   A  = 116.8
$end

$rem
   BASIS              6-31G*
   PURCAR             2            Non-spherical (6D)
   METHOD             CIS(D)
   EE_SINGLETS        [2,2,2,2]
   EE_TRIPLETS        [2,2,2,2]
   CC_TRANS_PROP      1
   CCMAN2             FALSE
$end
```

7.9.7 RI-CIS(D), SOS-CIS(D), and SOS-CIS(D_0): Job Control

These methods are activated by setting the *$rem* keyword METHOD to RICIS(D), SOSCIS(D), and SOSCIS(D0), respectively. Other keywords are the same as in CIS method explained in Section 7.2.2. As these methods rely on the RI approximation, AUX_BASIS needs to be set by following the same guide as in RI-MP2 (Section 6.6).

METHOD

Excited state method of choice

TYPE:

STRING

DEFAULT:

None

OPTIONS:

RICIS(D)	Activate RI-CIS(D)
SOSCIS(D)	Activate SOS-CIS(D)
SOSCIS(D0)	Activate SOS-CIS(D_0)

RECOMMENDATION:

None

CIS_N_ROOTS

Sets the number of excited state roots to find

TYPE:

INTEGER

DEFAULT:

0 Do not look for any excited states

OPTIONS:

n $n > 0$ Looks for n excited states

RECOMMENDATION:

None

CIS_SINGLETS

Solve for singlet excited states (ignored for spin unrestricted systems)

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for singlet states

FALSE Do not solve for singlet states.

RECOMMENDATION:

None

CIS_TRIPLETS

Solve for triplet excited states (ignored for spin unrestricted systems)

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for triplet states

FALSE Do not solve for triplet states.

RECOMMENDATION:

None

CIS_STATE_DERIV

Sets the excited state index for analytical gradient calculation for geometry optimizations and vibrational analysis with SOS-CIS(D_0)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Select the nth state.

RECOMMENDATION:

Check to see that the states do no change order during an optimization. For closed-shell systems, either CIS_SINGLETS or CIS_TRIPLETS must be set to false.

MEM_STATIC

Sets the memory for individual program modules

TYPE:

INTEGER

DEFAULT:

192 corresponding to 192 MB

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

At least $150(N^2 + N)D$ of MEM_STATIC is required (N: number of basis functions, D: size of a double precision storage, usually 8). Because a number of matrices with N^2 size also need to be stored, 32–160 MB of additional MEM_STATIC is needed.

MEM_TOTAL

Sets the total memory available to Q-CHEM

TYPE:

INTEGER

DEFAULT:

2000 2 GB

OPTIONS:

n User-defined number of megabytes

RECOMMENDATION:

The minimum memory requirement of RI-CIS(D) is approximately MEM_STATIC + $\max(3SVXD, 3X^2D)$ (S: number of excited states, X: number of auxiliary basis functions, D: size of a double precision storage, usually 8). However, because RI-CIS(D) uses a batching scheme for efficient evaluations of electron repulsion integrals, specifying more memory will significantly speed up the calculation. Put as much memory as possible if you are not sure what to use, but never put any more than what is available. The minimum memory requirement of SOS-CIS(D) and SOS-CIS(D_0) is approximately MEM_STATIC + $20X^2D$. SOS-CIS(D_0) gradient calculation becomes more efficient when $30X^2D$ more memory space is given. Like in RI-CIS(D), put as much memory as possible if you are not sure what to use. The actual memory size used in these calculations will be printed out in the output file to give a guide about the required memory.

SOS_FACTOR

Sets the scaling parameter c_T

TYPE:

INTEGER

DEFAULT:

1300000 corresponding to 1.30

OPTIONS:

n $c_T = n/1000000$

RECOMMENDATION:

Use the default

SOS_UFACTOR

Sets the scaling parameter c_U

TYPE:

INTEGER

DEFAULT:

151 For SOS-CIS(D), corresponding to 1.51

140 For SOS-CIS(D_0), corresponding to 1.40

OPTIONS:

n $c_U = n/100$

RECOMMENDATION:

Use the default

7.9.8 Examples

Example 7.60 Input for an RI-CIS(D) calculation.

```
$molecule
   0 1
   C     0.667472     0.000000     0.000000
   C    -0.667472     0.000000     0.000000
   H     1.237553     0.922911     0.000000
   H     1.237553    -0.922911     0.000000
   H    -1.237553     0.922911     0.000000
   H    -1.237553    -0.922911     0.000000
$end

$rem
   METHOD          ricis(d)
   BASIS           aug-cc-pVDZ
   MEM_TOTAL       1000
   MEM_STATIC      100
   AO2MO_DISK      1000
   AUX_BASIS       rimp2-aug-cc-pVDZ
   PURECART        1111
   CIS_N_ROOTS     10
   CIS_SINGLETS    true
   CIS_TRIPLETS    false
$end
```

Example 7.61 Input for an SOS-CIS(D) calculation.

```
$molecule
   0 1
   C    -0.627782     0.141553     0.000000
   O     0.730618    -0.073475     0.000000
   H    -1.133677    -0.033018    -0.942848
   H    -1.133677    -0.033018     0.942848
$end

$rem
   METHOD          soscis(d)
   BASIS           aug-cc-pVDZ
   MEM_TOTAL       1000
   MEM_STATIC      100
   AO2MO_DISK      500000     ! 0.5 Terabyte of disk space available
   AUX_BASIS       rimp2-aug-cc-pVDZ
   PURECART        1111
   CIS_N_ROOTS     5
   CIS_SINGLETS    true
   CIS_TRIPLETS    true
$end
```

Example 7.62 Input for an SOS-CIS(D_0) geometry optimization on S_2 surface.

```
$molecule
   0 1
   o
   h  1  r
   h  1  r  2  a

   r  0.95
   a  104.0
$end

$rem
   JOBTYPE           = opt
   METHOD            = soscis(d0)
   BASIS             = 6-31G**
   AUX_BASIS         = rimp2-VDZ
   PURECART          = 1112
   SET_STATE_DERIV   = 2
   CIS_N_ROOTS       = 5
   CIS_SINGLETS      = true
   CIS_TRIPLETS      = false
$end
```

7.10 Coupled-Cluster Excited-State and Open-Shell Methods

7.10.1 Introduction

EOM-CC and most of the CI codes are part of CCMAN and CCMAN2. CCMAN is a legacy code which is being phased out. All new developments and performance-enhancing features are implemented in CCMAN2. Some options behave differently in the two modules. Below we make an effort to mark which features are available in legacy code only.

7.10.2 Excited States via EOM-EE-CCSD and EOM-CCSDT

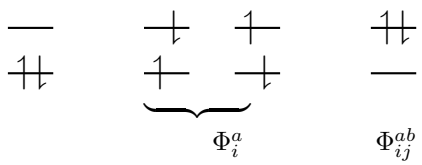
One can describe electronically excited states at a level of theory similar to that associated with coupled-cluster theory for the ground state by applying either linear response theory [130] or equation-of-motion methods. [235] A number of groups have demonstrated that excitation energies based on a coupled-cluster singles and doubles ground state are generally very accurate for states that are primarily single electron promotions. The error observed in calculated excitation energies to such states is typically 0.1–0.2 eV, with 0.3 eV as a conservative estimate, including both valence and Rydberg excited states. This, of course, assumes that a basis set large and flexible enough to describe the valence and Rydberg states is employed. The accuracy of excited state coupled-cluster methods is much lower for excited states that involve a substantial double excitation character, where errors may be 1 eV or even more. Such errors arise because the description of electron correlation of an excited state with substantial double excitation character requires higher truncation of the excitation operator. The description of these states can be improved by including triple excitations, as in EOM(2,3). Including triple excitations at CC as well as EOM levels results in the the EOM-CCSDT method (See 7.10.22 for more details) in which, the higher degree correlation correction is effectively incorporated in a more balanced way.

Q-CHEM includes coupled-cluster methods for excited states based on the coupled cluster singles and doubles (CCSD) and the singles, doubles and triples (CCSDT) methods described above.

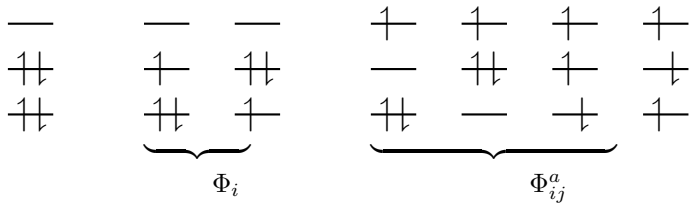
CCMAN code also includes the optimized orbital coupled-cluster doubles (OD) variant. OD excitation energies have been shown to be essentially identical in numerical performance to CCSD excited states.[142]

These methods, while far more computationally expensive than TDDFT, are nevertheless useful as proven high accuracy methods for the study of excited states of small molecules. Moreover, they are capable of describing both valence and Rydberg excited states, as well as states of a charge-transfer character. Also, when studying a series of related molecules it can be very useful to compare the performance of TDDFT and coupled-cluster theory for at least a small example to understand its performance. Along similar lines, the CIS(D) method described earlier as an economical correlation energy correction to CIS excitation energies is in fact an approximation to EOM-CCSD. It is useful to assess the performance of CIS(D) for a class of problems by benchmarking against the full coupled-cluster treatment. Finally, Q-CHEM includes extensions of EOM methods to treat ionized or electron attachment systems, as well as di- and triradicals.

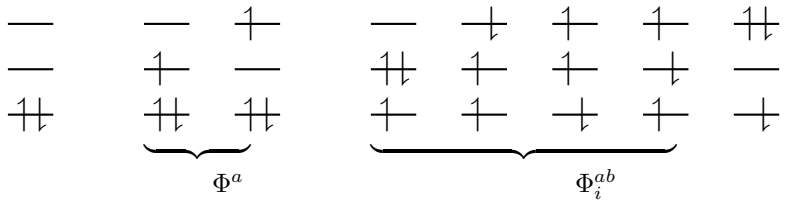
EOM-EE $\Psi(M_S = 0) = R(M_S = 0)\Psi_0(M_S = 0)$



EOM-IP $\Psi(N) = R(-1)\Psi_0(N + 1)$



EOM-EA $\Psi(N) = R(+1)\Psi_0(N - 1)$



EOM-SF $\Psi(M_S = 0) = R(M_S = -1)\Psi_0(M_S = 1)$

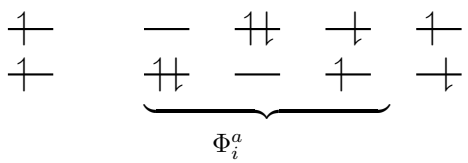


Figure 7.1: In the EOM formalism, target states Ψ are described as excitations from a reference state Ψ_0: $\Psi = R\Psi_0$, where R is a general excitation operator. Different EOM models are defined by choosing the reference and the form of the operator R. In the EOM models for electronically excited states (EOM-EE, upper panel), the reference is the closed-shell ground state Hartree-Fock determinant, and the operator R conserves the number of α and β electrons. Note that two-configurational open-shell singlets can be correctly described by EOM-EE since both leading determinants appear as single electron excitations. The second and third panels present the EOM-IP/EA models. The reference states for EOM-IP/EA are determinants for $N + 1/N - 1$ electron states, and the excitation operator R is ionizing or electron-attaching, respectively. Note that both the EOM-IP and EOM-EA sets of determinants are spin-complete and balanced with respect to the target multi-configurational ground and excited states of doublet radicals. Finally, the EOM-SF method (the lowest panel) employs the high-spin triplet state as a reference, and the operator R includes spin-flip, *i.e.*, does not conserve the number of α and β electrons. All the determinants present in the target low-spin states appear as single excitations, which ensures their balanced treatment both in the limit of large and small HOMO/LUMO gaps. Other EOM methods available in Q-CHEM are EOM-2SF, EOM-DIP, and EOM-DEA.

7.10.3 EOM-XX-CCSD and CI Suite of Methods

Q-CHEM features the most complete set of EOM-CCSD models,[140] enabling accurate, robust, and efficient calculations of electronically excited states (EOM-EE-CCSD or EOM-EE-OD);[131,142,150,226,235]; ground and excited states of diradicals and triradicals (EOM-SF-CCSD and EOM-SF-OD);[137,150] ionization potentials and electron attachment energies, as well as problematic doublet radicals and cation or anion radicals (EOM-IP/EA-CCSD).[192,229,236] The EOM-DIP-CCSD, EOM-2SF-CCSD, and EOM-DEA-CCSD methods are available as well. Conceptually, EOM is very similar to configuration interaction (CI): target EOM states are found by diagonalizing the similarity transformed Hamiltonian $\bar{H} = e^{-T} H e^{T}$,

$$\bar{H} R = E R, \tag{7.86}$$

where T and R are general excitation operators with respect to the reference determinant $|\Phi_0\rangle$. In the EOM-CCSD models, T and R are truncated at single and double excitations, and the amplitudes T satisfy the CC equations for the reference state $|\Phi_0\rangle$:

$$\langle \Phi_i^a | \bar{H} | \Phi_0 \rangle = 0 \tag{7.87}$$

$$\langle \Phi_{ij}^{ab} | \bar{H} | \Phi_0 \rangle = 0 \tag{7.88}$$

The computational scaling of EOM-CCSD and CISD methods is identical, *i.e.*, $\mathcal{O}(N^6)$, however EOM-CCSD is numerically superior to CISD because correlation effects are "folded in" in the transformed Hamiltonian, and because EOM-CCSD is rigorously size-intensive.

By combining different types of excitation operators and references $|\Phi_0\rangle$, different groups of target states can be accessed as explained in Fig. 7.1. For example, electronically excited states can be described when the reference $|\Phi_0\rangle$ corresponds to the ground state wave function, and operators $\hat{R}$ conserve the number of electrons and a total spin.[235] In the ionized/electron attached EOM models,[192,236] operators R are not electron conserving (*i.e.*, include different number of creation and annihilation operators)—these models can accurately treat ground and excited states of doublet radicals and some other open-shell systems. For example, singly ionized EOM methods, *i.e.*, EOM-IP-CCSD and EOM-EA-CCSD, have proven very useful for doublet radicals whose theoretical treatment is often plagued by symmetry breaking. Finally, the EOM-SF method[137,150] in which the excitation operators include spin-flip allows one to access diradicals, triradicals, and bond-breaking.[141]

Q-CHEM features EOM-EE/SF/IP/EA/DIP/DEA /DSF-CCSD methods for both closed and open-shell references (RHF/UHF/ROHF), including frozen core/virtual options. For EE, SF, IP, and EA, a more economical flavor of EOM-CCSD is available (EOM-MP2 family of methods). All EOM models take full advantage of molecular point group symmetry. Analytic gradients are available for RHF and UHF references, for the full orbital space, and with frozen core/virtual orbitals.[151] Properties calculations (permanent and transition dipole moments and angular momentum projections, $\langle \hat{S}^2 \rangle$, $\langle \hat{R}^2 \rangle$, *etc.*) are also available. The current implementation of the EOM-XX-CCSD methods enables calculations of medium-size molecules, *e.g.*, up to 15–20 heavy atoms. Using RI approximation (Section 6.12.9) or Cholesky decomposition (Section 6.12.10) helps to reduce integral transformation time and disk usage enabling calculations on much larger systems. EOM-MP2 and EOM-MP2t variants are also less computationally demanding. The computational cost of EOM-IP calculations can be considerably reduced (with negligible decline in accuracy) by truncating virtual orbital space using FNO scheme (see Section 7.10.13).

7.10.3.1 EOM-CC and Projection-Based Embedding

Due to the high computational cost of this method, the application of EOM-CCSD to large systems is difficult in terms of both computational time and resources. One strategy to overcome this issue is to combine EOM-CC with projection-based embedding[160,200]. This method performs remarkably well for the calculation of ionized, core-ionized and valence excitation energies. Rydberg states are also described with sufficient accuracy, while it is not advised to

employ projection-based embedding combined with EOM-EA-CCSD. Theory and job commands for projection-based embedding are reported in Section 11.6.

7.10.3.2 Legacy Features Available in CCMAN

The CCMAN module of Q-CHEM includes two implementations of EOM-IP-CCSD. The proper implementation[206] is used by default is more efficient and robust. The EOM_FAKE_IPEA keyword invokes is a pilot implementation in which EOM-IP-CCSD calculation is set up by adding a very diffuse orbital to a requested basis set, and by solving EOM-EE-CCSD equations for the target states that include excitations of an electron to this diffuse orbital. The implementation of EOM-EA-CCSD in CCMAN also uses this trick. Fake IP/EA calculations are only recommended for Dyson orbital calculations and debug purposes. (CCMAN2 features proper implementations of EOM-IP and EOM-EA (including Dyson orbitals)).

A more economical CI variant of EOM-IP-CCSD, IP-CISD is also available in CCMAN. This is an $\mathcal{O}(N^5)$ approximation of IP-CCSD, and can be used for geometry optimizations of problematic doublet states.[81]

7.10.4 EOM-XX-CC2

A more economical flavor of EOM-CCSD is CC2 linear response theory[49], which can also be interpreted as EOM-CC2. The double amplitudes for the reference state are approximated using the CC2 equations (see Section 6.12.4) and the equations for the target states are modified accordingly. This means that Eqs. 7.86 to 7.88 are valid for EOM-XX-CC2 as well but the elements of $\bar{H}$ are defined differently.

Q-CHEM features EOM-EE/SF/IP/EA-CC2 methods for both closed and open-shell references (RHF/UHF/ROHF), including frozen core option. The RI approximation can also be applied by specifying an auxiliary basis set. Only energies and no state or transition properties are available at the moment.

7.10.5 Spin-Flip Methods for Di- and Triradicals

The spin-flip method[137–139] addresses the bond-breaking problem associated with a single-determinant description of the wave function. Both closed and open shell singlet states are described within a single reference as spin-flipping, (*e.g.*, $\alpha \rightarrow \beta$ excitations from the triplet reference state), for which both dynamical and non-dynamical correlation effects are smaller than for the corresponding singlet state. This is because the exchange hole, which arises from the Pauli exclusion between same-spin electrons, partially compensates for the poor description of the coulomb hole by the mean-field Hartree-Fock model. Furthermore, because two α electrons cannot form a bond, no bond breaking occurs as the internuclear distance is stretched, and the triplet wave function remains essentially single-reference in character. The spin-flip approach has also proved useful in the description of di- and tri-radicals as well as some problematic doublet states.

The spin-flip method is available for the CIS, CIS(D), CISD, CISDT, OD, CCSD, and EOM-(2,3) levels of theory and the spin complete SF-XCIS (see Section 7.2.3.2). An $\mathcal{O}(N^7)$ non-iterative triples corrections are also available. For the OD and CCSD models, the following non-relaxed properties are also available: dipoles, transition dipoles, eigenvalues of the spin-squared operator ($\langle \hat{S}^2 \rangle$), and densities. Analytic gradients are also for SF-CIS and EOM-SF-CCSD methods. Construction of effective Hamiltonians in Heisenberg and Hubbard spaces from EOM-SF wave functions is described in the Section 13.6. To invoke a spin-flip calculation the SF_STATES *$rem* should be used, along with the associated *$rem* settings for the chosen level of correlation by using METHOD (recommended) or using older keywords (CORRELATION, and, optionally, EOM_CORR). Note that the high multiplicity triplet or quartet reference states should be used.

Several double SF methods have also been implemented.[44] To invoke these methods, use DSF_STATES.

7.10.6 EOM-DIP-CCSD

Double-ionization potential (DIP) is another non-electron-conserving variant of EOM-CCSD.[143,144,268] In DIP, target states are reached by detaching two electrons from the reference state:

$$\Psi_k = \hat{R}_{N-2}\Psi_0(N+2), \tag{7.89}$$

and the excitation operator R has the following form:

$$\hat{R} = \frac{1}{2}\sum_{ij} r_{ij} ji + \frac{1}{6}\sum_{ijka} r^a_{ijk} a^\dagger kji. \tag{7.90}$$

As a reference state in the EOM-DIP calculations one usually takes a well-behaved closed-shell state. EOM-DIP is a useful tool for describing molecules with electronic degeneracies of the type "$2n-2$ electrons on n degenerate orbitals". The simplest examples of such systems are diradicals with two-electrons-on-two-orbitals pattern. Moreover, DIP is a preferred method for four-electrons-on-three-orbitals wave functions.

Accuracy of the EOM-DIP-CCSD method is similar to accuracy of other EOM-CCSD models, *i.e.*, 0.1–0.3 eV. The scaling of EOM-DIP-CCSD is $\mathcal{O}(N^6)$, analogous to that of other EOM-CCSD methods. However, its computational cost is less compared to, *e.g.*, EOM-EE-CCSD, and it increases more slowly with the basis set size. An EOM-DIP calculation is invoked by using DIP_STATES, or DIP_SINGLETS and DIP_TRIPLETS. In certain circumstances, the DIP_AA_STATES, DIP_BB_STATES, DIP_BA_STATES keywords can be used.

Note: The performance of EOM-DIP may be poor if the reference state is unstable with respect to electron detachment. See Section 7.10.12 for details.

Note: In some applications of EOM-DIP-CCSD, only $2h$ operators were included in the EOM part. These calculations correspond to energies obtained from EOM_PRECONV_DOUBLES = TRUE calculation.

7.10.7 EOM-DEA-CCSD

In the EOM-DEA method, the target states are described by $2p$ and $3p1h$ operators acting on $N-2$ electron reference[204]:

$$\Psi_k = \hat{R}_{N+2}\Psi_0(N-2), \tag{7.91}$$

and the excitation operator R has the following form:

$$\hat{R} = \frac{1}{2}\sum_{ab} r_{ab} a^\dagger b^\dagger + \frac{1}{6}\sum_{iabc} r^{abc}_i a^\dagger b^\dagger c^\dagger i. \tag{7.92}$$

EOM-DEA is useful for calculating diradical states including excited states beyond the SF manifold. In calculations of neutral diradicals, EOM-DEA should use +2 charged reference state. EOM-DEA is also suitable for describing certain types of doubly excited states, such as $\ldots(\pi)^0(\pi^*)^2$ in ethylene. An EOM-DEA calculation is invoked by using DEA_STATES, or DEA_SINGLETS and DEA_TRIPLETS. In more exotic calculations, such as EOM-DEA for open-shell references, DEA_AA_STATES, DEA_BB_STATES, and DEA_AB_STATES keywords might be useful. Both EOM-CCSD and EOM-MP2 variants are available.

Note: In some applications of EOM-DEA-CCSD, only $2p$ operators were included in the EOM part. These calculations correspond to energies obtained from EOM_PRECONV_DOUBLES = TRUE calculation.

7.10.8 EOM-CC Calculations of Core-Level States

The core-valence separation (CVS) scheme[47] allows one to extend standard methods for excited and ionized states to the core-level states. In this approach, the excitations involving core electrons are decoupled from the rest of the configurational space. This allows one to reduce computational costs and decouple the highly excited core states from the continuum. Currently, CVS is implemented within EOM-EE/SF/IP-CCSD for energies and transition properties (oscillator strengths, NTOs, Dyson orbitals, exciton descriptors, ECD moments). CVS-EOM-EE-CCSD can be used to model NEXAFS, while CVS-EOM-IP-CCSD can be used to model XPS and XES. These methods can also be used to compute transient absorption spectra, *e.g.*, valence pump/X-ray probe experiments.[258] The calculation of non-linear spectra, such as RIXS, is also possible. L-edge spectra (XAS and XPS) can be described by using state-interaction approach in which spin-orbit coupling is evaluated using non-relativistic CVS-EOM-EE states[259]. Auger spectra can be computed using CVS-EOM-EE combined with the explicit treatment of the continuum via Feshbach-Fano approach[230] (see Section 7.10.10).

In Q-CHEM, a slightly different version of CVS-EOM-EE-CCSD than the original theory by Coriani and Koch[54] is implemented: the reference coupled-cluster amplitudes do not include core electrons.[257] To distinguish this method from the original,[54] we refer in what follows to the Q-CHEM implementation as frozen-core-ground-state/core-valence-separated EOM (FC-CVS-EOM) approach.[257]

In the FC-CVS-EOM approach the ground-state parameters (amplitudes and Lagrangian multipliers) are computed within the frozen-core approximation, whereas the core-excitation energies and transition strengths are obtained imposing that at least one index in the EOM excitation (and ionization) operators refer to a core occupied orbital. Likewise, CVS is enforced in the calculation of the response states in RIXS.[191]

To ensure the best convergence of EOM equations, the calculation is *edge-specific* with respect to the highest lying edges (or deepest lying core orbitals): the frozen-core and CVS spaces are selected for each edge such that the core orbitals we are addressing in the excited state calculations are explicitly frozen in the ground state calculation and specifically included in the EOM calculation. Examples 7.63 and 7.64 below illustrate this point.

Although the convergence of FC-CVS-EOM is much more robust that that of regular EOM-CCSD, sometimes calculations would collapse to low-lying artificial states. If this happens, rerun the calculation using CVS_EOM_SHIFT to specify an approximate onset of the edge.

To invoke the CVS approximation, use METHOD = CCSD and CVS_EE_STATES instead of EE_STATES to specify the desired target states (likewise, CVS_EE_SINGLETS and CVS_EE_TRIPLETS can be used in exactly the same way as in regular EOM calculations). For ionized states, use CVS_IP_STATES or CVS_IP_ALPHA/CVS_IP_BETA. Spin-flip version can be deployed by using CVS_SF_STATES (this may be needed for computing SOCs and L-edge spectra). Preconverging single amplitudes can be invoked by CVS_EOM_PRECONV_SINGLES. Transition properties and Dyson orbitals can be computed either within CVS manifold or between CVS and valence manifolds (see Section 7.10.28 for definition of Dyson orbitals). CVS-EOM-CCSD is only available with CCMAN2.

Note: (1) Core electrons must be frozen in CVS-EOM calculations. The exact definition of the core depends on the edge, so using default values may be not appropriate.
(2) The default setting (N_FROZEN_CORE = FC) does not work correctly in QM/MM calculations. One should specify the number of frozen core orbitals explicitly.

It is possible to set up customized CVS calculations for ionization/excitation originating from a particular core orbital. To do so, one should first use *$reorder_mo* feature to reorder orbitals such that the desired core orbital appears the first in the list and then run a CVS-EOM calculation with N_FROZEN_CORE = 1.

Note: *$reorder_mo* works only with old SCF code (GEN_SCFMAN=FALSE).

It is also possible to split the frozen-core space into CVS-active and CVS-inactive subspaces using the keyword

N_FC_CVS_INACTIVE, which defines the size of the CVS-active subspace as N_FROZEN_CORE minus N_FC_CVS_INACTIVE. These CVS-active orbitals are arranged such that they precede the valence orbital subspace but follow the CVS-inactive orbitals in the molecular orbital ordering (use *$reorder_mo* section to customize if required).

CVS_EOM_SHIFT

Specifies energy shift in CVS-EOM calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n corresponds to $n \times 10^{-3}$ hartree shift (*i.e.*, 11000 = 11 hartree); solve for eigenstates around this value.

RECOMMENDATION:

Improves the stability of the calculations.

N_FC_CVS_INACTIVE

Number of frozen-core CVS inactive orbitals

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n, where $0 \leq$ n $\leq$ total FC orbitals

RECOMMENDATION:

Useful in cluster calculations.

CVS_EOM_PRECONV_SINGLES

When not zero, singly excited vectors are converged prior to a full excited states calculation (CVS states only). Sets the maximum number of iterations for pre-converging procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not pre-converge

1 pre-converge singles

RECOMMENDATION:

Sometimes helps with problematic convergence.

CVS_EE_SINGLETS

Sets the number of singlet core-excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_EE_TRIPLETS

Sets the number of triplet core-excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_SF_STATES

Sets the number of core-level spin-flip target states roots to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i SF states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_IP_STATES

Sets the number of core-ionized states to find. By default, β electron will be removed.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP states.

OPTIONS:

[i,j,k...] Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_IP_ALPHA

Sets the number of ionized target states derived by removing α electron ($M_S = -\frac{1}{2}$).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/α states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_IP_BETA

Sets the number of ionized target states derived by removing β electron ($M_S = \frac{1}{2}$, default for CVS-IP).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/β states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

7.10.8.1 EOM-CC Calculations of RIXS

RIXS is a coherent two-photon process involving core-level states.[191] The calculations of RIXS cross-sections require solving response equations, in the same fashion as in calculations of 2PA cross-sections (see Section 7.10.20.6). Currently, only calculations of RIXS cross-sections between the CCSD reference and the EOM-CCSD target states are possible. Because of the resonant nature of RIXS, damped response theory is used to handle singularities in the resolvent. In addition, CVS is enforced on the response vectors to eliminate their coupling with the ionization continuum. To set up RIXS calculation, one needs to set METHOD = EOM-CCSD and to specify the number of excited states to be included by using EE_STATES (note that only 2PA bright states need to be included) and to activate CVS by using CVS_EE_STATES asking for zero states. RIXS calculations is deployed by activating CC_EOM_RIXS. The option for performing wave-function analysis (*e.g.*, NTO analysis) of the complex RIXS one-particle transition density matrices is also available. Conceptual details can be found in Ref. 188. Finally, one can request to compute REXS (cross-section for the "elastic" peak) by CC_REF_PROP = 1 (by default, REXS is not calculated) This feature is only available within fc-CVS-EOM-EE-CCSD.

CC_EOM_RIXS

Whether or not the RIXS scattering moments and cross-sections will be calculated.

TYPE:

INTEGER

DEFAULT:

0 do not compute RIXS cross-sections

OPTIONS:

1 Perform RIXS within fc-CVS-EOM-EE-CCSD using the response wave functions of the CCSD reference state only

2 Perform RIXS within fc-CVS-EOM-EE-CCSD response theory along with the wave-function analysis of RIXS transition density matrices

11 Perform RIXS within the standard EOM-EE-CCSD using the response wave functions of the CCSD reference state only

12 Use σ-intermediates for RIXS response calculations within the standard EOM-EE-CCSD

RECOMMENDATION:

Use 1 to deploy fc-CVS-EOM-EE-CCSD with robust convergence

Note: Since the RIXS response solutions within EOM-EE-CCSD often do not converge, fc-CVS-EOM-EE-CCSD RIXS calculations with CC_EOM_RIXS = 1 are recommended for smooth convergence. All other options are experimental.

To specify pumping frequencies and damping factor ϵ, use *$rixs* section:

```
$rixs
damped_epsilon 0.005       Damping factor in hartree.
omega_1 2306503 500 10 0   Pumping frequencies: starting w (cm-1), increment (cm-1),
                           number of frequency points, algo (always 0 for now)
omega_2 2200000 600 20 0   Emitted frequencies:  starting w (cm-1), increment (cm-1),
                           number of frequency points, algo (always 0 for now)
$end
```

Damping factor (DAMPED_EPSILON) is specified in atomic units (0.005 is a good choice). OMEGA_1 specifies the starting pumping frequency (in cm^{-1}), the increment in frequency scan (in cm^{-1}), number of frequency points, and the algorithm for response equation (use zero until further notice). OMEGA_2 should be specified only for generating 2D RIXS scans. Example 7.70 illustrates the setup of RIXS calculations.

By default, the damped RIXS response wave functions are computed using a new iterative, Davidson-like subspace procedure called the DALTON solver.[50,128] It is controlled by the DALTON_XCONV, DALTON_PRECOND_START, DALTON_MAXITER, and DALTON_MAXSPACE keywords. Damped RIXS response wave functions can also be computed using the older DIIS procedure, by setting DAMPED_DALTON_SOLVER = FALSE. The DIIS solver is controlled by the CC_MAX_ITER, CC_DIIS_START, CC_DIIS_SIZE, CC_EOM_2PA_ECONV, and CC_EOM_2PA_XCONV.

Note:

- For better convergence of the response equations with DIIS solver, increase CC_DIIS_SIZE (25 is recommended) and consider using a larger damping factor.
- The DALTON solver uses square of the norm of the residual in estimating convergence (as in the Davidson solver), whereas the DIIS solver uses the norm of the difference of the response vector between iterations.

7.10.8.2 Examples

The following examples illustrate how to perform these calculations.

- Example 7.63: the $1s$ orbital on the oxygen atom is frozen in the CCSD calculation (N_FROZEN_CORE = FC). In the EOM calculation, the CVS approximation is invoked (CVS_EE_SINGLETS), so that the core-excitation energies are obtained as the lowest excitations. The calculation of the oscillator strengths and rotatory strengths is activated by selecting CC_TRANS_PROP = 1 and the LIBWFA analysis is invoked by STATE_ANALYSIS = TRUE (see Section 10.2.12). Note that rotatory strengths will be zero for non-chiral systems.

- Example 7.64: CVS-EOM-EE-CCSD calculations in a two-edge molecule (carbon monoxide). In the present implementation, the calculation should be done separately for each edge. The first job computes carbon-edge states. Since the carbon $1s$ orbital is the highest in energy (among the core $1s$ orbitals of the molecule), the input for the C-edge is similar to example 7.63. Both the oxygen's and the carbon's $1s$ orbitals are frozen in the reference CCSD calculation. In the EOM part, the carbon core-excited states are automatically selected. In this case, using default frozen core settings (N_FROZEN_CORE = FC) is equivalent to specifying N_FROZEN_CORE = 2. In the second input, the oxygen edge is computed. As the core-orbitals of oxygen lie deeper, the frozen core and CVS selection specifically targets the oxygen edge by using a smaller core. The $1s$ orbital of the oxygen atom is selected by N_FROZEN_CORE = 1. If the molecule has other edges, the deepest lying core orbitals, up to and including those of the edge of interest, should be selected by an appropriate value of N_FROZEN_CORE.

- Example 7.65: calculation of the XAS spectrum of ammonia using additional set of Rydberg functions placed on the nitrogen; the exponents of the Rydberg functions were generated using Kaufmann's recipe[129]. Note that when ghost atoms are used, the symmetry is turned off. It can be forced on by using FORCE_SYMMETRY_ON keyword. Alternatively, this calculation can be set up by adding additional diffuse functions to all (or selected) atoms using even-tempered exponents.

- Example 7.66: calculation of the XES spectrum of benzene.

- Examples 7.67 and 7.68: calculations of Dyson orbitals between core-excited and core-ionized states and between core-excited and valence-ionized states.

- Example 7.70: setup of RIXS calculations.

Calculations of L-edge spectra using state-interaction approach[259] entails a two-step procedure. First, Q-CHEM computes necessary CVS-EOM states and SOCs. In the second step, the spin-orbit perturbed spectrum is computed by a

post-processing script. Scripts, detailed instructions, and examples can be found elsewhere.[256]

Example 7.63 FC-CVS-EOM-EE-CCSD calculation of the first six dipole-allowed core excitation energies and their intensities at the oxygen edge of water. Wave-function analysis is also performed.

```
$molecule
   0 1
   O    0.0000   0.0000   0.1173
   H    0.0000   0.7572  -0.4692
   H    0.0000  -0.7572  -0.4692
$end

$rem
   METHOD              = eom-ccsd
   BASIS               = aug-cc-pVDZ
   CVS_EE_SINGLETS     = [3,0,2,1]
   N_FROZEN_CORE       = fc
   CC_TRANS_PROP       = true
   EOM_PRECONV_SINGLES = true
   STATE_ANALYSIS      = true !invoke libwa to compute NTOs and exciton descriptors
   MOLDEN_FORMAT       = true
   NTO_PAIRS           = 3
   POP_MULLIKEN        = true
$end
```

Example 7.64 FC-CVS-EOM-EE-CCSD calculations of the first two dipole-allowed core excitation energies per irreducible representation and their intensities at (1) carbon and (2) oxygen edges of carbon monoxide.

```
$molecule
   0 1
   O    0.0000   0.0000    0.913973
   C    0.0000   0.0000   -1.218243
$end

$rem
   METHOD               =  eom-ccsd
   BASIS                =  aug-cc-pVDZ
   INPUT_BOHR           =  true
   CVS_EE_SINGLETS      =  [2,0,2,2]
   N_FROZEN_CORE        =  fc            !For C K-edge
   EOM_PRECONV_SINGLES  =  true
   CC_TRANS_PROP        =  true
$end

@@@

$molecule
   read
$end

$rem
   METHOD               =  eom-ccsd
   BASIS                =  aug-cc-pVDZ
   CVS_EE_SINGLETS      =  [2,0,2,2]
   N_FROZEN_CORE        =  1             !For O K-edge
   EOM_PRECONV_SINGLES  =  true
   CC_TRANS_PROP        =  true
$end
```

Example 7.7.65 Calculation of XAS spectrum of ammonia using additional Rydberg functions placed on the ghost atom.

View input online

Example 7.66 Calculation of XES spectrum of benzene

```
$comment
RI-MP2/cc-pVTZ optimized geometry. XES calculation.
$end

$molecule
0 1
    H         2.4750347531    0.0000000000    0.0000000000
    C         1.3935929418    0.0000000000    0.0000000000
    C         0.6967964709    1.2068868901    0.0000000000
    H         1.2375173766    2.1434429715    0.0000000000
    C        -0.6967964709    1.2068868901    0.0000000000
    H        -1.2375173766    2.1434429715    0.0000000000
    C        -1.3935929418    0.0000000000    0.0000000000
    H        -2.4750347531    0.0000000000    0.0000000000
    C        -0.6967964709   -1.2068868901    0.0000000000
    H        -1.2375173766   -2.1434429715    0.0000000000
    C         0.6967964709   -1.2068868901    0.0000000000
    H         1.2375173766   -2.1434429715    0.0000000000
$end

$rem
   BASIS         =  6-31G*
   METHOD        =  eom-ccsd
   IP_STATES     = [3,2,1,1,0,1,2,2] !All valence Koompans-like ionized states
                                     !except for 3 lowest ones
   CVS_IP_STATES = [2,1,0,0,0,0,1,2] !All core-ionized states
   CC_TRANS_PROP = 2 !Compute transitions between all pairs of EOM states
   CC_MEMORY     = 8000  !8 GB
$end
```

Example 7.67 Calculation of Dyson orbitals between FC-CVS-EOM-EE-CCSD and FC-CVS-EOM-IP-CCSD manifolds.

```
$comment
   CVS-IP/CVS-EE Dyson orbitals, formaldehyde
$end

$molecule
   0 1
   C
   H   1  1.096135
   H   1  1.096135  2  116.191164
   O   1  1.207459  2  121.904418  3  -180.000000  0
$end

$rem
   METHOD           =  eom-ccsd
   BASIS            =  cc-pVDZ   ! Please do not use BASIS2
   SCF_CONVERGENCE  =  8
   CVS_IP_STATES    =  [1,0,0,0]
   CVS_EE_STATES    =  [1,0,1,0]
   CC_DO_DYSON      =  true
   CC_TRANS_PROP    =  2 !Compute all EOM-to-EOM transitions
$end
```

Example 7.68 Calculation of Dyson orbitals between FC-CVS-EOM-EE-CCSD and EOM-IP-CCSD manifolds.

```
$comment
   IP/CVS-EE Dyson orbitals, formaldehyde
$end

$molecule
   0 1
   C
   H  1  1.096135
   H  1  1.096135  2  116.191164
   O  1  1.207459  2  121.904418  3  -180.000000  0
$end

$rem
   METHOD           =  eom-ccsd
   BASIS            =  cc-pVDZ
   SCF_CONVERGENCE  =  8
   IP_STATES        =  [1,0,0,0]   ! Valence a1 hole
   CVS_EE_STATES    =  [1,0,0,0]
   CC_DO_DYSON      =  true
   CC_TRANS_PROP    =  2 !Compute all EOM-to-EOM transitions
$end
```

Example 7.69 Calculation of Dyson orbitals between FC-CVS-EOM-EE-CCSD and FC-CVS-EOM-IP-CCSD manifolds.

```
$comment
CVS-IP/CVS-EE Dyson orbitals, formaldehyde
$end

$molecule
   0 1
   C
   H  1  1.096135
   H  1  1.096135  2  116.191164
   O  1  1.207459  2  121.904418  3  -180.000000  0
$end

$rem
   BASIS           =  cc-pVDZ
   SCF_CONVERGENCE = 8
   METHOD          = eom-ccsd
   IP_STATES       = [1,0,0,0]   ! Valence a1 hole
   CVS_EE_STATES   = [1,0,0,0]
   CC_DO_DYSON     = true
   CC_TRANS_PROP   = true        ! required to activate a Dyson orbitals job
$end
```

Example 7.70 Calculation of RIXS/REXS for benzene (10 excited states per 2PA active irrep)

```
$comment
RI-MP2/cc-pVTZ optimized geometry.
Pump XAS transition peak A at 285.97 eV, only one frequency point.
$end

$molecule
   0 1
   H      2.4750347531    0.0000000000    0.0000000000
   C      1.3935929418    0.0000000000    0.0000000000
   C      0.6967964709    1.2068868901    0.0000000000
   H      1.2375173766    2.1434429715    0.0000000000
   C     -0.6967964709    1.2068868901    0.0000000000
   H     -1.2375173766    2.1434429715    0.0000000000
   C     -1.3935929418    0.0000000000    0.0000000000
   H     -2.4750347531    0.0000000000    0.0000000000
   C     -0.6967964709   -1.2068868901    0.0000000000
   H     -1.2375173766   -2.1434429715    0.0000000000
   C      0.6967964709   -1.2068868901    0.0000000000
   H      1.2375173766   -2.1434429715    0.0000000000
$end

$rem
   BASIS         = 6-31(+,+)G**
   METHOD        = eom-ccsd
   CVS_EE_STATES = [0,0,0,0,0,0,0,0]       just to invoke CVS
   EE_STATES     = [10,10,10,10,0,0,0,0]   10 states in each 2PA active irrep
   CC_REF_PROP   = 1     ! Calculate REXS in addition to RIXS
   CC_EOM_RIXS   = 1     ! Activate RIXS calculation using fc-CVS-EOM-EE-CCSD
   CC_DIIS_SIZE  = 25    ! Use for better convergence of response equations
   CC_MEMORY     = 8000  !8 GB
   mem_total = 8500
$end

$rixs
   omega_1         2306503 500 1 0
   damped_epsilon  0.005
$end
```

Example 7.71 FC-CVS-EOM-EE-CCSD and FC-CVS-EOM-IP-CCSD calculation for a water dimer with core excitations and ionization arising from the energetically lowest core orbital with the other core orbital kept CVS inactive. The first input reorders MOs. The second input defines CVS active and inactive orbitals before computing core-level states and properties.

```
$molecule
0 1
          O      -4.9920582819    -0.1246131900     2.2067549109
          H      -5.5873205924    -0.9397949066     2.3252790559
          H      -4.9827539950     0.0559440223     1.1016406903
          O      -6.5474306925    -1.1803213469    -1.0374284134
          H      -5.7367207770    -0.6839235793    -0.7295169422
          H      -6.2716956294    -2.1185636950    -1.1190969125
$end

$rem
   GEN_SCFMAN = false
   use_libqints = true
   CC_MEMORY = 4000
   MEM_STATIC = 600
   BASIS  =  6-31g
   METHOD  =  hf
   n_frozen_core = 0
   thresh =  14          Does not converge with defaults
   scf_algorithm gdm     Does not converge with defaults
$end

$reorder_mo
2 1 3
2 1 3
$end

@@@

$molecule
0 1
          O      -4.9920582819    -0.1246131900     2.2067549109
          H      -5.5873205924    -0.9397949066     2.3252790559
          H      -4.9827539950     0.0559440223     1.1016406903
          O      -6.5474306925    -1.1803213469    -1.0374284134
          H      -5.7367207770    -0.6839235793    -0.7295169422
          H      -6.2716956294    -2.1185636950    -1.1190969125
$end

$rem
   CC_MEMORY = 4000
   MEM_STATIC = 600
   BASIS  =  6-31g
   thresh =  14          Does not converge with defaults
   n_frozen_core = 2
   scf_guess = read
   max_scf_cycles = 0
   method = eom-ccsd
   cvs_ee_states = [2]
   cvs_ip_states = [1]
   n_fc_cvs_inactive = 1
   cc_trans_prop = 1
   cc_ref_prop = 1
   cc_eom_prop = 1
$end
```

7.10.9 EOM-CC Calculations of Metastable States

While conventional coupled-cluster and equation-of-motion methods allow one to tackle electronic structure ranging from well-behaved closed shell molecules to various open-shell and electronically excited species,[140] meta-stable electronic states, so-called resonances, present a difficult case for theory. By using complex scaling and complex absorbing potential techniques, we extended these powerful methods to describe auto-ionizing states, such as transient anions, highly excited electronic states, and core-ionized species.[27,122,123] CC and EOM-CC calculations can also be carried out using complex basis functions (CBFs), as described in Sections 4.9.5 and 8.7. In addition, users can employ stabilization techniques using charged sphere and scaled atomic charges options.[144] These methods are only available within CCMAN2. The complex CC/EOM code is engaged by COMPLEX_CCMAN; the specific parameters should be specified in the *$complex_ccman* section.

COMPLEX_CCMAN

Requests complex-scaled or CAP-augmented CC/EOM calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Engage complex CC/EOM code.

RECOMMENDATION:

Not available in CCMAN. Need to specify CAP strength or complex-scaling parameter in *$complex_ccman* section.

The *$complex_ccman* section is used to specify the details of the complex-scaled/CAP calculations, as illustrated below. If user specifies CS_THETA, complex scaling calculation is performed.

```
$complex_ccman
   CS_THETA  10   Complex-scaling parameter theta=0.01, r->r exp(-i*theta)
   CS_ALPHA  10   Real part of the scaling parameter alpha=0.01,
!                 r->alpha r exp(-itheta)
$end
```

Alternatively, for CAP calculations, the CAP parameters need to be specified.

```
$complex_ccman
   CAP_ETA  1000  CAP strength in 10-5 a.u. (0.01)
   CAP_X    2760  CAP onset along X in 10^-3 bohr (2.76 bohr)
   CAP_Y    2760  CAP onset along Y in 10^-3 bohr (2.76 bohr)
   CAP_Z    4880  CAP onset along Z in 10^-3 bohr (4.88 bohr)
   CAP_TYPE 1     Use cuboid cap (CAP_TYPE=0/2 will use spherical/Voronoi CAP)
$end
```

One can also add real absorbing potential by using CAP_RE_ETA; it follows the same format as CAP_ETA. For example, this setup would add purely real absorbing potential with η=0.01:

```
$complex_ccman
   CAP_ETA     0000  CAP strength in 10-5 a.u. (0.00)
   CAP_RE_ETA  1000  real CAP strength in 10-5 a.u. (0.01)
   CAP_X    2760  CAP onset along X in 10^-3 bohr (2.76 bohr)
   CAP_Y    2760  CAP onset along Y in 10^-3 bohr (2.76 bohr)
   CAP_Z    4880  CAP onset along Z in 10^-3 bohr (4.88 bohr)
   CAP_TYPE 1     Use cuboid cap (CAP_TYPE=0/2 will use spherical/Voronoi CAP)
$end
```

The CAP_TYPE field specifies the type of the CAP. The current options are: spherical CAP (CAP_TYPE = 0), cuboid CAP (CAP_TYPE = 1), and smooth Voronoi[233] CAP (CAP_TYPE = 2). In the calculations with a Voronoi CAP, the onset is specified by the CAP_X variable.

CS_THETA is specified in radian$\times$ 10^{-3}. CS_ALPHA, CAP_X/Y/Z are specified in a.u. $\times$ 10^{-3}, *i.e.*, CS_THETA = 10 means $\theta = 0.01$; CAP_ETA is specified in units of 10^{-5} E_h. The CAP is calculated by numerical integration and the default grid is ($N_r = 99, N_\Omega = 590$). For testing the accuracy of numerical integration, the numerical overlap matrix is calculated and compared to the analytical one. If the performance of the default grid is poor, the grid type can be changed using the keyword XC_GRID (see Section 5.5 for further details). When CAP calculations are performed, CC_EOM_PROP = 1 by default; this is necessary for calculating first-order perturbative correction.

EOM-CC with complex basis functions CBFs (see Section 4.9.5) can be enabled by setting COMPLEX_CCMAN = TRUE and enabling complex basis functions with COMPLEX_EXPONENTS = TRUE. As with mean-field calculations the complex basis must be specified as in described in Section 8.7.

Advanced users may find the following options useful. Several ways of conducing complex calculations are possible, *i.e.*, complex scaling/CAPs can be either engaged at all levels (HF, CCSD, EOM), or not. When applied at post Hartree-Fock level, CAP can either be added to all blocks of the Fock matrix or restricted to the virtual-virtual block only. The latter approach, known as projected CAP,[220] improves the stability of the calculation results with respect to CAP onset by reducing a CAP-induced perturbation on the target states through the occupied orbital space.

This type of CAP projection is currently implemented only for EE/EA calculations and is invoked by setting PROJ_CAP key in the *$complex_ccman* section as follows. PROJ_CAP = 1 deploys CAP/EOM-CCSD with projected CAP added

at the CCSD and EOM steps. PROJ_CAP = 2 deploys CAP/EOM-CCSD/MP2/MP2T with projected CAP added at the EOM step. The latter implies that T-amplitudes (Sec. 7.10.3) are obtained from a real-valued calculation (for zero CAP strength) and can be reused to generate complex eigenvalue trajectories by specifying ETA_STEP and NSTEPS parameters in *$complex_ccman*.

PROJ_CAP = 3 deploys another form of CAP projection[234] in which the CAP Hamiltonian is projected onto the subspace spanned by a set of pre-computed EOM eigenvectors. By default, the excited state eigenvectors are obtained from a single real-valued calculation, and the CAP matrix represented in the state basis is printed in the output for each irreducible representation. This functionality is available for all EOM-CC models for which transition properties between EOM target states are available. To generate eigenvalue trajectories, CAP_ETA should be set to a non-zero CAP strength, and subsequent points are specified using the ETA_STEP and NSTEPS parameters in *$complex_ccman*. Trajectories are written to a separate output file for each irreducible representation. Additionally, first-order perturbative corrections can be obtained by setting PROJ_PROP = 1. Note that when PROJ_PROP = 1, the initial set of real eigenvectors are obtained using the complex valued code at zero CAP strength. As such, first-order perturbative corrections are only available for complex EOM-CC models. The complete set of one-particle state and transition OPDMs between each pair of states (which is all that is required for CAP projection) will be exported to checkpoint file for further analysis when GUI = 2 and PROJ_CAP = 3.

By default, if COMPLEX_CCMAN is specified, the EOM calculations are conducted using complex code. Other parameters are set up as follows:

```
$complex_ccman
  CS_HF   = true
  CS_CCSD = true
$end
```

Alternatively, the user can disable complex HF. These options are experimental and should only be used by advanced users. For CAP-EOM-CC, only CS_HF = TRUE and CS_CCSD = TRUE is implemented.

Non-iterative triples corrections are available for all complex scaled and CAP-augmented CC/EOM-CC models and requested in analogy to regular CC/EOM-CC (see Section 7.10.26 for details).

To take account of the impact of the environment on electronic resonances CAP-EOM-CCSD and CBF-EOM-CCSD can be combined with projection-based embedding,[200] as described in Sections 7.10.3.1 and 11.6.

Molecular properties and transition moments are requested for complex scaled or CAP-augmented CC/EOM-CC calculations in analogy to regular CC/EOM-CC (see Section 7.10.20 for details). Natural orbitals and natural transition orbitals can be computed and the exciton wave-functions can be analyzed, similarly to real-valued EOM-CCSD (same keywords are used to invoke the analysis). Analytic gradients are available for complex CC/EOM-CC only for cuboid CAPs (CAP_TYPE = 1) introduced at the HF level (CS_HF = TRUE), as described in Ref. 17. The frozen core approximation is disabled for CAP-CC/EOM-CC gradient calculations. Geometry optimization can be requested in the same way as in regular CC/EOM-CC (see Section 7.10.20 for details).

7.10.9.1 Complex-valued RI-EOM-CCSD

To reduce the memory requirements, the RI-approximation can be invoked to decompose the electron repulsion integrals. The code has the same structure as for real-valued RI-EOM-CCSD[64] but uses a complex-valued RI decomposition[255]. The RI-approximation is available for CCSD and EOM-IP/EA/EE/SF-CCSD in combination with complex-scaling, complex basis functions and complex absorbing potentials. It can be invoked by setting AUX_BASIS to your favourite auxiliary basis set (*e.g.* RIMP2-aug-cc-pVTZ). For calculations with complex basis functions, COMPLEX_AUX_BASIS needs to specified as well, which is usually the same as AUX_BASIS. When using user-defined auxiliary basis sets, AUX_BASIS is specified in *$aux_basis* while COMPLEX_AUX_BASIS is specified in *$aux_zbasis*.

Complex-valued RI integrals are currently only available for energies and unrelaxed one-electron properties. It is also possible to use the RI-approximation for the determination of partial Auger decay widths by decomposing the CCSD energy or by the ACP method (see sections 7.10.10 and 7.10.11).

7.10.10 Auger Spectra and Lifetimes of Core-Level States

Certain types of resonances can be described by using real-valued EOM-CC wave functions via Feshbach–Fano approach.[66,70] In this section we describe the application of Feshbach–Fano approach to core-excited and core-ionized states.[230,231] Core-hole states, which are Feshbach resonances, are subject to autoionization—commonly known as Auger decay. Auger Electron Spectroscopy (AES) measures kinetic energy and intensity of ejected electrons. The energy of the Auger electrons are computed as differences between the intial core-hole state and final decay states. For example, for regular Auger decay:

$$E^{\mu n}_{\text{Auger}} = E^{\text{CoreIP}}_{n} - E^{\text{DIP}}_{\mu}, \tag{7.93}$$

where E^{CoreIP}_{n} and E^{DIP}_{μ} are energies of the initial (core-hole) and final (doubly ionized valence) states, respectively. The intensities of each decay channel are given by partial decay widths, $\Gamma_{\mu n}$. In the crudest approximation, one can assign each decay channel unit intensity—so-computed Auger spectra reflect density of doubly ionized states of the system. For more rigorous treatments, the decay widths can be computed by either the Feshbach–Fano or complex-variable approaches[230] for electronic resonances (the latter approach is described in sections 7.10.10 and 7.10.11).

The Feshbach–Fano theory[66,70] invokes two projection operators, $\hat{Q}$ and $\hat{P}$, which decompose the total wavefunction into bound-like and continuum-like components. In the case of core-level states this separation is enabled by invoking the CVS scheme and frozen-core approximation in the calculations of initial and final states in the Auger process (more details about CVS can be found in Section 7.10.8).

The initial (bound-like) state Ψ_0 is a core-hole ionized or core-hole excited state, which can be described by CVS-EOM-CC. The final (continuum-like) state χ_{μ,E_k} is represented by an antisymmetrized product of a stable channel state Ψ_μ (described by an appropriate EOM-CC model) and a continuum orbital ϕ_k, $\chi_{\mu,E_k} \sim \mathcal{A}\{\phi_k \Psi_\mu\}$. Note that Ψ_μ is a state with one electron less than Ψ_0. Two essential parameters defining AES are the rate of the decay into a channel μ, given as

$$\Gamma_\mu = 2\pi \langle \Psi^L_0 | \hat{H} - E_0 | \chi^R_{\mu,E_k} \rangle \langle \chi^L_{\mu,E_k} | \hat{H} - E_0 | \Psi^R_0 \rangle, \tag{7.94}$$

and partial energy correction Δ_μ to the zero-order resonance position E_0, defined as

$$\Delta_\mu = P.V. \int_0^\infty \frac{\langle \Psi^L_0 | \hat{H} - E_0 | \chi^R_{\mu,E} \rangle \langle \chi^L_{\mu,E} | \hat{H} - E_0 | \Psi^R_0 \rangle}{E_0 - E_\mu - E} dE. \tag{7.95}$$

In the expressions above $\hat{H}$ is the electronic Hamiltonian, E_μ is the energy of the channel state Ψ_μ, E_k is the energy of the ejected electron ($E_k = E_0 - E_\mu$), L/R superscripts denote left and right EOM-CCSD wavefunctions, and $P.V.$ stands for the Cauchy principle value. Calculations of Γ_μ are activated with the CC_DO_FESHBACH keyword. By default, the continuum orbital ϕ_k is approximated with a plane wave.[230,231]

It is also possible to model ϕ_k with a Coulomb wave by setting CC_FESHBACH_CW = 1. This option requires an additional input section *$coulomb_wave*, which provides the Coulomb wave parameters for the expansion of the Coulomb wave in terms of product of a plane wave and Gaussian-type functions (PW-CGTOs), as detailed in Ref. 230. The

parameters are listed below.

```
$coulomb_wave
   lmax   =   4      ! Maximum angular momentum in the pseudo partial wave expansion
   ncoeff =   6      ! Number of contraction coefficient for each \emph{l}
   Rmin   =   0.05   ! PW-CGTO grid in atomic unit
   Rmax   =   10
   Rmid   =   2
   Charge =   6.2    ! Effective charge of the Coulomb wave
   cwave_center = 1  ! Atom index for the position of Coulomb wave. Starts from 1.
   print_cw = false  ! For printing the partial wave and the PW-CGTO basis
$end
```

The Coulomb wave is approximated by the linear combination of PW-CGTOs. Typically, *lmax* = 4-6 and *ncoeff* = 6-8 provide a good approximation; increasing these values significantly increases computational cost. The grid is set up by the *Rmin*, *Rmax*, and *Rmid* parameters. If *Rmid* is specified, the grid spacing is set to 0.05 au from *Rmin* to *Rmid* and 0.1 au from *Rmid* to *Rmax*. If *Rmid* is not provided, a uniform grid spacing of 0.1 au is used. The grid should be large enough to accurately represent the continuum function in the space where bound-domain orbitals involved in the decay have considerable amplitudes. The *cwave_center* sets the position of the Coulomb wave. By default, it is at the origin (*cwave_center = 0*). One can place it at a specific atom or any other location, such as the midpoint of a bond, by defining a ghost atom (Gh) in the input and setting its index as the *cwave_center*. The *print_cw* option enables printing the Coulomb wave, its fitted representation (for each *l*), and the corresponding Coulomb basis. The user can check the *Fit square error:* in the qchem output file and adjust these parameters accordingly by varying *ncoeff*.

It is also possible to provide the direct expansion of the Coulomb wave in terms of PW-CGTOs from external optimization (for the given effective charge and kinetic energy) in the input section *$coulomb_wave*. This is enabled by setting CC_FESHBACH_CW = 3.

The implementation of Feshbach widths includes numerical integration over all possible directions of the emitted electron (**k**-vector). This integration over the sphere is carried out using Lebedev's quadrature, with the default order of 5. For molecules with delocalized core-hole states (e.g., benzene), higher-order quadrature may be needed. The order of the quadrature is controlled by CC_FESHBACH_INT_ORDER. Examples 7.72 and 7.73 below illustrate calculations of auger decay rates using plane-wave and Coulomb wave treatments.

For non-resonant Auger decay, the initial state can be conveniently computed by CVS-EOM-IP-CCSD and its stable decay channels can be computed by EOM-DIP-CCSD. A section of the input invoking Auger decay rates calculation for an atom can be given as:

```
$rem
   JOBTYPE          sp
   METHOD           eom-ccsd
   basis            6-31G*
   CVS_EOM_IP_BETA [1,0,0,0,0,0,0,0] !This is the initial core-hole state
   DIP_TRIPLETS    [0,0,0,0,0,1,1,1] !These are the final triplet decay channels
   DIP_SINGLETS    [3,1,1,1,0,1,1,1] !These are the final singlet decay channels
   CC_DO_DYSON      1                !Needed for Feshbach-type calculations
   CC_DO_FESHBACH   1
$end
```

In resonant Auger decay, the initial state can be computed by CVS-EOM-EE-CCSD and the corresponding decay channels can be computed by EOM-IP-CCSD. By default, Feshbach calculations are performed for all possible state pairs that include an energetically allowed decay channel. This is not practical if, for example, the core-hole state of interest is not the lowest state in the given symmetry, or when the Coulomb wave is used to model the continuum orbital. In such a case, the user can specify pairs of states for Feshbach calculations using the *$trans_prop* section with $dyson$ as the requested property:

```
$trans_prop
   state_list
   cvs_ip_beta  1 1 !state 1: CVS_IP with irrep = 1 and istate = 1
   dip_singlets 1 3 !state 2: DIP_SINGLET state with irrep = 1 and istate = 3
   dip_triplets 6 1 !state 3: DIP_TRIPLET state with irrep = 6 and istate = 1
   end_list
   state_pair_list
   1 2   ! transition 1 <-> 2
   1 3   ! transition 1 <-> 3
   end_pairs
   calc dyson
$end
```

Calculations of energy correction Δ_μ are invoked by setting CC_DO_FESHBACH = 2, and are currently available only within the plane-wave approximation.

The integrals in Eq. (7.94) are evaluated analytically. Integration in Eq. (7.95) is done numerically, and is split into two or three intervals to bypass the singularity at $E = E_0 - E_\mu$. The upper limits of those intervals are set to default values related to E_0. They can also be customized (except for the first interval) by setting CC_FESHBACH_DELTA_INTB = XX and/or CC_FESHBACH_DELTA_INTC = YY where XX and/or YY are desired upper integration limits in units of eV.

A molecular orbital description of the Auger process (or other two-electron decay processes such as intermolecular Coulomb and electron-transfer-mediated decay) can be obtained from the singular-value decompostion of the two-particle Dyson amplitudes (Γ_μ)[126]. The procedure yields several three-orbital sets, which represent the core-vacancy state and the valence decay states. These sets, called Natural Auger Orbitals, provide the most compact description of the two-electron decay process[126]. The calculations of NAOs can be invoked within Feshbach calculations by the CC_DO_NAO keyword. The NAOs are printed in the MOLDEN format in a subdirectory created in the working directory. The "energies" are the squares of singular values (not normalized), core-hole NAOs are assigned populations of 1, and valence decay NAOs are assigned populations of 0. The default threshold for the NAOs to be printed is 0.20.

It can be controlled by the user using the WFA_ORB_THRESH keyword (see Section 10.2). This feature is illustated in examles 7.74, 7.75, and 7.76 below.

Note: NAOs are computed using $\Gamma_{r\beta}^{p\alpha,q\beta}$ block of the two-body Dyson amplitude. This means that in the non-resonant Auger decay calculations, the core hole should correspond to removing β-electron.

CC_DO_FESHBACH

Activates calculation of resonance widths using Feshbach-Fano approach.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not invoke Feshbach-Fano calculation

1 invoke Feshbach-Fano calculation of the resonance width

2 invoke Feshbach-Fano calculation of the resonance width and resonance shift

RECOMMENDATION:

Initial and final states should be correctly specified.

CC_FESHBACH_CW

Activates Coulomb wave description of the ejected electron.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use plane wave

1 Use Coulomb wave generated internally

3 Use Coulomb basis provided by the user in the *$coulomb_wave* section

RECOMMENDATION:

Additional details need to be specified in *$coulomb_wave* section.

CC_FESHBACH_INT_ORDER

Controls k-vector integration grid in calculations of resonance widths using Feshbach-Fano approach.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n corresponds to the Lebedev quadrature order

RECOMMENDATION:

Use default, unless tighter convergence is desired (16 gives fully converged widths).

CC_FESHBACH_DELTA_INTB

Specifies integration limits in calculation of energy shift in Feshbach-Fano calculations.

TYPE:

INTEGER

DEFAULT:

Preset

OPTIONS:

n corresponds to energy limit in eV

RECOMMENDATION:

Use default.

CC_FESHBACH_DELTA_INTC

Specifies integration limits in calculation of energy shift in Feshbach-Fano calculations.

TYPE:

INTEGER

DEFAULT:

Preset

OPTIONS:

n corresponds to energy limit in eV

RECOMMENDATION:

Use default.

CC_DO_NAO

Activates calculation of NAOs within Feshbach–Fano calculation of the decay widths.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not compute NAOs

1 compute NAOs

RECOMMENDATION:

Initial and final states should be correctly specified.

7.10.10.1 Examples

Examples 7.72 and 7.73 illustrate calculation of resonant Auger decay of core-ionized water molecule. The initial state is described by CVS-EOM-IP-CCSD and the decay channels are described by EOM-DIP-CCSD. Example 7.72 uses a plane-wave representation of the ejected electron. In example 7.73, the autoionizing electron is described by the Coulomb wave, represented by a pseudo-partial wave expansion over PW-CGTO functions. Examples 7.74, 7.75, and

7.76 illustarte calculations of NAOs for regular and resonant Auger decay.

Example 7.72 Calculation of Auger decay rates of core-ionized water molecule to selected singlet and triplet final states. Continuum orbital is a plane wave.

```
$molecule
0 1
   O         0.0000    0.000    0.0000
   H        -0.7528    0.000   -0.5917
   H         0.7528    0.000   -0.5917
$end

$rem
   METHOD            ccsd
   BASIS             6-311+G(3df)
   CVS_EOM_IP_BETA   [1,0,0,0]
   DIP_SINGLETS      [4,1,2,2]
   DIP_TRIPLETS      [1,1,2,2]
   CC_DO_DYSON       1
   CC_DO_FESHBACH    1
$end
```

Example 7.73 Calculation of Auger decay rates of core-ionized water molecule to selected singlet and triplet final states. Continuum orbital is approximated by a Coulomb wave.

```
$molecule
0 1
   O         0.0000    0.000    0.0000
   H        -0.7528    0.000   -0.5917
   H         0.7528    0.000   -0.5917
$end

$rem
   METHOD            ccsd
   BASIS             6-311+G(3df)
   CVS_EOM_IP_BETA   [1,0,0,0]
   DIP_TRIPLETS      [1,1,2,2]
   CC_DO_DYSON       1
   CC_DO_FESHBACH    1
   CC_FESHBACH_CW    1
$end

$trans_prop
state_list
  cvs_ip_beta  1 1
  dip_triplets 3 2
end_list
state_pair_list
  1 2   ! transition 1 <-> 2
end_pairs
calc dyson
$end

$coulomb_wave
lmax = 4
ncoeff = 6
Rmin = 0.05
Rmax = 10
Rmid = 2
Charge = 4.9
cwave_center = 1  !CW is centered on oxygen (atom #1)
print_cw = false
$end
```

Example 7.74 Calculation of Natural Auger Orbitals for regular (non-resonant) Auger decay in water for both singlet and triplet decay channels.

```
$comment
NAO calculation - regular Auger decay in water
$end

$molecule
0 1
O         0.0000    0.000    0.0000
H        -0.7528    0.000   -0.5917
H         0.7528    0.000   -0.5917
$end


$rem
jobtype              sp
method               eom-ccsd
basis                6-31g
cvs_eom_ip_beta      [1,0,0,0]  !Initial core-hole state
dip_singlets         [1,1,0,0]  !Final singlet decay channels
dip_triplets         [0,1,0,0]  !Final triplet decay channels
cc_do_dyson          1          !Needed for Feshbach-type calculations
cc_do_feshbach       1          !Needed for Feshbach-type calculations
cc_do_nao            1          !Needed for natural Auger orbitals
wfa_orb_thresh       1          !Setting the threshold for singular value dceomposition
mem_total            4000
$end
```

Example 7.75 Calculation of Natural Auger Orbitals for regular (non-resonant) Auger decay in benzene for both singlet and triplet decay channels, and for several core-hole states.

```
$comment
NAO calculation - regular Auger decay in benzene
$end

$molecule
  H   2.4750347531   0.0000000000   0.0000000000
  C   1.3935929418   0.0000000000   0.0000000000
  C   0.6967964709   1.2068868901   0.0000000000
  H   1.2375173766   2.1434429715   0.0000000000
  C  -0.6967964709   1.2068868901   0.0000000000
  H  -1.2375173766   2.1434429715   0.0000000000
  C  -1.3935929418   0.0000000000   0.0000000000
  H  -2.4750347531   0.0000000000   0.0000000000
  C  -0.6967964709  -1.2068868901   0.0000000000
  H  -1.2375173766  -2.1434429715   0.0000000000
  C  0.6967964709   -1.2068868901   0.0000000000
  H  1.2375173766   -2.1434429715   0.0000000000
$end

$rem
jobtype              sp
method               ccsd
basis                6-31g
cvs_eom_ip_beta      [1,0,0,0,0,0,1,1] !Initial core-hole state
dip_singlets         [2,0,0,1,0,0,0,0] !Final singlet decay channel
dip_triplets         [1,0,0,1,0,0,0,0] !Final triplet decay channel
cc_do_dyson          1                 !Needed for Feshbach-type calculation
cc_do_feshbach       1                 !Needed for Feshbach-type calculation
cc_do_nao            1                 !Needed for natural Auger orbital calculation
mem_total            8000
$end
```

Example 7.76 Calculation of Natural Auger Orbitals for resonant Auger decay in water.

```
$comment
NAO calculation - resonant Auger decay in water
$end

$molecule
0 1
O          0.0000    0.000    0.0000
H         -0.7528    0.000   -0.5917
H          0.7528    0.000   -0.5917
$end


$rem
jobtype              sp
method               eom-ccsd
basis                6-31g
cvs_ee_states        [1,0,0,0]  !Initial core-hole state
ip_states            [1,0,1,0]  !Final  decay channels
cc_do_dyson          1          !Needed for Feshbach-type calculations
cc_do_feshbach       1          !Needed for Feshbach-type calculations
cc_do_nao            1          !Needed for natural Auger orbitals
mem_total            4000
$end
```

7.10.11 Partial Auger Decay Widths from Complex-Variable Calculations

Auger decay in core-ionized atoms and molecules can be described by Feshbach–Fano approach (see Section 7.10.10) as well as by complex-variable extentions of the CCSD or EOM-CCSD methods.[171,172] In this latter approach, energies and widths Γ (which are proportional to the decay rates) of core-hole states are computed as the difference of complex energies of ground and core-vacant states.

Both in Feshbach–Fano and complex-variable calculations of the AES, the Auger intensities are proportional to the rate of adecay into a particular channel. Hence, one need to compute partial decay widths Γ_{ij}, which add up to the total width Γ. Therein, i and j refer to the valence orbitals from which the electrons involved in the decay process are removed. Below we describe two different ways of computing partial decay width using complex-variable CC/EOM-CC approaches.

7.10.11.1 Decomposition of the Coupled-Cluster energy

The energy of a complex-variable coupled-cluster singles and doubles wave function can be written as

$$E - \mathrm{i}\frac{\Gamma}{2} = E_{\mathrm{CCSD}} = E_{\mathrm{HF}} + \sum_{ijab}\left(\frac{1}{4}t_{ij}^{ab} + \frac{1}{2}t_i^a t_j^b\right)\langle ij||ab\rangle. \tag{7.96}$$

If the reference state has a core-hole, Auger decay-like transitions leading to doubly ionized states are double excitations from two valence orbitals i and j to the core-hole a and a virtual orbital b, which represents emission into the continuum when using a complex-variable method.[171] From equation 7.96 it is evident that we can obtain the contribution from one of these valence orbital combinations, i.e., the partial width, from the amplitude and two-electron integral tensors

as

$$\frac{\Gamma_{ij}}{2} = -\mathrm{Im}\left(\sum_b \left(t_{ij}^{ab} + 2t_i^a t_j^b\right) \langle ij||ab\rangle\right). \tag{7.97}$$

Since these quantities are determined in every CCSD calculation, the computational cost for such a partial width calculation is negligible. This procedure is implemented in the *ccman2* module of Q-CHEM and can be invoked by setting the CC_PW variable to 1. Open decay channels are determined by comparing the orbital energies. The output contains a list of all combinations of two valence orbitals and their partial widths.

If the core-vacancy is produced through core-ionization in the closed-shell ground state of a molecule or atom, the combination of i_α and j_β describes the same target state as i_β and j_α. In the current implementation, these two channels are automatically combined to a single decay width: in the output, one of the orbitals characterizing the decay channel is always an alpha orbital and the inverted spin case is implicitly contained.

Note: Core electrons must not be frozen in such calculations—N_FROZEN_CORE should be set to 0. The core hole must be in a β orbital.

CC_PW

Activates calculation of partial Auger decay widths via decomposition of the imaginary part of the coupled-cluster energy of a complex-variable CCSD calculation on a core-ionized state. Currently, this is implemented for states resulting from ionization of a β core electron of a closed-shell system.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not invoke energy decomposition into partial Auger decay widths

1 invoke energy decomposition into partial Auger decay widths

RECOMMENDATION:

Use to compute partial widths for a complex-variable calculation on a core-vacant state. An appropriate complex-scaled basis set has to be chosen in order to capture Auger decay and the optimal scaling angle needs to be determined.[171,172]

Example 7.7.77 Calculation of Auger partial decay widths of the core-ionized neon atom.

View input online

7.10.11.2 Auger Channel Projectors

An alternative recipe to obtain partial decay widths using complex-variable methods is to restrict the excitation manifold so that excitations describing decay via a certain channel are no longer included. The projectors which accomplish this have been dubbed Auger Channel Projectors (ACP).[172] A calculation with a decay channel projected out yields a different energy and decay width than one with the channel present, and the difference in the decay width represents the partial decay width of that channel.

ACP-EOM-IP-CCSD calculations are accelerated by using the EOM-IP-CCSD solution with the full excitation manifold as a guess. For this purpose, the Maximum Overlap Method is used to ensure convergence to the same roots as in the initial EOM-IP-CCSD calculation. This is invoked in the *ccman2* module of Q-CHEM by setting the EOM_ACP variable to 1. The output contains a list of all open decay channels and their partial widths for each core-ionized state.

If the core-vacancy is produced through core-ionization in the closed-shell ground state of a molecule or atom, the combination of i_α and j_β describes the same target state as i_β and j_α. In the current implementation, these two channels are automatically combined to a single decay width: in the output, one of the orbitals characterizing the decay channel is always an alpha orbital and the inverted spin case is implicitly contained.

Note: Core electrons must not be frozen in such calculations—N_FROZEN_CORE should be set to 0. The core-valence separation must not be invoked.

EOM_ACP
Activates calculation of partial Auger decay widths by recomputation of the EOM-CCSD state with an Auger Channel Projector applied. Currently, this is implemented for EOMIP-CCSD calculations with a closed-shell reference.

TYPE:
INTEGER

DEFAULT:
0

OPTIONS:
0 do not run ACP-EOM-CCSD calculations
1 determine partial Auger decay widths by running ACP-EOM-CCSD calculations

RECOMMENDATION:
Use to compute partial widths for a complex-variable calculation which produces a core-vacant state. An appropriate complex-scaled basis set has to be chosen in order to capture Auger decay and the optimal scaling angle needs to be determined.[171,172]

Example 7.7.78 Calculation of Auger partial decay widths of the core-ionized neon atom.

View input online

Example 7.7.79 Calculation of Auger partial decay widths using an open-shell reference and frozen core.

View input online

7.10.12 Charge Stabilization for EOM-DIP and Other Methods

The performance of EOM-DIP deteriorates when the reference state is unstable with respect to electron detachment,[143,144] which is usually the case for dianion reference states that are employed to describe neutral diradicals by EOM-DIP. These states are often characterized by occupied Hartree-Fock energy levels having positive (unbound) eigenvalues, corresponding to a wave function that is not normalizable. (These are essentially discretized continuum solutions, represented crudely in a Gaussian basis set.[104]) Similar problems are encountered by all excited-state methods when dealing with excited states lying above ionization or electron-detachment thresholds.

To remedy this problem, one can employ charge stabilization methods.[104,143,144] This approach, which can be used with any electronic structure method, introduces an additional Coulomb potential to stabilize the wave function. There are three ways to do this:

- Scaling the nuclear charges, which is accomplished using the *$rem* variable SCALE_NUCLEAR_CHARGE.
- User-defined nuclear charges, activated by setting CHARGE_STABILIZE = TRUE and specifying the new charges in a *$nuclear_charges* input section. The format for the *$nuclear_charges* input section is the same as that of the *$van_der_waals* section, which is described in Section 11.2.10.

- Adding a "charged cage", *i.e.*, an array of point charges around the molecule, which is activated by setting ADD_CHARGED_CAGE = TRUE. Two types of cages (spherical and dodecahedral) are available; the shape, radius, number of points, and total charge of the cage are set by the user.

In the case of EOM methods, a perturbative estimate of the effect of the external Coulomb potential on the EOM energy will be computed when target state densities are calculated, *e.g.*, when CC_EOM_PROP = TRUE. Charge stabilization techniques can be used with other methods including ground state DFT (in order to describe meta-stable ground states) and TDDFT (to improve the description of auto-ionizing resonances). For methods other than EOM, no special correction is applied and one simply obtains the "ordinary" electronic structure but with modified nuclear charges or in the presence of additional point charges. In such cases, it may be advisable to perform several calculations with differing values of the nuclear charges in order to extrapolate the results to the true atomic numbers.[104] In so doing, only calculations for which stabilization is sufficient to obtain a bound-state wave function should be used in the extrapolation. Calculations that result in unbound occupied levels (SCF eigenvalues $\epsilon_i > 0$) should not be taken seriously, as they represent orthogonalized discretized continuum states and not true bound-state solutions. Results from such solutions will vary strongly with respect to the choice of basis set but in ways that are essentially meaningless.[104]

The following descriptions and examples illustrate all three mechanisms of charge stabilization.

SCALE_NUCLEAR_CHARGE

Scale the nuclear charges.

TYPE:

INTEGER

DEFAULT:

0 do not scale (use true atomic numbers)

OPTIONS:

N scale the nuclear charges in a way that adds a charge of N/100 (in a.u.)

RECOMMENDATION:

For EOM methods a perturbative correction can be added in conjunction with this option (as noted above), but for other electronic structure methods once simply gets a traditional calculation but with modified nuclear charges.

Example 7.80 EOM-DIP excitation energies and properties using a charged cage for stabilization.

```
$molecule
  -2 1
   C   0.000000     0.000000     0.106788
   H  -0.989216     0.000000    -0.320363
   H   0.989216     0.000000    -0.320363
$end

$rem
   METHOD                     eom-ccsd
   BASIS                      6-311g(d,p)
   SCF_ALGORITHM              diis_gdm
   SCF_CONVERGENCE            8
   CC_T_CONV                  8
   EOM_DAVIDSON_CONVERGENCE   5
   CC_SYMMETRY                false
   DIP_SINGLETS               [1]    ! Compute one EOM-DIP singlet state
   DIP_TRIPLETS               [1]    ! Compute one EOM-DIP triplet state
   CC_EOM_PROP                true   ! Compute excited state properties
   ADD_CHARGED_CAGE           2      ! 1 for dodecahedral, 2 for spherical
   CAGE_RADIUS                225    ! Radius = 2.25 A
   CAGE_CHARGE                500    ! Total Charge = 5 a.u.
   CAGE_POINTS                100    ! Place 100 point charges
   INTEGRAL_SYMMETRY          false ! charged cage may violate point-group symmetry
$end
```

Example 7.81 Stabilization of SO_4^{2-} by modified nuclear charges.

```
$comment
   Charge stabilization of an unbound anion (sulfate) by changing nuclear charge
   for S.  Format for nuclear charges section is same as van_der_waals section.
$end

$molecule
   -2 1
   S     0.0000000000    0.0000000000    0.0000000000
   O     0.8960432838    0.8960432838    0.8960432838
   O    -0.8960432838   -0.8960432838    0.8960432838
   O    -0.8960432838    0.8960432838   -0.8960432838
   O     0.8960432838   -0.8960432838   -0.8960432838
$end

$rem
   METHOD             mp2
   BASIS              aug-cc-pvdz
   CHARGE_STABILIZE   1
$end

$nuclear_charges
   1
   16 16.5
$end
```

7.10.13 Frozen Natural Orbitals in CC, IP-CC, and SF-CC Calculations

Large computational savings are possible if the virtual space is truncated using the frozen natural orbital (FNO) approach (see Section 6.15.1). An extension of the FNO approach to ionized states within the EOM-CC formalism has

also been introduced and benchmarked.[145] In addition to ground-state coupled-cluster calculations, FNOs can also be used in EOM-IP-CCSD, EOM-IP-CCSD(dT/fT) and EOM-IP-CC(2,3). In IP-CC the FNOs are computed for the reference (neutral) state and then are used to describe several target (ionized) states of interest. Different truncation scheme are described in Section 6.15.1.

To reduce the cost of EOM-SF-CCSD calculations, a special variant of FNO—open-shell frozen natural orbital approximation (OSFNO)—has been introduced.[213] This approach is a two-step scheme. First, the open-shell orbitals of the reference are found by singular value decomposition of the overlap matrix of alpha occupied and beta virtual orbitals. These orbitals contain the main amplitudes of the EOM-SF wave functions. Then, after separation of the open-shell orbitals, the rest of the virtual space is transformed through singular value decomposition of the singlet part of the MP2 density matrix (in alpha-beta spin orbital pairs). Benchmarks in Ref.[213] show that this scheme achieves speedups similar to FNO, while introducing very small errors to the relative energies of both covalent and ionic EOM-SF states. In particular, the errors in singlet–triplet gaps for single molecule magnets are less than 18 cm^{-1} for a typical OSFNO truncation at 99% of total population. Properties also show small errors. OSFNO is activated with CC_OSFNO = true rem variable. CC_FNO_THRESH and CC_FNO_USEPOP keywords have the same usage as in conventional FNO.

Because of the limitation of the implementation, point-group symmetry cannot be used with FNO/OSFNO and will be disabled. Please, adjust your input consistently with CC_SYMMETRY = FALSE.

OSFNO can be combined with orbital localization to produce effective Hamiltonians, as described in the Section 13.6.

CC_OSFNO

Activation of OSFNO. Available only for open-shell references.

TYPE:

LOGICAL

DEFAULT:

FALSE do not activate

OPTIONS:

TRUE activate

RECOMMENDATION:

Use for EOM-SF-CCSD calculations from open-shell references. Available in CCMAN2 only.

Example 7.82 An EOM-SF-CCSD job, illustrating usage of OSFNO for energies and properties.

```
$molecule
   0 3
   C
   H 1 rCH
   H 1 rCH 2 aHCH

   rCH = 1.0775
   aHCH = 133.29
$end

$rem
   METHOD           eom-ccsd
   BASIS            cc-PVTZ
   SF_STATES        [4]
   CC_SYMMETRY      false
   CC_EOM_PROP      1
   CC_EOM_PROP_TE   1
   CC_OSFNO         true
   CC_FNO_THRESH    9900
$end
```

7.10.14 Single-Precision Arithmetic in EOM-CC Calculations

Similar to ground-state CCSD calculations described in Section 6.18, single precision can be used in EOM-CC and EOM-MP2 calculations.[210] Currently, the following variants of EOM are supported: EE, SF,IP, EA; both in standard and RI/CD implementations, for energies and properties evaluation. If you wish to use single-precision version of EOM, please first read Section 6.18 for basic setup of single-precision coupled-cluster calculation. Here we describe only additional EOM-specific keywords.

Precision selection is controlled by the EOM_SINGLE_PREC keyword: 0 corresponds to double-precision calculation and 1 corresponds to single-precision calculation. EOM-specific convergence criteria are controlled by the same keywords as in the double precision, but the same rule as for CCSD applies: too tight thresholds may cause issues with convergence. The default Davidson threshold 10^{-5} works well for most cases.[210]

The keyword CC_SP_DM controls calculation of intermediates, density matrices, and $\hat{S}^2$ for EOM calculations in the same manner as for CCSD, which is described in Section 6.18.

Calculations of analytical gradients require solving amplitude-response equations, which can be done on single precision as well; this is activated by EOM_ARESP_SINGLE_PREC = 1. For using single precision in calculating response equations for 2PA calculations, use the CC_EOM_2PA_SINGLE_PREC keyword (see also Section 7.10.20.6).

EOM_SINGLE_PREC

Precision selection for EOM-CC/MP2 calculations. Available in CCMAN2 only.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

2 single-precision calculation is followed by double-precision clean-up iterations

RECOMMENDATION:

Do not set too tight convergence criteria when use single precision

CC_EOM_2PA_SINGLE_PREC

Precision selection for 2PA response equations. Available in CCMAN2 only.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

RECOMMENDATION:

NONE

Note: Remember to adjust convergence thresholds when using single precision.

Example 7.83 A job evaluating EOM-EA-CCSD energies for formaldehyde anion using single-precision execution combined with CD

```
$comment
   Formaldehyde anion, single-precision calculation
$end

$molecule
   0 1
   C
   H  1  1.127888
   H  1  1.127888  2  100.546614
$end

$rem
   BASIS                 = cc-pvdz
   METHOD                = ccsd
   CHOLESKY_TOL          = 3
   EA_STATES             = [1,0,0,0]
   CC_REF_PROP           = 1   Compute properties of the CCSD reference
   !SP keywords
   CC_SINGLE_PREC        = 1
   CC_SP_T_CONV          = 4
   CC_SP_E_CONV          = 6
   CC_ERASE_DP_INTEGRALS = 0 ! set 1 to save disk space
   CC_SP_DM              = 1
   !EOM-specific keyword
   EOM_SINGLE_PREC       = 1
$end
```

Example 7.84 Geometry optimization of a triplet excited state of uracil-water complex in single-precision setup

```
$molecule
   0 1
   N     0.034130    -.986909    0.000000
   N    -1.173397     .981920    0.000000
   C    -1.218805    -.408164    0.000000
   C    -0.007302    1.702153    0.000000
   C     1.196200    1.107045    0.000000
   C     1.289085    -.345905    0.000000
   O     2.310232    -.996874    0.000000
   O    -2.257041   -1.026495    0.000000
   H     0.049329   -1.997961    0.000000
   H    -2.070598    1.437050    0.000000
   H    -0.125651    2.776484    0.000000
   H     2.111671    1.674079    0.000000
   O     1.747914   -1.338382   -3.040233
   H     2.180817   -1.817552   -2.333676
   H     0.813180   -1.472188   -2.883392
$end

$rem
   JOB_TYPE              = opt
   METHOD                = ccsd
   BASIS                 = cc-pvdz
   CC_STATE_TO_OPT       = [1,1]
   MEM_TOTAL             = 30000
   EE_TRIPLETS           = [1]
   CC_SP_T_CONV          = 4
   CC_SP_E_CONV          = 6
   CC_SINGLE_PREC        = 1
   EOM_SINGLE_PREC       = 1
   CC_SP_DM              = 1
   CC_EOM_PROP           = 1
   EOM_ARESP_SINGLE_PREC = 1
$end
```

7.10.15 Approximate EOM-CC Methods

7.10.15.1 EOM-MP2 and EOM-MP2T

Approximate EOM-CCSD models with T-amplitudes obtained at the MP2 level offer reduced computational cost compared to the full EOM-CCSD since the computationally demanding $\mathcal{O}(N^6)$ CCSD step is eliminated from the calculation. Two methods of this type are implemented in Q-CHEM. The first is invoked with the keyword METHOD = EOM-MP2. Its formulation and implementation follow the original EOM-CCSD(2) approach developed by Stanton and coworkers.[237] The second method can be requested with the METHOD = EOM-MP2T keyword and is similar to EOM-MP2, but it accounts for the additional terms in $\bar{H}$ that appear because the MP2 T-amplitudes do not satisfy the CCSD equations. EOM-MP2 *ansatz* is implemented for IP/EA/EE/SF energies, state properties, and interstate properties (EOM-EOM, but not REF-EOM). EOM-MP2t is available for the IP/EE/EA energy calculations only.

7.10.15.2 EOM-CCSD-S(D) and EOM-MP2-S(D)

These are very light-weight EOM methods in which the EOM problem is solved in the singles block and the effect of doubles is evaluated perturbatively. The $\bar{H}$ is evaluated by using either CCSD or MP2 amplitudes, just as in the regular EOM calculations. The EOM-MP2-S(D) method, which is similar in level of correlation treatment to SOS-CIS(D), is particularly fast. These methods are implemented for IP and EE states. For valence states, the errors for absolute ionization or excitation energies against regular EOM-CCSD are about 0.4 eV and appear to be systematically blue-shifted; the EOM-EOM energy gaps look better. The calculations are set as in regular EOM-EE/IP, but using METHOD = EOM-CCSD-SD(D) or METHOD = EOM-MP2-SD(D). State properties and EOM-EOM transition properties can be computed using these methods (reference-EOM properties are not yet implemented). These methods are designed for treating core-level states.[219]

Note: These methods are still in the experimental stage.

7.10.16 EOM-CC Guess Formation and Iterative Diagonalization

An EOM-CC eigenproblem is solved by an iterative diagonalization procedure that avoids full diagonalization and only looks for several eigenstates, as specified by the XX_STATES keywords. The default procedure is based on the modified Davidson diagonalization algorithm.[150] In addition to several keywords that control the convergence of algorithm, memory usage, and fine details of its execution, there are several important keywords that allow user to specify how the target state selection will be performed.

By default, the diagonalization looks for several lowest eigenstates, as specified by XX_STATES. The guess vectors are generated as singly excited determinants selected by using Koopmans' theorem; the number of guess vectors is equal to the number of target states. If necessary, the user can increase the number of singly excited guess vectors (EOM_NGUESS_SINGLES) and include doubly excited guess vectors (EOM_NGUESS_DOUBLES).

Note: In CCMAN2, if there is not enough singly excited guess vectors, the algorithm adds doubly excited guess vectors. In CCMAN, doubly excited guess vectors are generated only if EOM_NGUESS_DOUBLES is invoked.

The user can request to pre-converge singles (solve the equations in singles-only block of the Hamiltonian. This is done by using EOM_PRECONV_SINGLES. In EOM-DEA/DIP calculations, one can pre-converge $2p/2h$ amplitudes by using EOM_PRECONV_DOUBLES. In the CVS suite of methods, this option is invoked by CVS_EOM_PRECONV_SINGLES.

Note: In CCMAN, the user can pre-converge both singles and doubles blocks (EOM_PRECONV_SINGLES and. EOM_PRECONV_DOUBLES)

If a state (or several states) of a particular character is desired (*e.g.*, $\mathrm{HOMO} \rightarrow \mathrm{LUMO} + 10$ excitation or $\mathrm{HOMO} - 10$ ionization), the user can specify this by using EOM_USER_GUESS keyword and *$eom_user_guess* section, as illustrated by an example below. The algorithm will attempt to find an eigenstate that has the maximum overlap with this guess vector. The multiplicity of the state is determined as in the regular calculations, by using the XX_SINGLETS and EE_TRIPLETS keywords. This option is useful for looking for high-lying states such as core-ionized or core-excited states. It is only available with CCMAN2.

The examples below illustrate how to use user-specified guess in EOM calculations:

```
$eom_user_guess
  4  Corresponds to 4(OCC)->5(VIRT) transition.
  5
$end
```

or

```
$eom_user_guess
  1  5   Ex. states corresponding to 1(OCC)->5(VIRT) and 1(OCC)->6(VIRT)
  1  6
$end
```

In IP/EA calculations, only one set of orbitals is specified:

```
$eom_user_guess
  4 5 6
$end
```

If IP_STATES is specified, this will invoke calculation of the EOM-IP states corresponding to the ionization from 4th, 5th, and 6th occupied MOs. If EA_STATES is requested, then EOM-EA equations will be solved for a root corresponding to electron-attachment to the 4th, 5th, and 6th virtual MOs.

For these options to work correctly, user should make sure that XX_STATES requests a sufficient number of states. In case of symmetry, one can request several states in each irrep, but the algorithm will only compute those states which are consistent with the user guess orbitals.

Alternatively, the user can specify an energy shift by EOM_SHIFT. In this case, the solver looks for the XX_STATES eigenstates that are closest to this energy; the guess vectors are generated accordingly, using Koopmans' theorem. This option is useful when highly excited states (*i.e.*, interior eigenstates) are desired.

7.10.17 EOM-CC Job Control

It is important to ensure there are sufficient resources available for the necessary integral calculations and transformations. For CCMAN/CCMAN2 algorithms, these resources are controlled using the *$rem* variables CC_MEMORY, MEM_STATIC and MEM_TOTAL (see Section 6.17).

The exact flavor of correlation treatment within equation-of-motion methods is defined by METHOD (see Section 7.1). For EOM-CCSD, once should set METHOD to EOM-CCSD, for EOM-MP2, METHOD = EOM-CCSD, *etc.*. In addition, a specification of the number of *target states* is required through XX_STATES (XX designates the type of the target states, *e.g.*, EE, SF, IP, EA, DIP, DSF, *etc.*). Users must be aware of the point group symmetry of the system being studied and also the symmetry of the initial and target states of interest, as well as symmetry of transition. It is possible to turn off the use of symmetry by CC_SYMMETRY. If set to FALSE the molecule will be treated as having C_1 symmetry and all states will be of A symmetry.

Note:

1. In finite-difference calculations, the symmetry is turned off automatically, and the user must ensure that XX_STATES is adjusted accordingly.

2. In CCMAN, mixing different EOM models in a single calculation is only allowed in Dyson orbitals calculations. In CCMAN2, different types of target states can be requested in a single calculation.

Below we describe alternative way to specify correlation treatment in EOM-CC/CI calculations. These keywords will be eventually phased out. By default, the level of correlation of the EOM part of the wave function (*i.e.*, maximum excitation level in the EOM operators R) is set to match CORRELATION, however, one can mix different correlation levels for the reference and EOM states by using EOM_CORR. To request a CI calculation, set CORRELATION = CI and select type of CI expansion by EOM_CORR. The table below shows default and allowed CORRELATION and EOM_CORR combinations.

CORRELATION	Default EOM_CORR	Allowed EOM_CORR	Target states	CCMAN / CCMAN2
CI	none	CIS, CIS(D)	EE, SF	y/n
		CISD	EE, SF, IP	y/n
		SDT, DT	EE, SF, DSF	y/n
CIS(D)	CIS(D)	N/A	EE, SF	y/n
CCSD, OD	CISD		EE, SF, IP, EA, DIP	y/y
		SD(fT)	EE, IP, EA	n/y
		SD(dT), SD(fT)	EE, SF, fake IP/EA	y/n
		SD(dT), SD(fT), SD(sT)	IP	y/n
		SDT, DT	EE, SF, IP, EA, DIP, DSF	y/n

Table 7.1: Default and allowed CORRELATION and EOM_CORR combinations as well as valid target state types. The last column shows if a method is available in CCMAN or CCMAN2.

Table 7.10.17 shows the correct combinations of CORRELATION and EOM_CORR for standard EOM and CI models.

The most relevant EOM-CC input options follow.

EE_STATES

Sets the number of excited state roots to find. For closed-shell reference, defaults into EE_SINGLETS. For open-shell references, specifies all low-lying states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EE_SINGLETS

Sets the number of singlet excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

Method	CORRELATION	EOM_CORR	Target states selection
CIS	CI	CIS	EE_STATES
			EE_SINGLETS, EE_TRIPLETS
SF-CIS	CI	CIS	SF_STATES
CIS(D)	CI	CIS(D)	EE_STATES
			EE_SINGLETS, EE_TRIPLETS
SF-CIS(D)	CI	CIS(D)	SF_STATES
CISD	CI	CISD	EE_STATES
			EE_SINGLETS, EE_TRIPLETS
SF-CISD	CI	CISD	SF_STATES
IP-CISD	CI	CISD	IP_STATES
CISDT	CI	SDT	EE_STATES
			EE_SINGLETS, EE_TRIPLETS
SF-CISDT	CI	SDT or DT	SF_STATES
EOM-EE-CCSD	CCSD		EE_STATES
			EE_SINGLETS, EE_TRIPLETS
EOM-SF-CCSD	CCSD		SF_STATES
EOM-IP-CCSD	CCSD		IP_STATES
EOM-EA-CCSD	CCSD		EA_STATES
EOM-DEA-CCSD	CCSD		DIP_STATES
			DEA_SINGLETS, DEA_TRIPLETS
EOM-DIP-CCSD	CCSD		DIP_STATES
			DIP_SINGLETS, DIP_TRIPLETS
EOM-2SF-CCSD	CCSD	SDT or DT	DSF_STATES
EOM-EE-(2,3)	CCSD	SDT	EE_STATES
			EE_SINGLETS, EE_TRIPLETS
EOM-SF-(2,3)	CCSD	SDT	SF_STATES
EOM-IP-(2,3)	CCSD	SDT	IP_STATES
EOM-SF-CCSD(dT)	CCSD	SD(dT)	SF_STATES
EOM-SF-CCSD(fT)	CCSD	SD(fT)	SF_STATES
EOM-IP-CCSD(dT)	CCSD	SD(dT)	IP_STATES
EOM-IP-CCSD(fT)	CCSD	SD(fT)	IP_STATES
EOM-IP-CCSD(sT)	CCSD	SD(sT)	IP_STATES

Table 7.2: Commonly used EOM and CI models. "SINGLETS" and "TRIPLETS" are only available for closed-shell references.

EE_TRIPLETS

Sets the number of triplet excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

SF_STATES

Sets the number of spin-flip target states roots to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i SF states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DSF_STATES

Sets the number of doubly spin-flipped target states roots to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DSF states.

OPTIONS:

$[i, j, k \ldots]$ Find i doubly spin-flipped states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

IP_STATES

Sets the number of ionized target states roots to find. By default, β electron will be removed (see EOM_IP_BETA).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

IP_ALPHA

Sets the number of ionized target states derived by removing α electron ($M_S = \frac{1}{2}$).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/α states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

IP_BETA

Sets the number of ionized target states derived by removing β electron ($M_S = \frac{1}{2}$, default for EOM-IP).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/β states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EA_STATES

Sets the number of attached target states roots to find. By default, β electron will be attached (see EA_BETA).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any EA states.

OPTIONS:

$[i, j, k \ldots]$ Find i EA states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EA_ALPHA

Sets the number of attached target states derived by attaching α electron ($M_S = \frac{1}{2}$, default in EOM-EA).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any EA states.

OPTIONS:

$[i, j, k \ldots]$ Find i EA states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EA_BETA

Sets the number of attached target states derived by attaching β electron ($M_S = \frac{1}{2}$, EA-SF).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any EA states.

OPTIONS:

$[i, j, k \ldots]$ Find i EA states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_STATES

Sets the number of DIP roots to find. For closed-shell reference, defaults into DIP_SINGLETS. For open-shell references, specifies all low-lying states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_SINGLETS

Sets the number of singlet DIP roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any singlet DIP states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP singlet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_TRIPLETS

Sets the number of triplet DIP roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP triplet states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP triplet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_AA_STATES

Sets the number of $M_S = -1$ DIP roots (remove two α electrons) to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP $M_S = -1$ states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_BB_STATES

Sets the number of $M_S = +1$ DIP roots (remove two β electrons) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP $M_S = +1$ states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_AB_STATES

Sets the number of $M_S = 0$ DIP roots (remove one α and one β electron) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP $M_S = 0$ states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_STATES

Sets the number of DEA roots to find. For closed-shell reference, defaults into DEA_SINGLETS.

For open-shell references, specifies all low-lying states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_SINGLETS

Sets the number of singlet DEA roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any singlet DEA states.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA singlet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_TRIPLETS

Sets the number of triplet DEA roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA triplet states.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA triplet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_AA_STATES

Sets the number of $M_S = 1$ DEA roots (two α electrons) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA $M_S = 1$ transitions.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA $\alpha\alpha$ states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_BB_STATES

Sets the number of $M_S = -1$ DEA roots (two β electrons) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA $M_S = -1$ transitions.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA $\beta\beta$ states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_AB_STATES

Sets the number of $M_S = 0$ DEA roots (one α and one β electron) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA $M_S = 0$ transitions.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA $\alpha\beta$ states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

Note: It is a symmetry of a *transition* rather than that of a *target state* which is specified in excited state calculations. The symmetry of the target state is a product of the symmetry of the reference state and the transition. For closed-shell molecules, the former is fully symmetric and the symmetry of the target state is the same as that of transition, however, for open-shell references this is not so.

Note: In earlier versions of Q-CHEM, we used EOM_XX_STATES. These keywords were replaced by XX_STATES for uniformity (e.g., they are also used in ADC methods to specify target ADC states). For backward compatibility, the old keywords are aliased to the new ones, so old inputs would still work. However, these old keywords eventually will be depreciated. It is recommended to switch to the new ones.

Note: For the XX_STATES options, Q-CHEM will increase the number of roots if it suspects degeneracy, or change it to a smaller value, if it cannot generate enough guess vectors to start the calculations.

EOM_FAKE_IPEA

If TRUE, calculates fake EOM-IP or EOM-EA energies and properties using the diffuse orbital trick. Default for EOM-EA and Dyson orbital calculations in CCMAN.

TYPE:

LOGICAL

DEFAULT:

FALSE (use proper EOM-IP code)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None. This feature only works for CCMAN.

Note: When EOM_FAKE_IPEA is set to TRUE, it can change the convergence of Hartree-Fock iterations compared to the same job without EOM_FAKE_IPEA, because a very diffuse basis function is added to a center of symmetry *before* the Hartree-Fock iterations start. For the same reason, BASIS2 keyword is incompatible with EOM_FAKE_IPEA. In order to read Hartree-Fock guess from a previous job, you must specify EOM_FAKE_IPEA (even if you do not request for any correlation or excited states) in that previous job. Currently, the second moments of electron density and Mulliken charges and spin densities are incorrect for the EOM-IP/EA-CCSD target states.

EOM_USER_GUESS

Specifies if user-defined guess will be used in EOM calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Solve for a state that has maximum overlap with a trans-n specified in *$eom_user_guess*.

RECOMMENDATION:

The orbitals are ordered by energy, as printed in the beginning of the CCMAN2 output. Not available in CCMAN.

EOM_SHIFT

Specifies energy shift in EOM calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n corresponds to $n \cdot 10^{-3}$ hartree shift (*i.e.*, 11000 = 11 hartree); solve for eigenstates around this value.

RECOMMENDATION:

Not available in CCMAN.

EOM_NGUESS_DOUBLES

Specifies number of excited state guess vectors which are double excitations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Include n guess vectors that are double excitations

RECOMMENDATION:

This should be set to the expected number of doubly excited states, otherwise they may not be found.

EOM_NGUESS_SINGLES

Specifies number of excited state guess vectors that are single excitations.

TYPE:

INTEGER

DEFAULT:

Equal to the number of excited states requested

OPTIONS:

n Include n guess vectors that are single excitations

RECOMMENDATION:

Should be greater or equal than the number of excited states requested, unless .

EOM_PRECONV_SINGLES

When not zero, singly excited vectors are converged prior to a full excited states calculation. Sets the maximum number of iterations for pre-converging procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not pre-converge

1 pre-converge singles

RECOMMENDATION:

Sometimes helps with problematic convergence.

Note: In CCMAN, setting EOM_PRECONV_SINGLES = N would result in N Davidson iterations pre-converging singles.

EOM_PRECONV_DOUBLES

When not zero, doubly excited vectors are converged prior to a full excited states calculation. Sets the maximum number of iterations for pre-converging procedure

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not pre-converge

N Perform N Davidson iterations pre-converging doubles.

RECOMMENDATION:

Occasionally necessary to ensure a doubly excited state is found. Also used in DSF, DIP, and DEA calculations instead of EOM_PRECONV_SINGLES

Note: Not available for EOM-EE in CCMAN2.

EOM_PRECONV_SD

When not zero, EOM vectors are pre-converged prior to a full excited states calculation. Sets the maximum number of iterations for pre-converging procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not pre-converge

N perform N Davidson iterations pre-converging singles and doubles.

RECOMMENDATION:

Occasionally necessary to ensure that all low-lying states are found. Also, very useful in EOM(2,3) calculations.

None

Note: Not available in CCMAN2.

EOM_DAVIDSON_CONVERGENCE

Convergence criterion for the RMS residuals (square of the norm) of excited-state vectors.

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Use the default. Normally this value be the same as EOM_DAVIDSON_THRESHOLD.

EOM_DAVIDSON_THRESHOLD

Specifies threshold for including a new expansion vector in the iterative Davidson diagonalization. Their norm must be above this threshold.

TYPE:

INTEGER

DEFAULT:

00103 Corresponding to 0.00001

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-(de+2)}$, *i.e.*, 02505->2.5×10^{-6}

RECOMMENDATION:

Use the default unless converge problems are encountered. Should normally be set to the same values as EOM_DAVIDSON_CONVERGENCE, if convergence problems arise try setting to a value slightly larger than EOM_DAVIDSON_CONVERGENCE.

EOM_DAVIDSON_MAXVECTORS

Specifies maximum number of vectors in the subspace for the Davidson diagonalization.

TYPE:

INTEGER

DEFAULT:

60

OPTIONS:

n Up to n vectors per root before the subspace is reset

RECOMMENDATION:

Larger values increase disk storage but accelerate and stabilize convergence.

EOM_DAVIDSON_MAX_ITER

Maximum number of iteration allowed for Davidson diagonalization procedure.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User-defined number of iterations

RECOMMENDATION:

Default is usually sufficient

EOM_IPEA_FILTER

If TRUE, filters the EOM-IP/EA amplitudes obtained using the diffuse orbital implementation (see EOM_FAKE_IPEA). Helps with convergence.

TYPE:

LOGICAL

DEFAULT:

FALSE (EOM-IP or EOM-EA amplitudes will not be filtered)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None

Note: Not available in CCMAN2.

CC_FNO_THRESH

Initialize the FNO truncation and sets the threshold to be used for both cutoffs (OCCT and POVO).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

range 0000-10000

$abcd$ Corresponding to $ab.cd\%$

RECOMMENDATION:

None

CC_FNO_USEPOP

Selection of the truncation scheme.

TYPE:

INTEGER

DEFAULT:

1 OCCT

OPTIONS:

0 POVO

RECOMMENDATION:

None

SCALE_NUCLEAR_CHARGE

Scales charge of each nuclei by a certain value. The nuclear repulsion energy is calculated for the unscaled nuclear charges.

TYPE:

INTEGER

DEFAULT:

0 No scaling.

OPTIONS:

n A total positive charge of (1+$n/100$)e is added to the molecule.

RECOMMENDATION:

NONE

ADD_CHARGED_CAGE

Add a point charge cage of a given radius and total charge.

TYPE:

INTEGER

DEFAULT:

0 No cage.

OPTIONS:

0 No cage.

1 Dodecahedral cage.

2 Spherical cage.

RECOMMENDATION:

Spherical cage is expected to yield more accurate results, especially for small radii.

CAGE_RADIUS

Defines radius of the charged cage.

TYPE:

INTEGER

DEFAULT:

225

OPTIONS:

n radius is $n/100$ Å.

RECOMMENDATION:

None

CAGE_POINTS

Defines number of point charges for the spherical cage.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n Number of point charges to use.

RECOMMENDATION:

None

CAGE_CHARGE

Defines the total charge of the cage.

TYPE:

INTEGER

DEFAULT:

400 Add a cage charged +4e.

OPTIONS:

n Total charge of the cage is $n/100$ a.u.

RECOMMENDATION:

None

7.10.18 Examples

Example 7.85 EOM-EE-OD and EOM-EE-CCSD calculations of the singlet excited states of formaldehyde.

```
$molecule
   0 1
   O
   C  1  R1
   H  2  R2  1  A
   H  2  R2  1  A  3  180.

   R1  =  1.4
   R2  =  1.0
   A   =  120.
$end

$rem
   METHOD      eom-od
   BASIS       6-31+g
   EE_STATES   [2,2,2,2]
$end

@@@

$molecule
   read
$end

$rem
   METHOD        eom-ccsd
   BASIS         6-31+g
   EE_SINGLETS   [2,2,2,2]
   EE_TRIPLETS   [2,2,2,2]
$end
```

Example 7.86 EOM-EE-CCSD calculations of the singlet excited states of water using Cholesky decomposition.

```
$molecule
   0 1
   O           0.000000    0.000000   -0.069336
   H          -0.759081    0.000000   -0.665332
   H           0.759081    0.000000   -0.665332
$end

$rem
   METHOD          eom-ccsd
   BASIS           aug-cc-pVDZ
   PURECART        1112
   N_FROZEN_CORE   fc
   CC_T_CONV       4
   CC_E_CONV       6
   CHOLESKY_TOL    2    using CD/1e-2 threshold
   EE_SINGLETS     [2,2,0,0]
$end
```

Example 7.87 EOM-SF-CCSD calculations for methylene from high-spin $^3B_2$ reference.

```
$molecule
   0 3
   C
   H  1   1.1167
   H  1   1.1167   2   102.07
$end

$rem
   METHOD      eom-ccsd
   BASIS       6-31G*
   SCF_GUESS   core
   SF_STATES   [2,0,0,2]   Two singlet A1 states and singlet and triplet B2 states
$end
```

Example 7.88 EOM-SF-MP2 calculations for SiH_2 from high-spin $^3B_2$ reference. Both energies and properties are computed.

```
$molecule
   0 3
   Si
   H  1 1.5145
   H  1 1.5145 2 92.68
$end

$rem
   BASIS            = cc-pVDZ
   UNRESTRICTED     = true
   SCF_CONVERGENCE  = 8
   METHOD           = eom-mp2
   SF_STATES        = [1,1,0,0]
   CC_EOM_PROP_TE   = true   ! Compute &lt;S^2&gt; of excited states
$end
```

Example 7.89 EOM-IP-CCSD calculations for NO_3 using closed-shell anion reference.

```
$molecule
   -1 1
   N
   O  1  r1
   O  1  r2   2 A2
   O  1  r2   2 A2    3 180.0

   r1 = 1.237
   r2 = 1.237
   A2 = 120.00
$end

$rem
   METHOD       eom-ccsd
   BASIS        6-31G*
   IP_STATES    [1,1,2,1]    ground and excited  states of the radical
$end
```

Example 7.90 EOM-IP-CCSD calculation using FNO with OCCT=99%.

```
$molecule
   0 1
   O
   H  1  1.0
   H  1  1.0  2  100.
$end

$rem
   METHOD          eom-ccsd
   BASIS           6-311+G(2df,2pd)
   IP_STATES       [3,0,0,0]
   CC_FNO_THRESH   9900        99% of the total natural population recovered
$end
```

Example 7.91 EOM-EE-CC2, EOM-EA-CC2, EOM-IP-CC2 calculations for water.

```
$molecule
0 1
   O     0.00000000    0.00000000   -0.69415386
   H    -1.34476814    0.00000000    0.34707675
   H     1.34476814    0.00000000    0.34707675
$end

$rem
   INPUT_BOHR       true
   METHOD           eom-cc2
   BASIS            6-31g
   EE_STATES         [1,1,1,1]
   EA_STATES         [1,0,1,1]
   IP_STATES         [1,0,1,1]
$end
```

Example 7.92 EOM-EE-MP2T calculation of the H_2 excitation energies.

```
$molecule
   0 1
   H   0.0000   0.0000   0.0000
   H   0.0000   0.0000   0.7414
$end

$rem
   THRESH     16
   BASIS      cc-pvdz
   METHOD     eom-mp2t
   EE_STATES  [3,0,0,0,0,0,0,0]
$end
```

Example 7.93 EOM-IP-MP2 calculation of the three low lying ionized states of the phenolate anion.

```
$molecule
   0 1
   C    -0.189057    -1.215927    -0.000922
   H    -0.709319    -2.157526    -0.001587
   C     1.194584    -1.155381    -0.000067
   H     1.762373    -2.070036    -0.000230
   C     1.848872     0.069673     0.000936
   H     2.923593     0.111621     0.001593
   C     1.103041     1.238842     0.001235
   H     1.595604     2.196052     0.002078
   C    -0.283047     1.185547     0.000344
   H    -0.862269     2.095160     0.000376
   C    -0.929565    -0.042566    -0.000765
   O    -2.287040    -0.159171    -0.001759
   H    -2.663814     0.725029     0.001075
$end

$rem
   METHOD      eom-mp2
   BASIS       6-31+g(d)
   THRESH      14
   IP_STATES   [3]
$end
```

Example 7.94 EOM-EA-CCSD calculation of CN using user-specified guess.

```
$molecule
   +1 1
   C
   N  1  1.1718
$end

$rem
   METHOD          = eom-ccsd
   BASIS           = 6-311+g*
   EA_STATES       = [1,1,1,1]
   CC_EOM_PROP     = true
   EOM_USER_GUESS = true   ! attach to HOMO, HOMO+1, and HOMO+3
$end

$eom_user_guess
   1  2  4
$end
```

Example 7.95 DSF-CIDT calculation of methylene starting with quintet reference.

```
$molecule
   0 5
   C
   H 1 CH
   H 1 CH 2 HCH

   CH  = 1.07
   HCH = 111.0
$end

$rem
   METHOD               cisdt
   BASIS                6-31G
   DSF_STATES           [0,2,2,0]
   EOM_NGUESS_SINGLES   0
   EOM_NGUESS_DOUBLES   2
   CCMAN2               0
$end
```

Example 7.96 EOM-EA-CCSD job for cyano cation. We first do Hartree–Fock calculation for the cation in the basis set with one extremely diffuse orbital (EOM_FAKE_IPEA) and use these orbitals in the second job. We need to make sure that the diffuse orbital is occupied using the OCCUPIED keyword. No SCF iterations are performed as the diffuse electron and the molecular core are uncoupled. The attached states show up as "excited" states in which electron is promoted from the diffuse orbital to the molecular ones.

```
$molecule
   +1 1
   C
   N 1 bond

   bond   1.1718
$end

$rem
   METHOD            hf
   BASIS             6-311+G*
   PURECART          111
   SCF_CONVERGENCE   8
   EOM_FAKE_IPEA     true
   THRESH            11
   S2THRESH          11
$end

@@@

$molecule
   0 2
   C
   N 1 bond

   bond   1.1718
$end

$rem
   BASIS             6-311+G*
   PURECART          111
   SCF_GUESS         read
   MAX_SCF_CYCLES    0
   METHOD            eom-ccsd
   CC_DOV_THRESH     2501   use thresh for CC iters with convergence problems
   EE_STATES         [2,0,0,0]
   EOM_FAKE_IPEA     true
$end

$occupied
   1 2 3 4 5 6 14
   1 2 3 4 5 6
$end
```

Example 7.97 EOM-DIP-CCSD calculation of methylene with charged cage stabilization.

```
$molecule
   -2 1
   C   0.000000     0.000000     0.106788
   H  -0.989216     0.000000    -0.320363
   H   0.989216     0.000000    -0.320363
$end

$rem
   METHOD                   = eom-ccsd
   BASIS                    = 6-311g(d,p)
   SCF_ALGORITHM            = diis_gdm
   CC_SYMMETRY              = false
   DIP_SINGLETS             = [1]  ! Compute one EOM-DIP singlet state
   DIP_TRIPLETS             = [1]  ! Compute one EOM-DIP triplet state
   EOM_DAVIDSON_CONVERGENCE = 5
   CC_EOM_PROP              = true ! Compute excited state properties
   ADD_CHARGED_CAGE         = 2    ! Install a charged sphere around the molecule
   CAGE_RADIUS              = 225  ! Radius = 2.25 A
   CAGE_CHARGE              = 500  ! Charge = +5 a.u.
   CAGE_POINTS              = 100  ! Place 100 point charges
   CC_MEMORY                = 256  ! Use 256Mb of memory, increase for larger jobs
   INTEGRAL_SYMMETRY        = false
$end
```

Example 7.98 EOM-EE-CCSD calculation of excited states in NO^- using scaled nuclear charge stabilization method.

```
$molecule
   -1 1
    N  -1.08735    0.0000    0.0000
    O   1.08735    0.0000    0.0000
$end

$rem
   METHOD               = eom-ccsd
   BASIS                = 6-31g
   INPUT_BOHR           = true
   CC_SYMMETRY          = false
   EE_SINGLETS          = [2]  ! Compute two EOM-EE singlet excited states
   EE_TRIPLETS          = [2]  ! Compute two EOM-EE triplet excited states
   CC_REF_PROP          = true ! Compute ground state properties
   CC_EOM_PROP          = true ! Compute excited state properties
   CC_MEMORY            = 256  ! Use 256Mb of memory, increase for larger jobs
   SCALE_NUCLEAR_CHARGE = 180  ! Adds +1.80e charge to the molecule
   INTEGRAL_SYMMETRY    = false
$end
```

Example 7.99 EOM-DEA-CCSD calculation of ozone with EOM_PRECONV_DOUBLES.

```
$molecule
   +2 1
   O
   O 1 1.2724
   O 2 1.2724 1 116.8
$end

$rem
   METHOD              = eom-ccsd
   BASIS               = 6-31G*
   DEA_SINGLETS        = [1,0,0,0]
   DEA_TRIPLETS        = [0,0,0,1]
   EOM_PRECONV_DOUBLES = true
$end
```

Example 7.100 EOM-EE-CCSD calculation for formamide with user-specified guess requesting the EE transition from the occupied orbital number 12 (2 A'') to the virtual orbital number 1 (11 A').

```
$molecule
   0 1
   N          1.0706214490   -0.1462996030    0.0000000000
   C         -0.1838756809    0.3832287690    0.0000000000
   O         -1.2178351723   -0.2734201303    0.0000000000
   H          1.8945772136    0.4351761203    0.0000000000
   H          1.1761147729   -1.1515954431    0.0000000000
   H         -0.1740335498    1.4879608698    0.0000000000
$end

$rem
   METHOD                     EOM-CCSD
   BASIS                      6-31+G(d,p)
   CC_MEMORY                  3000   ccman2 memory
   MEM_STATIC                 250
   CC_T_CONV                  4      T-amplitudes convergence threshold
   CC_E_CONV                  6      Energy convergence threshold
   EE_STATES                  [0,1]  Calculate 1 A" states
   EOM_DAVIDSON_CONVERGENCE   5      Convergence threshold for the Davidson procedure
   !EOM_USER_GUESS             true   Use user guess from $eom_user_guess section
$end

$eom_user_guess
   12
   1
$end
```

Example 7.101 CAP-augmented EOM-EA-CCSD calculation for N_2^-. aug-cc-pVTZ basis augmented by the $3s3p3d$ diffuse functions placed in the COM. Two EA states are computed for CAP strength $\eta = 0.002$.

```
$molecule
   0 1
   N   0.0  0.0 -0.54875676501
   N   0.0  0.0  0.54875676501
   Gh  0.0  0.0  0.0
$end

$rem
   COMPLEX_CCMAN     1             engage complex_ccman
   METHOD            EOM-CCSD
   BASIS             gen           use general basis
   EA_STATES         [0,0,2,0,0,0,0,0]  compute electron attachment energies
   CC_MEMORY         2000          ccman2 memory
   MEM_TOTAL         4000
   CC_EOM_PROP       true          compute excited state properties
   FORCE_SYMMETRY_ON true
$end

$complex_ccman
   CS_HF           1     Use complex HF
   CAP_ETA         200   Set strength of CAP potential 0.002
   CAP_X           2760  Set length of the box along x dimension
   CAP_Y           2760  Set length of the box along y dimension
   CAP_Z           4880  Set length of the box along z dimension
   CAP_TYPE        1     Use cuboid CAP
$end

$basis
N    0
aug-cc-pvtz
****
Gh   0
S    1    1.000000
   2.88000000E-02    1.00000000E+00
S    1    1.000000
   1.44000000E-02    1.00000000E+00
S    1    1.000000
   0.72000000E-02    1.00000000E+00
P    1    1.000000
   2.45000000E-02    1.00000000E+00
P    1    1.000000
   1.22000000E-02    1.00000000E+00
P    1    1.000000
   0.61000000E-02    1.00000000E+00
D    1    1.000000
   0.755000000E-01   1.00000000E+00
D    1    1.000000
   0.377500000E-01   1.00000000E+00
D    1    1.000000
   0.188750000E-01   1.00000000E+00
****
$end
```

Example 7.102 CAP-EOM-EE calculation of water, with wave-function analysis of state and transition properties.

```
$molecule
   0 1
   O   0.00000000     0.00000000     0.13594219
   H   0.00000000    -1.44761450    -1.07875060
   H   0.00000000     1.44761450    -1.07875060
$end

$rem
   METHOD                     eom-ccsd
   BASIS                      6-31G**
   CC_MEMORY                  2000
   MEM_TOTAL                  2500
   SCF_CONVERGENCE            12
   CC_CONVERGENCE             11
   EOM_DAVIDSON_CONVERGENCE   11
   CC_EOM_PROP                TRUE
   CC_FULLRESPONSE            FALSE
   CC_TRANS_PROP              TRUE
   COMPLEX_CCMAN              1
   EE_STATES                  [1,0,2,0]
   INPUT_BOHR                 TRUE
   ! WFA KEYWORDS
   STATE_ANALYSIS             true
   MOLDEN_FORMAT              true
   NTO_PAIRS                  4
   POP_MULLIKEN               true
$end

$complex_ccman
CS_HF           1
CAP_TYPE        1
CAP_ETA         10000
CAP_X           2000
CAP_Y           2500
CAP_Z           2500
$end
```

Example 7.103 CAP-EOM-EA-CCSD calculation of π^* shape resonance in CO^- using smooth Voronoi CAP.

```
$molecule
   0  1
   C    0.0000   0.0000    0.564
   O    0.0000   0.0000   -0.564
   Gh   0.0000   0.0000    0.000
$end

$rem
   METHOD             eom-ccsd
   BASIS              gen
   EA_STATES          [0,0,5,0]
   COMPLEX_CCMAN      1
   XC_GRID            000099000590
   N_FROZEN_CORE      fc
   FORCE_SYMMETRY_ON  true
$end

$complex_ccman
   CS_HF      1
   CAP_TYPE   2
   CAP_ETA    640
   CAP_X      2765
$end

$basis
C     0
S    8    1.000000
   8.23600000E+03    5.31000000E-04
   1.23500000E+03    4.10800000E-03
   2.80800000E+02    2.10870000E-02
   7.92700000E+01    8.18530000E-02
   2.55900000E+01    2.34817000E-01
   8.99700000E+00    4.34401000E-01
   3.31900000E+00    3.46129000E-01
   3.64300000E-01   -8.98300000E-03
S    8    1.000000
   8.23600000E+03   -1.13000000E-04
   1.23500000E+03   -8.78000000E-04
   2.80800000E+02   -4.54000000E-03
   7.92700000E+01   -1.81330000E-02
   2.55900000E+01   -5.57600000E-02
   8.99700000E+00   -1.26895000E-01
   3.31900000E+00   -1.70352000E-01
   3.64300000E-01    5.98684000E-01
S    1    1.000000
   9.05900000E-01    1.00000000E+00
S    1    1.000000
   1.28500000E-01    1.00000000E+00
S    1    1.000000
   4.40200000E-02    1.00000000E+00
P    3    1.000000
   1.87100000E+01    1.40310000E-02
   4.13300000E+00    8.68660000E-02
   1.20000000E+00    2.90216000E-01
P    1    1.000000
   3.82700000E-01    1.00000000E+00
P    1    1.000000
   1.20900000E-01    1.00000000E+00
P    1    1.000000
   3.56900000E-02    1.00000000E+00
D    1    1.000000
   1.09700000E+00    1.00000000E+00
D    1    1.000000
   3.18000000E-01    1.00000000E+00
D    1    1.000000
```

Example 7.104 Projected CAP-EOM-EE-CCSD calculation of H_2, with CAP added at the CCSD step.

```
$molecule
   0 1
   H   0.0000   0.0000   0.0000
   H   0.0000   0.0000   0.7414
$end

$rem
   METHOD          eom-ccsd
   BASIS           cc-pvdz
   EE_STATES       [5,0,0,0,0,0,0,0]
   COMPLEX_CCMAN   1
   THRESH          16
$end

$complex_ccman
   PROJ_CAP  1
   CS_HF     0
   CAP_ETA   1000
   CAP_X     2000
   CAP_Y     2000
   CAP_Z     2000
$end
```

Example 7.105 Projected CAP-EOM-EA-MP2T $\eta-$trajectory generation for the first three states of H_2^-, with CAP applied at the EOM step.

```
$molecule
   0 1
   H   0.0000   0.0000   0.0000
   H   0.0000   0.0000   0.7414
$end

$rem
   METHOD          eom-mp2t
   BASIS           cc-pvdz
   EA_STATES       [3,0,0,0,0,0,0,0]
   COMPLEX_CCMAN   1
   THRESH          16
$end

$complex_ccman
   PROJ_CAP  2
   ETA_STEP  100      ETA step
   CAP_ETA   1000      Initial ETA value
   NSTEPS    20         Number of steps along the trajectory
   CAP_X     2000
   CAP_Y     2000
   CAP_Z     2000
$end
```

Example 7.106 Projected CAP-EOM-EA $\eta-$trajectory generation. CAP is projected onto the first five states of N_2^-.

```
$molecule
   0 1
   N   0.0  0.0 -0.54875676501
   N   0.0  0.0  0.54875676501
   Gh  0.0  0.0  0.0
$end

$rem
   COMPLEX_CCMAN     1
   METHOD            EOM-CCSD
   BASIS             gen
   EA_STATES         [0,0,5,0,0,0,0,0]
   FORCE_SYMMETRY_ON true
$end

$complex_ccman
   PROJ_CAP        3      Project CAP onto set of real EOM eigenvectors
   !PROJ_PROP      1      Request first-order perturbative correction for each point
   CAP_ETA         50
   ETA_STEP        50
   NSTEPS          100
   CAP_X           2760
   CAP_Y           2760
   CAP_Z           4880
   CAP_TYPE        1
$end

$basis
N    0
aug-cc-pvtz
****
Gh   0
S    1    1.000000
   2.88000000E-02    1.00000000E+00
S    1    1.000000
   1.44000000E-02    1.00000000E+00
S    1    1.000000
   0.72000000E-02    1.00000000E+00
P    1    1.000000
   2.45000000E-02    1.00000000E+00
P    1    1.000000
   1.22000000E-02    1.00000000E+00
P    1    1.000000
   0.61000000E-02    1.00000000E+00
D    1    1.000000
   0.755000000E-01   1.00000000E+00
D    1    1.000000
   0.377500000E-01   1.00000000E+00
D    1    1.000000
   0.188750000E-01   1.00000000E+00
****
$end
```

Example 7.107 CBF-EOM-EA-CCSD calculations of water.

```
$molecule
0 1
H          0.00000        0.53835       -0.78304
O          0.00000       -0.01840        0.00000
H          0.00000        0.53835        0.78304
$end

$rem
   METHOD                      ccsd
   BASIS                       cc-pvdz ! important: use gen basis set
   COMPLEX_BASIS               aug-cc-pvdz ! important: use gen basis set
   PURECART                    1111
   EA_STATES                   [1,0,0,1]
   COMPLEX_CCMAN               true
   COMPLEX_EXPONENTS           true
   COMPLEX_THETA               100
   COMPLEX_SCF                 1
   COMPLEX_SCF_GUESS           true
   GEN_SCFMAN                  true
   COMPLEX_METSCF              true
   COMPLEX_N_ELECTRONS         false
   N_FROZEN_CORE               false
   SCF_CONVERGENCE             11
   CC_CONVERGENCE              10
   EOM_DAVIDSON_CONVERGENCE    10
   THRESH                      14
$end

$complex_ccman
cs_alpha 100
$end
```

Example 7.108 CAP-EOM-EE-CC2 calculation of water using Resolution-of-the-Identity approximation.

```
$molecule
   0 1
   H          0.00000        0.53835       -0.78304
   O          0.00000       -0.01840        0.00000
   H          0.00000        0.53835        0.78304
$end

$rem
   METHOD                      cc2
   BASIS                       cc-pvdz
   AUX_BASIS                   rimp2-cc-pvdz
   EA_STATES                   [1,0,0,1]
   COMPLEX_CCMAN               true
   N_FROZEN_CORE               false
   SCF_CONVERGENCE             11
   CC_CONVERGENCE              10
   EOM_DAVIDSON_CONVERGENCE    10
$end

$complex_ccman
CS_HF     1
CAP_ETA   1000
CAP_X     2760
CAP_Y     2760
CAP_Z     4970
CAP_TYPE  1
$end
```

Example 7.109 Formaldehyde, calculating EOM-IP-CCSD-S(D) and EOM-IP-MP2-S(D) energies of 3 valence ionized states.

```
$molecule
   0 1
   C
   H  1  1.096135
   H  1  1.096135  2  116.191164
   O  1  1.207459  2  121.904418  3  -180.000000 0
$end

$rem
   METHOD       eom-ccsd-s(d)
   BASIS        6-31G*
   IP_STATES    [1,0,1,1]
$end

@@@

$molecule
   read
$end

$rem
   METHOD       eom-mp2-s(d)
   BASIS        6-31G*
   IP_STATES    [1,0,1,1]
$end
```

Example 7.110 Formaldehyde, calculating EOM-EE-CCSD states with C-PCM method.

```
$molecule
   0 1
   O
   C   1  1.4
   H   2  1.0  1  120.
   H   2  1.0  1  120.  3  180.0
$end

$rem
   METHOD           eom-ccsd
   BASIS            cc-pvdz
   EE_STATES        [4]
   SOLVENT_METHOD   pcm
$end

$pcm
   theory          cpcm
$end

$solvent
   dielectric      4.34
   dielectric_infi 1.829
$end
```

Example 7.111 NO_2^-, calculating EOM-IP-CCSD states with C-PCM method.

```
$molecule
   -1 1
   N1
   O2 N1 RNO
   O3 N1 RNO O2 AONO

   RNO  = 1.305
   AONO = 106.7
$end

$rem
   METHOD           eom-ccsd
   BASIS            cc-pvdz
   IP_STATES        [2]
   SOLVENT_METHOD   pcm
$end

$pcm
   theory          cpcm
$end

$solvent
   dielectric      4.34
   dielectric_infi 1.829
$end
```

7.10.19 Non-Hartree-Fock Orbitals in EOM Calculations

In cases of problematic open-shell references, *e.g.*, strongly spin-contaminated doublet, triplet or quartet states, one may choose to use DFT orbitals. This can be achieved by first doing DFT calculation and then reading the orbitals and turning Hartree-Fock off (by setting SCF_GUESS = READ MAX_SCF_CYCLES = 0 in the CCMAN or CCMAN2 job). In CCMAN, a more convenient way is just to specify EXCHANGE, *e.g.*, if EXCHANGE = B3LYP, B3LYP orbitals will be computed and used. See Section 6.15 for additional discussion of alternative orbital choices in correlated wave function calculations.

Note: Using non-HF exchange in CCMAN2 is not possible.

7.10.20 Analytic Gradients and Properties for CCSD and EOM-XX-CCSD

The coupled-cluster package in Q-CHEM can calculate properties of target EOM states including permanent dipoles, angular momentum projections, static polarizabilities, $\langle \hat{S}^2 \rangle$ and $\langle \hat{R}^2 \rangle$ values, g-tensors (CCSD only), nuclear gradients (and geometry optimizations). The target state of interest is selected by CC_STATE_TO_OPT in the *$rem* section, which specifies the symmetry and the number of the EOM state. In addition to state properties, calculations of various inter-state properties are available, including transition dipoles, angular momentum matrix elements, two-photon absorption transition moments (and cross-sections), spin–orbit couplings, electronic circular dichroism (ECD) rotatory strengths, Dyson orbitals, and photo-ionization cross sections.

Analytic gradients are available for the CCSD and all EOM-CCSD methods for both closed- and open-shell references (UHF and RHF only), including frozen core/virtual functionality [151] and RI/Cholesky representation of the two-electron integrals (see also Section 6.16). These calculations should be feasible whenever the corresponding single-point energy calculation is feasible.

Note: Gradients for ROHF and non-HF (*e.g.*, B3LYP) orbitals are not yet available.

For the CCSD and EOM-CCSD wave functions, Q-CHEM can calculate permanent and transition dipole moments, oscillator strengths, ECD rotatory strengths [both in length gauge (origin dependent) and in velocity gauge (origin independent)], $\langle \hat{R}^2 \rangle$ (as well as XX, YY and ZZ components separately, which is useful for assigning different Rydberg states, *e.g.*, $3p_x$ vs. $3s$, *etc.*), and the $\langle \hat{S}^2 \rangle$ values. Calculation of g-tensors is available for CCSD wave functions (see Section 10.10.4 and example 10.54).

Note: g-tensors can also be computed with state-interaction approach using EOM-CC and RAS-CI wave-functions,. as described in Ref. 127

Interface of the CCSD and EOM-CCSD codes with the NBO 5.0 package is also available. Furthermore, excited state analyses can be requested for EOM-CCSD excited states. For EOM-MP2, only state properties (dipole moments, angular momentum projections, $\langle \hat{R}^2 \rangle$, $\langle \hat{S}^2 \rangle$ are available). Similar functionality is available for some EOM-OD and CI models (CCMAN only).

Analysis of the real- and complex-valued EOM-CC wave functions can also be performed; see Sections 7.10.29 and 10.2.12. NTO analysis for EOM-IP/DIP /EA/DEA/SF states is, obviously, only available for the transitions between the EOM states, so CC_STATE_TO_OPT keyword needs to be used, as in calculations of transition properties.

Users must be aware of the point group symmetry of the system being studied and also the symmetry of the excited (target) state of interest. It is possible to turn off the use of symmetry using the CC_SYMMETRY. If set to FALSE the molecule will be treated as having C_1 symmetry and all states will be of A symmetry.

Q-CHEM allows flexible control of interstate properties calculations, by using CC_TRANS_PROP rem or rem section *$trans_prop*: the user can request the transitions between all computed EOM target states and the reference state (CC_TRANS_PROP = 1) or the calculations of all transition properties between all computed EOM target states (CC_TRANS_PROP = 2). By default, the reference state is the CCSD reference. To compute transition properties relative to a particular EOM state, use CC_STATE_TO_OPT.

By default, only one-electron properties are computed. To activate calculations of two-electron properties, such as NACs, SOCs, 2PA, additional keywords should be activated, as described below.

The *$trans_prop* input section allows the user to specify precisely which properties and for which pairs of states to computed. When the *$trans_prop* section is present in the input, it overrides the setting of the CC_TRANS_PROP *$rem*

variable. However, for *$trans_prop* to work, CC_TRANS_PROP *does* need to be set.

```
$trans_prop
   state_list          ! Start a list of states
   ee_singlets 1 1     ! state 1: EE singlet with irrep = 1 and istate = 1
   ee_triplets 1 2     ! state 2: EE triplet with irrep = 1 and istate = 2
   ref                 ! state 3: Reference state (can be CC or MP2, but the latter NYI
                       ! in transition prop driver)
   end_list            ! End list
   state_pair_list     ! Start to specify pairs of states,
   3 1                 ! transition from state 3 to state 1 (known bug here: CC state
                       ! needs to be 1st one)
   3 2                 ! transition from state 3 to state 2 (known bug here: CC state
                       ! needs to be 1st one)
   end_pairs           ! End list of pairs
   calc nac            ! Compute NAC for all transition pairs listed before this keyword
   state_list          ! Start another list of states (user is able to request multiple
                       ! state lists for multiple tasks)
   ref                 ! reference state
   ee_singlets 0 0     ! zero means all requested irreps/istate in $rem
   end_list
   calc dipole soc     ! Compute transition dipole and SOC
   calc opdm_norm      ! Compute norm of transition OPDM
$end
```

Notes about the *$trans_prop* input section:

1. `calc` computes properties for the *first* pair list (or state list) *before* it.
2. The pair list is *optional*: if there is no pair list, all possible combinations within the state list will be considered.
3. Options after `calc` include: `nac`, `soc`, `dyson`, `2pa`, `dipole`, `default`, `pcm`, `opdm_norm`, `wfa`, `ang`, `linmom`, `dipole linmom`, `ecd`. Currently, only some of them are implemented.

Note: The *$trans_prop* section is a new feature and is still under development — use at your own risk. Eventually, this section will replace other controls and will become a default.

7.10.20.1 Calculations of Angular Momentum Matrix Elements Using EOM-CC Wave Functions

Angular momentum matrix elements can be useful for analyzing wave functions, assigning state characters, or identifying term symmetry labels in linear molecules. As other one-electron properties, these matrix elements are evaluated using one-particle state and transition density matrices. Angular momentum is a gauge-dependent property, meaning that its value depends on the origin of the gauge. The gauge position can be controlled with *$gauge_origin* input section. The default position of the gauge corresponds to the $(0, 0, 0)$ of the coordinate system.

```
$gauge_origin            Position of the gauge origin for
   1.00  0.200  0.500       angular momentum calculations in angstroms.
$end
```

7.10.20.2 EOM-CC Oscillator Strengths in Velocity and Mixed Gauges

The oscillator strength of one-photon absorption is usually calculated in the length gauge ("lg")

$$f^{\mathrm{lg}} = \frac{2\omega_{fn}}{3} \sum_{\alpha=x,y,z} \langle n|\hat{\mu}_\alpha|f\rangle\langle f|\hat{\mu}_\alpha|n\rangle, \tag{7.98}$$

where $\hat{\mu}$ is the electric dipole moment operator, but it can also be expressed in velocity gauge ("vg") and mixed gauge ("mx") as follows

$$f^{\mathrm{vg}} = \frac{2}{3\omega_{fn}} \sum_{\alpha=x,y,z} \langle n|\hat{p}_\alpha|f\rangle\langle f|\hat{p}_\alpha|n\rangle, \tag{7.99}$$

$$f^{\mathrm{mx}} = \frac{2i}{3} \sum_{\alpha=x,y,z} \langle n|\hat{\mu}_\alpha|f\rangle\langle f|\hat{p}_\alpha|n\rangle, \tag{7.100}$$

where $\hat{p}$ is the linear momentum operator.

All three gauges are implemented as part of the EOM-CCSD transition properties in Q-CHEM. Whereas f^{lg} is computed as the default transition property, the other two must be activated. This is done in the *$trans_prop* input section by specifying `calc linmom` for velocity gauge only, or `calc dipole linmom` for all three gauges. This is illustrated in Example 7.115.

Note that:

- If the calculation of ECD is activated in the input, f^{lg}, f^{vg}, and f^{mx} will also be computed;
- If the energy separation between the two states is very small (threshold = 10^{-6}), as in case of (near)-degenerate states, a warning is issued and the calculation of the velocity gauge strength is skipped.

7.10.20.3 Calculations of EOM-CC Rotatory Strengths (ECD and XCD) in Length and Velocity Gauges

Q-CHEM affords ECD and XCD calculations using EOM-CC wavefunctions using formalism and implementation described in Ref. 6. The rotatory strength of ECD within CC theory is defined as:

$$R^{\mathrm{lg}}_{nf} = -\frac{1}{4} \sum_{\alpha=x,y,z} \left\{ \langle \hat{\mu}^{nf}_\alpha \rangle \langle \hat{L}^{fn}_\alpha \rangle - \langle \hat{L}^{nf}_\alpha \rangle \langle \hat{\mu}^{fn}_\alpha \rangle \right\}, \tag{7.101}$$

$$R^{\mathrm{vg}}_{fn} = -\frac{1}{4\omega_{nf}} \sum_{\alpha=x,y,z} \left\{ \langle \hat{p}^{nf}_\alpha \rangle \langle \hat{L}^{fn}_\alpha \rangle + \langle \hat{L}^{nf}_\alpha \rangle \langle \hat{p}^{fn}_\alpha \rangle \right\} \tag{7.102}$$

in the length ("lg") and velocity ("vg") formulations, respectively. $\hat{\mu}$ is the electric dipole moment operator and $\hat{p}$ is the linear momentum operator. The length-gauge expression in Eq. (7.101) is gauge-origin dependent, whereas Eq. (7.102) is gauge-origin independent.

ECD is available for ground-state and excited-state EOM-EE-CCSD transitions in the valence and core-excited regimes. The latter is based on the fc-CVS-EOM-EE-CCSD framework. The ECD calculations are not available for EOM-IP/EOM-DIP calculations yet.

The computation of rotatory strengths is activated in the input by either activating CC_EOM_ECD or via *$trans_prop* section, as illustrated in Example 7.116. In the latter case, the user can either request the calculation of the transition dipole moment, linear momentum, and angular momentum operators together ("`calc dipole linmom ang`") or request ECD only ("`calc ecd`").

Note that:

- Rotatory strengths will be computed in both length and velocity gauge;

– Oscillator strengths in length, velocity, and mixed gauges will be computed as well;

– If the energy separation between the two states is very small (threshold = 10^{-6}), as in the case of (near)-degenerate s states, a warning is issued and the calculation of the velocity gauge strength is skipped.

7.10.20.4 Calculations of Spin-Orbit Couplings Using EOM-CC Wave Functions

Calculations of spin-orbit couplings (SOCs) for EOM-CC wave functions is available in CCMAN2.[65,211] We employ a perturbative approach in which SOCs are computed as matrix elements of the respective part of the Breit–Pauli Hamiltonian using zero-order non-relativistic wave functions. There are two versions of the code that can be engaged by using CALC_SOC. The old code computes SOC as a matrix element between a pair of states that need to be explicitly computed; thus, it is able to produce only one matrix element per transition. Although this approach can be used to extract the whole matrix through rotations of the molecule,[212] the relative phases of these matrix elements are lost. The new version of the code solves this issue and also improves performance.[211] The new code actively uses Wigner-Eckart's theorem to compute the entire spin-orbit matrix. We partition excitation operators by their spin properties. Since the spin-orbit operator is a triplet operator, we can consider the following triplet excitation operators

$$T^{1,1}_{p,q} = -a^\dagger_{p\alpha} a_{q\beta} \tag{7.103}$$

$$T^{1,0}_{p,q} = \frac{1}{\sqrt{2}} \left(a^\dagger_{p\alpha} a_{q\alpha} - a^\dagger_{p\beta} a_{q\beta} \right) \tag{7.104}$$

$$T^{1,-1}_{p,q} = a^\dagger_{p\beta} a_{q\alpha} \tag{7.105}$$

If we apply the Wigner-Eckart theorem, the following matrix elements are decomposed into the Clebsch-Gordan coefficients and reduced matrix elements:

$$\langle \Psi_I SM | T^{1,k}_{p,q} | \Psi_J S'M' \rangle = \langle S'M' | 1k | SM \rangle \langle \Psi_I S || T^{1,\cdot}_{p,q} || \Psi_J S' \rangle, \tag{7.106}$$

where S and S' denote spins, and M and M' denote spin projections of Ψ_I and Ψ_J, respectively. The reduced matrix elements form a new, spinless triplet transition density matrix, which we will denote as $u^{1,\cdot}$. The spin-orbit reduced matrix elements are obtained as a trace with $u^{1,\cdot}$:

$$\langle \Psi_I S || H_{L_-} || \Psi_J S' \rangle = -\frac{1}{2} \sum_{p,q} h^{SO}_{L_-;p,q} u^{1,\cdot}_{p,q} \tag{7.107}$$

$$\langle \Psi_I S || H_{L_0} || \Psi_J S' \rangle = \frac{\sqrt{2}}{2} \sum_{p,q} h^{SO}_{L_z;p,q} u^{1,\cdot}_{p,q} \tag{7.108}$$

$$\langle \Psi_I S || H_{L_+} || \Psi_J S' \rangle = +\frac{1}{2} \sum_{p,q} h^{SO}_{L_+;p,q} u^{1,\cdot}_{p,q}, \tag{7.109}$$

Applying Wigner–Eckart's theorem again, all possible matrix elements are generated from reduced matrix elements. This algorithm is available for the EOM-EE/SF/IP/EA/DIP /DEA wave functions for MP2 and CCSD references, as well as between the CCSD reference and EOM-EE/SF. Examples 7.126, 7.127, 7.128 illustrate calculations of SOCs

for various EOM-CC models.

Example 7.7.112 Example of an output of the new SOC code for a transition between triplet and singlet states (open-shell reference). Only A→B part is shown. First, one-electron reduced matrix elements and the actual matrix elements are printed. Note the proper symmetry of these elements. Second, mean-field reduced matrix elements are printed. Note a small violation of symmetry, which is restored for the averaged (symmetrized) reduced matrix elements. The actual matrix elements are printed in the end (in this version the symmetrized reduced matrix elements were used for its construction). Since EOM-CC methods are non-Hermitian, A→B and B→A transitions lead to, in general, different properties. In the end of the considered transition the matrix, averaged between A→B and B→A is printed with phases, corresponding to A→B transition. To form a Hermitian Hamiltonian matrix, the user should take the averaged matrix, plug it into a corresponding block of a Hamiltonian matrix; the missing blocks are recovered from other transitions and complex conjugation.

```
  A(Ket)->B(Bra) transition SO matrices
Analyzing Sz ans S^2 of the pair of states...
Ket state:  Computed S^2 = 2.019877 will be treated as 2.000000 Sz = 0.000000
Bra state:  Computed S^2 = 0.019216 will be treated as 0.000000 Sz = 0.000000
Clebsch-Gordan coefficient: <1.000,0.000;1.000,0.000|0.000,0.000> = 0.577
One-electron SO (cm-1)
Reduced matrix elements:
<S|| Hso(L-) ||S'> = (-2.660965,-22.180075)
<S|| Hso(L0) ||S'> = (0.000000,-20.019517)
<S|| Hso(L+) ||S'> = (-2.660965,22.180075)
SOCC = 21.593626
Actual matrix elements:
      |Sz=-1.00>           |Sz=0.00>           |Sz=1.00>
<Sz=0.00|(1.536309,12.805673)(0.000000,-11.558274)(1.536309,-12.805673)
Mean-field SO (cm-1)
Reduced matrix elements:
<S|| Hso(L-) ||S'> = (-2.069801,-9.910572)
<S|| Hso(L0) ||S'> = (0.000000,-10.692261)
<S|| Hso(L+) ||S'> = (-2.005570,9.957712)
Singlet part of <S|| Hso(L0) ||S'> = (-0.000000,0.037101) (excluded from all matrix elements)
L-/L+ Averaged reduced matrix elements:
<S|| Hso(L-) ||S'> = (-2.037685,-9.934142)
<S|| Hso(L+) ||S'> = (-2.037685,9.934142)
SOCC = 10.328006
Actual matrix elements:
      |Sz=-1.00>           |Sz=0.00>           |Sz=1.00>
<Sz=0.00|(1.176458,5.735480)(0.000000,-6.173180)(1.176458,-5.735480)
```

The algorithm is activated by setting CALC_SOC to 1 or 2. The execution relies on a non-zero Clebsch-Gordan coefficient (otherwise, it is not possible to extract the necessary information from the transition density matrix). If the Clebsch-Gordan coefficient is zero, a warning will be printed, and the transition will be skipped. Usually it is possible to select the states, leading to non-zero Clebsch-Gordan coefficients, by selecting SF or EE, alpha or beta electrons in case of EA or IP. Determination of state multiplicities is based on $\hat{S}^2$. In case of a strong spin contamination, a warning is printed, and the closest multiplicity is assumed.

In the case of open-shell references, the resulting reduced matrix elements may violate symmetries implied by the spin-orbit operator. Setting CALC_SOC to 1 will produce these reduced matrix elements, but the actual spin-orbit matrix will be produced from the *averaged* reduced matrix elements, in which the desired symmetry is restored. Setting CALC_SOC to 2 will form spin-orbit matrix without averaging of reduced matrix elements. Although in practice the difference between these two options is small, the averaging may simplify the analysis of splittings. Examples 7.125 and 7.126 illustrate calculation of SOC using SOMF in the acetylene-O complex.

Note: In the case of ECP, attempts to compute SOCs using the above algorithms (with CALC_SOC=1 or 2) will, most likely, result in unphysically low SOC values. Instead, we recommend using effective charges (CALC_SOC=4).

To activate the NTO analysis of the spinless one-particle transition density matrices[209] and to compute spin-orbit integrals over NTOs, set STATE_ANALYSIS = TRUE , MOLDEN_FORMAT = TRUE, in addition to CALC_SOC = 1, 2, or 4.

Q-CHEM also affords computing SOCs using one-electron SO operator with effective nuclear charges[69]. The rational for this approach is that the effect of the mean-field two-electron part of the SOC operator can be approximated by screened nuclear charges. In the case of all-electron calculations (no ECP), this approach is less rigorous than SOMF, but it does not use two-electron integrals at all and is, therefore, less expensive. Moreover, using effective charges allows one to obtain reasonable values of SOCs with ECPs. This calculation is activated by using CALC_SOC=4. Q-CHEM output will contain one-electron SOCs computed with the original and screened (effective) nuclear charges. In the case of open-shell references, SOCs are computed using L_+/L_- averaging (as for the CALC_SOC=1 case). By default, the calculations use effective charges from Fedorov *et al*[69] (tabulated for lithium through astatine, except lanthanides): for all-electron calculations, the effective charges from the 6-31G basis are used, and for ECPs, SBKJC charges are used. Alternatively, the user can provide user-defined effective charges via input section *$soc_eff_charges*, as follows:

```
$soc_eff_charges
   8.0  6.0   !use eff. charge=6 for oxygen (Z=8)
   17.0 11.0 !use eff. charge=11 for chlorine (Z=17)
$end
```

Examples 7.129 and 7.130 illustrate this capability.

The legacy code has the full two-electron treatment and the mean-field approximation. To enable the SOC calculation, transition properties between EOM states must be enabled via CC_TRANS_PROP, and SOC requested setting CALC_SOC to 3. By default, one-electron and mean-field two-electron couplings will be computed. Full two-electron coupling calculation is activated by setting CC_EOM_PROP_TE.

As with other EOM transition properties, the initial EOM state is set by CC_STATE_TO_OPT, and couplings are computed between that state and all other EOM states. In the absence of CC_STATE_TO_OPT, SOCs are computed between the reference state and all EOM-EE or EOM-SF states.

Note: In a spin-restricted case, such as EOM-EE calculations using closed-shell reference state, SOCs between the singlet and triplet EOM manifolds cannot be computed (only SOCs between the reference state and EOM triplets can be calculated). To compute SOCs between EOM-EE singlets and EOM-EE triplets, run the same job with UNRESTRICTED = TRUE, such that triplets and singlets appear in the same manifold.

Note: The most flexible control for computing transition properties is afforded by the *$trans_prop* input section, as described in Section 7.10.20.

7.10.20.5 Calculations of Non-Adiabatic Couplings Using EOM-CC Wave Functions

Calculations of nonadiabatic (derivative) couplings (NACs) for EOM-CC wave functions is available in CCMAN2. We employ Szalay's approach in which couplings are computed by a modified analytic gradient code, via "summed states":[243]

$$h_{IJ}^{x} \equiv \langle \Psi_I | H^x | \Psi_J \rangle = \frac{1}{2} \left(G_{(I+J)} - G_I - G_J \right), \tag{7.110}$$

where, G_I, G_J, and G_{IJ} are analytic gradients for states I, J, and a fictitious summed state $|\Psi_{I+J}\rangle \equiv |\Psi_I\rangle + |\Psi_J\rangle$. Currently, NACs for EE/IP/EA are available.[67]

Note: Note that the individual components of the NAC vector depend on the molecular orientation.

7.10.20.6 Transition moments and cross-sections for two-photon absorption within EOM-EE-CCSD

Calculation of transition moments and cross-sections for two-photon absorption for EOM-EE-CCSD wave functions is available in Q-CHEM (CCMAN2 only). Both CCSD-EOM and EOM-EOM transitions can be computed. The formalism is described in Ref. 186. This feature is available both for canonical and RI/CD implementations. The undamped canonical 2PA cross sections between EOM-DEA-CCSD states are also available. Relevant keywords are CC_EOM_2PA (turns on the calculation, controls NTO calculation), CC_STATE_TO_OPT (used for EOM-EOM transitions); additional customization can be performed using the *$2pa* section. The quantity printed out is the microscopic cross-section δ^{TPA} (also known as rotationally averaged 2PA strength), as defined in Eq. (30) of Ref. 186.

The *$2pa* section is used to specify the range of frequency-pairs satisfying the resonance condition. If *$2pa* section is absent in the input, the transition moments are computed for 2 degenerate photons with total energy matching the excitation energy of each target EOM state (for CCSD-EOM) or each EOM-EOM energy difference (for EOM-EOM transitions): $2h\nu = E_{ex}$ For resonant or near-resonant calculations (*i.e.*, when one of the photons is in resonance with an excited state), damped response theory should be used; such calculations are deployed by adding DAMPED_EPSILON in the *$2pa* section (the value of ϵ is specified in hartrees, 0.001 is usually sufficient).

```
$2pa                      Non-degenerate resonant 2PA
   N_2PA_POINTS 6         Number of frequency pairs
   OMEGA_1 500000 10000   Scans 500 cm$^{-1}$ to 550 cm$^{-1}$
                          in steps of 10 cm$^{-1}$
$end
```

N_2PA_POINTS is the number of frequency pairs across the spectrum. The first value associated with OMEGA_1 is the frequency $\times 1000$ in cm^{-1} at the start of the spectrum and the second value is the step size $\times 1000$ in cm^{-1}. The frequency of the second photon at each step is determined within the code as the excitation energy minus OMEGA_1.

To gain insight into computed cross sections for 2PA, one can perform NTO analysis of the response one-particle density matrices[187]. To activate NTO analysis of the 2PA response one-particle transition density matrices, set STATE_ANALYSIS = TRUE, MOLDEN_FORMAT = TRUE (to export the orbitals as MOLDEN files), NTO_PAIRS (specifies the number of orbitals to print). The NTO analysis will be performed for the full 2PA response one-particle transition density matrices as well as the normalized ωDMs (see Ref. 187 for more details).

By default, both undamped and damped 2PA response wave functions are computed using a new iterative Davidson-like subspace procedure called the DALTON solver.[50,128] It is controlled by the DALTON_XCONV, DALTON_PRECOND_START, DALTON_MAXITER, and DALTON_MAXSPACE keywords. Damped 2PA response wave functions can also be computed using the older DIIS procedure by setting DAMPED_DALTON_SOLVER = FALSE. The DIIS solver is controlled by the CC_MAX_ITER, CC_DIIS_START, CC_DIIS_SIZE, CC_EOM_2PA_ECONV, and CC_EOM_2PA_XCONV.

Note: The DALTON solver uses square of the norm of the residual in estimating convergence (as in the Davidson solver), whereas the DIIS solver uses the norm of the difference of the response vector between iterations.

2PA calculations can be carried out using single precision, by activating CC_EOM_2PA_SINGLE_PREC keyword, in addition to the keywords activating single-precision execution of CC and EOM-CC equations (see Section 7.10.14. Capabilities for computing other non-linear properties include RIXS (see Section 7.10.8.1) and static and dynamic polarizabilities (Section 7.10.20.8).

7.10.20.7 Calculation of Optical Rotation for CCSD Wave Function

Q-CHEM affords calculations of the optical rotation (OR) tensor and the related specific optical rotation[202] (at a chosen frequency corresponding to the polarimeter's operating lamp) for a CCSD wave function (in CCMAN2)[6]. The theory is based on the expectation-value approach, *i.e.*, the sum-over-states expressions recast into closed-form expressions

using response states[6]. The calculation involves computing regular or damped response wave functions. Length gauge, velocity gauge, and modified velocity gauge are used[203].

The calculation of the regular (i.e., undamped) optical rotation is deployed by setting CC_OPTROT to 5 and involves computing (undamped) density matrices using zero-order and response wave functions. If the frequency of probing linearly-polarized (lp) light is nearly resonant with an excitation energy, the undamped response calculations diverge. By augmenting the frequency with a phenomenological imaginary damping factor (e.g., 0.001 hartrees), the corresponding damped response wave functions are computed.

In the length (velocity) gauge, the real (imaginary) part of the damped optical rotation tensor corresponds to the differential absorption of the two circularly polarized components of the lp light, i.e., to the electronic circular dichroism (ECD). The imaginary (real) part corresponds to the dispersion (ORD). Currently, these features are available for the canonical implementation only.

Note: CC_FULLRESPONSE must be set to 0 and CC_REF_PROP must be set to TRUE for these calculations.

The response equations for OR are solved using the same solver as the response equations for polarizabilities (see Section 7.10.20.8 for relevant keywords).

A typical *$opt_rot* section input for a regular (undamped) optical rotation calculation at the sodium D-line (589.6 nm = 16960.651 cm^{-1}) is:

```
$opt_rot
omega_1  16960651 0 1    !One calculation at D-line
damped_epsilon 0.0       ! Disables damped OR [mandatory].
length_gauge <0 or 1>    ! Computes length-gauge OR if set to 1
velocity_gauge <0 or 1>  ! Computes (modified) velocity gauge OR if set to 1
$end
```

For a damped calculation, the frequency range, damping factor, and gauge choice are selected using the *$opt_rot* section as follows:

```
$opt_rot
omega_1  500000 10000 6  !Scans 500 cm$^{-1}$ to 550 cm$^{-1}$
                         !      in steps of 10 cm$^{-1}$
damped_epsilon 0.01      ! (Optional) Use for damped CCSD OR only.
                         ! Sets a damping factor of 0.01 hartrees.
length_gauge <0 or 1>    ! Computes length-gauge OR if set to 1
velocity_gauge <0 or 1>  !Computes (modified) velocity gauge OR if set to 1
$end
```

Note: Like other transition properties, ECD/XCD spectra can also be computed from the rotatory strengths between EOM-CC/CVS-EOM-CC states. See Section 7.10.20.

7.10.20.8 Calculations of Static and Dynamic Polarizabilities for CCSD and EOM-CCSD Wave Functions

Calculation of the static and dynamic dipole polarizabilities for CCSD and EOM-EE/SF-CCSD wave functions is available in CCMAN2. Polarizabilities are calculated as second derivatives of the CCSD or EOM-CCSD energy[185] as well as using the expectation-value approach, *i.e.*, the sum-over-states expression for the exact wave function[190]. In the analytic derivative approach (or the response-theory formalism), only the response of the cluster amplitudes is taken into the account; orbital relaxation is not included. CC_FULLRESPONSE must be set to 2 to turn off the orbital relaxation but allow amplitude response in EOM calculations, if the polarizabilities are to be computed with this approach (both CC_POL and EOM_POL must equal 1 or 2). CC_POL and EOM_POL = 3 or 4 coupled with CC_FULLRESPONSE = 0 will

enable expectation-value polarizabilities. Note that CC_REF_PROP and CC_EOM_PROP/ must be set to TRUE for these calculations.

Dynamic polarizabilities are enabled using the *$dyn_pol* section, which specifies the frequency range. Without this section, only the static polarizabilities will be computed. CCSD and EOM-EE/SF-CCSD dynamic polarizabilities for imaginary frequencies are also available through the use of the *$dyn_pol* section. CCSD static and dynamic polarizabilities within the damped response theory are also available, which can be turned on by specifying the *damped_epsilon* variable in the *$dyn_pol* section.

```
$dyn_pol
   500000 10000 6  !Scans 500 cm$^{-1}$ to 550 cm$^{-1}$
                   !      in steps of 10 cm$^{-1}$
damped_epsilon 10  !(Optional) Use for damped CCSD polarizabilities only.
                   !Sets a damping factor of 10/1000 hartrees.
$end
```

Currently, these features are available for the canonical implementation only.

Note: EOM-CCSD polarizabilities are available for the EE and SF wave functions only.

By default, both undamped and damped polarizability response wave functions are computed using a new iterative Davidson-like subspace procedure called the DALTON solver.[50,128] It is controlled by the DALTON_XCONV, DALTON_PRECOND_START, DALTON_MAXITER, and DALTON_MAXSPACE keywords. Damped polarizability response wave functions can also be computed using the older DIIS solver by setting DAMPED_DALTON_SOLVER = FALSE. The DIIS solver is controlled by CC_MAX_ITER, CC_DIIS_START, CC_DIIS_SIZE, and CC _CONVERGENCE.

Note: The DALTON solver uses square of the norm of the residual in estimating convergence (as in the Davidson solver), whereas the DIIS solver uses the norm of the difference of the response vector between iterations. The default threshold for convergence in response calculations (2PA, polarizability, RIXS) is 5.

7.10.20.9 Calculations of Static and Dynamic First Hyperpolarizabilities for CCSD Wave Functions

Calculation of the static and dynamic dipole first hyperpolarizabilities for CCSD wave functions is available in CCMAN2 (currently, canonical implementation only). First hyperpolarizabilities are computed within the damped-response expectation-value approach[189]. Setting CC_1HPOL to 1 deploys the algorithm that computes this property with the least number of response wave-function calculations. Setting CC_1HPOL to 3, computes this property by constructing the second-order response density matrices, which involves computing a large number of second-order response wave functions. These second-order response density matrices are then used to compute second-order response natural orbitals using LIBWFA for quantitative wave-function and orbital analyses. Calculations of first hyperpolarizabilities require the *$1hpol* section, which specifies the frequency ranges of the two photons (ω_1 and ω_2; $\omega_3 = \omega_1 + \omega_2$), as well as the corresponding damping factors.

Note: CC_FULLRESPONSE must be set to 0 and CC_REF_PROP must be set to TRUE for these calculations.

```
$1hpol
omega_1 0000000 1500000 3 0.01 !Scans omega_1 from 0 cm$^{-1}$ to (3-1)*1500 cm$^{-1}
                               ! in steps of 1500 cm$^{-1}
                               ! damping factor epsilon_1 is 0.01 hartrees
omega_2 0000000 2500000 4 0.01 !Scans omega_2 from 0 cm$^{-1}$ to (4-1)*2500 cm$^{-1}
                               ! in steps of 2500 cm$^{-1}
                               ! damping factor epsilon_2 is 0.01 hartrees
omega_3 0.01                   ! damping factor epsilon_3 is 0.01 hartrees
$end
```

Example 7.124 computes static and dynamic first hyperpolarizability for CCSD/STO-3G wave function for LiH using the framework of damped response theory and the expectation-value approach. The property for the following set of 3 photons is calculated (frequencies in cm^{-1}): (0, 0; 0), (2500, 0; -2500), (5000, 0; -5000), (0, 3500; -3500), (0, 7000; -7000), (2500, 3500; -6000), (2500, 7000; -9500), (5000, 3500; -8500), (5000, 7000; -12000).

7.10.20.10 Photoionization/photodetachment cross sections and PADs

Q-CHEM can compute quantities describing photo-ionization/photodetachment following the theoretical framework based on Dyson orbitals[83]. One can use computed Dyson orbitals and then calculate cross sections using the stand-alone EZDYSON code[83] or request such calculations to be done by Q-CHEM; the latter version is more efficient. These calculations require Dyson orbitals, which connect the initial and target states, and a wave-function of the free electron. Currently, only plane-wave treatment is available. This feature is only available in *ccman2*.

Total and angular-resolved photoionization cross sections can be computed by using the following set-up. First, the user should request calculation of appropriate initial and target states and the corresponding Dyson orbitals (using CC_DO_DYSON keyword) or the corresponding setting in *$trans_prop* section). Second, CC_FREE_ELECTRON should be set to TRUE. Third, the parameters for the cross sections—such as polarization of the light, energy range of photo-electrons, and details of orientational averaging—should be specified in the *$free_electron* section.

```
$free_electron
N_ENERGY_POINTS 20  !number of points in the kinetic energy grid
KE_MIN 10   !minimum kinetic energy of the free-electron in 1E-2  eV (here, 0.1 eV)
KE_MAX 1010  !maximum kinetic energy of the free-electron in 1E-2  eV  (here, 10.1 eV)
SPIN_DEG 2  !spin degeneracy factor
ORB_DEG 1  !orbital degeneracy factor
POL_X 0  !X-component of the polarization vector in 1E-2 (here, 0)
POL_Y 0  !Y-component of the polarization vector in 1E-2 (here, 0)
POL_Z 100  !Z-component of the polarization vector in 1E-2 (here, 1)
CIRC_POL 0  !if circular polarization is requested
DO_AVERAGE 0  !if orientational averaging to be performed
N_THETA 50  !number of theta-points in the angular grid
N_PHI 10  !number of phi-points in the angular grid
$end
```

Using similar setup, one can also compute overlap between a Dyson orbital and a plane wave. This calculation is activated by the DYSON_PW_COUPLING keyword (CC_DO_DYSON should also be set to TRUE). The parameters for the caluclation, such as number of $|\mathbf{k}|$-values and whether averaging over different $\mathbf{k}$-orientations should be performed,

are specified in *$free_electron* section. Examples 7.163 and 7.164 and excerpts below illustrate this capability. The default is no averaging and five grind points for $|\mathbf{k}|$.

```
$free_electron
k_avg  1  !perform averaging over k-orientations using default grid of 50( theta)x10(phi)
k_grid_points 6  !Coupling computed for |k|-values  from 0.0 eV to 0.1x(k_grid_points -1)
$end
```

Note: The grid of $|\mathbf{k}|$ values is computed as $\sqrt{2E}$ where E runs from 0.0 eV to 0.1*(k_grid_points-1) eV.

```
$free_electron
k_x  100  ! kx=1 (k_x/100)
k_y  100  ! ky=1
k_z  100  ! kz=1
k_grid_points 6  ! coupling computed for  |k|-values  from 0.0 eV to 0.1x(k_grid_points -1)
$end
```

Note: The magnitude of $|\mathbf{k}|$ does not matter, only direction does.

Photoelectron angular distribution (PAD) — $\Omega(\theta, \phi)$, where θ and ϕ are angular coordinates of the photoelectron's vector $\mathbf{k}$ in the lab frame — will be written in separate files for each ionized state (i.e., pad_1A1, pad_1B2, etc). For a linear polarization, asymmetry parameter β will be calculated along with the cross section. Input parameters for the calculation are specified through the *$free_electron* section. The default values are: SPIN_DEG=2, ORB_DEG=1, and DO_AVERAGE=0 (no averaging). The calculations require a number of θ and ϕ points to be specified by the user (defaults are 50 and 10 respectively) to determine the directions of the wave-vector $\mathbf{k}$ for computing PAD. The user should specify either the three components of the polarization (for linear polarization) or the circular polarization; the two options are mutually exclusive. For circular-polarized light, z is the direction of light propargation. If DO_AVERAGE is set to 1, averaging over all molecular orinations is performed and polarization of light is assumed to be along the z-axis (the polarization setting will be ignored). These features are illustrated in examples 7.133-7.137 below.

CC_FREE_ELECTRON

Specifies whether properties involving free electrons will be computed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

DYSON_PW_COUPLING

Calculate overlap between Dyson orbital and plane wave.

TYPE:

LOGICAL

DEFAULT:

FALSE (do not compute coupling)

OPTIONS:

TRUE/FLSE

RECOMMENDATION:

Activate when needed. CC_DO_DYSON must be turned on.

7.10.21 EOM-CC Optimization and Properties Job Control

CC_STATE_TO_OPT

Specifies which state to optimize (or from which state compute EOM-EOM inter-state properties).

TYPE:

INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[i,j] optimize the jth state of the ith irrep.

RECOMMENDATION:

None

Note: The state number should be smaller or equal to the number of excited states calculated in the corresponding irrep.

Note: If analytic gradients are not available, the finite difference calculations will be performed and the symmetry will be turned off. In this case, CC_STATE_TO_OPT should be specified assuming C_1 symmetry, *i.e.*, as [1,N] where N is the number of state to optimize (the states are numbered from 1).

CC_EOM_PROP

Whether or not the non-relaxed (expectation value) one-particle EOM-CCSD target state properties will be calculated. Available properties currently include permanent dipole moment, angular momentum projections, the second moments ($\langle x^2 \rangle$, $\langle y^2 \rangle$, and $\langle z^2 \rangle$) of the electron density along with $\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$. This option is incompatible with JOBTYPE = FORCE, OPT, or FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (EOM-CCSD equations for the left eigenvectors) need to be solved for properties, approximately doubling the cost of calculation for each irrep. The cost of the one-particle properties calculation itself is low. The one-particle density of an EOM-CCSD target state can be analyzed with NBO or LIBWFA packages by specifying the state with CC_STATE_TO_OPT and requesting NBO = TRUE and CC_EOM_PROP = TRUE.

CC_TRANS_PROP

Whether or not the transition dipole moment (in atomic units) and oscillator strength and rotatory strength (in atomic units) for the EOM-CCSD target states will be calculated. By default, the transition dipole moment, angular momentum matrix elements, and rotatory strengths are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition dipole moment, angular momentum matrix elements, and rotatory strengths between a set of EOM-CCSD states and another EOM-CCSD state, the CC_STATE_TO_OPT must be specified for this state.

TYPE:

INTEGER

DEFAULT:

0 (no transition properties will be calculated)

OPTIONS:

1 (calculate transition properties between all computed EOM state and the reference state)

2 (calculate transition properties between all pairs of EOM states)

RECOMMENDATION:

Additional equations (for the left EOM-CCSD eigenvectors plus lambda CCSD equations in case of transition properties between the CCSD reference and EOM-CCSD target states are requested) need to be solved for transition properties, approximately doubling the computational cost. The cost of the transition properties calculation itself is low.

Note: When the *$trans_prop* section is present in the input, it overrides the setting of the CC_TRANS_PROP *$rem* variable. However, for *$trans_prop* to work, CC_TRANS_PROP *does* need to be set.

CC_EOM_ECD

Whether or not the ECD transition moments will be calculated. By default, the transition moments are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition moments between a set of EOM-CCSD states and another EOM-CCSD state, the CC_STATE_TO_OPT must be specified for this state.

TYPE:

LOGICAL

DEFAULT:

FALSE (do not compute ECD transition moments)

OPTIONS:

TRUE Compute ECD transition moments.

RECOMMENDATION:

Activate for chiral molecules only.

CC_EOM_2PA

Whether or not the transition moments and cross-sections for two-photon absorption will be calculated. By default, the transition moments are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition moments between a set of EOM-CCSD states and another EOM-CCSD state, the CC_STATE_TO_OPT must be specified for this state. If 2PA NTO analysis is requested, the CC_EOM_2PA value is redundant as long as CC_EOM_2PA > 0.

TYPE:

INTEGER

DEFAULT:

0 (do not compute 2PA transition moments)

OPTIONS:

1 Compute 2PA using the fastest algorithm (use $\tilde{\sigma}$-intermediates for canonical and σ-intermediates for RI/CD response calculations).

2 Use σ-intermediates for 2PA response equation calculations.

3 Use $\tilde{\sigma}$-intermediates for 2PA response equation calculations.

RECOMMENDATION:

Additional response equations (6 for each target state) will be solved, which increases the cost of calculations. The cost of 2PA moments is about 10 times that of energy calculation. Use the default algorithm. Setting CC_EOM_2PA > 0 turns on CC_TRANS_PROP.

CC_EOM_2PA_XCONV

Convergence criterion for the response vectors (norm of the difference) of the DIIS solver for damped response equations in 2PA and RIXS calculations.

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion.

RECOMMENDATION:

Use the default in double precision. May reduce in single precision.

DAMPED_DALTON_SOLVER

Boolean for using the new Davidson-like solver (DALTON) for damped (CCSD polarizabilities and hyperpolarizabilities and EOM-CCSD 2PA and RIXS cross sections) response equations.

TYPE:

LOGICAL

DEFAULT:

TRUE (Use the new DALTON solver)

OPTIONS:

FALSE If the old DIIS solver is desired for the above properties.

RECOMMENDATION:

Use the new solver for faster convergence relative to DIIS.

DALTON_XCONV

Convergence criterion for the residuals (square norm) of the DALTON solver for response equations.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion.

RECOMMENDATION:

Use the default in double precision. May reduce to 5 in single precision.

DALTON_MAXITER

Maximum number of iteration allowed for the DALTON solver for response equations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n User-defined number of iterations.

RECOMMENDATION:

Default is usually sufficient

DALTON_MAXSPACE

Specifies maximum number of vectors in the subspace for the DALTON solver for response equations.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

n Up to n vectors per root before the subspace is reset.

RECOMMENDATION:

Larger values increase disk storage but accelerate and stabilize convergence.

DALTON_PRECOND_START

Specifies the iteration number in the DALTON procedure for response equations from which the preconditioner is applied to the residuals.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n User-defined iteration number.

RECOMMENDATION:

Use default.

CALC_SOC

Whether or not the spin-orbit couplings between CC/EOM/ADC/CIS/TDDFT electronic states will be calculated. In the CC/EOM-CC suite, by default the couplings are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate couplings between EOM states, CC_STATE_TO_OPT must specify the initial EOM state. If NTO analysis is requested, analysis of spinless transition density matrices will be performed and the spin-orbit integrals over NTO pairs will be printed.

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE (no spin-orbit couplings will be calculated)

OPTIONS:

0/FALSE (no spin-orbit couplings will be calculated)

1/TRUE Activates SOC calculation. EOM-CC/EOM-MP2 only: spin-orbit couplings will be computed with the new code with L+/L- averaging

2 EOM-CC/EOM-MP2 only: spin-orbit couplings will be computed with the new code without L+/L- averaging

3 EOM-CC/EOM-MP2 only: spin-orbit couplings will be computed with the legacy code

4 One-electron spin-orbit couplings will be computed with effective nuclear charges (with L+/L- averaging for EOM-CC/MP2)

RECOMMENDATION:

CCMAN2 supports several variants of SOC calculation for EOM-CC/EOM-MP2 methods. One-electron and mean-field two-electron SOCs will be computed by default. To enable full two-electron SOCs, two-particle EOM properties must be turned on (see CC_EOM_PROP_TE).

CALC_NAC

Whether or not nonadiabatic couplings will be calculated for the EOM-CC, CIS, and TDDFT wave functions.

TYPE:

INTEGER

DEFAULT:

0 (do not compute NAC)

OPTIONS:

1 NYI for EOM-CC

2 Compute NACs using Szalay's approach (this what needs to be specified for EOM-CC).

RECOMMENDATION:

Additional response equations will be solved and gradients for all EOM states and for summed states will be computed, which increases the cost of calculations. Request only when needed and do not ask for too many EOM states.

CC_POL

Specifies the approach for calculating the polarizability of the CCSD wave function.

TYPE:

INTEGER

DEFAULT:

0 (CCSD polarizability will not be calculated)

OPTIONS:

1 (analytic-derivative or response-theory mixed symmetric-asymmetric approach)
2 (analytic-derivative or response-theory asymmetric approach)
3 (expectation-value approach with right response intermediates)
4 (expectation-value approach with left response intermediates)
13 (damped-response expectation-value approach with right response intermediates)
14 (damped-response expectation-value approach with left response intermediates)
15 (damped-response expectation-value approach with first-order response density matrices)
23 (expectation-value approach with right response intermediates and imaginary frequencies)

RECOMMENDATION:

CCSD polarizabilities are expensive since they require solving three/six (for static) or six/twelve (for dynamic) additional response equations. Do no request this property unless you need it.

EOM_POL

Specifies the approach for calculating the polarizability of the EOM-CCSD wave function.

TYPE:

INTEGER

DEFAULT:

0 (EOM-CCSD polarizability will not be calculated)

OPTIONS:

1 (analytic-derivative or response-theory mixed symmetric-asymmetric approach)
2 (analytic-derivative or response-theory asymmetric approach)
3 (expectation-value approach with right response intermediates)
4 (expectation-value approach with left response intermediates)
23 (expectation-value approach with right response intermediates and imaginary frequencies)

RECOMMENDATION:

EOM-CCSD polarizabilities are expensive since they require solving three/nine (for static) or six/eighteen (for dynamic) additional response equations. Do no request this property unless you need it.

CC_1HPOL

Specifies the approach for calculating the first hyperpolarizability of the CCSD wave function.

TYPE:

INTEGER

DEFAULT:

0 (CCSD first hyperpolarizability will not be calculated)

OPTIONS:

1 (damped-response expectation-value approach with only first-order response wave functions)
3 (damped-response expectation-value approach with second-order response density matrices for wave-function and natural orbital analyses)

RECOMMENDATION:

CCSD first hyperpolarizabilities are expensive since they require solving a huge number of first- and second-order response equations. Do no request this property unless you need it.

CC_EOM_PROP_TE

Request for calculation of non-relaxed two-particle EOM-CC properties. The two-particle properties currently include $\langle \hat{S}^2 \rangle$. The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is inferior compared to the cost of $\langle \hat{S}^2 \rangle$. The variable CC_EOM_PROP must be also set to TRUE. Alternatively, CC_CALC_SSQ can be used to request $\langle \hat{S}^2 \rangle$ calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are computationally expensive since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation). Do not request the two-particle properties unless you really need them.

CC_FULLRESPONSE

Fully relaxed properties (with or without orbital relaxation terms) will be computed. The variable CC_EOM_PROP must be also set to TRUE.

TYPE:

INTEGER

DEFAULT:

0 (no amplitude and orbital response will be calculated)

OPTIONS:

1 (both amplitude and orbital response will be calculated)

2 (only amplitude response will be calculated)

RECOMMENDATION:

Not available for non-UHF/RHF references. Only available for EOM/CI methods for which analytic gradients are available.

CC_SYMMETRY

Controls the use of symmetry in coupled-cluster calculations

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Use the point group symmetry of the molecule

FALSE Do not use point group symmetry (all states will be of A symmetry).

RECOMMENDATION:

It is automatically turned off for any finite difference calculations, *e.g.* second derivatives.

STATE_ANALYSIS

Activates excited state analyses using LIBWFA.

TYPE:

LOGICAL

DEFAULT:

FALSE (no excited state analyses)

OPTIONS:

TRUE, FALSE

RECOMMENDATION:

Set to TRUE if excited state analysis is required, but also if plots of densities or orbitals are needed. For details see Section 10.2.12.

G_TENSOR

Activates g-tensor calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Don't calculate g-tensor

TRUE (or 1) Calculate g-tensor.

RECOMMENDATION:

None.

Note: g-Tensor calculations are only available for CCSD.

7.10.21.1 Examples

Example 7.113 Geometry optimization for the excited open-shell singlet state, $^1B_2$, of methylene followed by the calculations of the fully relaxed one-electron properties using EOM-EE-CCSD.

```
$molecule
   0 1
   C
   H  1 rCH
   H  1 rCH 2 aHCH

   rCH    = 1.083
   aHCH   = 145.
$end

$rem
   JOBTYPE                   OPT
   METHOD                    EOM-CCSD
   BASIS                     cc-pVTZ
   SCF_GUESS                 CORE
   SCF_CONVERGENCE           9
   EE_SINGLETS               [0,0,0,1]
   EOM_NGUESS_SINGLES        2
   CC_STATE_TO_OPT           [4,1]
   EOM_DAVIDSON_CONVERGENCE  9    use tighter convergence for EOM amplitudes
$end

@@@

$molecule
   read
$end

$rem
   METHOD               EOM-CCSD
   BASIS                cc-pVTZ
   SCF_GUESS            READ
   EE_SINGLETS          [0,0,0,1]
   EOM_NGUESS_SINGLES   2
   CC_EOM_PROP          1  calculate properties for EOM states
   CC_FULLRESPONSE      1  use fully relaxed properties
$end
```

Example 7.114 Property and transition property calculation on the lowest singlet state of CH_2 using EOM-SF-CCSD.

```
$molecule
   0 3
   C
   H  1 rch
   H  1 rch 2 ahch

   rch  = 1.1167
   ahch = 102.07
$end

$rem
   METHOD             eom-ccsd
   BASIS              cc-pvtz
   SCF_GUESS          core
   SCF_CONVERGENCE    9
   SF_STATES          [2,0,0,3]   Get three 1^B2 and two 1^A1 SF states
   CC_EOM_PROP        1
   CC_TRANS_PROP      1
   CC_STATE_TO_OPT    [4,1] First EOM state in the 4th irrep
$end
```

Example 7.115 Calculation of EOM transition strength using length, momentum, and mixed gauges.

```
$comment
EOM-CC oscillator strength using three gauges
$end

$molecule
0 1
C       1.2509987    -0.0000000     0.0000000
C      -1.2509987     0.0000000    -0.0000000
H       2.3262529     1.8903377     0.4190778
H       2.3262529    -1.8903377    -0.4190778
H      -2.3262529     1.8903377    -0.4190778
H      -2.3262529    -1.8903377     0.4190778
$end

$rem
    method = ccsd
    input_bohr = true
    ee_singlets [0,0,2,2]
    basis = 6-31g
    cc_trans_prop = true
    cc_ref_prop = 1
    cc_eom_prop = 1
$end

$trans_prop
  state_list
    ref
    ee_singlets 0 0
  end_list
  calc dipole linmom
$end
```

Example 7.116 Calculation of ECD using EOM-CCSD wavefunctions.

```
$comment
Calculation of ECD  using EOM-CC wave-functions
$end

$molecule
0 1
C       1.2509987    -0.0000000     0.0000000
C      -1.2509987     0.0000000    -0.0000000
H       2.3262529     1.8903377     0.4190778
H       2.3262529    -1.8903377    -0.4190778
H      -2.3262529     1.8903377    -0.4190778
H      -2.3262529    -1.8903377     0.4190778
$end

$rem
method = ccsd
input_bohr = true
ee_singlets [0,0,2,2]
basis = 6-31g
cc_trans_prop = true
cc_ref_prop = 1
cc_eom_prop = 1
cc_eom_ecd  = 1 ! keyword to activate ECD
$end

@@@
$comment
Calculation of ECD  using EOM-CC wave-functions
using trans_prop section
$end

$molecule
0 1
C       1.2509987    -0.0000000     0.0000000
C      -1.2509987     0.0000000    -0.0000000
H       2.3262529     1.8903377     0.4190778
H       2.3262529    -1.8903377    -0.4190778
H      -2.3262529     1.8903377    -0.4190778
H      -2.3262529    -1.8903377     0.4190778
$end

$rem
method = ccsd
input_bohr = true
ee_singlets [0,0,2,2]
basis = 6-31g
cc_trans_prop = true
cc_ref_prop = 1
cc_eom_prop = 1
$end

$trans_prop
state_list
    ref
    ee_singlets 0 0
    end_list
    calc ecd
$end
```

Example 7.7.117 Geometry optimization with tight convergence for the $^2A_1$ excited state of CH_2Cl, followed by calculation of non-relaxed and fully relaxed permanent dipole moment and $\langle \hat{S}^2 \rangle$.

View input online

Example 7.118 CCSD calculation on three A_2 and one B_2 state of formaldehyde. Transition properties will be calculated between the third A_2 state and all other EOM states.

```
$molecule
   0  1
   O
   C  1  1.4
   H  2  1.0  1  120
   H  2  1.0  1  120  3 180
$end

$rem
   BASIS              6-31+G
   METHOD             EOM-CCSD
   EE_STATES          [0,3,0,1]
   CC_STATE_TO_OPT    [2,3]
   CC_TRANS_PROP      true
$end
```

Example 7.119 EOM-IP-CCSD geometry optimization of X $^2B_2$ state of H_2O^+.

```
$molecule
   0 1
   H     0.774767     0.000000     0.458565
   O     0.000000     0.000000    -0.114641
   H    -0.774767     0.000000     0.458565
$end

$rem
   JOBTYPE            opt
   METHOD             eom-ccsd
   BASIS              6-311G
   IP_STATES          [0,0,0,1]
   CC_STATE_TO_OPT    [4,1]
$end
```

Example 7.7.120 CAP-EOM-EA-CCSD geometry optimization of the $^2B_1$ anionic resonance state of formaldehyde. The applied basis is aug-cc-pVDZ augmented by 3s3p diffuse functions on heavy atoms.

View input online

Example 7.121 Calculating resonant 2PA with degenerate photons.

```
$molecule
   0 1
   O
   H  1 0.959
   H  1 0.959 2 104.654
$end

$rem
   METHOD          eom-ccsd
   BASIS           aug-cc-pvtz
   EE_SINGLETS     [1,0,0,0]   1A_1 state
   CC_TRANS_PROP   1    Compute transition properties
   CC_EOM_2PA      1    Calculate 2PA cross-sections using the fastest algorithm
$end
```

Example 7.122 Non-degenerate, resonant 2PA scan over a range of frequency pairs.

```
$molecule
   0 1
   O
   H  1 0.959
   H  1 0.959 2 104.654
$end

$rem
   METHOD          eom-ccsd
   BASIS           aug-cc-pvdz
   EE_SINGLETS     [2,0,0,0]  !Two A_1 states
   CC_TRANS_PROP   1          !Calculate transition properties
   CC_EOM_2PA      1          !Calculate 2PA cross-sections using the fastest algorithm
$end

$2pa
   n_2pa_points 11
   omega_1 500000 5000
$end
```

Example 7.123 Resonant 2PA with degenerate photons between two excited states.

```
$molecule
   0 1
   O
   H  1 0.959
   H  1 0.959 2 104.654
$end

$rem
   METHOD          eom-ccsd
   BASIS           aug-cc-pvtz
   EE_SINGLETS     [2,0,0,0] Two A_1 states
   CC_STATE_TO_OPT [1,1]     "Reference" state for transition properties is 1A_1 state
   CC_TRANS_PROP   1         Compute transition properties
   CC_EOM_2PA      1         Calculate 2PA cross-sections using the fastest algorithm
$end
```

Example 7.124 Calculation of 1st hyperpolarizability for CCSD wave function for LiH using the framework of damped response theory and the expectation-value approach for a range of frequencies.

```
$comment
Calculation of static and dynamical first hyperpolarizability for CCSD wave function
for LiH with STO-3G basis set using the framework of damped response theory and
the expectation-value approach.
The property for the following set of photons is calculated:
(omega_1, omega_2; omega_3) in cm-1
(0, 0; 0), (2500, 0; -2500), (5000, 0; -5000), (0, 3500; -3500),
(0, 7000; -7000), (2500, 3500; -6000), (2500, 7000; -9500),
(5000, 3500; -8500), (5000, 7000; -12000)
$end

$molecule
0 1
H
Li 1 1.6
$end

$rem
METHOD           ccsd
BASIS            sto-3g
CC_REF_PROP      1         ! required for CCSD property calculation
CC_1HPOL         1         ! computes first hyperpolarizability using first-order response wav
MEM_STATIC       400
CC_MEMORY        2000
THRESH           13
SCF_CONVERGENCE  10
CC_CONVERGENCE   9
$end

$1hpol
omega_1 0 2500000 3 0.01  ! scans the first photon frequency from 0 cm-1 to 5000 cm-1 in 3-1=2
omega_2 0 3500000 3 0.01  ! scans the second photon frequency from 0 cm-1 to 7000 cm-1 in 3-1=
omega_3 0.01              ! damping for the third photon is 0.01 hartrees
$end
```

Example 7.125 Mean-field spin-orbit calculation between two excited triplets in acetylene-O intermediate.

```
$molecule
   0 1
   C        -0.0303943366   -0.3149506151   -0.0436827067
   H        -0.1031279784   -1.4353675705   -0.1647400816
   O         1.0178175761    0.2350702146    0.2517598501
   C        -1.3252768442    0.1905302054   -0.4205132671
   H        -2.0767072171    0.1461814657    0.3842573052
$end

$rem
   BASIS             cc-pvdz
   METHOD            eom-ccsd
point_group_symmetry False
   SF_STATES         [2]
   CC_STATE_TO_OPT   [1,1]
   THRESH            14
   CALC_SOC          1
   CC_TRANS_PROP     1
$end
```

Example 7.126 Mean-field spin-orbit calculation between the reference singlet and excited triplet states for acetylene-O intermediate.

```
$molecule
0 1
  C        -0.0303943366   -0.3149506151   -0.0436827067
  H        -0.1031279784   -1.4353675705   -0.1647400816
  O         1.0178175761    0.2350702146    0.2517598501
  C        -1.3252768442    0.1905302054   -0.4205132671
  H        -2.0767072171    0.1461814657    0.3842573052
$end

$rem
   BASIS           cc-pvdz
   METHOD          eom-ccsd
point_group_symmetry False
   EE_TRIPLETS     [1]
   THRESH          14
   CALC_SOC        1
   CC_TRANS_PROP   1
$end
```

Example 7.127 Computation of spin-orbit couplings between closed-shell singlet and $M_S = 1$ triplet state in NH using EOM-SF-CCSD.

```
$molecule
   0 3
   N
   H N 1.0450
$end

$rem
   METHOD          = eom-ccsd
   BASIS           = 6-31g
   SF_STATES       = [1,2,0,0]
   CC_TRANS_PROP   = true
   CALC_SOC        = 3         ! legacy code
   CC_STATE_TO_OPT = [1,1]
$end
```

Example 7.128 SOC calculations between EOM-DEA states.

```
$comment
SOC calculations between EOM-DEA states.
$end

$molecule
+2 1
Si
H  1 1.530
F  1 1.605 2 97.000
$end

$rem
BASIS = 6-31G*
jobtype = sp
method = eom-ccsd
n_frozen_core  0
dea_singlets = [1,1]     Ms=0 triplet
dea_triplets = [0,1]     Ms=-1 triplets
CALC_SOC = 2             Activates SOC calculation with new code
CC_EOM_PROP = 1
CC_TRANS_PROP = true
$end
```

Example 7.129 Computation of spin-orbit couplings in neutral hextet (tpa)Fe complex (stripped ligands) by EOM-EA-MP2 method using pseudopotentials and effective nuclear charges.

```
$comment
Spin-orbit coupling calculation with SBKJC ecp using default effective nuclear
charges.  Computes SOC in neutral hextet (tpa)Fe geometry (stripped ligands)
by EOM-EA-MP2 method and high-spin reference.
$end

$molecule
0 6
  N   -0.0330663   -0.2576466    1.3744726
  N    2.0052862   -0.4826730   -0.5758819
  N   -1.9801232   -0.6573608   -0.6407513
  N   -0.0634263    1.4157074   -0.9056826
 Fe    0.0133915   -0.5560750   -0.6002859
  H   -0.0630940   -1.1802863    1.8620650
  H   -0.8780689    0.2958440    1.6401738
  H    2.3459249   -0.6439674   -1.5498484
  H    2.3506743   -1.2708551    0.0162081
  H   -2.2816187   -0.8324531   -1.6252623
  H   -2.3957415    0.2480261   -0.3283621
  H    0.0565056    1.5985967   -1.9267087
  H    0.7257691    1.8766147   -0.3993792
  H    0.8165651    0.2643868    1.6852415
$end

$rem
   METHOD                = EOM-MP2
   EOM_EA_BETA           = [5]
   MAX_SCF_CYCLES        = 300
   SCF_ALGORITHM         = gdm
   SCF_GUESS             = autosad
   EOM_PRECONV_SINGLES   = 1
   CALC_SOC              = 4
   CC_TRANS_PROP         = 2
   BASIS  =   SBKJC           [  use effective charges of ecps in soc calculation]
   ECP = fit-SBKJC
$end
```

Example 7.130 Computation of spin-orbit coupling in ClO using all-electron basis set, user-provided effective nuclear charges, and EOM-IP wave-functions.

```
$comment
Spin-orbit coupling calculation using all-electron basis set
and user-provided effective nuclear charges by
using EOM-IP wave-functions.
$end

$molecule
-1 1
  Cl   -0.9937913   -0.6696391   -1.9087016
  O    0.3415336   -0.1593825   -1.2619353
$end

$rem
jobtype             = sp
method              = eom-ccsd
basis               = 6-31G
print_general_basis = 1
scf_convergence     = 12
cc_convergence      = 10
eom_davidson_conv   = 8
eom_ip_alpha        = [0,0,1,1]
cc_eom_prop         = 1
cc_ref_prop         = 1
cc_trans_prop       = 1
cc_state_to_opt     = [3,1]
calc_soc            = 4
$end

$soc_eff_charges
8.0 6.0
17.0 11.0
$end
```

Example 7.131 Computation of nonadiabatic couplings between EOM-EE states within triplet (first job) and singlet (second job) manifolds.

```
$molecule
   +1 1
   H          0.00000        0.00000        0.0
   He         0.00000        0.00000        3.0
$end

$rem
   JOBTYPE                  = FORCE
   METHOD                   = EOM-CCSD
   BASIS                    = cc-pVDZ
   INPUT_BOHR               = true
   EE_TRIPLETS              = [2]
   CC_EOM_PROP              = true
   POINT_GROUP_SYMMETRY     = false Do not reorient molecule and turn off YYYYYYYYYYY
   CALC_NAC                 = 2     Invoke Szalay NAC
   EOM_DAVIDSON_CONVERGENCE = 9     tight davidson convergence
   SCF_CONVERGENCE          = 9     Hartree-Fock convergence threshold 1e-9
   CC_CONVERGENCE           = 9
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE                  = FORCE
   METHOD                   = EOM-CCSD
   BASIS                    = cc-pVDZ
   INPUT_BOHR               = true
   EE_STATES                = [2]   singlets
   POINT_GROUP_SYMMETRY     = false Do not reorient molecule and turn off YYYYYYYYYYY
   CALC_NAC                 = 2     Invoke Szalay NAC
   EOM_DAVIDSON_CONVERGENCE = 9     tight davidson convergence
   SCF_CONVERGENCE          = 9     Hartree-Fock convergence threshold 1e-9
   CC_CONVERGENCE           = 9
$end
```

Example 7.132 Calculation of the static dipole polarizability of the CCSD wave function of Helium.

```
$molecule
   0 1
   He
$end

$rem
   METHOD            ccsd
   BASIS             cc-pvdz
   CC_REF_PROP       1
   CC_POL            2
   CC_DIIS_SIZE      15
   CC_FULLRESPONSE   1
$end
```

Example 7.133 Water molecule: Calculating photoionization cross-section and photoelectron angular distribution with averaging over molecular orientations.

```
$comment
Calculating photoionization cross-section and photoelectron angular distribution
with averaging over molecular orientations.
$end

$molecule
 0 1
 O     0.000000000000     0.000000000000    -0.004762594898
 H     0.000000000000     0.801842648150    -0.560344690289
 H     0.000000000000    -0.801842648150    -0.560344690289
$end

$rem
    JOB_TYPE = SP
    METHOD = EOM-CCSD
    BASIS = 6-31G*
    EOM_IP_ALPHA = [1,0,0,1]
    CC_DO_DYSON = TRUE
    CC_FREE_ELECTRON = 1
    MEM_TOTAL = 4000
$end

$free_electron
    N_ENERGY_POINTS 10
    N_THETA 25
    N_PHI 10
    SPIN_DEG 2
    ORB_DEG 1
    DO_AVERAGE 1
    KE_MIN 10
    KE_MAX 1010
$end
```

Example 7.134 Water molecule: Calculating photoionization cross-section, asymmetry parameter and photoelectron angular distribution with a Z-polarized light.

```
$comment
Calculating photoionization cross-section, asymmetry parameter, and PAD
with a Z-polarized light.
$end

$molecule
 0 1
 O     0.000000000000     0.000000000000    -0.004762594898
 H     0.000000000000     0.801842648150    -0.560344690289
 H     0.000000000000    -0.801842648150    -0.560344690289
$end

$rem
    JOB_TYPE = SP
    METHOD = EOM-CCSD
    BASIS = 3-21G
    purecart = 1111
    EOM_IP_ALPHA = [1,0,0,1]
    CC_DO_DYSON = TRUE
    CC_FREE_ELECTRON = 1
    MEM_TOTAL = 4000
$end

$free_electron
    N_ENERGY_POINTS 10
    N_THETA 25
    N_PHI 10
    SPIN_DEG 2
    ORB_DEG 1
    POL_X 0
    POL_Y 0
    POL_Z 100
    KE_MIN 10
    KE_MAX 1010
$end
```

Example 7.135 Water molecule: Calculating photoionization cross-section and photoelectron angular distribution with averaging over molecular orientations.

```
$comment
Calculating photoionization cross-section and PAD
with averaging over molecular orientations.
$end

$molecule
 0 1
 O     0.000000000000     0.000000000000    -0.004762594898
 H     0.000000000000     0.801842648150    -0.560344690289
 H     0.000000000000    -0.801842648150    -0.560344690289
$end

$rem
    JOB_TYPE = SP
    METHOD = EOM-CCSD
    BASIS = 6-31G*
    EOM_IP_ALPHA = [1,0,0,1]
    CC_DO_DYSON = TRUE
    CC_FREE_ELECTRON = 1
    MEM_TOTAL = 4000
$end

$free_electron
    N_ENERGY_POINTS 10
    N_THETA 25
    N_PHI 10
    SPIN_DEG 2
    ORB_DEG 1
    DO_AVERAGE 1
    KE_MIN 10
    KE_MAX 1010
$end
```

Example 7.136 Calculating photoionization cross-sections for the ionization of oxygen 1s (core) of water with Z-polarized light.

```
$comment
Calculating photoionization cross-sections for the ionization of oxygen 1s (core)
of water with Z-polarized light.
$end

$molecule
 0 1
 O     0.000000000000     0.000000000000    -0.004762594898
 H     0.000000000000     0.801842648150    -0.560344690289
 H     0.000000000000    -0.801842648150    -0.560344690289
$end

$rem
    JOB_TYPE = SP
    METHOD = EOM-CCSD
    BASIS = 3-21G
    purecart = 1111
    CVS_IP_STATES = [1,0,0,0]
    N_FROZEN_CORE = FC
    CC_DO_DYSON = TRUE
    CC_FREE_ELECTRON = 1
    MEM_TOTAL = 4000
$end

$free_electron
    N_ENERGY_POINTS 10
    N_THETA 25
    N_PHI 10
    SPIN_DEG 2
    ORB_DEG 1
    POL_X 0
    POL_Y 0
    POL_Z 100
    KE_MIN 10
    KE_MAX 1010
$end
```

Example 7.137 Calculating photoionization cross-section, asymmetry parameter, and PAD with a circularly polarized light.

```
$comment
Calculating photoionization cross-section, asymmetry parameter, and
PAD with a circularly polarized light.
$end

$molecule
 0 1
C -0.28956361 0.08176958 0.92701577
C 1.93885200 -1.15600766 -0.10800484
C -2.79852375 -0.19094696 -0.27656907
O 1.53613696 1.47730615 -0.46998636
H 1.71860882 -2.32670888 -1.80795335
H 3.52273536 -1.64526642 1.13875511
H -0.27744501 0.50560530 2.96417187
H -2.56905279 -0.59002862 -2.30639911
H -3.89960425 1.56511541 -0.07933409
H -3.88892566 -1.73605572 0.59599929
$end

$rem
    JOB_TYPE = SP
    METHOD = EOM-CCSD
    BASIS = 3-21G
    INPUT_BOHR = TRUE
    EOM_IP_ALPHA = [2]
    CC_DO_DYSON = TRUE
    CC_FREE_ELECTRON = 1
    MEM_TOTAL = 4000
$end

$free_electron
    N_ENERGY_POINTS 10
    N_THETA 15
    N_PHI 10
    SPIN_DEG 2
    ORB_DEG 1
    CIRC_POL 1
    DO_AVERAGE 0
    KE_MIN 10
    KE_MAX 1010
$end
```

7.10.22 EOM-CCSDT variants for Exclusively High Accuracy (CCMAN2 only)

The equation-of-motion singles, doubles and triples (EOM-CCSDT) method [164] can be viewed as an extension of EOM-CCSD method in which full triples are employed at the CC level (in addition to singles and doubles) and at the EOM level. Conceptually, EOM-CCSD and EOM-CCSDT are similar; the difference lies only in the truncation level. The CC Hamiltonian matrix being non-Hermitian, the left and right eigenvectors are not Hermitian conjugates of each other but can be chosen to form a mutually biorthogonal set. Using Eq. 7.86, the right eigenvectors can be obtained and this is sufficient for obtaining energies (and excitation energies) of the corresponding target states. For gradients and properties, one must in addition solve for the left eigenvectors of the respective states.

In EOM-CCSDT, the CCSDT Hamiltonian matrix is constructed up to the triples block so as to give one-, two-, and three-body eigenvectors of the CC Hamiltonian. As a result, the EOM-CCSDT computed energies and properties are

highly accurate as compared even to the EOM-CCSD ones, by almost an order of magnitude. For example, EOM-CCSDT provides an accurcy of ~ 0.01 eV for singly excited states and single bond-breaking, and an accuracy of 0.1–0.2 eV for doubly excited states. In contrast, at the EOM-CCSD levels these errors are on the order of 0.1–0.2 eV and > 1 eV, respectively. EOM-CCSDT has $\mathcal{O}(N^8)$ scaling versus $\mathcal{O}(N^6)$ for EOM-CCSD. Unlike EOM-CC(2,3), the EOM-CCSDT approach is rigorously size-intensive, as is EOM-CCSD, but it is expensive enough that it can only be used with careful memory management. Currently, the SF[164], EE, IP, EA, DIP, and DEA[165] variants of EOM-CCSDT are available, for single point energy computations only, and can be requested by setting METHOD = EOM-CCSDT. EOM-CCSDT is implemented within the new coupled-cluster suite (CCMAN2) both within double and within singe precision. Other job-control variables are similar to EOM-CCSD ones.

Illustrative examples employing EOM-CCSDT for computing excitation energies are given below.

Example 7.138 Computation of first excited singlet (B_1) and first excited triplet (B_2) energies of BH using EOM-EE-CCSDT/STO-3G in a single-precision setup.

```
$comment
Computation of energies singlet and a triplet excited states of BH
using EOM-EE-CCSDT/STO-3G
$end

$molecule
0 1
B
H  1  1.234347
$end

$rem
basis=sto-3g
job_type=sp
method eom-ccsdt
cc_single_prec=1
eom_single_prec=1
n_frozen_core fc
ee_singlets=[0,0,1,0]
ee_triplets=[0,0,0,1]
$end
```

Example 7.139 Computation of singlet ground state and lowest triplet state energies of BH using EOM-SF-CCSDT/ STO-3G in a double-precision setup.

```
$comment
Computation of spin-flipping excitation energies of
BH using EOM-SF-CCSDT/STO-3G, UHF reference
$end

$molecule
0 3
B
H 1 1.191857
$end

$rem
jobtype=sp
method eom-ccsdt
basis=sto-3g
sf_states=[1,0,0,1]
eom_nguess_singles=4
n_frozen_core fc
$end
```

Illustrative examples employing EOM-CCSDT method for computing ionization energies/electron affinities are given below.

Example 7.140 Computation of the singlet ground state and lowest four IP state energies of N_2 using EOM-IP-CCSDT/6-31G* in a double-precision setup.

```
$molecule
0 1
N
N 1  1.097685
$end

$rem
method=eom-ccsdt
IP_states=[4]
basis=6-31G*
cc_symmetry=false
EOM_davidson_convergence=6
job_type=sp
$end
```

Example 7.141 Computation of singlet ground state and lowest four EA state energies of C_2 using EOM-EA-CCSDT/6-31G* in a double-precision setup.

```
$molecule
0 1
C
C 1  1.243
$end

$rem
method=eom-ccsdt
EA_states=[4]
basis=6-31G*
cc_symmetry=false
EOM_davidson_convergence=6
job_type=sp
$end
```

Example 7.142 Computation of singlet ground state and lowest two DIP state energies of O using EOM-DIP-CCSDT/6-31G in a double-precision setup.

```
$molecule
-2 1
O
$end

$rem
method=eom-ccsdt
basis=6-31g
DIP_STATES=[2]
cc_symmetry=false
$end
```

Example 7.143 Computation of singlet ground state and lowest two DEA state energies of C using EOM-DEA-CCSDT/6-31G in a double-precision setup.

```
$molecule
2  1
C
$end

$rem
method=eom-ccsdt
basis=6-31g
DEA_STATES=[2]
cc_symmetry=false
$end
```

7.10.23 EOM(2,3) Methods for Higher-Accuracy and Problematic Situations (CCMAN only)

In the EOM-CC(2,3) approach,[115] the transformed Hamiltonian $\bar{H}$ is diagonalized in the basis of the reference, singly, doubly, and triply excited determinants, *i.e.*, the excitation operator R is truncated at triple excitations. The excitation operator T, however, is truncated at double excitation level, and its amplitudes are found from the CCSD equations, just like for EOM-CCSD [or EOM-CC(2,2)] method.

The accuracy of the EOM-CC(2,3) method closely follows that of full EOM-CCSDT [which can be also called EOM-CC(3,3)], whereas computational cost of the former model is less.

The inclusion of triple excitations is necessary for achieving chemical accuracy (1 kcal/mol) for ground state properties. It is even more so for excited states. In particular, triple excitations are crucial for doubly excited states,[115] excited states of some radicals and SF calculations (diradicals, triradicals, bond-breaking) when a reference open-shell state is heavily spin-contaminated. Accuracy of EOM-CCSD and EOM-CC(2,3) is compared in Table 7.10.23.

System	**EOM-CCSD**	**EOM-CC(2,3)**
Singly-excited electronic states	0.1–0.2 eV	0.01 eV
Doubly-excited electronic states	$\geq$ 1 eV	0.1–0.2 eV
Severe spin-contamination of the reference	$\sim$ 0.5 eV	$\leq$ 0.1 eV
Breaking single bond (EOM-SF)	0.1–0.2 eV	0.01 eV
Breaking double bond (EOM-2SF)	$\sim$ 1 eV	0.1–0.2 eV

Table 7.3: Performance of the EOM-CCSD and EOM-CC(2,3) methods

The applicability of the EOM-EE/SF-CC(2,3) models to larger systems can be extended by using their active-space variants, in which triple excitations are restricted to semi-internal ones.

Since the computational scaling of EOM-CC(2,3) method is $\mathcal{O}(N^8)$, these calculations can be performed only for relatively small systems. Moderate size molecules (10 heavy atoms) can be tackled by either using the active space implementation or tiny basis sets. To achieve high accuracy for these systems, energy additivity schemes can be used. For example, one can extrapolate EOM-CCSDT/large basis set values by combining large basis set EOM-CCSD calculations with small basis set EOM-CCSDT ones.

Running the full EOM-CC(2,3) calculations is straightforward, however, the calculations are expensive with the bottlenecks being storage of the data on a hard drive and the CPU time. Calculations with around 80 basis functions are possible for a molecule consisting of four first row atoms (NO dimer). The number of basis functions can be larger for smaller systems.

Note: In EE calculations, one needs to always solve for at least one low-spin root in the first symmetry irrep in order to obtain the correlated EOM energy of the reference. The triples correction to the total reference energy must be used to evaluate EOM-(2,3) excitation energies.

Note: EOM-CC(2,3) works for EOM-EE, EOM-SF, and EOM-IP/EA. In EOM-IP, "triples" correspond to $3h2p$ excitations, and the computational scaling of EOM-IP-CC(2,3) is less.

7.10.24 Active-Space EOM-CC(2,3): Tricks of the Trade (CCMAN only)

Active space calculations are less demanding with respect to the size of a hard drive. The main bottlenecks here are the memory usage and the CPU time. Both arise due to the increased number of orbital blocks in the active space calculations. In the current implementation, each block can contain from 0 up to 16 orbitals of the same symmetry irrep, occupancy, and spin-symmetry. For example, for a typical molecule of C_{2v} symmetry, in a small/moderate basis set (*e.g.*, TMM in 6-31G*), the number of blocks for each index is:

occupied: $(\alpha+\beta)\times(a_1+a_2+b_1+b_2)=2\times 4=8$
virtuals: $(\alpha+\beta)\times(2a_1+a_2+b_1+2b_2)=2\times 6=12$
(usually there are more than 16 a_1 and b_2 virtual orbitals).

In EOM-CCSD, the total number of blocks is $O^2V^2=8^2\times 12^2=9216$. In EOM-CC(2,3) the number of blocks in the EOM part is $O^3V^3=8^3\times 12^3=884736$. In active space EOM-CC(2,3), additional fragmentation of blocks occurs to distinguish between the restricted and active orbitals. For example, if the active space includes occupied and virtual orbitals of all symmetry irreps (this will be a very large active space), the number of occupied and virtual blocks for each index is 16 and 20, respectively, and the total number of blocks increases to 3.3×10^7. Not all of the blocks contain real information, some blocks are zero because of the spatial or spin-symmetry requirements. For the C_{2v} symmetry group, the number of non-zero blocks is about 10–12 times less than the total number of blocks, *i.e.*, 3×10^6. This is the number of non-zero blocks in *one* vector. Davidson diagonalization procedure requires (2*MAX_VECTORS + 2*NROOTS) vectors, where MAX_VECTORS is the maximum number of vectors in the subspace, and NROOTS is the number of the roots to solve for. Taking NROOTS = 2 and MAX_VECTORS = 20, we obtain 44 vectors with the total number of non-zero blocks being 1.3×10^8.

In CCMAN implementation, each block is a logical unit of information. Along with real data, which are kept on a hard drive at all the times except of their direct usage, each non-zero block contains an auxiliary information about its size, structure, relative position with respect to other blocks, location on a hard drive, and so on. The auxiliary information about blocks is *always* kept in memory. Currently, the approximate size of this auxiliary information is about 400 bytes per block. It means, that in order to keep information about one vector (3×10^6 blocks), 1.2 GB of memory is required! The information about 44 vectors amounts 53 GB. Moreover, the huge number of blocks significantly slows down the code.

To make the calculations of active space EOM-CC(2,3) feasible, we need to reduce the total number of blocks. One way to do this is to reduce the symmetry of the molecule to lower or C_1 symmetry group (of course, this will result in more expensive calculation). For example, lowering the symmetry group from C_{2v} to C_s would results in reducing the total number of blocks in active space EOM-CC(2,3) calculations in about $2^6=64$ times, and the number of non-zero blocks in about 30 times (the relative portion of non-zero blocks in C_s symmetry group is smaller compared to that in C_{2v}).

Alternatively, one may keep the MAX_VECTORS and NROOTS parameters of Davidson's diagonalization procedure as small as possible (this mainly concerns the MAX_VECTORS parameter). For example, specifying MAX_VECTORS = 12 instead of 20 would require 30% less memory.

One more trick concerns specifying the active space. In a desperate situation of a severe lack of memory, should the two previous options fail, one can try to modify (increase) the active space in such a way that the fragmentation of active and restricted orbitals would be less. For example, if there is one restricted occupied b_1 orbital and one active occupied B_1 orbital, adding the restricted b_1 to the active space will reduce the number of blocks, by the price of increasing the number of FLOPS. In principle, adding extra orbital to the active space should increase the accuracy of calculations, however, a special care should be taken about the (near) degenerate pairs of orbitals, which should be handled in the same way, *i.e.*, both active or both restricted.

7.10.25 Job Control for EOM-CC(2,3)

EOM-CC(2,3) is invoked by METHOD=EOM-CC(2,3). The following options are available:

EOM_PRECONV_SD

Solves the EOM-CCSD equations, prints energies, then uses EOM-CCSD vectors as initial vectors in EOM-CC(2,3). Very convenient for calculations using energy additivity schemes.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Do n SD iterations

RECOMMENDATION:

Turning this option on is recommended

CC_REST_AMPL

Forces the integrals, T, and R amplitudes to be determined in the full space even though the CC_REST_OCC and CC_REST_VIR keywords are used.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do apply restrictions

TRUE Do not apply restrictions

RECOMMENDATION:

None

CC_REST_TRIPLES

Restricts R_3 amplitudes to the active space, *i.e.*, one electron should be removed from the active occupied orbital and one electron should be added to the active virtual orbital.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Applies the restrictions

RECOMMENDATION:

None

CC_REST_OCC

Sets the number of restricted occupied orbitals including frozen occupied orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Restrict n occupied orbitals.

RECOMMENDATION:

None

CC_REST_VIR

Sets the number of restricted virtual orbitals including frozen virtual orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Restrict n virtual orbitals.

RECOMMENDATION:

None

To select the active space, orbitals can be reordered by specifying the new order in the *$reorder_mo*section. The section consists of two rows of numbers (α and β sets), starting from 1, and ending with n, where n is the number of the last orbital specified.

The following example *$reorder_mo* section shows orbitals 16 and 17 swapped for both α and β electrons:

```
$reorder_mo
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   17  16
   1 2 3 4 5 6 7 8 9 10 11 12 13 14 15   17  16
$end
```

7.10.25.1 Examples

Example 7.144 EOM-SF(2,3) calculations of methylene.

```
$molecule
   0 3
   C
   H 1 CH
   H 1 CH 2 HCH

   CH  = 1.07
   HCH = 111.0
$end

$rem
   METHOD             eom-cc(2,3)
   BASIS              6-31G
   SF_STATES          [2,0,0,2]
   N_FROZEN_CORE      1
   N_FROZEN_VIRTUAL   1
   EOM_PRECONV_SD     20 Get EOM-CCSD energies first (max_iter=20).
$end
```

Example 7.145 This is active-space EOM-SF(2,3) calculations for methane with an elongated CC bond. HF MOs should be reordered as specified in the *$reorder_mo*section such that active space for triples consists of sigma and sigma* orbitals.

```
$molecule
   0 3
   C
   H  1  CH
   H  1  CHX  2  HCH
   H  1  CH   2  HCH  3  A120
   H  1  CH   2  HCH  4  A120

   CH   = 1.086
   HCH  = 109.4712206
   A120 = 120.
   CHX  = 1.8
$end

$rem
   METHOD            eom-cc(2,3)
   BASIS             6-31G*
   SF_STATES         [1,0]
   N_FROZEN_CORE     1
   EOM_PRECONV_SD    20   does eom-ccsd first, max_iter=20
   CC_REST_TRIPLES   1   triples are restricted to the active space only
   CC_REST_AMPL      0   ccsd and eom singles and doubles are full-space
   CC_REST_OCC       4   specifies active space
   CC_REST_VIR       17  specifies active space
   PRINT_ORBITALS    10  (number of virtuals to print)
$end

$reorder_mo
   1 2 5 4 3
   1 2 3 4 5
$end
```

Example 7.146 EOM-IP-CC(2,3) calculation of three lowest electronic states of water cation.

```
$molecule
0 1
   H    0.774767     0.000000     0.458565
   O    0.000000     0.000000    -0.114641
   H   -0.774767     0.000000     0.458565
$end

$rem
   METHOD        eom-cc(2,3)
   BASIS         6-311G
   IP_STATES     [1,0,1,1]
$end
```

7.10.26 Non-Iterative Triples Corrections to EOM-CCSD and CCSD

The effect of triple excitations to EOM-CCSD energies can be included via perturbation theory in an economical $\mathcal{O}(N^7)$ computational scheme. Using EOM-CCSD wave functions as zero-order wave functions, the second order triples correction to the μth EOM-EE or SF state is:

$$\Delta E_{\mu}^{(2)} = -\frac{1}{36} \sum_{i,j,k} \sum_{a,b,c} \frac{\tilde{\sigma}_{ijk}^{abc}(\mu)\sigma_{ijk}^{abc}(\mu)}{D_{ijk}^{abc} - \omega_{\mu}} \tag{7.111}$$

where i, j and k denote occupied orbitals, and a, b and c are virtual orbital indices. ω_{μ} is the EOM-CCSD excitation energy of the μth state. The quantities $\tilde{\sigma}$ and σ are:

$$\begin{aligned} \tilde{\sigma}_{ijk}^{abc}(\mu) &= \langle \Phi_0 | (L_{1\mu} + L_{2\mu})(He^{(T_1+T_2)})_c | \Phi_{ijk}^{abc} \rangle \\ \sigma_{ijk}^{abc}(\mu) &= \langle \Phi_{ijk}^{abc} | [He^{(T_1+T_2)}(R_{0\mu} + R_{1\mu} + R_{2\mu})]_c | \Phi_0 \rangle \end{aligned} \tag{7.112}$$

where, the L and R are left and right eigen-vectors for μth state. Two different choices of the denominator, D_{ijk}^{abc}, define the (dT) and (fT) variants of the correction. In (fT), D_{ijk}^{abc} is just Hartree-Fock orbital energy differences. A more accurate (but not fully orbital invariant) (dT) correction employs the complete three body diagonal of $\bar{H}$, $\langle \Phi_{ijk}^{abc} | (He^{(T_1+T_2)})_C | \Phi_{ijk}^{abc} \rangle$, D_{ijk}^{abc}as a denominator. For the reference (*e.g.*, a ground-state CCSD wave function), the (fT) and (dT) corrections are identical to the CCSD(2)$_T$ and CR-CCSD(T)$_L$ corrections of Piecuch and coworkers.[205]

The EOM-SF-CCSD(dT) and EOM-SF-CCSD(fT) methods yield a systematic improvement over EOM-SF-CCSD bringing the errors below 1 kcal/mol. For theoretical background and detailed benchmarks, see Ref. 166.

Similar corrections are available for EOM-IP-CCSD,[167] where triples correspond to $3h2p$ excitations and EOM-EA-CCSD, where triples correspond to $2h3p$ excitations.

Note: Due to the orbital non-invariance problem, using (dT) correction is discouraged.

Note: EOM-IP-CCSD(fT) correction is now available both in CCMAN and CCMAN2 .

7.10.26.1 Job Control for Non-Iterative Triples Corrections

Triples corrections are requested by using METHOD or EOM_CORR:

METHOD

Specifies the calculation method.

TYPE:

STRING

DEFAULT:

No default value

OPTIONS:

EOM-CCSD(DT)	EOM-CCSD(dT), available for EE, SF, and IP
EOM-CCSD(FT)	EOM-CCSD(fT), available for EE, SF, IP, and EA
EOM-CCSD(ST)	EOM-CCSD(sT), available for IP

RECOMMENDATION:

None

EOM_CORR

Specifies the correlation level.

TYPE:

STRING

DEFAULT:

None No correction will be computed

OPTIONS:

SD(DT)	EOM-CCSD(dT), available for EE, SF, and IP
SD(FT)	EOM-CCSD(fT), available for EE, SF, IP, and EA
SD(ST)	EOM-CCSD(sT), available for IP

RECOMMENDATION:

None

Note: In CCMAN2, EOM-IP-CCSD(fT) can be computed with or without USE_LIBPT = TRUE.

7.10.26.2 Examples

Example 7.147 EOM-EE-CCSD(fT) calculation of CH^+.

```
$molecule
   1 1
   C
   H  C  1.13092
$end

$rem
   METHOD                 eom-ccsd(ft)
   BASIS                  general
   EE_STATES              [1,0,1,1]
   EOM_DAVIDSON_MAX_ITER  60  increase number of Davidson iterations
$end

$basis
 H   0
 S   3  1.00
       19.24060000         0.3282800000E-01
       2.899200000         0.2312080000
      0.6534000000         0.8172380000
 S   1  1.00
      0.1776000000          1.000000000
 S   1  1.00
      0.0250000000          1.000000000
 P   1  1.00
       1.00000000          1.00000000
 ****
 C   0
 S   6  1.00
       4232.610000         0.2029000000E-02
       634.8820000         0.1553500000E-01
       146.0970000         0.7541100000E-01
       42.49740000         0.2571210000
       14.18920000         0.5965550000
       1.966600000         0.2425170000
 S   1  1.00
       5.147700000          1.000000000
 S   1  1.00
      0.4962000000          1.000000000
 S   1  1.00
      0.1533000000          1.000000000
 S   1  1.00
      0.0150000000          1.000000000
 P   4  1.00
       18.15570000         0.1853400000E-01
       3.986400000         0.1154420000
       1.142900000         0.3862060000
      0.3594000000         0.6400890000
 P   1  1.00
      0.1146000000          1.000000000
 P   1  1.00
      0.0110000000          1.000000000
 D   1  1.00
      0.750000000          1.00000000
 ****
$end
```

Example 7.148 EOM-SF-CCSD(dT) calculations of methylene.

```
$molecule
   0 3
   C
   H 1 CH
   H 1 CH 2 HCH

   CH  = 1.07
   HCH = 111.0
$end

$rem
   METHOD            eom-ccsd(dt)
   BASIS             6-31G
   SF_STATES         [2,0,0,2]
   N_FROZEN_CORE     1
   N_FROZEN_VIRTUAL  1
   CCMAN2            false   !only works in ccman1
$end
```

Example 7.149 EOM-IP-CCSD(dT) calculations of Mg.

```
$molecule
   0 1
   Mg    0.000000     0.000000    0.000000
$end

$rem
   N_FROZEN_CORE  1
   CORRELATION    ccsd
   EOM_CORR       sd(dt)
   BASIS          6-31g
   IP_STATES      [1,0,0,0,0,1,1,1]
   CCMAN2         false               NYI in ccman2
$end
```

7.10.27 Potential Energy Surface Crossing Minimization

EOM-CCSD can also be used as the excited-state method to find minimum energy crossing points (MECPs). See Section 9.8.3 for more general information about MECP calculations.

The potential energy surface crossing optimization procedure finds energy minima of crossing seams. On the seam, the potential surfaces are degenerate in the subspace perpendicular to the plane defined by two vectors: the gradient difference

$$\mathbf{g} = \frac{\partial}{\partial \mathbf{q}}(E_1 - E_2) \tag{7.113}$$

and the derivative coupling

$$\mathbf{h} = \left\langle \Psi_1 \left| \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right| \Psi_2 \right\rangle \tag{7.114}$$

At this time Q-CHEM is unable to locate crossing minima for states which have non-zero derivative coupling. Fortunately, this does not occur often. Minima on the seams of conical intersections of states of different multiplicity can be found as their derivative coupling is zero. Minima on the seams of intersections of states of different point group symmetry can be located as well.

To run a PES crossing minimization, CCSD and EOM-CCSD methods must be employed for the ground and excited state calculations respectively.

Note: MECP optimization is only available for methods with analytic gradients. Finite-difference evaluation of two gradients is not possible.

7.10.27.1 Job Control Options

Note: When performing coupled-cluster-based MECP calculations, one must use the XOPT_STATE_1 and XOPT_STATE_2 keywords and not MECP_STATE1 and MECP_STATE2.

XOPT_STATE_1, XOPT_STATE_2

Specify two electronic states the intersection of which will be searched.

TYPE:

[INTEGER, INTEGER, INTEGER]

DEFAULT:

No default value (the option must be specified to run this calculation)

OPTIONS:

[spin, irrep, state]

spin = 0	Addresses states with low spin, see also EE_SINGLETS or IP_STATES,EA_STATES.
spin = 1	Addresses states with high spin, see also EE_TRIPLETS.
irrep	Specifies the irreducible representation to which the state belongs; for example, in the C_{2v} point group, irreps are ordered 1, 2, 3, 4 for A_1, A_2, B_1, and B_2, respectively.
state	Specifies the state number within the irreducible representation, state = 1 means the lowest excited state, state = 2 is the second excited state, *etc.*.
0, 0, -1	Ground state.

RECOMMENDATION:

Only intersections of states with different spin or symmetry can be calculated at this time.

Note: The spin can only be specified when using closed-shell RHF references. In the case of open-shell references all states are treated together, see also EE_STATES. For example, in spin-flip calculations use spin = 0 regardless of what is the actual multiplicity of the target state.

XOPT_SEAM_ONLY

Orders an intersection seam search only, no minimization is to be performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	Find a point on the intersection seam and stop.
FALSE	Perform a minimization of the intersection seam.

RECOMMENDATION:

In systems with a large number of degrees of freedom it might be useful to locate the seam first by setting this option to TRUE and using that geometry as a starting point for the minimization.

7.10.27.2 Examples

Example 7.150 Optimization of the intersection of $\tilde{A}^1B_2$ and $\tilde{B}^1A_2$ states of the N_3^+ ion at the EOM-EE-CCSD level.

```
$molecule
   1 1
   N1
   N2 N1 rnn
   N3 N2 rnn N1 annn

   rnn=1.46
   annn=70.0
$end

$rem
   JOBTYPE                opt
   METHOD                 eom-ccsd
   BASIS                  6-31g
   EE_SINGLETS            [0,2,0,2]   C2v point group symmetry
   XOPT_STATE_1           [0,4,1]     1B2 low spin state
   XOPT_STATE_2           [0,2,2]     2A2 low spin state
   XOPT_SEAM_ONLY         true        Find the seam only
   GEOM_OPT_TOL_GRADIENT  100
$end

$opt
CONSTRAINT                Set constraints on the N-N bond lengths
   stre  1  2  1.46
   stre  2  3  1.46
ENDCONSTRAINT
$end

@@@

$molecule
   READ
$end

$rem
   JOBTYPE                opt         Optimize the intersection seam
   METHOD                 eom-ccsd
   BASIS                  6-31g
   EE_SINGLETS            [0,2,0,2]
   XOPT_STATE_1           [0,4,1]
   XOPT_STATE_2           [0,2,2]
   GEOM_OPT_TOL_GRADIENT  30
   GEOM_OPT_DRIVER        optimize
$end
```

Example 7.151 Optimization of the intersection of $\tilde{A}^2A_1$ and $\tilde{B}^2B_1$ states of the NO_2 molecule at the EOM-IP-CCSD level.

```
$molecule
   -1  1
   N1
   O2  N1  rno
   O3  N1  rno  O2  aono

   rno  = 1.3040
   aono = 106.7
$end

$rem
   JOBTYPE                opt         Optimize the intersection seam
   UNRESTRICTED           true
   METHOD                 eom-ccsd
   N_FROZEN_CORE          0
   BASIS                  6-31g
   IP_STATES              [1,0,1,0]   C2v point group symmetry
   EOM_FAKE_IPEA          1
   XOPT_STATE_1           [0,1,1]     1A1 low spin state
   XOPT_STATE_2           [0,3,1]     1B1 low spin state
   GEOM_OPT_TOL_GRADIENT  30          Tighten gradient tolerance
   CCMAN2                 false
$END
```

7.10.28 Dyson Orbitals for Ionized or Attached States within the EOM-CCSD Formalism

Dyson orbitals can be used to compute total photodetachment/photoionization cross-sections, as well as angular distribution of photoelectrons. A Dyson orbital is the overlap between the N-electron molecular wave function and the $N-1/N+1$ electron wave function of the corresponding cation/anion:

$$\phi^d(1) = \frac{1}{N-1}\int \Psi^N(1,\ldots,n)\,\Psi^{N-1}(2,\ldots,n)\,d2\cdots dn \tag{7.115}$$

$$\phi^d(1) = \frac{1}{N+1}\int \Psi^N(2,\ldots,n+1),\Psi^{N+1}(1,\ldots,n+1)\,d2\cdots d(n+1) \tag{7.116}$$

For the Hartree-Fock wave functions and within Koopmans' approximation, these are just the canonical HF orbitals. For correlated wave functions, Dyson orbitals are linear combinations of the reference molecular orbitals:

$$\phi^d = \sum_p \gamma_p \phi_p \tag{7.117}$$

$$\gamma_p = \left\langle \Psi^N \middle| p^+ \middle| \Psi^{N-1} \right\rangle \tag{7.118}$$

$$\gamma_p = \left\langle \Psi^N \middle| p \middle| \Psi^{N+1} \right\rangle \tag{7.119}$$

The calculation of Dyson orbitals is straightforward within the EOM-IP/EA-CCSD methods, where cation/anion and initial molecule states are defined with respect to the same MO basis. Since the left and right CC vectors are not the same, one can define correspondingly two Dyson orbitals (left and right):

$$\begin{aligned}\gamma_p^R &= \left\langle \Phi_0 e^{T_1+T_2} L^{EE} \middle| p^+ \middle| R^{IP} e^{T_1+T_2}\Phi_0 \right\rangle \\ \gamma_p^L &= \left\langle \Phi_0 e^{T_1+T_2} L^{IP} \middle| p \middle| R^{EE} e^{T_1+T_2}\Phi_0 \right\rangle\end{aligned} \tag{7.120}$$

The norm of these orbitals is proportional to the one-electron character of the transition.

Dyson orbitals also offer qualitative insight visualizing the difference between molecular and ionized/attached states. In ionization/photodetachment processes, these orbitals can be also interpreted as the wave function of the leaving electron. For additional details, see Refs. 193 and 194. Dyson orbitals can be used for computing total and differential photoelectron cross-sections using a stand-alone EZDYSON code[83] or built-in Q-CHEM's functionality (see section 7.10.20.10).

Dyson orbitals can be computed both for valence states and core-level states;[258] see Section 7.10.8 for calculations of Dyson orbitals within the FC-CVS-EOM framework.

Q-CHEM can also compute the overlap of a Dyson orbital with a plane wave, $\langle \phi^d | e^{i\mathbf{k}r} \rangle$, using the DYSON_PW_COUPLING keyword. The user can request calculations of the overlap for several $|\mathbf{k}|$-values and either perform averaging over $\mathbf{k}$-drirections or perform the calculation for a fixed direction of $\mathbf{k}$. This capability is illustrated by examples 7.163 and 7.164 and explained in section 7.10.20.10.

7.10.28.1 Dyson Orbitals Job Control

The calculation of Dyson orbitals is implemented for the ground (reference) and excited states ionization/electron attachment. To obtain the ground state Dyson orbitals one needs to run an EOM-IP/EA-CCSD calculation, request transition properties calculation by setting CC_TRANS_PROP = TRUE and CC_DO_DYSON = TRUE. The Dyson orbitals decomposition in the MO basis is printed in the output, for all transitions between the reference and all IP/EA states. At the end of the file, also the coefficients of the Dyson orbitals in the AO basis are available.

Two implementations of Dyson orbitals are currently available: (i) the original implementation in CCMAN; and (ii) new implementation in CCMAN2. The CCMAN implementation is using a diffuse orbital trick (*i.e.*, EOM_FAKE_IPEA will be automatically set to TRUE in these calculations). Note: this implementation has a bug affecting the values of norms of Dyson orbitals (the shapes are correct); thus, using this code is strongly discouraged. The CCMAN2 implementation has all types of initial states available: Dyson orbitals from ground CC, excited EOM-EE, and spin-flip EOM-SF states; it is fully compatible with all helper features for EOM calculations, like FNO, RI, Cholesky decomposition. The CCMAN2 implementation can use a user-specified EOM guess (using EOM_USER_GUESS keyword and *$eom_user_guess* section), which is recommended for highly excited states (such as core-ionized states). In addition, CCMAN2 can calculate Dyson orbitals involving meta-stable states (see Section 7.10.9) and core-level states (see Section 7.10.8).

For calculating Dyson orbitals between excited or spin-flip states from the reference configuration and IP/EA states, same CC_TRANS_PROP = TRUE and CC_DO_DYSON = TRUE keywords have to be added to the combination of usual EOM-IP/EA-CCSD and EOM-EE-CCSD or EOM-SF-CCSD calculations. (However, note the separate keyword CC_DO_DYSON_EE = TRUE for CCMAN.) The IP_STATES keyword is used to specify the target ionized states. The attached states are specified by EA_STATES. The EA-SF states are specified by EOM_EA_BETA. The excited (or spin-flipped) states are specified by EE_STATES and SF_STATES. The Dyson orbital decomposition in MO and AO bases is printed for each EE/SF-IP/EA pair of states first for reference, then for all excited states in the order: CC-IP/EA1, CC-IP/EA2,. . ., EE/SF1 - IP/EA1, EE/SF1 - IP/EA2,. . ., EE/SF2 - IP/EA1, EE/SF2 - IP/EA2,. . ., and so on. CCMAN implementation keeps reference transitions separate, in accordance with separating keywords.

CC_DO_DYSON
CCMAN2: starts all types of Dyson orbitals calculations. Desired type is determined by requesting corresponding EOM-XX transitions CCMAN: whether the reference-state Dyson orbitals will be calculated for EOM-IP/EA-CCSD calculations.
TYPE:
LOGICAL
DEFAULT:
FALSE (the option must be specified to run this calculation)
OPTIONS:
TRUE/FALSE
RECOMMENDATION:
none

CC_DO_DYSON_EE
Whether excited-state or spin-flip state Dyson orbitals will be calculated for EOM-IP/EA-CCSD calculations with CCMAN.
TYPE:
LOGICAL
DEFAULT:
FALSE (the option must be specified to run this calculation)
OPTIONS:
TRUE/FALSE
RECOMMENDATION:
none

Dyson orbitals are most easily visualized by setting IQMOL_FCHK = TRUE (equivalently, GUI = 2) and reading the resulting checkpoint file into IQMOL. In addition to the canonical orbitals, the Dyson orbitals will appear under the Surfaces item in the Model View. For step-by-step instructions, see the EZDYSON manual.[82] Alternatively Dyson orbitals can be plotted using IANLTY = 200 and the *$plots* utility. Only the sizes of the box need to be specified, followed by a line of zeros:

```
$plots
   comment
   10   -2   2
   10   -2   2
   10   -2   2
   0     0   0    0
$plots
```

All Dyson orbitals on the Cartesian grid will be written in the resulting `plot.mo` file (only CCMAN). For RHF(UHF) reference, the columns order in `plot.mo` is: $\phi_1^{lr}\alpha$ $(\phi_1^{lr}\beta)$ $\phi_1^{rl}\alpha$ $(\phi_1^{rl}\beta)$ $\phi_2^{lr}\alpha$ $(\phi_2^{lr}\beta)$ ...

In addition, setting the MAKE_CUBE_FILES keyword to TRUE will create cube files for Dyson orbitals which can be viewed with Visual Molecular Dynamics (VMD)[1,118] or other programs; see Section 10.5.5 for details. This option is available for CCMAN and CCMAN2. The Dyson orbitals will be written to files `mo.1.cube`, `mo.2.cube`, ... in the order ϕ_1^{lr} ϕ_1^{rl} ϕ_2^{lr} ϕ_2^{rl} .... For meta-stable states, the real and imaginary parts of the Dyson orbitals are written to separate files in the order $\mathrm{Re}(\phi_1^{lr})$ $\mathrm{Re}(\phi_1^{rl})$ $\mathrm{Re}(\phi_2^{lr})$ $\mathrm{Re}(\phi_2^{rl})$ ... $\mathrm{Im}(\phi_1^{lr})$ $\mathrm{Im}(\phi_1^{rl})$ $\mathrm{Im}(\phi_2^{lr})$ $\mathrm{Im}(\phi_2^{rl})$ ...

Note: Visualization via the MOLDEN format is not available.

7.10.28.2 Examples

Example 7.152 Plotting grd-ex and ex-grd state Dyson orbitals for ionization of the oxygen molecule. The target states of the cation are ^2A_g and $^2B_{2u}$. Works for CCMAN only.

```
$molecule
   0 3
   O    0.000  0.000  0.000
   O    1.222  0.000  0.000
$end

$rem
   BASIS          6-31G*
   METHOD         eom-ccsd
   IP_STATES      [1,0,0,0,0,0,1,0] Target EOM-IP states
   CC_TRANS_PROP  true  request transition OPDMs to be calculated
   CC_DO_DYSON    true  calculate Dyson orbitals
   IANLTY         200
$end

$plots
plots excited states densities and trans densities
   10   -2   2
   10   -2   2
   10   -2   2
   0     0   0    0
$plots
```

Example 7.153 Plotting ex-ex state Dyson orbitals between the 1st $^2A_1$ excited state of the HO radical and the the 1st A_1 and A_2 excited states of HO^-. Works for CCMAN only.

```
$molecule
   -1 1
   H    0.000   0.000  0.000
   O    1.000   0.000  0.000
$end

$rem
   METHOD             eom-ccsd
   BASIS              6-31G*
   IP_STATES          [1,0,0,0]  states of HO radical
   EE_STATES          [1,1,0,0]  excited states of HO-
   CC_TRANS_PROP      2          calculate transition properties
   CC_DO_DYSON        true       calculate Dyson orbitals for ionization from ex. states
   IANLTY             200
$end

$plots
plot excited states densities and trans densities
   10   -2   2
   10   -2   2
   10   -2   2
   0     0   0    0
$plots
```

Example 7.154 Dyson orbitals for ionization of CO molecule; A_1 and B_1 ionized states requested.

```
$molecule
   0 1
   O
   C  O  1.131
$end

$rem
   CORRELATION           CCSD
   BASIS                 cc-pVDZ
   PURECART              111        5d, will be required for ezDyson
   IP_STATES             [1,0,1,0]  (A1,A2,B1,B2)
   CCMAN2                true
   CC_DO_DYSON           true
   CC_TRANS_PROP         true       necessary for Dyson orbitals job
   PRINT_GENERAL_BASIS   true       will be required for ezDyson
$end
```

Example 7.155 Dyson orbitals for ionization of H_2O; core (A_1) state requested — ionization from O(1s).

```
$molecule
   0 1
   O
   H1  O  0.955
   H2  O  0.955  H1  104.5
$end

$rem
   CORRELATION           CCSD
   BASIS                 cc-pVTZ
   PURECART              111        5d, will be required for ezDyson
   IP_STATES             [1,0,0,0]  (A1,A2,B1,B2)
   EOM_USER_GUESS        1          on, further defined in $eom_user_guess
   CCMAN2                true
   CC_DO_DYSON           true
   CC_TRANS_PROP         true       necessary for Dyson orbitals job
   PRINT_GENERAL_BASIS   true       will be required for ezDyson
   N_FROZEN_CORE         false
$end

$eom_user_guess
   1
$end
```

Example 7.156 Dyson orbitals for ionization of NO molecule using EOM-EA and a closed-shell cation reference; A_1 and B_2 states requested.

```
$molecule
   +1 1
   N   0.00000  0.00000  0.00000
   O   0.00000  0.00000  1.02286
$end

$rem
   CORRELATION            CCSD
   BASIS                  aug-cc-pVTZ
   PURECART               111       5d, will be required for ezDyson
   EA_STATES              [1,0,0,1] (A1,A2,B1,B2)
   CCMAN2                 true
   CC_DO_DYSON            true
   CC_TRANS_PROP          true      necessary for Dyson orbitals job
   PRINT_GENERAL_BASIS    true      will be required for ezDyson
$end
```

Example 7.7.157 Dyson orbitals for detachment from the meta-stable $^2\Pi_g$ state of N_2^-.

View input online

Example 7.158 Dyson orbitals for ionization of triplet O_2 and O_2^- at slightly stretched (relative to the equilibrium O_2 geometry); B_{3g} states are requested.

```
$comment
   EOM-IP-CCSD/6-311+G* and EOM-EA-CCSD/6-311+G* levels of theory,
   UHF reference.  Start from O2:
    1) detach electron - ionization of neutral (alpha IP).
    2) attach electron, use EOM-EA w.f. as initial state
       - ionization of anion (beta EA).
$end

$molecule
   0 3
   O   0.00000  0.00000  0.00000
   O   0.00000  0.00000  1.30000
$end

$rem
   CORRELATION           CCSD
   BASIS                 6-311(3+)G*
   PURECART              2222          6d, will be required for ezDyson
   EOM_IP_ALPHA          [0,0,0,1,0,0,0,0]  (Ag,B1g,B2g,B3g,Au,B1u,B2u,B3u)
   EOM_EA_BETA           [0,0,0,1,0,0,0,0]  (Ag,B1g,B2g,B3g,Au,B1u,B2u,B3u)
   CCMAN2                true
   CC_DO_DYSON           true
   CC_TRANS_PROP         true          necessary for Dyson orbitals job
   PRINT_GENERAL_BASIS   true          will be required for ezDyson
$end
```

Example 7.159 Dyson orbitals for ionization of formaldehyde from the first excited state AND from the ground state.

```
$molecule
   0 1
   O     1.535338855      0.000000000     -0.438858006
   C     1.535331598     -0.000007025      0.767790994
   H     1.535342484      0.937663512      1.362651452
   H     1.535342484     -0.937656488      1.362672535
$end

$rem
   CORRELATION           CCSD
   BASIS                 6-31G*
   PURECART              2222    6d, will be required for ezDyson
   CCMAN2                true    new Dyson code
   EE_STATES             [1]
   EOM_IP_ALPHA          [1]
   EOM_IP_BETA           [1]
   CC_TRANS_PROP         true    necessary for Dyson orbitals job
   CC_DO_DYSON           true
   PRINT_GENERAL_BASIS   true    will be required for ezDyson
$end
```

Example 7.160 Dyson orbitals for core ionization of Li atom use Li^+ as a reference, get neutral atom via EOM-EA get 1st excitation for the cation via EOM-EE totally: core ionization AND 1st ionization of Li atom.

```
$molecule
   +1 1
   Li   0.00000  0.00000  0.00000
$end

$rem
   CORRELATION            CCSD
   BASIS                  6-311+G*
   PURECART               2222          6d, will be required for ezDyson
   CCMAN2                 true          new Dyson code
   EE_STATES              [1,0,0,0,0,0,0,0]
   EA_STATES              [1,0,0,0,0,0,0,0]
   EOM_NGUESS_SINGLES     5             to converge to the lowest EA state
   CC_TRANS_PROP          true          necessary for Dyson orbitals job
   CC_DO_DYSON            true
   PRINT_GENERAL_BASIS    true          will be required for ezDyson
$end
```

Example 7.161 Dyson orbitals for ionization of CH_2 from high-spin triplet reference and from the lowest SF state.

```
$molecule
   0 3
   C
   H  1 rCH
   H  1 rCH 2 aHCH

   rCH    = 1.1167
   aHCH   = 102.07
$end

$rem
   CORRELATION           CCSD
   BASIS                 6-31G*
   SCF_GUESS             core
   CCMAN2                true    new Dyson code
   CC_SYMMETRY           false
   SF_STATES             [1]
   EOM_IP_ALPHA          [2]     one should be careful to request
   EOM_EA_BETA           [2]     meaningful spin for EA/IP state(s)
   CC_TRANS_PROP         true    necessary for Dyson orbitals job
   CC_DO_DYSON           true
   IQMOL_FCHK            true    generate formatted checkpoint file for IQMol
$end
```

Example 7.162 Dyson orbitals for ionization of SO^- using EOM-EA to describe the anion states and EOM-SF to describe the neutral; both sets of EOM states are generated using neutral triplet reference.

```
$comment
SO-, calculating Dyson orbitals using EOM-EA to describe
the anion states and EOM-SF to describe the neutral;
both sets of EOM states are generated using triplet reference.
$end

$molecule
0 3
S    0.0000000    0.0000000   -0.5241891
O    0.0000000    0.0000000    1.0676951
$end

$rem
JOBTYPE SP
METHOD            EOM-CCSD
BASIS                   6-31G*
PURECART 111 needed for ezDyson
EA_BETA         [0,0,0,1]  anion state
SF_STATES               [2,2,0,0]  neutral states
CC_DO_DYSON true
CC_TRANS_PROP true
PRINT_GENERAL_BASIS true  needed for ezDyson
$end

$trans_prop
state_list
ea_beta   4 1      !state 1
sf_states 1 1    !state 2
sf_states 1 2    !state 3
sf_states 2 1
sf_states 2 2
end_list
state_pair_list
1 2   ! transition 1 <-> 2
1 3
1 4
1 5
end_pairs
calc dyson
$end
```

Example 7.163 Calculation of overlap between Dyson orbital of HCl- computed with EOM-EA and plane wave for a fixed direction of k.

```
$comment
Computes coupling between a Dyson orbital and a plane wave for HCl-
for fixed direction of k-vector: k = [1,1,1] (no averaging) and
for 6 values of |k|.
$end

$molecule
0 1
H 0 0 0
Cl 0 0 1.60
$end

$rem
basis 6-31+G*
method eom-ccsd
ea_states = [1,0,0,0]
CC_DO_DYSON true
DYSON_PW_COUPLING true
$end

$free_electron
k_x  100  ! the value of kx is  1
k_y  100  ! the value of ky is  1
k_z  100  ! the value of kz is  1
k_grid_points 6  !Coupling will be calculated for 6 |k|-values
$end
```

Example 7.164 Calculation of overlap between Dyson orbital of N2- computed with EOM-EA and plane wave averaged over k-orientations.

```
$comment
Computes coupling betwee a Dyson orbital of N2- and a plane wave with
averaging over k-directions
$end


$molecule
0 1
N 0 0 0
N 0 0 1.42
$end

$rem
basis 6-31+G*
method eom-ccsd
CC_DO_DYSON true
EA_STATES = [0,0,1,0,0,0,0,0]
DYSON_PW_COUPLING true
$end


$free_electron
k_avg  1        !perform averaging over k-orientations using default grid
k_grid_points 6  !Coupling will be calculated for |k|-values  from 0.0 eV to 0.1x(k_grid_point
$end
```

7.10.29 Interpretation of EOM/CI Wave Functions and Orbital Numbering

Analysis of the leading wave function amplitudes is always necessary for determining the character of the state (*e.g.*, HOMO $\rightarrow$ LUMO excitation, open-shell diradical, *etc.*). The CCMAN module print out leading EOM/CI amplitudes using its internal orbital numbering scheme, which is printed in the beginning. The typical CCMAN EOM-CCSD output looks like:

```
Root 1 Conv-d yes Tot Ene= -113.722767530 hartree (Ex Ene 7.9548 eV),
U1^2=0.858795, U2^2=0.141205 ||Res||=4.4E-07
Right U1:
       Value                  i            ->    a
       0.5358                 7( B2  ) B   ->   17( B2  ) B
       0.5358                 7( B2  ) A   ->   17( B2  ) A
      -0.2278                 7( B2  ) B   ->   18( B2  ) B
      -0.2278                 7( B2  ) A   ->   18( B2  ) A
```

This means that this state is derived by excitation from occupied orbital #7 (which has b_2 symmetry) to virtual orbital #17 (which is also of b_2 symmetry). The two leading amplitudes correspond to $\beta \rightarrow \beta$ and $\alpha \rightarrow \alpha$ excitation (the spin

part is denoted by A or B). The orbital numbering for this job is defined by the following map:

```
The orbitals are ordered and numbered as follows:
Alpha orbitals:
Number  Energy    Type    Symmetry  ANLMAN number  Total number:
  0    -20.613    AOCC      A1       1A1     1
  1    -11.367    AOCC      A1       2A1     2
  2     -1.324    AOCC      A1       3A1     3
  3     -0.944    AOCC      A1       4A1     4
  4     -0.600    AOCC      A1       5A1     5
  5     -0.720    AOCC      B1       1B1     6
  6     -0.473    AOCC      B1       2B1     7
  7     -0.473    AOCC      B2       1B2     8

  0      0.071    AVIRT     A1       6A1     9
  1      0.100    AVIRT     A1       7A1     10
  2      0.290    AVIRT     A1       8A1     11
  3      0.327    AVIRT     A1       9A1     12
  4      0.367    AVIRT     A1      10A1     13
  5      0.454    AVIRT     A1      11A1     14
  6      0.808    AVIRT     A1      12A1     15
  7      1.196    AVIRT     A1      13A1     16
  8      1.295    AVIRT     A1      14A1     17
  9      1.562    AVIRT     A1      15A1     18
 10      2.003    AVIRT     A1      16A1     19
 11      0.100    AVIRT     B1       3B1     20
 12      0.319    AVIRT     B1       4B1     21
 13      0.395    AVIRT     B1       5B1     22
 14      0.881    AVIRT     B1       6B1     23
 15      1.291    AVIRT     B1       7B1     24
 16      1.550    AVIRT     B1       8B1     25
 17      0.040    AVIRT     B2       2B2     26
 18      0.137    AVIRT     B2       3B2     27
 19      0.330    AVIRT     B2       4B2     28
 20      0.853    AVIRT     B2       5B2     29
 21      1.491    AVIRT     B2       6B2     30
```

The first column is CCMAN's internal numbering (*e.g.*, 7 and 17 from the example above). This is followed by the orbital energy, orbital type (frozen, restricted, active, occupied, virtual), and orbital symmetry. Note that the orbitals are blocked by symmetries and then ordered by energy within each symmetry block, (*i.e.*, first all occupied a_1, then all a_2, *etc.*), and numbered starting from 0. The occupied and virtual orbitals are numbered separately, and frozen orbitals are excluded from CCMAN numbering. The two last columns give numbering in terms of the final ANLMAN printout (starting from 1), *e.g.*, our occupied orbital #7 will be numbered as $1B_2$ in the final printout. The last column gives the absolute orbital number (all occupied and all virtuals together, starting from 1), which is often used by external visualization routines.

CCMAN2 numbers orbitals by their energy within each irrep keeping the same numbering for occupied and virtual orbitals. This numbering is exactly the same as in the final printout of the SCF wave function analysis. Orbital energies

are printed next to the respective amplitudes. For example, a typical CCMAN2 EOM-CCSD output will look like that:

```
EOMEE-CCSD transition 2/A1
Total energy = -75.87450159 a.u.  Excitation energy = 11.2971 eV.
R1^2 = 0.9396  R2^2 = 0.0604  Res^2 = 9.51e-08

Amplitude    Orbitals with energies
 0.6486       1 (B2) A                 ->    2 (B2) A
             -0.5101                        0.1729
 0.6486       1 (B2) B                 ->    2 (B2) B
             -0.5101                        0.1729
-0.1268       3 (A1) A                 ->    4 (A1) A
             -0.5863                        0.0404
-0.1268       3 (A1) B                 ->    4 (A1) B
             -0.5863                        0.0404
```

which means that for this state, the leading EOM amplitude corresponds to the transition from the first b_2 orbital (orbital energy -0.5101) to the second b_2 orbital (orbital energy 0.1729).

The most complete analysis of EOM-CC calculations is afforded by deploying a general wave-function analysis tool contained in the LIBWFA module and described in Section 10.2.12. The EOM-CC state analysis is activated by setting STATE_ANALYSIS = TRUE. In addition, keywords controlling calculations of state and interstate properties should be set up accordingly.

Note: Wave function analysis is only available for CCMAN2.

Example 7.165 Wave function analysis of the EOM-IP states (He_3^+).

```
$molecule
   0 1
   He
   He   1       R1
   He   2       R1      1       A

   R1 = 1.236447
   A  = 180.00
$end

$rem
   METHOD            = EOM-CCSD
   BASIS             = 6-31G
   IP_STATES         = [1,0,0,0,0,1,0,0]
   CC_EOM_PROP       = true   Analyze state properties (state OPDM)
   CC_REF_PROP       = true   Analyze reference state and difference OPDM
   CC_STATE_TO_OPT   = [1,1]  Compute transition properties wrt 1st EOM state of 1st irrep
   CC_TRANS_PROP     = true   Analyze transitions (transition OPDM)
   STATE_ANALYSIS    = true
   MOLDEN_FORMAT     = true
   NTO_PAIRS         = 2
$end
```

7.10.30 Interface with OpenFermion Package for Quantum Computing

Q-CHEM provides a capability to dump all quantities required for interfacing with the *OpenFermion* package [121] (and similar softwares) for quantum computation. This feature is controlled by the PRINT_QIS keyword. Setting it to TRUE activates the dump: Fock operator, two-electron integrals, and energies of molecular orbitals are saved in separate files

in the home directory. *OpenFermion* (development version, soon to be checked into official github repository) contains an import module, which understands how to read these files, and several unit tests. Example 7.166 illustrates this capability for water molecule.

Note: Core should not be frozen.

PRINT_QIS

Requests to dump stuff needed for OpenFermion.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print stuff for QIS in user directory.

RECOMMENDATION:

Beware of size of the files.

Example 7.166 CCSD/STO-3G calculation of water molecule with print-out for OpenFermion.

```
$comment
Water molecule in STO-3G basis
$end

$molecule
0 1
  H    0.5355326   -0.0489506    0.7341984
  O    0.0563012    0.0114119   -0.1303093
  H   -0.9024750   -0.0253772    0.1144725
$end

$rem
method = ccsd
basis = sto-3g
print_qis = true
gui = 2
n_frozen_core = 0
$end
```

7.11 The ADC(n) Family of Correlated Excited-State Methods

7.11.1 Introduction

The ADC(n) family of correlated excited state methods is a series of size-extensive excited state methods based on perturbation theory. Each order n of ADC presents the excited state equivalent to the well-known nth order Møller-Plesset perturbation theory for the ground state. Currently, the ADC variants ADC(0), ADC(1), ADC(2)-s, ADC(2)-x and ADC(3) are implemented in Q-CHEM.[98,269] The resolution-of-the-identity approximation can be used with any ADC variant. Additionally, there are spin-opposite scaling versions of both ADC(2) variants available.[136,269] Core-excited states for the simulation of X-ray absorption spectra can be computed exploiting the core-valence separation (CVS) approximation. Currently, the CVS-ADC(1), CVS-ADC(2)-s, CVS-ADC(2)-x and CVS-ADC(3) methods are available.[262–264,269] Ionized and electron-attached states can be computed using the non-Dyson IP- and EA-ADC methods. Currently, the IP-ADC(2), IP-ADC(3), EA-ADC(2) and EA-ADC(3) methods are implemented.[58–60,62]

7.11.2 The Algebraic Diagrammatic Construction (ADC) Scheme

The Algebraic Diagrammatic Construction (ADC) of the polarization propagator is an excited state method originating from Green's function theory. It has first been derived employing the diagrammatic perturbation expansion of the polarization propagator using the Møller-Plesset partition of the Hamiltonian.[223] An alternative derivation is available in terms of the intermediate state representation (ISR),[224] which will be presented in the following.

As starting point for the derivation of ADC equations via ISR serves the exact N electron ground state $\left|\Psi_0^N\right\rangle$. From $\left|\Psi_0^N\right\rangle$ a complete set of correlated excited states is obtained by applying physical excitation operators $\hat{C}_J$.

$$\left|\bar{\Psi}_J^N\right\rangle = \hat{C}_J \left|\Psi_0^N\right\rangle \tag{7.121}$$

with

$$\left\{\hat{C}_J\right\} = \left\{c_a^\dagger c_i; c_a^\dagger c_b^\dagger c_i c_j, i<j, a<b; \ldots\right\} \tag{7.122}$$

Yet, the resulting excited states do not form an orthonormal basis. To construct an orthonormal basis out of the $|\bar{\Psi}_J^N\rangle$ the Gram-Schmidt orthogonalization scheme is employed successively on the excited states in the various excitation classes starting from the exact ground state, the singly excited states, the doubly excited states *etc.*. This procedure eventually yields the basis of intermediate states $\{|\tilde{\Psi}_J^N\rangle\}$ in which the Hamiltonian of the system can be represented forming the Hermitian ADC matrix

$$M_{IJ} = \left\langle \tilde{\Psi}_I^N \middle| \hat{H} - E_0^N \middle| \tilde{\Psi}_J^N \right\rangle \tag{7.123}$$

Here, the Hamiltonian of the system is shifted by the exact ground state energy E_0^N. The solution of the secular ISR equation

$$\mathbf{MX} = \mathbf{X\Omega}, \quad \text{with} \quad \mathbf{X}^\dagger\mathbf{X} = \mathbf{1} \tag{7.124}$$

yields the exact excitation energies Ω_n as eigenvalues. From the eigenvectors the exact excited states in terms of the intermediate states can be constructed as

$$\left|\Psi_n^N\right\rangle = \sum_J X_{nJ} \left|\tilde{\Psi}_J^N\right\rangle \tag{7.125}$$

This also allows for the calculation of dipole transition moments via

$$T_n = \left\langle \Psi_n^N \middle| \hat{\mu} \middle| \Psi_0^N \right\rangle = \sum_J X_{nJ}^\dagger \left\langle \tilde{\Psi}_J^N \middle| \hat{\mu} \middle| \Psi_0^N \right\rangle, \tag{7.126}$$

as well as excited state properties via

$$O_n = \left\langle \Psi_n^N \middle| \hat{o} \middle| \Psi_n^N \right\rangle = \sum_{I,J} X_{nI}^\dagger X_{nJ} \left\langle \tilde{\Psi}_I^N \middle| \hat{o} \middle| \Psi_J^N \right\rangle, \tag{7.127}$$

where O_n is the property associated with operator $\hat{o}$.

Up to now, the exact N-electron ground state has been employed in the derivation of the ADC scheme, thereby resulting in exact excitation energies and exact excited state wave functions. Since the exact ground state is usually not known, a suitable approximation must be used in the derivation of the ISR equations. An obvious choice is the nth order Møller-Plesset ground state yielding the nth order approximation of the ADC scheme. The appropriate ADC equations have been derived in detail up to third order.[249,251,252] Due to the dependency on the Møller-Plesset ground state the nth order ADC scheme should only be applied to molecular systems whose ground state is well described by the respective MP(n) method.

As in Møller-Plesset perturbation theory, the first ADC scheme which goes beyond the non-correlated wave function methods in Section 7.2 is ADC(2). ADC(2) is available in a *strict* and an *extended* variant which are usually referred to as ADC(2)-s and ADC(2)-x, respectively. The strict variant ADC(2)-s scales with the 5th power of the basis set. The quality of ADC(2)-s excitation energies and corresponding excited states is comparable to the quality of those obtained with CIS(D) (Section 7.9) or CC2. More precisely, excited states with mostly single excitation character are

well-described by ADC(2)-s, while excited states with double excitation character are usually found to be too high in energy. The ADC(2)-x variant which scales as the sixth power of the basis set improves the treatment of doubly excited states, but at the cost of introducing an imbalance between singly and doubly excited states. As result, the excitation energies of doubly excited states are substantially decreased in ADC(2)-x relative to the states possessing mostly single excitation character with the excitation energies of both types of states exhibiting relatively large errors. Still, ADC(2)-x calculations can be used as a diagnostic tool for the importance doubly excited states in the low-energy region of the spectrum by comparing to ADC(2)-s results. A significantly better description of both singly and doubly excited states is provided by the third order ADC scheme ADC(3). The accuracy of excitation energies obtained with ADC(3) is almost comparable to CC3, but at computational costs that scale with the sixth power of the basis set only.[98]

7.11.3 IP- and EA-ADC

Similar to ADC scheme of the polarization propagator (ADC), the procedure can also be applied to the $(N-1)$- and $(N+1)$-parts of the electron propagator, which is given in its spectral representation as

$$G_{pq}(\omega) = \underbrace{\sum_n \frac{\langle \Psi_0^N | c_p | \Psi_n^{N+1} \rangle \langle \Psi_n^{N+1} | c_q^\dagger | \Psi_0^N \rangle}{\omega + E_0^N - E_n^{N+1}}}_{(N+1)\text{-electron (EA) part}} + \underbrace{\sum_n \frac{\langle \Psi_0^N | c_q^\dagger | \Psi_n^{N-1} \rangle \langle \Psi_n^{N-1} | c_p | \Psi_0^N \rangle}{\omega + E_n^{N-1} - E_0^N}}_{(N-1)\text{-electron (IP) part}} . \tag{7.128}$$

Doing so, the (non-Dyson) IP- and EA-ADC methods up to third order of perturbation theory have been derived.[225,250] As in the case of the ADC scheme of the polarization propagator, the derivation of the same working equations is possible via the ISR formalism, the only difference to the procedure shown in the previous section 7.11.2 being the excitation operators

$$\left\{\hat{C}_J^{N-1}\right\} = \left\{c_i; c_a^\dagger c_i c_j, i<j; \ldots\right\} \qquad \text{IP-ADC} \tag{7.129}$$

$$\left\{\hat{C}_J^{N+1}\right\} = \left\{c_a^\dagger; c_a^\dagger c_b^\dagger c_i, a<b; \ldots\right\} \qquad \text{EA-ADC} \tag{7.130}$$

replacing the electron number-conserving one $\hat{C}_J$ in Eq. (7.121).

Diagonalization of the IP- and EA-ADC secular matrices $\mathbf{M}$ yields electron-detachment energies (or ionization potentials, IPs) and electron-attachment energies (or negative electron affinities, EAs), respectively. In addition, relative spectral intensities of $(N-1)$- and $(N+1)$-transitions are accessible as pole strengths P_n, which are computed according to

$$P_n = \sum_p |x_{pn}|^2 , \tag{7.131}$$

where the x_{pn} are the spectroscopic factors computed by means of the IP- and EA-ADC eigenvectors $\mathbf{X}$ and the matrix of effective transition amplitudes $\mathbf{f}$ using the relations

$$x_{pn}^{N-1} = \sum_J X_{Jn} f_{Jp}^{N-1} = \sum_J X_{Jn} \left\langle \tilde{\Psi}_n^{N-1} \middle| c_p \middle| \Psi_0^N \right\rangle \qquad \text{IP-ADC} \tag{7.132}$$

$$x_{pn}^{N+1} = \sum_J X_{Jn} f_{Jp}^{N+1} = \sum_J X_{Jn} \left\langle \tilde{\Psi}_J^{N+1} \middle| c_p^\dagger \middle| \Psi_0^N \right\rangle \qquad \text{EA-ADC.} \tag{7.133}$$

When requesting electron-detached or electron-attached states, the pole strengths are automatically computed. For IP- and EA-ADC(2) calculations, second-order pole strengths are used [*i.e.*, the IP- and EA-ADC(2) $\mathbf{f}$ matrix is employed in their computation]. As suggested in Ref. 225, for computational reasons the same second-order pole strengths are computed in case of strict IP- and EA-ADC(3) calculations, *i.e.*, using the second-order ground state density throughout the $\mathbf{M}$ matrix equations, ADC_DENSITY_ORDER = 2. When requesting a higher-order ground state density to be used, *e.g.*, by setting ADC_DENSITY_ORDER = 3 (corresponding to the $\Sigma(4)$ scheme[250]) or ADC_DENSITY_ORDER = 4 (corresponding to the $\Sigma(4+)$ scheme,[250] also denoted as standard IP- and EA-ADC(3) schemes), the corresponding

pole strengths are used, *i.e.*, third-order pole strenghts in case of $\Sigma(4)$ and improved third-order pole strengths in case of $\Sigma(4+)$.

The spectroscopic factors also allow for computing the Dyson orbitals $|\phi_n\rangle$ connected to electron detachment and attachment processes

$$|\phi_n\rangle = \sum_p x_{pn} \, |\varphi_p\rangle \, , \tag{7.134}$$

where in the latter relation the φ_p denote HF orbitals. Dyson orbital output is triggered by ADC_DO_DYSON = TRUE together with STATE_ANALYSIS = TRUE. Also see Section 10.2.12 for further details.

7.11.4 Resolution of the Identity ADC Methods

Similar to MP2 and CIS(D), the ADC equations can be reformulated using the resolution-of-the-identity (RI) approximation. This significantly reduces the cost of the integral transformation and the storage requirements. Although it does not change the overall computational scaling of $O(N^5)$ for ADC(2)-s or $O(N^6)$ for ADC(2)-x with the system size, employing the RI approximation will result in computational speed-up of calculations of larger systems.

The RI approximation can be used with all available ADC methods. It is invoked as soon as an auxiliary basis set is specified using AUX_BASIS.

7.11.5 Spin Opposite Scaling ADC(2) Models

The spin-opposite scaling (SOS) approach originates from MP2 where it was realized that the same spin contributions can be completely neglected, if the opposite spin components are scaled appropriately. In a similar way it is possible to simplify the second order ADC equations by neglecting the same spin contributions in the ADC matrix, while the opposite-spin contributions are scaled with appropriate semi-empirical parameters.[103,136,267]

Starting from the SOS-MP2 ground state the same scaling parameter $c_T = 1.3$ is introduced into the ADC equations to scale the t_2 amplitudes. This alone, however, does not result in any computational savings or substantial improvements of the ADC(2) results. In addition, the opposite spin components in the ph/2p2h and 2p2h/ph coupling blocks have to be scaled using a second parameter c_c to obtain a useful SOS-ADC(2)-s model. With this model the optimal value of the parameter c_c has been found to be 1.17 for the calculation of singlet excited states.[267]

To extend the SOS approximation to the ADC(2)-x method yet another scaling parameter c_x for the opposite spin components of the off-diagonal elements in the 2p2h/2p2h block has to be introduced. Here, the optimal values of the scaling parameters have been determined as $c_c = 1.0$ and $c_x = 0.9$ keeping c_T unchanged.[136]

The spin-opposite scaling models can be invoked by setting METHOD to either SOSADC(2) or SOSADC(2)-x. By default, the scaling parameters are chosen as the optimal values reported above, *i.e.*, $c_T = 1.3$ and $c_c = 1.17$ for ADC(2)-s and $c_T = 1.3$, $c_c = 1.0$, and $c_x = 0.9$ for ADC(2)-x. However, it is possible to adjust any of the three parameters by setting ADC_C_T, ADC_C_C, or ADC_C_X, respectively.

7.11.6 Core-Excitation ADC Methods

Core-excited electronic states are located in the high energy X-ray region of the spectrum. Thus, to compute core-excited states using standard diagonalization procedures, which usually solve for the energetically lowest-lying excited states first, requires the calculation of a multitude of excited states. This is computationally very expensive and only feasible for calculations on very small molecules and small basis sets.

The core-valence separation (CVS) approximation solves the problem by neglecting the couplings between core and valence excited states *a priori*.[11,47] Thereby, the ADC matrix acquires a certain block structure which allows to solve only for core-excited states. The application of the CVS approximation is justified, since core and valence excited states are energetically well separated and the coupling between both types of states is very small. To achieve the separation of core and valence excited states the CVS approximation forces the following types of two-electron integrals to zero

$$\begin{aligned} \langle Ip|qr\rangle = \langle pI|qr\rangle = \langle pq|Ir\rangle = \langle pq|rI\rangle = 0 \\ \langle IJ|pq\rangle = \langle pq|IJ\rangle = 0 \\ \langle IJ|Kp\rangle = \langle IJ|pK\rangle = \langle Ip|JK\rangle = \langle pI|JK\rangle = 0, \end{aligned} \tag{7.135}$$

where capital letters I, J, K refer to core orbitals while lower-case letters p, q, r denote non-core occupied or virtual orbitals.

The core-valence approximation is currently available of ADC models up to third order (including the extended variant).[262–264] It can be invoked by setting METHOD to the respective ADC model prefixed by CVS. Besides the general ADC related keywords, two additional keywords in the *$rem* block are necessary to control CVS-ADC calculations:

- ADC_CVS = TRUE switches on the CVS-ADC calculation
- CC_REST_OCC = n controls the number of core orbitals included in the excitation space. The integer n corresponds to the n energetically lowest core orbitals.

Example: cytosine with the molecular formula $C_4H_5N_3O$ includes one oxygen atom. To calculate O 1s core-excited states, CC_REST_OCC has to be set to 1, because the 1s orbital of oxygen is the energetically lowest. To obtain the N 1s core excitations, the integer has to be set to 4, because the 1s orbital of the oxygen atom is included as well, since it is energetically below the three 1s orbitals of the nitrogen atoms. Accordingly, to simulate the C(1s) XAS spectrum of cytosine, CC_REST_OCC must be set to 8.

To obtain the best agreement with experimental data, one should use the CVS-ADC(2)-x method in combination with at least a diffuse triple-ζ basis set.[262–264]

7.11.7 Spin-Flip ADC Methods

The spin-flip (SF) method[137–139,149] is used for molecular systems with few-reference wave functions like diradicals, bond-breaking, rotations around single bonds, and conical intersections. Starting from a triplet ($m_s = 1$) ground state reference a spin-flip excitation operator $\{\hat{C}_J\} = \{c^\dagger_{a\beta}c_{i\alpha};\ c^\dagger_{a\beta}c^\dagger_{b\sigma}c_{i\alpha}c_{j\sigma},\quad \text{a} < \text{b}, \text{i} < \text{j}\}$ is introduced, which flipped the spin of one electron while singlet and ($m_s = 0$) triplet excited target states are yielded. The spin-flip method is implemented for the ADC(2) (strict and extended) and the ADC(3) methods.[149] Note that high-spin ($m_s = 1$) triplet states can be calculated with the SF-ADC method as well using a closed-shell singlet reference state. The number of spin-flip states that shall be calculated is controlled with the *$rem* variable SF_STATES.

7.11.8 CAP/ADC Methods for the Description of Metastable Electronic States

For the description of metastable electronic states and the calculation of positions and widths of such electronic resonances, the *complex absorbing potential* (CAP) methodology[217] has been combined with all available non-CVS ADC methods using a subspace projection approach.[61,234]

In this approach, the CAP is projected onto the subspace spanned by a number of converged (ADC, IP-ADC or EA-ADC) states. For this purpose one-electron state and state-to-state transition densities computed using the second-order ISR are exploited.

The generation of CAP trajectories and determination of the resonance parameters can be done *a posteriori* (see Ref. 221 for details), *i.e.*, only a single electronic structure calculation has to be performed. As a distinct feature of this approach, a series of different CAP onsets can be handled in a single ADC calculation.

CAP/ADC calculations are invoked by setting ADC_CAP = 1, automatically implying a CAP strength of $\eta = 1$ (or CAP_ETA = 100000). Different CAP types can be employed, however, it is generally recommended to use a smoothed Voronoi CAP,[233] which is requested by setting CAP_TYPE = 2. For this CAP type, a series of different onsets can be controlled using the CAP_X, CAP_X_STEP and CAP_X_END keywords. For example, subspace-projected CAP/ADC output for onset values of 2.0, 3.0 and 4.0 a.u. can be obtained by setting CAP_X = 2000, CAP_X_STEP = 1000 and CAP_X_END = 4000.

For further details on different CAP types and their control, also see Section 7.10.9.

7.11.9 Properties and Visualization

The calculation of excited, ionized and electron-attached states using the ADCMAN module yields by default the usual excitation energies, ionization potentials and electron-attachment energies together with the respective excitation amplitudes, as well as the transition dipole moments, oscillator strengths (or pole strengths in case of IP- and EA-ADC calulations), and the norm of the doubles part of the amplitudes (if applicable). In addition, the calculation of excited, ionized and electron-attached state properties, like dipole moments, and transition properties between these states can be requested by setting the *$rem* variables ADC_PROP_ES and ADC_PROP_ES2ES, respectively.

Resonant two-photon absorption cross-sections of excited states can be computed as well, using either sum-over-states expressions or the matrix inversion technique. The calculation via sum-over-state expressions is automatically activated, if ADC_PROP_ES2ES is set. The accuracy of the results, however, strongly depends on the number of states which are included in the summation, *i.e.* the number of states computed. At least, 20-30 excited states (per irreducible representation) are required to yield useful results for the two-photon absorption cross-sections. Alternatively, the resonant two-photon absorption cross-sections can be calculated by setting ADC_PROP_TPA to TRUE. In this case, the computation of a large number of excited states is avoided and there is no dependence on the number of excited states. Instead, an additional linear matrix equation has to be solved for every excited state for which the two-photon absorption cross-section is computed. Thus, the obtained resonant two-photon absorption cross-sections are usually more reliable. The quantity printed out is the microscopic cross-section (also known as rotationally averaged 2PA strength). Specifically, the value $30 \times \delta^m_{TP}$ is printed out where δ^m_{TP} is defined in Eq. (13) of Ref. 269. The quantity printed out is the microscopic cross-section (also known as rotationally averaged 2PA strength). Specifically, the value $30 \times \delta^m_{TP}$ is printed out where δ^m_{TP} is defined in Eq. (13) of Ref. 269.

Furthermore, the ADCMAN module allows for the detailed analysis of the excited states and export of various types of excited state related orbitals and densities. This can be activated by setting the keyword STATE_ANALYSIS. Details on the available analyses and export options can be found in section 10.2.12.

7.11.10 Excited States in Solution with ADC/SS-PCM

ADCMAN is interfaced to the versatile polarizable-continuum model (PCM) implemented in Q-CHEM (Section 11.2.3) and may thus be employed for the calculation and analysis of excited-state wave functions and transitions in solution, or more general in dielectric environments. The interface follows the state-specific approach, and supports a self-consistent equilibration of the solvent field for long-lived excited states commonly referred to as *equilibrium solvation*, as well as the calculation of perturbative corrections for vertical transitions, known as *nonequilibrium solvation*; see also section 11.2.3.3. Combining both approaches, virtually all photochemically relevant processes can be modeled, including ground- and excited-state absorption, fluorescence, phosphorescence, as well as photochemical reactivity.

Requiring only the electron-densities of ground- and excited states of the solute as well as and the dielectric constant and refractive index of the solvent, ADC/SS-PCM is straightforward to set up and supports all orders and variants of ADC for which densities are available via the ISR. This includes all levels of canonical ADC, SOS-ADC, SF-ADC for electronically complicated situations, as well as CVS-ADC for the description of core-excited states with and without the resolution of the identity and frozen-core approximation, restricted closed-shell references as well unrestricted open-shell. The computed solvent-relaxed wave functions can be visualized and analyzed using the interface to LIBWFA.

Although we give a short introduction to the theory in this section, it is limited to a brief, qualitative overview with only the most important equations, leaving out major aspects such as the polarization work. For an overview of the theory see Ref. 105. A more detailed treatment of the theory can be found in Refs. 182 and 183. See also Sections 11.2.3 and 11.2.3.3 of this manual.

7.11.10.1 Modeling the Absorption Spectrum in Solution

(A) Theory

Let us begin with a brief review of the theoretical and technical aspects of the calculation of absorption spectra in solution. For this purpose, one would typically employ the perturbative, state-specific approach in combination with an ADC of second or third order. The first step is a self-consistent reaction field calculation [SCRF, Hartree-Fock with a PCM, Eq. (7.136)].

$$E_0 = \langle 0|\hat{H}^{\text{vac}} + \hat{R}(0)|0\rangle \tag{7.136}$$

The PCM is formally represented by the reaction- or solvent-field operator $\hat{R}$. In practice, $\hat{R}$ is a set of point charges placed on the molecular surface, which are optimized together with the orbitals during the SCF procedure. Note that since $\hat{R}$ accounts for the self-induced polarization of the solute, it depends on solute's wave function (here the ground state), which will in the following be indicated in the subscript $\hat{R}_0$.

After the SCRF is converged, the final surface charges and the respective operator can, according to the Franck-Condon principle, be separated into a "slow" solvent-nuclei related ($\epsilon - n^2$) and "fast", solvent-electron related (n^2) component (using Marcus partitioning, eq. (7.137)), and are stored on disk.

$$\hat{R}(0) \equiv \hat{R}_0^{\text{tot}} = \hat{R}_0^{\text{f}} + \hat{R}_0^{\text{s}} \tag{7.137}$$

$$q^f = \frac{n^2-1}{\epsilon-1} q^{total} \quad q^s = \frac{\epsilon-n^2}{\epsilon-1} q^{total} = q - q^f \tag{7.138}$$

The "polarized" MOs resulting from the SCRF step are subjected to ordinary MP/ADC calculations, which yield the correlation energy for the ground and excitation energies for the excited states, which are both added to the HF energy to obtain the total MP ground and ADC excited-state energies. However, since the MOs contain the interaction with the complete "frozen" solvent-polarization of the SCF ground-state density ($\hat{R}^{\text{tot}}$, *i.e.*, *both* components), the resulting excitation energies violate the Franck-Condon principle, which requires the solvents electronic degrees of freedom (fast component of the polarization) to be relaxed. Furthermore, the solvent field has been obtained for the SCF density, which more often than not provides a poor description of the electrostatic nature of the solute and may in turn lead to systematic errors in the excitation energies.

To account for the relaxation of the solvent electrons and bring the excitation/excited-state energies in accordance with Franck-Condon, we employ the perturbative ansatz shown in eq. (7.139), in which the fast component of the ground-state solvent field $\hat{R}_0^{\text{f}}$ is replaced by the respective quantity computed for the excited-state density $\hat{R}_{\text{i}}^{\text{f}}$. In this framework, the energy of an excited state i computed with the polarized MOs can be identified as the zeroth-order energy [Eq. (7.141)], while the first-order term becomes the difference between the interaction of the zeroth-order excited-state density with the fast component of the ground- and excited-state solvent fields given in eq. (7.142).

$$E_{\text{i}}^{\text{NEq}} = \langle i|\hat{H}^{\text{vac}} + \hat{R}_0^{\text{tot}} + \lambda(\hat{R}_{\text{i}}^{\text{f}} - \hat{R}_0^{\text{f}})|i\rangle \tag{7.139}$$

$$E_i^{\mathrm{NEq}(0)} = E_i^{(0)} + \lambda E_i^{(1)} + \ldots \tag{7.140}$$

$$E_i^{(0)} = \langle i^{(0)}|\hat{H}^{\mathrm{vac}} + \hat{R}_0^{\mathrm{tot}}|i^{(0)}\rangle \tag{7.141}$$

$$E_i^{(1)} = \langle i^{(0)}|\hat{R}_{\mathrm{i}}^{\mathrm{f}} - \hat{R}_0^{\mathrm{f}}|i^{(0)}\rangle \tag{7.142}$$

After the ADC iterations have converged, $\hat{R}_{\mathrm{i}}^{\mathrm{f}}$ is computed from the respective excited-state density and $\hat{R}_0^{\mathrm{f}}$ read from disk to form the first-order correction. Adding this so-called ptSS-term to the zeroth order energy, one arrives at the vertical energy in the nonequilibrium limit $E_{\mathrm{i}}^{\mathrm{NEq}}$. Since the ptSS-term accounts for the response of the electron density of the implicit solvent molecules to excitation of the solute, its always negative and thus lowers the energy of the excited state with respect to the ground state.

To eliminate problems resulting from the poor description of the ground-state solvent field at the SCF level of theory, we use an additive correction that is based on the MP2 ground-state density. In a nutshell, it replaces the interaction between the potential of the difference density of the excited state $\hat{V}_{i^{(0)}} - \hat{V}_{0^{MP}}$ with the SCF solvent field $\hat{Q}_{0^{HF}}$ by the respective interaction with the MP solvent field $\hat{Q}_{0^{MP}}$:

$$E_{PTD} = (\hat{V}_{i^{(0)}} - \hat{V}_{0^{MP}}) \cdot \hat{Q}_{0^{MP}} \tag{7.143}$$

$$- (\hat{V}_{i^{(0)}} - \hat{V}_{0^{MP}}) \cdot \hat{Q}_{0^{HF}} \tag{7.144}$$

We will in the following refer to this as "perturbative, density-based" (PTD) correction and denote the respective approach as ptSS(PTD). Accordingly, the non-PTD corrected results will be denoted ptSS(PTE).

In addition to the ptSS-PCM discussed above, a perturbative variant of the so-called linear response corrections (termed ptLR presented in Ref. 182) is also available. In contrast to the ptSS approach, the ptLR corrections depend on the transition rather than the state (difference) densities. Although the ptLR corrections are always computed and printed, we discourage their use with the correlated ADC variants (2 and higher), for which the ptSS approach is better suited. The ptSS and ptLR approaches are also available for TDDFT as described in Section 11.2.3.3.

A detailed, comprehensive introduction to the theory and implementation, as well as extensive benchmark data for the nonequilibrium formalism in combination with all orders of ADC can be found in Ref. 182.

(B) Usage

The calculation of vertical transition energies with the ptSS-PCM approach is fairly straightforward. One just needs to activate the PCM (set SOLVENT_METHOD in the *$rem* block to PCM), enable the nonequilibrium functionality (set NONEQUILIBRIUM in the *$pcm* block to TRUE), and specify the solvent parameters, *i.e.*, dielectric constant ϵ (DIELECTRIC) and squared refractive index n^2 (DIELECTRIC_INFI) in the *$solvent* block.

Note: Symmetry will be disabled for all calculations with a PCM.

In the output of a ptSS-PCM calculation with any correlated ADC variant, multiple values are given for the total and transition energies depending in the level of theory. For the correlated ADC variants these include:

- *Zeroth-order results*, direct outcome of the ADC calculation with solvated orbitals, excited states are ordered according to this value.
- *First-order results*, incl. only ptSS nonequilibrium corrections, termed ptSS(PTE).
- *Corrected first-order results*, incl. also the correlation correction, termed ptSS(PTD).
- *Scaled and corrected first-order results*, incl. an empirical scaling of the correlation correction, termed ptSS(PTD*).

We recommend to use the corrected first-order results since the PTD correction generally yields the most accurate results. The ptSS(PTD) approach is typically a very good approximation to the fully consistent, but more expensive PTED scheme, in which the solvent field is made self-consistent with the MP2 density (see next section for the PTED scheme

and sample-jobs for a comparison of the two). Oscillator strengths are computed using the ptSS(PTD) energies. The PTD* approach includes an empirical scaling of the PTD correction that was developed to improve the solvatochromic-shifts of a series of nitro-aromatics with a cc-pVDZ basis.[182] However, we later found that for most other systems, the scaling slightly worsens the agreement with the fully consistent PTED scheme. In addition to the compiled total and transition energies, all contributing terms (ptSS, PTD etc.) are given separately. An even more verbose output detailing all the integrals contributing to the 1st order corrections can be obtained by increasing PCM_PRINT to 1.

Note: The zeroth-order results are by no means identical to the gas-phase excitation energies, and in turn the ptSS-term is not the solvatochromic shift.

(C) Tips and Tricks

To model the absorption spectrum in polar solvents, it is advisable to use a structure optimized in the presence of a PCM since the influence can be quite significant.

It should also be taken into account that a PCM, being a purely electrostatic model, lacks at the description of explicit interactions such as hydrogen bonds. If fairly strong H-bond donor or acceptor moieties are present, or a protic or Lewis-basic solvent is to be modeled, consider adding one or two explicit solvent molecules per donor/acceptor site, and optimize them together with the molecule in the presence of a PCM. A systematic investigation of this aspect for two representative examples can be found in Ref. 182.

If large differences between the HF and MP description of the molecule exist (PTD terms > 0.2 eV), it is advisable to employ the iterative ptSS(PTED) scheme described in the next section. Due to the inverse nature of the systematic errors of ADC(2) and ADC(3), the best guess for the excitation energy in solution is usually the average of both values.

For the PCM, we recommend the formally exact and slightly more expensive integral-equation formalism (IEF)-PCM variant (**Theory** to **IEFPCM** in the *$pcm* block) in place of the approximate C-PCM, and otherwise default parameters.

7.11.10.2 Modeling Emission, Excited-State Absorption and Photochemical Reactivity

(A) Theory

To model emission/absorption of solvent-equilibrated excited states and/or to investigate their photochemical reactivity, both components of the polarization have to be relaxed with respect to the desired state. This becomes evident considering that a full solvent-field equilibration (including the nuclear component) is essentially a geometry optimization for the implicit solvent, and should thus be employed whenever the geometry of the solute is optimized for the desired excited state. The Hamiltonian for a solvent-equilibrated state $|k\rangle$ simply reads

$$E_{\mathrm{k}}^{\mathrm{EqS}} = \langle k|\hat{H}^{\mathrm{vac}} + \hat{R}_{\mathrm{k}}|k\rangle \; . \tag{7.145}$$

Since the interaction with solvent field of any state has to be introduced to the MOs in the SCF step of a calculation, a solvent-field equilibration for excited states (and correlated ground states) is an iterative procedure requiring multiple SCF+ADC calculations until convergence is achieved. This also means that a guess (typically from a ptSS calculation) for the solvent field has to generated and used in the first SCF step. The MOs resulting from this first SCF are subjected to an ADC calculation, providing a first excited-state density, for which a new solvent field is computed and employed in the SCF step of the second iteration. This procedure is repeated until the solvent field (charges) and energies are converged and ultimately provides the total energy and wave function of the solvent-equilibrated excited state $|k\rangle$, as well as the out-of-equilibrium wave functions of other states. However, as the name already suggest, this state-specific approach yields a meaningful energy only for the solvent-equilibrated reference state $|k\rangle$. All other states have to be treated in the nonequilibrium limit and subjected to the formalism presented in the previous section to be consistent with the Franck-Condon principle. The respective generalization of the perturbative ansatz for the Hamiltonian for the ith out-of-equilibrium state in the field of the equilibrated state $|k\rangle$ reads as

$$E_{\mathrm{i}}^{\mathrm{NEq(k)}} = \langle i|\hat{H}^{\mathrm{vac}} + \hat{R}_{\mathrm{k}}^{\mathrm{tot}} + \lambda(\hat{R}_{\mathrm{i}}^{\mathrm{f}} - \hat{R}_{\mathrm{k}}^{\mathrm{f}})|i\rangle \; , \tag{7.146}$$

which can be solved using the procedure introduced in the previous section.

While most of the technical aspects concerning the application of the model will be covered in the following, we highly recommend to read at least the formalism and implementation section of Ref. 183 before using the model.

(B) Usage

The main switch for the state-specific equilibrium solvation (SS-PCM) is the variable EQSOLV in the *$pcm* block. Setting it to TRUE will cause the SCF+ADC calculation to be carried out using the solvent field of the first excited state (if that is found on disk), while any integer > 1 triggers the automatic solvent-field optimization and is interpreted as the maximum number of steps. We recommend to use EQSOLV = 15. Note that to use the SS-PCM, the PCM (SOLVENT_METHOD = PCM in *$rem*) and its nonequilibrium functionalities (NONEQUILIBRIUM = TRUE in *$pcm*) have to be activated as well. Since the solvent-field iterations require an initial guess, a SS-PCM calculation is always the second (or third, . . .) step of a multi-step job.

Note: Any SS-PCM calculation requires a preceding converged ptSS-PCM calculation (*i.e.*, EQSOLV = FALSE) for the desired state to provide a guess for the initial solvent field, or it will crash during the SCF.

To create the input-file for a multi step job, add "@@@" at the end of the input for the first job and append the input for the second job. See also section 3.5. Note that the solvent field used in the subsequent step is stored in the basis of the molecular-surface elements and thus, the geometry of the molecule as well as parameters that affect construction and discretization of the molecular cavity must not be changed between the jobs/steps. This, however, is not enforced.

The state for which the solvent field is to be optimized is specified using the variable EQSTATE in the *$pcm* block. A value of 0 refers to the MP ground-state (for PTED calculations), 1 selects the energetically lowest excited state (default), 2 the second lowest, and so on. The solvent field can be optimized for any singlet, triplet or spin-flip excited state. However, *only the desired class of states should be requested*, *i.e.*, either EE_SINGLETS or EE_TRIPLETS for singlet references, and either EE_STATES or SF_STATES for triplet references. To compute, *e.g.*, the phosphorescence energy, only triplet states should be requested and EQSTATE would typically be set to 1.

Note: Computing multiple classes of excited states during the solvent-field iterations will confuse the state-ordering logic of the program and yield the wrong results.

Convergence is controlled by EQS_CONV. Criteria are the SCF energy as well as the RMSD, MAD and largest single difference of the surface-charges. The convergence should not need to be modified. It is per default derived from the SCF convergence parameter (SCF_CONVERGENCE-4). The default value of 4 (since SCF_CONVERGENCE is 8 for ADC calculations) corresponds to an maximum energy change of 2.72 meV and will yield converged total energies for all states. Excitation energies and in particular the total energy of the reference state converge somewhat faster than the SCF energy, and a value of 3 may save some time for the computation, *e.g.*, for the emission energy of large solutes.

A typical ADC/SS-PCM calculation consists of three steps/subsequent jobs:

1. Generation of an initial guess for the solvent field using a nonequilibrium calculation (EQSTATE = FALSE). To save time, this would typically be done with a smaller basis set and lower convergence criteria (*e.g.*, ADC_DAVIDSON_CONV = 4).

2. Converging the solvent field for the desired state. In this step it is advisable to compute as few states as possible (maybe one more than EQSTATE to be aware of looming state crossings), and more importantly, only the desired class of excited states.

3. Computing all excited states of interest and their properties in the previously converged solvent field. If the reference state of the solvent-field equilibration is a singlet state, this is straightforward and any number of singlet/triplet states can be computed in this final step without further input. *If singlet states are to be computed in the solvent field of a triplet state*, the additional variable EQS_REF in *$pcm* has to be set to tell the program which state is to be treated as the reference state in this last step. For this purpose, EQS_REF is set to the desired

triplet state plus the number of *converged* singlet states. Hence, assuming 2 singlet states are to be calculated along with the triplets in previously converged solvent field of T_1, and furthermore that both singlets converge, EQS_REF needs to be set to 3 (this can not be realized using the variable EQSTATE, because we still want to use the solvent field of the lowest triplet state stored in the previous step).

The self-consistent SS-PCM can also be used to compute a fully consistent solvent field for the MP2 ground state by setting EQSTATE to 0. This is known as ptSS(PTED) approach and can improve vertical excitation energies if there are large differences between the electrostatic properties of the SCF and MP ground states (large PTD corrections). In most cases, however, the non-iterative PTD approach is a very good approximation to the PTED approach (see the sample jobs below). The output of a PTED calculation is essentially identical to that of a ptSS calculation.

The program possesses limited intelligence in detecting the type of the calculation (PTED or EQS/SS-PCM) as well as the target state of the solvent-field equilibration and will assemble and designate the ptSS corrections, total energies and transition energies accordingly. This logic will be confused if multiple classes of states (*e.g.* singlet and triplets) are computed simultaneously during the iterations, and/or if the ordering of the states changes. Nevertheless, in a final job for a previously converged solvent field *of a singlet reference* singlet and triplet states can be computed together yielding the correct output. However, if singlet states are computed in the converged solvent field of *a triplet reference* the additional variable EQS_REF has to be set (and only then, see above).

In the "HF/MP2/MP3 Summary" section, zeroth (without ptSS) and first (with ptSS) order total energies of the respective ground state in the solvent field of the target state are given along with the ptSS term for a vertical transition from the equilibrated state (emission). Note that to obtain the MP3 ground state energy during an ADC(3)/EQS calculation, the ptSS term has to be added manually (it is printed in the MP2 Summary since we use the MP2 density for this purpose).

In the "Excited-State Summary" section the reference state is distinguished from the out-of-equilibrium states. Respective total and transition energies are given along with the nonequilibrium corrections, transition moments and some remarks. Note that for emission, in contrast to absorption, the ptSS term increases the transition energy as it lowers the energy of the out-of-equilibrium ground state. The so-called "self-ptSS term" is a perturbative estimate of how much the solvent field of this state is off from its equilibrium. Although the line in the output changes depending on the value (from "not converged" to "reasonably converged" to "converged") it is not used in the actual convergence check. Note that the self-ptSS term is computed with n^2 (**dielectric_infi**) and not ϵ, as it probably should be. The self-ptSS term may be used to judge how well a solvent field computed with a different methodology (basis, ADC order/variant) fits. In such a case, values < 0.01 eV signal a reasonable agreement.

To calculate inter-state transition moments for excited-state absorption, the variable ADC_PROP_ES2ES has to be set to TRUE. Unfortunately, transition energies have to be computed manually from the (first-order) total energies given in the excited-state summary, since the transition energies given along with the state-to-state transition moments following the excite-state summary are incorrect (missing nonequilibrium terms).

The progress of the solvent-field iteration and their convergence is reported following the "Excited-State Summary".

(C) Tips and Tricks

To compute fluorescence and phosphorescence energies, solute geometry AND solvent field should both be optimized for the suspected emitting state. Since hardly any programs can perform excited-state optimizations with the SS-PCM solvent models, you will probably have to use gas-phase geometries. In our experience, the errors introduced by this approximation are small to negligible (typically < 0.1 eV) in non-polar solvents, but can become significant in polar solvents, in particular for polar charge-transfer states.

Concerning the predicted emission energies, we found that ADC(2)/SS-PCM typically over-stabilizes CT states, yielding emission energies that are too low. SOS-ADC(2) can improve this error, but does not eliminate it. In general, while

emission energies are more accurate with (SOS-)ADC(2)/SS-PCM than with ADC(3)/SS-PCM, the latter affords better relative state energies (see Ref. 183).

Keep in mind that the solute-solvent interaction of polar solvents with polar (*e.g.*, charge-transfer) states can become quite large (multiple eV), and may thus affect the ordering and nature of the excited states. This is quite typical for charge-transfer states even in remotely polar solvents. If they are not the lowest state in the nonequilibrium calculation, but say S_2, it is typically necessary to do one solvent-field iteration for the CT state (EQSOLV = 1 and EQSTATE = 2), which brings the CT down to become S_1, and then continue the iterations with EQSOLV = 15 and EQSTATE = 1. It is in general advisable to carefully check if the character and/or energetic ordering of the states changes during the procedure, in particular for any equilibration of the solvent field for all but the lowest excited state (*e.g.* S_2 or S_3). But even the solvent-field equilibration for a weakly polar S_1 in a polar solvent can cause a more polar state with the same dipole-vector to become the lowest state.

If the excited-states swap during the procedure, find out in which step the swap occurred and do only so many iterations (*i.e.*, set EQSOLV accordingly). In a following job, adjust the variable EQSTATE and continue the iterations. If states start to mix when they get close, it might help to first set an artificially large dielectric constant to induce the change in ordering, and then continue with the desired dielectric constant in a following job.

If performance is critical, the calculations may be accelerated by lowering the ADC convergence during the solvent-field iterations (set ADC_DAVIDSON_CONV = 5). The number of iterations may be reduced by first converging the solvent field with a smaller basis/at a lower level of ADC followed by another job with the full basis/level of ADC. However, in our experience ADC(2) and ADC(3) solvent fields for the same state differ quite significantly and the approach probably does not help much. In contrast, the solvent field computed with a smaller basis (*e.g.* 3-21G, SVP) is often a good approximation for that computed with a larger basis (*e.g.* def2-TZVP, see examples), such this may actually help. It is in general advisable to compute just as many states as is necessary during the solvent-field iterations and include higher lying excited states and triplets in the final job. In ADC(2) calculations for large systems, one should always employ the resolution-of-the-identity approximation.

To save time in PTED jobs, it is suggested to disable the computation of excited states during the solvent-field iterations of a PTED job by setting EE_SINGLETS (and/or EE_TRIPLETS/ EE_STATES) to 0 and compute the excited-states in a final job once the reaction field is converged.

7.11.11 Frozen-Density Embedding: FDE-ADC methods

FDE-ADC[214] is a method to include interactions between an embedded species and its environment into an ADC(n) calculation based on Frozen Density Embedding Theory (FDET).[265,266] FDE-ADC supports ADC and CVS-ADC methods of orders 2s,2x and 3 and regular ADC job control keywords also apply.

The FDE-ADC method starts with generating an embedding potential using a MP(n) density for the embedded system (A) and a DFT or HF density for the environment (B). A Hartree-Fock calculation is then carried out during which the embedding potential is added to the Fock operator. The embedded Hartree-Fock orbitals act as an input for the subsequent ADC calculation which yields the embedded properties (vertical excitation energies, oscillator strengths, etc.). Further information on the FDE-ADC method and FDE-ADC job control are described in Section 11.7.2.1.

7.11.12 ADC Job Control

For an ADC calculation it is important to ensure that there are sufficient resources available for the necessary integral calculations and transformations. These resources are controlled using the *$rem* variables MEM_STATIC and MEM_TOTAL. The memory used by ADC is currently 95% of the difference MEM_TOTAL − MEM_STATIC.

An ADC calculation is requested by setting the *$rem* variable METHOD to the respective ADC variant. Furthermore, the number of excited states to be calculated has to be specified using one of the *$rem* variables EE_STATES, EE_SINGLETS or EE_TRIPLETS. The former variable should be used for open-shell or unrestricted closed-shell calculations, while the latter two variables are intended for restricted closed-shell calculations. Even though not recommended, it is possible to use EE_STATES in a restricted calculation which translates into EE_SINGLETS, if neither EE_SINGLETS nor EE_TRIPLETS is set. Similarly, the use EE_SINGLETS in an unrestricted calculation will translate into EE_STATES, if the latter is not set as well. For IP- and EA-ADC calculations, the IP_STATES, EOM_IP_ALPHA, EOM_IP_BETA, EA_STATES, EOM_EA_ALPHA and EOM_EA_BETA are available to control the number and type of ionized or electron-attached states to calculate. IP_STATES and EA_STATES should be used in case of restricted calculations, while the EOM_[IP/EA]_[ALPHA/BETA] keywords control the number of α- and β-ionized and -electron-attached states to calculate in case of unrestricted or open-shell calculations.

All *$rem* variables to set the number of excited, ionized or electron-attached states accept either an integer number or a vector of integer numbers. A single number specifies that the same number of excited states are calculated for every irreducible representation the point group of the molecular system possesses (molecules without symmetry are treated as C_1 symmetric). In contrast, a vector of numbers determines the number of states for each irreducible representation explicitly. Thus, the length of the vector always has to match the number of irreducible representations. Hereby, the excited states are labeled according to the irreducible representation of the electronic transition which might be different from the irreducible representation of the excited state wave function. Users can choose to calculate any molecule as C_1 symmetric by setting CC_SYMMETRY = FALSE.

METHOD

Controls the order in perturbation theory of ADC.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

ADC(1)	Perform ADC(1) calculation.
ADC(2)	Perform ADC(2)-s, IP-ADC(2)-s or EA-ADC(2)-s calculation.
ADC(2)-x	Perform ADC(2)-x calculation.
ADC(3)	Perform ADC(3), IP-ADC(3) or EA-ADC(3) calculation.
SOS-ADC(2)	Perform spin-opposite scaled ADC(2)-s calculation.
SOS-ADC(2)-x	Perform spin-opposite scaled ADC(2)-x calculation.
CVS-ADC(1)	Perform ADC(1) calculation of core excitations.
CVS-ADC(2)	Perform ADC(2)-s calculation of core excitations.
CVS-ADC(2)-x	Perform ADC(2)-x calculation of core excitations.

RECOMMENDATION:

None

EE_STATES

Controls the number of excited states to calculate.

TYPE:

INTEGER/ARRAY

DEFAULT:

0 Do not perform an ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in case of unrestricted or open-shell calculations. In restricted calculations it can also be used, if neither EE_SINGLETS nor EE_TRIPLETS is given. Then, it has the same effect as setting EE_SINGLETS.

EE_SINGLETS

Controls the number of singlet excited states to calculate.

TYPE:

INTEGER/ARRAY

DEFAULT:

0 Do not perform an ADC calculation of singlet excited states

OPTIONS:

$n > 0$ Number of singlet states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in case of restricted calculations of singlet states. In unrestricted calculations it can also be used, if EE_STATES not set. Then, it has the same effect as setting EE_STATES.

EE_TRIPLETS

Controls the number of triplet excited states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not perform an ADC calculation of triplet excited states

OPTIONS:

$n > 0$ Number of triplet states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in case of restricted calculations of triplet states.

IP_STATES

Controls the number of ionized states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not perform an IP-ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of ionized states in case of restricted calculations.

EOM_IP_ALPHA

Controls the number of α-ionized states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute α-ionized states

OPTIONS:

$n > 0$ Number of α-ionized states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 α-ionized states for the first irrep, n_2 α-ionized states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of α-ionized states in case of unrestricted or open-shell calculations.

EOM_IP_BETA

Controls the number of β-ionized states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute β-ionized states

OPTIONS:

$n > 0$ Number of β-ionized states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 β-ionized states for the first irrep, n_2 β-ionized states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of β-ionized states in case of unrestricted or open-shell calculations.

EA_STATES

Controls the number of electron-attached states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not perform an EA-ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of electron-attached states in case of restricted calculations.

EOM_EA_ALPHA

Controls the number of α-electron-attached states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute α-electron-attached states

OPTIONS:

$n > 0$ Number of α-electron-attached states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 α-electron-attached states for the first irrep, n_2 α-electron-attached states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of α-electron-attached states in case of unrestricted or open-shell calculations.

EOM_EA_BETA

Controls the number of β-electron-attached states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute β-electron-attached states

OPTIONS:

$n > 0$ Number of β-electron-attached states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 β-electron-attached states for the first irrep, n_2 β-electron-attached states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of β-electron-attached states in case of unrestricted or open-shell calculations.

CC_SYMMETRY

Activates point-group symmetry in the ADC calculation.

TYPE:

LOGICAL

DEFAULT:

TRUE If the system possesses any point-group symmetry.

OPTIONS:

TRUE Employ point-group symmetry

FALSE Do not use point-group symmetry

RECOMMENDATION:

None

ADC_DENSITY_ORDER

Controls the order of the ground state density used for the computation of third-order ADC matrix elements (non-CVS methods only).

TYPE:

INTEGER

DEFAULT:

2 Use strict third-order ADC(3) schemes.

OPTIONS:

3 Use a third-order ground state density computed from the IP-ADC(3) effective transition moments and the corresponding fourth order static self-energy according to the $\Sigma(4)$ scheme

4 Use an improved third-order ground state density and the corresponding improved fourth-order static self-energy computed according to the self-consistent $\Sigma(4+)$ procedure

RECOMMENDATION:

In case of IP-ADC(3) calculations, employing the $\Sigma(4+)$ scheme provides more accurate ionization potentials and ionized state dipole moments.

ADC_DENSITY_MAXITER

When setting ADC_DENSITY_ORDER = 4, this keyword controls the maximum number of DIIS iterations carried out in the $\Sigma(4+)$ procedure.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Use the default value.

ADC_DIRECT

For third-order ADC methods, this keyword controls if some large intermediate tensor contractions should be carried out in advance and the result saved in memory for later use or if these quantities should be evaluated directly whenever they are encountered.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Directly evaluate some N^6-scaling tensor contractions. This will reduce the memory requirement by $\sim$10 %.

FALSE Precompute all possible N^6-scaling intermediates. This will speed up ADC(3) calculations considerably (by a factor of $\sim$3 in case of ADC(3) for N-electron excitations and somewhat less for IP- and EA-ADC(3)).

RECOMMENDATION:

Use the default value unless memory is the bottleneck.

ADC_STRICT_ISR

Controls how second-order ground state contributions are treated in the calculation of second- and third-order IP- and EA-ADC state properties using the second-order ISR formalism.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Scale the second-order part of the ground state contribution to one-electron properties of ionized/electron-attached states by the one-hole/one-particle character of the respective states as implied by the strict ISR derivation.

FALSE Use the full second-order ground state contribution for each ionized/electron-attached state property.

RECOMMENDATION:

Use the default value. Both options are, however, valid second-order treatments of ionized/electron-attached state properties and should yield very similar results for states with predominant one-hole/one-particle chaaracter.

ADC_DO_DYSON

Controls if Dyson orbitals are output in case of IP- and EA-ADC calculations. This keyword only takes effect when used together with STATE_ANALYSIS = TRUE. See Section 10.2.12 for further details.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Output Dyson orbitals as cube files.

FALSE Do not output Dyson orbitals.

RECOMMENDATION:

Set to TRUE if visualization of ionization/electron-attachment processes is desired.

ADC_CAP

Controls the type of CAP/ADC calculation to be performed.

TYPE:

INTEGER

DEFAULT:

0 Do not perform a CAP/ADC calculation.

OPTIONS:

1 Perform a subspace-projected CAP/ADC calculation.

RECOMMENDATION:

Set to 1 for the computation of CAP/ADC subspace projections.

CAP_X

For ADC methods, in combination with a smoothed Voronoi-CAP (CAP_TYPE = 2) or a spherical CAP (CAP_TYPE = 0), this keyword controls the lower limit for a series of CAP onsets, where the upper limit is given by CAP_X_END. The parameter value in a.u. is obtained by multiplying the given integer by 10^{-3}. In this case, the onset value defines the region around the molecule with zero CAP strength. In combination with a cuboid CAP (CAP_TYPE = 1) or in general for other electronic structure methods (see 7.10.9 for further details), this keyword controls the CAP onset in x direction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n > 0$ User-defined integer.

RECOMMENDATION:

Usually, values of 2000 to 4000 (corresponding to onset values between 2.0 and 4.0 a.u.) give reasonable results.

CAP_X_STEP

Controls the step size for a series of CAP onsets between CAP_X and CAP_X_END. The parameter value in a.u. is obtained by multiplying the given integer by 10^{-3}. Currently only used in ADC methods.

TYPE:

INTEGER

DEFAULT:

500 corresponding to 0.5 a.u.

OPTIONS:

$n > 0$ User-defined integer.

RECOMMENDATION:

None.

CAP_X_END

Controls the upper onset limit for a series of CAP onsets, where the lower limit is given by CAP_X. The parameter value in a.u. is obtained by multiplying the given integer by 10^{-3}. Currently only used in ADC methods.

TYPE:

INTEGER

DEFAULT:

CAP_X Do not compute a series of CAP onsets but only use a single CAP with an onset value of CAP_X.

OPTIONS:

$n >$ CAP_X User-defined integer.

RECOMMENDATION:

Use this keyword if CAP onset series are desired.

ADC_PROP_ES

Controls the calculation of excited, ionized or electron-attached state properties (currently only dipole moments and $\hat{r}^2$ expectation values).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate excited, ionized or electron-attached state properties.

FALSE Do not compute state properties.

RECOMMENDATION:

Set to TRUE, if properties are required.

ADC_PROP_ES2ES

Controls the calculation of transition properties between excited, ionized or electron-attached states (currently only transition dipole moments and oscillator strengths). For ADC for N-electron excitations, this keyword also controls the computation of two-photon absorption cross-sections of excited states using the sum-over-states expression.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate state-to-state transition properties.

FALSE Do not compute transition properties between excited, ionized or electron-attached states.

RECOMMENDATION:

Set to TRUE, if state-to-state properties (ADC, IP-ADC, EA-ADC) or sum-over-states two-photon absorption cross-sections (only ADC) are required.

ADC_PROP_TPA

Controls the calculation of two-photon absorption cross-sections of excited states using matrix inversion techniques.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate two-photon absorption cross-sections.

FALSE Do not compute two-photon absorption cross-sections.

RECOMMENDATION:

Set to TRUE, if to obtain two-photon absorption cross-sections.

STATE_ANALYSIS

Controls the analysis and export of excited, ionized or electron-attached state densities and orbitals (see 10.2.12 for details).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform excited state analyses.

FALSE No excited state analyses or export will be performed.

RECOMMENDATION:

Set to TRUE, if detailed analysis of the excited, ionized or electron-attached states is required or if density or orbital plots are needed.

ADC_C_T

Set the spin-opposite scaling parameter c_T for an SOS-ADC(2) calculation. The parameter value is obtained by multiplying the given integer by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

1300 Optimized value $c_T = 1.3$.

OPTIONS:

n Corresponding to $n \cdot 10^{-3}$

RECOMMENDATION:

Use the default.

ADC_C_C

Set the spin-opposite scaling parameter c_c for the ADC(2) calculation. The parameter value is obtained by multiplying the given integer by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

1170 Optimized value $c_c = 1.17$ for ADC(2)-s or

1000 $c_c = 1.0$ for ADC(2)-x

OPTIONS:

n Corresponding to $n \cdot 10^{-3}$

RECOMMENDATION:

Use the default.

ADC_C_X

Set the spin-opposite scaling parameter c_x for the ADC(2)-x calculation. The parameter value is obtained by multiplying the given integer by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

1300 Optimized value $c_x = 0.9$ for ADC(2)-x.

OPTIONS:

n Corresponding to $n \cdot 10^{-3}$

RECOMMENDATION:

Use the default.

ADC_NGUESS_SINGLES

Controls the number of excited state guess vectors which are single excitations, one-hole ionizations and one-particle electron-attachments in case of ADC, IP-ADC and EA-ADC, respectively. If the number of requested excited states exceeds the total number of guess vectors (singles and doubles), this parameter is automatically adjusted, so that the number of guess vectors matches the number of requested excited states.

TYPE:

INTEGER

DEFAULT:

Equals to the number of excited states requested.

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Increase if there are convergence problems.

ADC_NGUESS_DOUBLES

Controls the number of excited state guess vectors which are double excitations, two-hole-one-particle ionizations and one-hole-two-particle electron-attachments in case of ADC, IP-ADC and EA-ADC, respectively.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined integer.

RECOMMENDATION:

ADC_DO_DIIS

Activates the use of the DIIS algorithm for the calculation of ADC(2) excited states.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use DIIS algorithm.

FALSE Do diagonalization using Davidson algorithm.

RECOMMENDATION:

None.

ADC_DIIS_START

Controls the iteration step at which DIIS is turned on.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Set to a large number to switch off DIIS steps.

ADC_DIIS_SIZE

Controls the size of the DIIS subspace.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n User-defined integer

RECOMMENDATION:

None

ADC_DIIS_MAXITER

Controls the maximum number of DIIS iterations.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Increase in case of slow convergence.

ADC_DIIS_ECONV

Controls the convergence criterion for the excited state energy during DIIS.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

None

ADC_DIIS_RCONV

Convergence criterion for the residual vector norm of the excited state during DIIS.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

None

ADC_DAVIDSON_MAXSUBSPACE

Controls the maximum subspace size for the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

$5\times$ the number of excited states to be calculated.

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Should be at least 2–4$\times$ the number of excited states to calculate. The larger the value the more disk space is required.

ADC_DAVIDSON_MAXITER

Controls the maximum number of iterations of the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

60

OPTIONS:

n Number of iterations

RECOMMENDATION:

Use the default unless convergence problems are encountered.

ADC_DAVIDSON_CONV

Controls the convergence criterion of the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

$n \leq 12$ Corresponding to 10^{-n}.

RECOMMENDATION:

Use the default unless higher accuracy is required or convergence problems are encountered.

ADC_DAVIDSON_THRESH

Controls the threshold for the norm of expansion vectors to be added during the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

Twice the value of ADC_DAVIDSON_CONV, but at maximum 10^{-14}.

OPTIONS:

$n \leq 14$ Corresponding to 10^{-n}

RECOMMENDATION:

Use the default unless convergence problems are encountered. The threshold value 10^{-n} should always be smaller than the convergence criterion ADC_DAVIDSON_CONV.

ADC_PRINT

Controls the amount of printing during an ADC calculation.

TYPE:

INTEGER

DEFAULT:

1 Basic status information and results are printed.

OPTIONS:

0 Quiet: almost only results are printed.

1 Normal: basic status information and results are printed.

2 Debug: more status information, extended information on timings.

...

RECOMMENDATION:

Use the default.

ADC_CVS

Activates the use of the CVS approximation for the calculation of CVS-ADC core-excited states.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Activates the CVS approximation.

FALSE Do not compute core-excited states using the CVS approximation.

RECOMMENDATION:

Set to TRUE, if to obtain core-excited states for the simulation of X-ray absorption spectra. In the case of TRUE, the *$rem* variable CC_REST_OCC has to be defined as well.

CC_REST_OCC

Sets the number of restricted occupied orbitals including active core occupied orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Restrict n energetically lowest occupied orbitals to correspond to the active core space.

RECOMMENDATION:

Example: cytosine with the molecular formula $C_4H_5N_3O$ includes one oxygen atom. To calculate O 1s core-excited states, n has to be set to 1, because the 1s orbital of oxygen is the energetically lowest. To obtain the N 1s core excitations, the integer n has to be set to 4, because the 1s orbital of the oxygen atom is included as well, since it is energetically below the three 1s orbitals of the nitrogen atoms. Accordingly, to simulate the C 1s spectrum of cytosine, n must be set to 8.

SF_STATES

Controls the number of excited spin-flip states to calculate.

TYPE:

INTEGER

DEFAULT:

0 Do not perform a SF-ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in the case of a spin-flip calculation. SF-ADC is available for ADC(2)-s, ADC(2)-x and ADC(3).

Keywords for SS-PCM control in *$pcm*:

EQSOLV

Main switch of the self-consistent SS-PCM procedure.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No self-consistent SS-PCM.

1 Single SS-PCM calculation (SCF+ADC) with the solvent field found on disk.

$n >1$ Do a maximum of n automatic solvent-field iterations.

RECOMMENDATION:

We recommend to use 15 steps max. Typical convergence is 3-5 steps. In difficult cases 6-12. If the solvent-field iteration do not converge in 15 steps, something is wrong. *Make sure that a solvent field has been stored on disk by a previous job.*

EQSTATE

Specifies the state for which the solvent field is to be optimized.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 MP2 ground state (for PTED approach)

1 energetically lowest excited state

2 2nd lowest excited state

...

RECOMMENDATION:

Given that only one class of excited states is calculated, the state will be selected according to its energetic position shown in the "Exited-State Summary" of the output file. A maximum of 99 states is stored and can be selected.

EQS_CONV

Controls the convergence of the solvent-field iterations by setting the convergence criteria (a mixture of SCF energy and charge-vector). SCF energy criterion computes as $10^{-\mathrm{value}}\ E_h$.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

SCF_CONVERGENCE$-4 = 4$

OPTIONS:

3 May be sufficient for emission energies

4 Assured converged total energies (2.7 meV)

5 Really tight

RECOMMENDATION:

Use the default.

EQS_REF

Allows to specify which state is to be treated as the reference state in the ADC part of the calculation. Does in contrast to EQSTATE not affect which solvent field is loaded in the SCF step. Only has to be used when singlets are computed in the solvent field of a triplet reference. Note that (converged) singlets states are always counted before triplets, and thus to select T_1 in a calculation with EE_SINGLETS = 2 this has to be set to 3.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

Same as EQSTATE

OPTIONS:

1 First excited state

2 Second excited state

...

RECOMMENDATION:

Only needed when computing singlet states in the solvent field of a triplet reference.

7.11.13 Examples

Example 7.167 Input for an ADC(2)-s calculation of singlet exited states of methane with D2 symmetry. In total six excited states are requested corresponding to four (two) electronic transitions with irreducible representation B_1 (B_2).

```
$molecule
   0 1
   C
   H   1 r0
   H   1 r0   2  d0
   H   1 r0   2  d0   3  d1
   H   1 r0   2  d0   4  d1

   r0 = 1.085
   d0 = 109.4712206
   d1 = 120.0
$end

$rem
   METHOD              adc(2)
   BASIS               6-31g(d,p)
   MEM_TOTAL           4000
   MEM_STATIC          100
   EE_SINGLETS         [0,4,2,0]
$end
```

Example 7.168 Input for an unrestricted RI-ADC(2)-s calculation with C_1 symmetry using DIIS. In addition, excited state properties and state-to-state properties are computed.

```
$molecule
   0 2
   C    0.0    0.0   -0.630969
   N    0.0    0.0    0.540831
$end

$rem
   METHOD           adc(2)
   BASIS            aug-cc-pVDZ
   AUX_BASIS        rimp2-aug-cc-pVDZ
   MEM_TOTAL        4000
   MEM_STATIC       100
   CC_SYMMETRY      false
   EE_STATES        6
   ADC_DO_DIIS      true
   ADC_PROP_ES      true
   ADC_PROP_ES2ES   true
   ADC_PROP_TPA     true
$end
```

Example 7.169 Input for a restricted CVS-ADC(2)-x calculation with C_1 symmetry using 4 parallel CPU cores.

```
$molecule
   0 1
   N     1.0706214490   -0.1462996030    0.0000000000
   C    -0.1838756809    0.3832287690    0.0000000000
   O    -1.2178351723   -0.2734201303    0.0000000000
   H     1.8945772136    0.4351761203    0.0000000000
   H     1.1761147729   -1.1515954431    0.0000000000
   H    -0.1740335498    1.4879608698    0.0000000000
$end

$rem
   METHOD                      cvs-adc(2)-x
   BASIS                       6-31G*
   EE_SINGLETS                 5
   ADC_DAVIDSON_MAXSUBSPACE    60
   MEM_TOTAL                   10000
   MEM_STATIC                  1000
   THREADS                     4
   CC_SYMMETRY                 false
   ADC_DAVIDSON_THRESH         8
   ADC_DAVIDSON_MAXITER        900
   ADC_CVS                     true
   CC_REST_OCC                 4
   INTEGRAL_SYMMETRY           false
$end
```

Example 7.170 Input for a restricted SF-ADC(2)-s calculation of the first three spin-flip target states of cyclobutadiene without point group symmetry.

```
$molecule
   0 3
   C     0.000000     0.000000     0.000000
   C     1.439000     0.000000     0.000000
   C     1.439000     0.000000     1.439000
   C     0.000000     0.000000     1.439000
   H    -0.758726     0.000000    -0.758726
   H     2.197726     0.000000    -0.758726
   H     2.197726     0.000000     2.197726
   H    -0.758726     0.000000     2.197726
$end

$rem
   METHOD          adc(2)
   MEM_TOTAL       15000
   MEM_STATIC      1000
   CC_SYMMETRY     false
   BASIS           3-21G
   SF_STATES       3
$end
```

The following example demonstrates a CAP/EA-ADC(3) calculation of the $^2\Pi_g$ resonance of the dinitrogen anion, employing the $\Sigma(4+)$ static self-energy. In this calculation, a series of smoothed Voronoi CAPs with onsets between 2.0 a.u. and 5.0 a.u. is projected onto the subspace of the lowest five converged EA-ADC(3) vectors of B_{2g} symmetry (the largest Abelian subgroup of $D_{\infty h}$, which is D_{2h}, is used in the calculation). The basis set employed is aug-cc-pVQZ on nitrogen and additional $3s3p3d$ diffuse functions on the molecular center.

Example 7.171 Input for a standard CAP/EA-ADC(3) calculation of the $^2\Pi_g$ resonance of the dinitrogen anion.

```
$molecule
   0 1
   N  0.0 0.0  0.548756750
   N  0.0 0.0 -0.548756750
   Gh 0.0 0.0  0.0
$end

$rem
   METHOD               adc(3)
   ADC_DENSITY_ORDER    4
   EA_STATES            [0,0,5,0,0,0,0,0]
   BASIS                gen
   ADC_CAP              1
   CAP_TYPE             2
   CAP_X                2000
   CAP_X_END            5000
   CAP_X_STEP           500
   MEM_TOTAL            100000
   ADC_DAVIDSON_CONV    5
   ADC_DAVIDSON_THRESH  14
   ADC_DAVIDSON_MAXITER 100
   ADC_NGUESS_SINGLES   20
   THREADS              12
   FORCE_SYMMETRY_ON    TRUE
$end

$basis
N     0
aug-cc-pVQZ
****
Gh    0
S   1   1.00
      0.0273200             1.0000000
S   1   1.00
      0.0136600             1.0000000
S   1   1.00
      0.0068300             1.0000000
P   1   1.00
      0.0220100             1.0000000
P   1   1.00
      0.0110050             1.0000000
P   1   1.00
      0.0055025             1.0000000
D   1   1.00
      0.0555000             1.0000000
D   1   1.00
      0.0277500             1.0000000
D   1   1.00
      0.0138750             1.0000000
****
$end
```

The next example provides input for a restricted ADC(2)-x calculation of water with C_s symmetry. Four singlet A''

excited states and two triplet A' excited states are requested. For the first two states ($1\,^1A''$ and $1\,^3A'$) the transition densities as well as the attachment and detachment densities are exported into cube files.

Example 7.172 Restricted ADC(2)-x calculation of water with C_s symmetry.

```
$molecule
   0 1
   O   0.000   0.000   0.000
   H   0.000   0.000   0.950
   H   0.896   0.000  -0.317
$end

$rem
   METHOD            adc(2)-x
   BASIS             6-31g(d,p)
   THREADS           2
   MEM_TOTAL         3000
   MEM_STATIC        100
   EE_SINGLETS       [0,4]
   EE_TRIPLETS       [2,0]
   ADC_PROP_ES       true
   MAKE_CUBE_FILES   true
$end

$plots
Plot transition and a/d densities
   40 -3.0 3.0
   40 -3.0 3.0
   40 -3.0 3.0
   0 0 2 2
   1 2
   1 2
$end
```

The next sample provides input for a ADC(2)-s/ptSS-PCM calculation of the five lowest singlet-excited states of N,N-dimethylnitroaniline in diethyl ether. The PCM settings are all default values except **Theory**, which is set to **IEFPCM**

instead of the default **CPCM**.

Example 7.173 DC(2)-s/ptSS-PCM calculation of N, N-dimethylnitroaniline in diethyl ether.

```
$molecule
   0 1
   C   -4.263068      2.512843      0.025391
   C   -5.030982      1.361365      0.007383
   C   -4.428196      0.076338     -0.021323
   C   -3.009941      0.019036     -0.030206
   C   -2.243560      1.171441     -0.011984
   C   -2.871710      2.416638      0.015684
   H   -4.740854      3.480454      0.047090
   H   -2.502361     -0.932570     -0.052168
   H   -1.166655      1.104642     -0.020011
   H   -6.104933      1.461766      0.015870
   N   -5.178541     -1.053870     -0.039597
   C   -6.632186     -0.969550     -0.034925
   H   -6.998203     -0.462970      0.860349
   H   -7.038179     -1.975370     -0.051945
   H   -7.001616     -0.431420     -0.910237
   C   -4.531894     -2.358860     -0.066222
   H   -3.912683     -2.476270     -0.957890
   H   -5.298508     -3.126680     -0.075403
   H   -3.902757     -2.507480      0.813678
   N   -2.070815      3.621238      0.033076
   O   -0.842606      3.510489      0.025476
   O   -2.648404      4.710370      0.054545
$end

$rem
   THREADS                4
   METHOD                 adc(2)
   BASIS                  3-21G
   MEM_TOTAL              32000
   MEM_STATIC             2000
   ADC_PROP_ES            true
   ADC_PRINT              1
   EE_SINGLETS            5
   ADC_DAVIDSON_MAXITER   100
   PCM_PRINT              1       !increase print level
   SOLVENT_METHOD         pcm     !invokes PCM solvent model
$end

$pcm
   nonequilibrium         true
   theory                IEFPCM  !default is CPCM, IEFPCM is more accurate
   Solver                Inversion
   vdwScale              1.20
$end

$solvent
   dielectric            4.34  !epsilon of Et2O
   dielectric_infi       1.829 !n_square of Et2O
$end
```

The next job requires a rather complicated compound input file. The sample job computes ADC/SS-PCM EqS solvent-field equilibration for the first excited singlet state of peroxinitrite in water, which can be used to compute the fluorescence energy. After generating a starting point in the first job (using a smaller basis and lower ADC convergence criteria), the solvent-field iterations are carried out until convergence in the second job. In the third job, ADC(2) excited states are computed in the converged solvent field that was left on disk by the second Job. In the fourth job, we

additionally compute ADC(3) excited states. This mixed approach should in general be used with great caution. If the self-ptSS term of the reference state becomes too large (>0.01 eV) like it is the case here, the fully consistent approach should be used, meaning that the solvent reaction field should also be computed at the ADC(3) level. PCM settings are all default values except **Theory**, which is set to IEFPCM instead of the default CPCM.

Example 7.7.174 ADC/SS-PCM EQS solvent-field equilibration for the first excited singlet state of peroxinitrite in water.

View input online

The next sample job provides the input for a RI-ADC(2)/ptSS-PCM(PTED) calculation for the five lowest excited states of peroxinitrite in water. After generating a starting point in the first job, which also provides the ptSS(PTE) and ptSS(PTD) results for comparison, the solvent-field is equilibrated for the MP density in the second job. During the iterations, the calculation of excited states is disabled to speed up the calculation. In the third job, five excited states are computed at the RI-ADC(2)/ptSS(PTED) level of theory. Although the PTD corrections for this molecule are unusually large, a comparison of the PTE, PTD and PTD* results from the first job with the PTED results from the third job will reveal a reasonable agreement between the fully consistent PTED and the perturbative PTD approaches. In the fourth job, excited states are calculated with a larger basis set. The self-ptSS term of the MP ground state will be quite small, showing that the solvent-field computed with the smaller SVP basis is a good approximation.

Example 7.7.175 RI-ADC(2)/ptSS-PCM(PTED) calculation for the five lowest excited states of peroxinitrite in water.

View input online

7.12 Restricted Active Space Spin-Flip (RAS-SF) and Configuration Interaction (RAS-CI)

7.12.1 Introduction

The restricted active space spin-flip (RAS-SF) is a special form of configuration interaction that is capable of describing the ground and low-lying excited states with moderate computational cost in a single-reference formulation,[15,36,40,281] including strongly correlated systems. The RAS-SF approach is essentially a much lower computational cost alternative to Complete Active Space SCF (CASSCF) methods. RAS-SF typically works by performing a full CI calculation within an active space that is defined by the half-occupied orbitals of a restricted open shell HF (ROHF) reference determinant. In this way the difficulties of state-specific orbital optimization in CASSCF are bypassed. Single excitations into (hole) and out of (particle) the active space provide state-specific relaxation instead. Unlike most CI-based methods, RAS-SF is size-consistent, as well as variational, and, the increase in computational cost with system size is modest for a fixed number of spin flips. Beware, however, for the increase in cost as a function of the number of spin-flips is exponential! RAS-SF has been shown to be capable of tackling multiple low-lying electronic states in polyradicals and reliably predicting ground state multiplicities.[14,15,35,40,280,281]

RAS-SF can also be viewed as one particular case of a more general RAS-CI family of methods. For instance, instead of defining the active space via spin-flipping as above, initial orbitals of other types can be read in, and electronic excitations calculated this way may be viewed as a RAS-EE-CI method (though size-consistency will generally be lost). Similar to EOM-CC approaches (see Section 7.10), other target RAS-CI wave functions can be constructed starting from any electronic configuration as the reference and using a general excitation-type operator. For instance, one can construct an ionizing variant that removes an arbitrary number of particles that is RAS-nIP-CI. An electron-attaching variant is RAS-nEA-CI.[36]

Q-CHEM features two versions of RAS-CI code with different, complementary, functionality. One code (invoked by specifying CORRRELATION = RASCI) has been written by David Casanova;[36] below we will refer to this code as RASCI1. The second implementation (invoked by specifying CORRRELATION = RASCI2) is primarily due to Paul Zimmerman;[281] we will refer to it as RASCI2 below. Both codes can be used to computed several state specific and interstate properties, such as spin-orbit couplings (as described in Section 7.10.20.4).[31]

The RASCI1 code uses an integral-driven implementation (exact integrals) and spin-adaptation of the CI configurations which results in a smaller diagonalization dimension. The current Q-CHEM implementation of RASCI1 allows for the calculation of systems with any number of α and β number of electrons, with the multiplicity ($\langle \hat{S}^2 \rangle$) of each state being printed alongside the state energy. Shared memory parallel execution decreases compute time as all the underlying integrals routines are parallelized.

The RASCI2 code includes the ability to simulate closed and open shell systems (*i.e.*, even and odd numbers of electrons), fast integral evaluation using the resolution of the identity (RI) approximation, shared memory parallel operation, and analysis of the $\langle \hat{S}^2 \rangle$ values and natural orbitals. The natural orbitals are stored in the `$QCSCRATCH` directory in a folder called "`NOs`" in MOLDEN-readable format. Shared memory parallel is invoked as described in Section 2.2.1.1. A RASCI2 input requires the specification of an auxiliary basis set analogous to RI-MP2 computations (see Section 6.6.2). Otherwise, the active space as well as hole and particle excitations are specified in the same way as in RASCI1.

Note: Because RASCI2 uses the RI approximation, the total energies computed with the two codes will be slightly different. Energy differences between different states should closely match each other, however.

7.12.2 The Restricted Active Space (RAS) Scheme

In the RAS formalism, one divides the orbital space into three subspaces called RAS1, RAS2 and RAS3 (Fig. 7.2). The RAS-CI states are defined by the number of orbitals and the restrictions in each subspace.

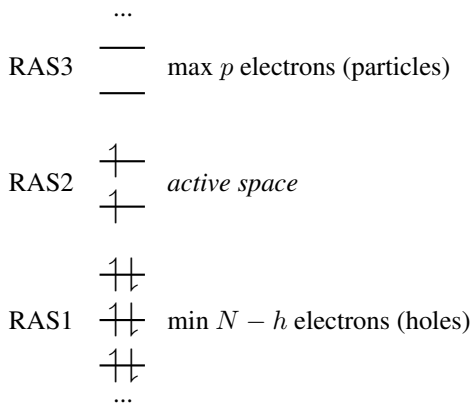


Figure 7.2: Orbital subspaces in RAS-CI employing a ROHF triplet reference.

The single reference RAS-CI electronic wave functions are obtained by applying a spin-flipping or excitation operator $\hat{R}$ on the reference determinant $|\phi^{(0)}\rangle$.

$$\left|\Psi^{\text{RAS}}\right\rangle = \hat{R}\left|\phi^{(0)}\right\rangle \tag{7.147}$$

The $\hat{R}$ operator must obey the restrictions imposed in the subspaces RAS1, RAS2 and RAS3, and can be decomposed as:

$$\hat{R} = \hat{r}^{\text{RAS2}} + \hat{r}^{h} + \hat{r}^{p} + \hat{r}^{hp} + \hat{r}^{2h} + \hat{r}^{2p} + \ldots \tag{7.148}$$

where $\hat{r}^{\text{RAS2}}$ contains all possible electronic promotions within the RAS2 space, that is, a reduced full CI, and the rest of the terms generate configurations with different number of holes (h super-index) in RAS1 and electrons in RAS3 (p super-index). The current implementation truncates this series up to the inclusion of hole and particle contributions, *i.e.*, the first three terms on the right hand side of Eq. (7.148).

7.12.3 Second-Order Perturbative Corrections to RAS-CI

In general, the RAS-CI family of methods within the $hole$ and $particle$ approximation is unable to capture the necessary amounts of dynamic correlation for the computation of relative energies with chemical accuracy. The missed correlation can be added on top of the RAS-CI wave function using multi-reference perturbation theory (MRPT).[37] The second order energy correction, *i.e.* RASCI(2), can be expressed as:

$$E^{(2)} = -\sum_{k} \frac{|\langle k|\hat{H}|0\rangle|^2}{E_k - E_0 + \epsilon} \tag{7.149}$$

where 0 indicates the zero-order space and $\{|k\rangle\}$ is the complementary set of determinants. There is no natural choice for the $\{E_k\}$ excited energies in MRPT, and two different models are available within the RASCI(2) approach, namely, either Davidson-Kapuy or Epstein-Nesbet partitioning. As it is common practice in many second-order MRPT corrections, the denominator energy differences in Eq. (7.149) can be level shifted with a parameter ϵ.

7.12.4 Short-Range Density Functional Correlation within RAS-CI

Alternatively, effective dynamic correlation can be introduced into the RAS-CI wave function by means of short-range density functional correlation energy. The idea relies on the different ability of wave function methods and DFT to treat non-dynamic and dynamic correlations. Concretely, the RAS-CI-srDFT (or RAS-srDFT) method[38] is based on the range separation of the electron-electron Coulomb operator ($\hat{V}_{ee}$) through the error function to describe long-range interactions,

$$\hat{V}_{ee}^{lr,\mu} = \sum_{i<j} \frac{\text{erf}(\mu r_{ij})}{r_{ij}} \tag{7.150}$$

$$\hat{V}_{ee}^{sr,\mu} = \hat{V}_{ee} - \hat{V}_{ee}^{lr,\mu} \tag{7.151}$$

where r_{ij} is the inter electronic distance and the parameter μ controls the extend of short- and long-range interactions. Such splitting of $\hat{V}_{ee}$ provides a well-defined approach to merge WFT with DFT by applying $\hat{V}_{ee}^{lr,\mu}$ to RAS-CI and $V_{ee}^{lr,\mu}$ to DFT. Within the RAS-srDFT approach, the energy of an electronic state can be expressed as:

$$E^{\text{RAS-}sr\text{DFT}} = \min_{\Psi^\mu} \left[\langle \Psi^\mu|\hat{T} + \hat{V}_{ne} + \hat{V}_{ee}^{lr,\mu}|\Psi^\mu\rangle + E_H^{sr,\mu}[\rho] + E_{xc}^{sr,\mu}[\rho]\right] \tag{7.152}$$

where $\rho \equiv \rho[\Psi^\mu]$, and $E_H^{sr,\mu}[\rho]$ and $E_{xc}^{sr,\mu}[\rho]$ are the short-range Hartree and exchange-correlation energy functionals, respectively. The RAS-CI wave function can be combined with different short-range exchange and correlation functionals (Sections 5.3.3 and 5.3.4).

7.12.5 Excitonic Analysis of the RAS-CI Wave Function

The RAS-CI wave function of multi-chromophoric systems can be decomposed in terms of local excitations (LE), charge resonances (CR) and multi-exciton configurations (ME).

$$|\Psi\rangle = |\Psi^{\text{LE}}\rangle + |\Psi^{\text{CR}}\rangle + |\Psi^{\text{ME}}\rangle \tag{7.153}$$

The wave function decomposition scheme combines the use of fragment (localized) orbitals with the computation of charge and spin cumulant indexes.[41,159] Such analysis is extremely useful in order to characterize in great detail the nature of the ground and excited states in the presence of two or more molecules or chromophoric moieties. The ME contributions can be further decomposed into different spin coupled configurations localized on each. Typically, the ME part of a singlet state in a molecular dimer can be decomposed as singlet-singlet (SS) and triplet-tripled (TT) contributions.

$$|\Psi^{\mathrm{ME}}\rangle \approx |\Psi^{\mathrm{SS}}\rangle + |\Psi^{\mathrm{TT}}\rangle \tag{7.154}$$

At the moment, this analysis is only available for the decomposition of spin singlet states.

7.12.6 Diabatization of RAS-CI Eigenstates

Characterization of RAS-CI electronic states can be alternatively performed by diabatization of multiple eigenstates of the RAS-CI Hamiltonian. Diabatic states can be obtained following a Boys,[240] Edmiston-Ruedenberg[241] or dipole-quadrupole (DQ) localization schemes.[116] See Section 10.14.1.3 for details. To activate the diabatization analysis of excited states in RAS-CI method it is necessary to define a section block (see *$localized_diabatization* keyword) that defines the list of excited states to be included in the diabatization (by increasing order). This is equivalent to the one used for the diabatization of CIS/TDDFT states.

7.12.7 Spin-flip CAS with Perturbative External Singles Corrections (CASMAN)

The CASMAN module allows one to run spin-flip CAS-based calculations, with perturbative corrections accounting for hole and particle excitations outside of the active space. (For variational hole and particle excitations, please refer to the section on LIBRASSF.) It also allows one to run with a limited number of spin-flips, which can be useful for large transition metal complexes in which the maximum number of spin-flips cannot be performed. For more details on this approach, please refer to the publication by Mayhall and Head-Gordon.[179]

Q-CHEM has several built-in excitation schemes. To use these, simply set CORRELATION to the appropriate value. The following methods are currently available: SF-CAS, SF-CAS(h,p), SF-CAS(S), SF-CAS(2x), SF-CAS(3x), and SF-CAS(3x)+D.

Alternatively, if a custom method is desired, one can set CORRELATION = SF-CAS and manually define the appropriate *$rem* variables as needed. Please see the section on CASMAN job control options for further details.

7.12.8 Direct RAS-nSF-IP/EA (LIBRASSF)

The LIBRASSF module enables the use of Fock-space CI RAS-nSF-IP/EA calculations. The CAS-nSF-IP/EA approach combines non-particle-number-conserving and non-spin-conserving operators to model systems with spin and spatial degeneracies. For example, the CAS-1SF-IP operator is defined as follows:

$$|\Psi\rangle = \sum_{ij\bar{a}} c_{ij}^{\bar{a}} \hat{a}_{ij}^{\bar{a}} |\Psi^{\mathrm{Ref}}\rangle \tag{7.155}$$

The LIBRASSF module can also perform traditional spin-flip and IP/EA calculations. External hole and particle excitations (RAS(h) and RAS(p), respectively) can also be added; this orbital relaxation is particularly important for IP/EA calculations due to the change in electron count.

Our implementation uses explicit expressions for unique changes in alpha and beta electron counts. These methods have been efficiently implemented using direct tensor contractions rather than alpha and beta strings. The code also

allows the user to select the appropriate number of spin-flips, limiting the number of determinants considered in cases with large active spaces.

Currently, the following schemes are implemented: CAS-1SF, RAS(h)-1SF, RAS(p)-1SF, RAS(h,p)-1SF, CAS-2SF, CAS-IP, RAS(h)-IP, CAS-EA, RAS(p)-EA, CAS-1SF-IP, RAS(h)-1SF-IP, CAS-1SF-EA, and RAS(p)-1SF-EA.

7.12.9 LIBRASSF Effective Hamiltonian Analysis

The Bloch effective Hamiltonian can be used in conjunction with a RAS-1SF calculation to determine the magnetic couplings of multi-site systems. The LIBRASSF module provides a variational version of the earlier perturbative approaches in the CASMAN code.[180] Note that this implementation is still in beta.

Doing the analysis within LIBRASSF is a two-step process. First, one must localize the active space orbitals using a Boys localization, indicated by setting CAS_LOCAL_ALGO and CAS_LOCAL to 1. (One can alternately use Pipek-Mezey by setting CAS_LOCAL to 2.) The localized RAS2 orbitals are automatically printed in Molden file format in the output file. The user can then visualize the localized orbitals and assign orbitals to sites manually as needed.

Once orbital assignment is complete, one can run a Bloch effective Hamiltonian calculation. This is done by setting RASSF_DO_BLOCH to true, and defining orbital assignments in the *$rassf_bloch_sites* section. This section has three lines. The first line specifies the total number of sites, the second line specifies the number of orbitals per site, and the third line is an ordered list of the orbitals belonging to each site. Note that orbital ordering starts at one. For a dichromium complex, where orbitals 3, 4, and 5 are localized on the first Cr atom and 1, 2, and 6 are on the second Cr, the section should therefore be defined as follows:

```
$rassf_bloch_sites
2
3 3
3 4 5 1 2 6
$end
```

7.12.10 State-Specific PCM with RAS-SF

The polarizable continuum model (PCM) is an efficient way to incorporate dielectric boundary conditions into a ground-state quantum chemistry calculation as described in Section 11.2.3, although some additional choices exist regarding how to model electronic excited states within this formalism.[105] For vertical transition energies computed using RAS-SF wave functions, both equilibrium and nonequilibrium versions of a state-specific PCM solvation correction are available.[5] Ground-state polarization is described using the solvent's static dielectric constant, and in the nonequilibrium solvation approach that polarization is modified upon vertical excitation using the solvent's optical dielectric constant.

The equilibrium formulation solves the state-specific Schrödinger equation

$$(\hat{H}_{\text{vac}} + \hat{R}_k)|\Psi_k\rangle = E_k|\Psi_k\rangle \tag{7.156}$$

where $\hat{R}_k$ is the reaction field operator for state k. Since $\hat{R}_k$ depends on $|\Psi_k\rangle$, this equation must be solved iteratively. The fully relaxed (equilibrium) excitation free energy is

$$\Delta G = G_k^{\text{eq}} - G_0 = \Delta E_k^{\text{eq}} - W_k + W_0 \; . \tag{7.157}$$

(It is a *free* energy because the continuum description of the solvent implicitly includes averaging over solvent degrees of freedom.) We use G to represent a free energy and W to represent polarization work, with subscripts to indicate the electronic state in question.

The nonequilibrium version of the solvation model uses a perturbative framework to solve the Schrödinger equation. This is described in more detail in Section 11.2.3.3. Briefly, this approach is based on a state-specific Hamiltonian that is partitioned according to

$$\hat{H}_k^{\text{noneq}} = \underbrace{\hat{H}_{\text{vac}} + \hat{R}_0^{\text{s+f}}}_{\hat{H}_0} + \lambda(\hat{R}_k^{\text{f}} - \hat{R}_0^{\text{f}}) \; . \tag{7.158}$$

The zeroth-order Hamiltonian ($\hat{H}_0$) includes the ground-state, equilibrium reaction-field operator $\hat{R}_0 \equiv R_0^{\text{s+f}} = \hat{R}_0^{\text{s}} + \hat{R}_0^{\text{f}}$. The solvent polarization is divided into "slow" (nuclear) and "fast" (electronic) components, but for equilibrium solvation both components are included. The perturbation (indicated by perturbation parameter λ corrects the fast polarization upon excitation, $|\Psi_0\rangle \rightarrow |\Psi_k\rangle$. This nonequilibrium correction is described via first-order perturbation theory. The result is a perturbative, state-specific (ptSS) correction to the excitation energy:

$$\Delta G_k^{\text{ptSS(1)}} = G_k - G_0 = \Delta E_k^{\text{ptSS(1)}} - W_{k(0)}^{\text{f}} + W_0^{\text{f}} + W_{0,k(0)} \tag{7.159}$$

For further details see Ref. 5.

7.12.11 Job Control for the RASCI1 Implementation

At present the RASCI1 and RASCI2 implementations employ different keywords (which will be reconciled in a future version). This subsection applies to RASCI1 (spin adapted algorithm using exact integrals).

The use of the RASCI1 methodology is controlled by setting the *$rem* variable CORRELATION to RASCI and EXCHANGE should be set to HF. The RASCI1 implementation is only compatible with the use of restricted orbitals, *i.e.*, UNRESTRICTED = FALSE.

The minimum input also requires specifying the desired (non-zero) value for RAS_ROOTS, the number of electrons in the "active" RAS2 space and the number of orbitals in RAS1 and RAS2 subspaces.

RAS_ROOTS

Sets the number of RAS-CI roots to be computed.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n $n > 0$ Compute n RAS-CI states

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ELEC

Sets the number of electrons in RAS2 (active electrons).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ELEC_ALPHA

Sets the number of spin-α electrons in RAS2 (active electrons).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ELEC_BETA

Sets the number of spin-β electrons in RAS2 (active electrons).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ACT

Sets the number of orbitals in RAS2 (active orbitals).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_OCC

Sets the number of orbitals in RAS1

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

These are the initial doubly occupied orbitals (RAS1) before including $hole$ type of excitations. The RAS1 space starts from the lowest orbital up to RAS_OCC, *i.e.* no frozen orbitals option available yet. Only works with RAS-CI.

RAS_DO_HOLE

Controls the presence of hole excitations in the RAS-CI wave function.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Include hole configurations (RAS1 to RAS2 excitations)

FALSE Do not include hole configurations

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_DO_PART

Controls the presence of particle excitations in the RAS-CI wave function.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Include particle configurations (RAS2 to RAS3 excitations)

FALSE Do not include particle configurations

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_AMPL_PRINT

Defines the absolute threshold ($\times 10^2$) for the CI amplitudes to be printed.

TYPE:

INTEGER

DEFAULT:

10 0.1 minimum absolute amplitude

OPTIONS:

n User-defined integer, $n \geq 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ACT_ORB

Sets the user-selected active orbitals (RAS2 orbitals).

TYPE:

INTEGER ARRAY

DEFAULT:

From RAS_OCC + 1 to RAS_OCC + RAS_ACT

OPTIONS:

$[i, j, k...]$ The number of orbitals must be equal to the RAS_ACT variable

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_NATORB

Controls the computation of the natural orbital occupancies.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute natural orbital occupancies for all states

FALSE Do not compute natural orbital occupancies

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_NATORB_STATE

Saves the natural orbitals of the ith RAS-CI computed state into the `.fchk` file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

i Saves the natural orbitals for the ith state

RECOMMENDATION:

None. Only works with RAS-CI and if GUI = 2.

RAS_GUESS_CS

Controls the number of closed shell guess configurations in RAS-CI.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Imposes to start with n closed shell guesses

RECOMMENDATION:

Only relevant for the computation of singlet states. Only works with RAS-CI.

RAS_SPIN_MULT

Specifies the spin multiplicity of the roots to be computed

TYPE:

INTEGER

DEFAULT:

1 Singlet states

OPTIONS:

0 Compute any spin multiplicity

$2n + 1$ User-defined integer, $n \geq 0$

RECOMMENDATION:

RAS_SPIN_MULT option is only available for $M_S = 0$ systems, that is, with the same number of α and β electrons.

RAS_PT2

Perform second-order perturbative correction to RAS-CI energy

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute RASCI(2) energy corrections

FALSE Do not compute RASCI(2) energy corrections

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_PT2_PARTITION

Specifies the partitioning scheme in RASCI(2)

TYPE:

INTEGER

DEFAULT:

1 Davidson-Kapuy (DK) partitioning

OPTIONS:

2 Epstein-Nesbet (EN) partitioning

0 Do both DK and EN partitionings

RECOMMENDATION:

Only for RAS-CI if RAS_PT2 is set to true.

RAS_PT2_VSHIFT

Defines the energy level shift ($\times 10^3\ au$) in RASCI(2)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined integer

RECOMMENDATION:

Only for RAS-CI if RAS_PT2 is set to true.

RAS_SRDFT

Perform short-range density functional RAS-CI calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute RASCI-srDFT states and energies

FALSE Do not perform a RASCI-srDFT calculation

RECOMMENDATION:

None. Only works with RAS-CI. RAS_SRDFT_EXC and RAS_SRDFT_COR need to be set.

RAS_SRDFT_EXC

Define short-range exchange functional

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use RAS_SRDFT_EXC = *NAME*, where *NAME* is one of the short-range exchange functionals listed in Section 5.3.3

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_SRDFT_COR

Define short-range correlation functional

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use RAS_SRDFT_COR = *NAME*, where *NAME* is one of the short-range correlation functionals listed in Section 5.3.4

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_OMEGA

Sets the Coulomb range-separation parameter within the RAS-CI-srDFT method.

TYPE:

INTEGER

DEFAULT:

400 ($\omega = 0.4$ bohr^{-1})

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None. Range-separation parameter is typical indicated by ω or μ. Only works with RAS-CI.

RAS_SRDFT_DAMP

Sets damping factor ($\alpha < 1$) in the RAS-CI-srDFT method.

TYPE:

REAL

DEFAULT:

0.5 ($\alpha = 0.5$)

OPTIONS:

α Damping factor $0 < \alpha < 1$

RECOMMENDATION:

Modify in case of convergence issues along the RAS-CI-srDFT iterations. Only works with RAS-CI

RAS_SRDFT_SA_ROOTS

Sets the list of roots used to build the state averaged reference density in RAS-CI-*sr*DFT.

TYPE:

INTEGER ARRAY

DEFAULT:

All computed states

OPTIONS:

$[i, j, k \ldots]$ List of states.

RECOMMENDATION:

None. Only works with RAS-CI

RAS_NFRAG

If $n > 0$ activates the excitation analysis in RAS-CI

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Number of fragments to be considered

RECOMMENDATION:

Only for RAS-CI. The printed information level is controlled by RAS_PRINT.

RAS_NFRAG_ATOMS

Sets the number of atoms in each fragment.

TYPE:

INTEGER ARRAY

DEFAULT:

None

OPTIONS:

$[i, j, k...]$ The sum of the numbers must be equal to the total number of atoms in the systems

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_FRAG_SETS

Defines the number of orbitals in each disjoint set to perform orbital localization.

TYPE:

INTEGER ARRAY

DEFAULT:

[NOcc,NAct,NVir] Number of orbitals within RAS1, RAS2 and RAS3 spaces

OPTIONS:

$[i, j, k...]$ Defines sets of canonical MOs to be localized into n fragments

RECOMMENDATION:

Setting within RAS1, RAS2 and RAS3 spaces alleviates the computational cost of the localization procedure. It might also result in improved fragment orbitals. Only works with RAS-CI.

CALC_SOC

Controls whether to calculate the SOC constants for EOM-CC, RAS-CI, TDDFT/TDA and TDDFT.

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform the SOC calculation.

TRUE Perform the SOC calculation.

RECOMMENDATION:

In addition to TRUE and FALSE, the EOM-CC code has more variants of SOC evaluation; see Section 7.10.20.4.

RAS_SOC_2E

Controls whether to compute two-electron mean-field contribution to RAS-CI SOC.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not compute two-electron mean-field contribution.

TRUE Compute two-electron mean-field contribution.

RECOMMENDATION:

None.

RAS_SOC_SYM_DENS

Controls whether to perform averaging of α and β densities.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not average α and β densities .

TRUE Average α and β densities.

RECOMMENDATION:

None.

7.12.12 Job Control Options for RASCI2

At present the RASCI1 and RASCI2 implementations employ different keywords (which will be reconciled in a future version). This subsection applies to RASCI2 (even and odd electron systems, determinant-driven algorithm using the resolution of the identity approximation).

The use of the RAS-CI2 methodology is controlled by setting the CORRELATION = RASCI2 and EXCHANGE = HF. The minimum input also requires specifying the desired (non-zero) value for RAS_N_ROOTS, and the number of active occupied and virtual orbital comprising the "active" RAS2 space. RASCI2 calculations also require specification of an auxiliary basis via AUX_BASIS.

RAS_N_ROOTS

Sets the number of RAS-CI roots to be computed.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n $n > 0$ Compute n RAS-CI states

RECOMMENDATION:

None. Only works with RASCI2

RAS_ACT_OCC

Sets the number of occupied orbitals to enter the RAS active space.

TYPE:

Integer

DEFAULT:

None

OPTIONS:

n user defined integer

RECOMMENDATION:

None. Only works with RASCI2

RAS_ACT_VIR

Sets the number of virtual orbitals to enter the RAS active space.

TYPE:

Integer

DEFAULT:

None

OPTIONS:

n user defined integer

RECOMMENDATION:

None. Only works with RASCI2.

RAS_ACT_DIFF

Sets the number of α versus β electrons and therefore controls the level of excitations used in calculations.

TYPE:

Integer

DEFAULT:

None

OPTIONS:

1	odd number of electrons or cations
0	even number of electrons
-1	anions
n	$n < -99$ triggers RAS2-SF at DDCI level of excitations $n = -451$ and $n = -452$ triggers restart mechanism that restores the last best guess for each state to the number of states requested

RECOMMENDATION:

Set to 0 would be appropriate for most singlet systems. Only works with RASCI2.

Note that other *$rem* variables that can be used to control the evaluation of RASCI2 calculations are MAX_CIS_CYCLES for the maximum number of Davidson iterations, and N_FROZEN_CORE and N_FROZEN_VIRTUAL to exclude core and/or virtual orbitals from the RASCI2 calculation. Please see the sample inputs below for more details.

RAS_CALC_SOC

Controls whether to calculate the SOC constants for RAS2 jobs only.

TYPE:

Integer/Logical

DEFAULT:

False

OPTIONS:

False	Do not perform the SOC calculation.
True	Perform the SOC calculation.

RECOMMENDATION:

This *$rem* variable is used to control the spin-orbit coupling (SOC) analysis section.

Note: For SOC calculations, RASCI2 method by default computes both the one-electron and two-electron mean-field contributions. SOC-NTO analyses can be triggered after the SOC analysis. Two additional parameters, MOLDEN_FORMAT and NTO_PAIRS, are needed for the NTO analysis. Details of the two variables are provided in the following *$rem* sections:

MOLDEN_FORMAT

Sets the output format of NTOs in RASCI2 SOC analysis to MOLDEN format.

TYPE:

Logical

DEFAULT:

False

OPTIONS:

True Append MOLDEN input file at the end of the Q-CHEM output file.

RECOMMENDATION:

Currently, SOC-NTO analysis in RASCI2 only works with MOLDEN. Other visualization tools are not supported at the moment. Please see the Visualizing Orbitals Using MOLDEN section for more information.

NTO_PAIRS

Controls the writing of hole/particle NTO pairs for SOC transitions calculated within the RASCI2 SOC analysis section.

TYPE:

Integer

DEFAULT:

0

OPTIONS:

N Write N NTO pairs per SOC transition.

RECOMMENDATION:

If activated ($N > 0$), a minimum of two NTO pairs will be printed for each transition. Increase the value of N if additional NTOs are desired. See Section 7.14.3 for information on visualizing NTOs.

RAS_PCM_INIT

Triggers state-specific PCM solvation within RAS-SF.

TYPE:

LOGICAL

DEFAULT:

FALSE no solvation (vacuum boundary conditions)

OPTIONS:

TRUE include solvation

RECOMMENDATION:

This triggers the nonequilibrium version of state-specific solvation. Additional keywords are necessary for the equilibrium version.

7.12.13 Job Control Options for CASMAN

In CASMAN, the built-in methods can be run by simply setting the value of CORRELATION accordingly:

CORRELATION

Sets the pre-defined method to use for CASMAN.

TYPE:

STRING

DEFAULT:

None.

OPTIONS:

SF-CAS	CAS-SF with no perturbative corrections (or a custom method)
SF-CAS(h,p)	CAS-SF with perturbative corrections for hole (h) and particle (p) excitations
SF-CAS(S)	CAS-SF with perturbative corrections for single (h,p,hp) excitations
SF-CAS(2x)	CAS-SF with perturbative corrections for 2x (h,p,hp,hh,pp) excitations
SF-CAS(3x)	CAS-SF with perturbative corrections for 3x (h,p,hp,hh,pp,hhp,hpp) excitations
SF-CAS(3x)+D	CAS-SF with perturbative corrections for (h,p,hp,hh,pp,hhp,hpp) + MP2 RAS1$\rightarrow$ RAS3 double excitations

RECOMMENDATION:

None.

If a custom excitation scheme is required, one can set CORRELATION = SF-CAS and set the following variables manually. (If you are using one of the built-in methods, setting these variables manually should be avoided, since they are automatically set up appropriately by Q-CHEM if CORRELATION is not SF-CAS.)

RAS_N_SPIN_FLIP

Sets the number of spin-flips.

TYPE:

INTEGER

DEFAULT:

Maximum number of spin-flips ($n = (\alpha - \beta)/2$)

OPTIONS:

n Do n spin-flips

RECOMMENDATION:

None.

CAS_DO_NDPT

Do non-degenerate perturbation theory?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do non-degenerate perturbation theory.

FALSE Do not use non-degenerate perturbation theory.

RECOMMENDATION:

None.

CAS_QDPT_ORDER

Order of terms kept in the quasi-degenerate perturbation theory denominator expansion.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

n Keep terms of order n in the denominator expansion.

RECOMMENDATION:

None.

CAS_DO_1X

Do perturbative hole (h) and particle (p) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative hole (h) and particle (p) correction

FALSE Do not do perturbative hole (h) and particle (p) correction

RECOMMENDATION:

None.

CAS_DO_SINGLES

Do perturbative singles (h,p,hp) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative singles correction

FALSE Do not do perturbative singles correction

RECOMMENDATION:

None.

CAS_DO_2x

Do perturbative 2x (h,p,hp,hh,pp) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative 2x correction

FALSE Do not do perturbative 2x correction

RECOMMENDATION:

None.

CAS_DO_3x

Do perturbative 3x (h,p,hp,hh,pp,hhp,hpp) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative 3x correction

FALSE Do not do perturbative 3x correction

RECOMMENDATION:

None.

CAS_DO_DOUBLES

Do perturbative (h,p,hp,hh,pp,hhp,hpp) correction + MP2 RAS1 $\rightarrow$ RAS3 doubles?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative (h,p,hp,hh,pp,hhp,hpp) + MP2 RAS1$\rightarrow$ RAS3 doubles correction

FALSE Do not do the correction

RECOMMENDATION:

None.

CAS_LEVEL_SHIFT

Use a denominator level-shift?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use the denominator level-shift

FALSE Do not use the denominator level-shift

RECOMMENDATION:

None.

CAS_SPARSE

Use a sparse matrix multiply when forming the effective Hamiltonian?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use sparse matrix multiply in forming effective Hamiltonian

FALSE Do not use sparse matrix multiply in forming effective Hamiltonian

RECOMMENDATION:

None. Can be useful for larger numbers of spin-flips.

7.12.14 Job Control Options for LIBRASSF

The LIBRASSF module can be enabled by setting EXCHANGE = HF, CORRELATION = RASCI2, and LIBRASSF = 1 in the *$rem* variable section. One should also indicate the number of roots by setting RAS_N_ROOTS appropriately. Although the module is called through RASCI2, most job control options are more similar to those found in RASCI1, with the exception of RAS_N_ROOTS.

The number of spin-flips and IP/EA are determined by the *$rem* variables RASSF_DELTA_ALPHA (the number of alpha electrons removed) and RASSF_DELTA_BETA (the number of beta electrons added). Hole and particle excitations can be added by setting the RAS_DO_HOLE and RAS_DO_PART keywords. Additionally, text files containing eigenvalues and eigenvectors can be printed via the RASSF_WRITE_EVALS and RASSF_WRITE_EVECS keywords.

RAS_N_ROOTS

Sets the number of RAS-CI roots to be computed.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n $n > 0$ Compute n RAS-CI states

RECOMMENDATION:

None.

RASSF_DELTA_ALPHA

Sets the number of alpha electrons to remove relative to the reference.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

0 Remove no alpha electrons (use for EA)

1 Remove one alpha electron (use for 1SF, IP)

2 Remove two alpha electrons (use for 2SF, 1SF-IP)

RECOMMENDATION:

None.

RASSF_DELTA_BETA

Sets the number of beta electrons to add relative to the reference.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

0 Add no beta electrons (use for IP)

1 Add one beta electron (use for 1SF, EA)

2 Add two beta electrons (use for 2SF, 1SF-EA)

RECOMMENDATION:

None.

RAS_DO_HOLE

Enables hole excitations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No hole excitations (use for CAS or RAS(p))

1 Single hole excitations (use for RAS(h) or RAS(h,p))

RECOMMENDATION:

None.

RAS_DO_PART

Enables particle excitations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No particle excitations (use for CAS or RAS(h))

1 Single particle excitations (use for RAS(p) or RAS(h,p))

RECOMMENDATION:

None.

N_FROZEN_CORE

Number of frozen core orbitals.

TYPE:

SIZE_T

DEFAULT:

0

OPTIONS:

n $n \geq 0$ Number of frozen core orbitals.

RECOMMENDATION:

None.

N_FROZEN_VIRTUAL

Number of frozen virtual orbitals.

TYPE:

SIZE_T

DEFAULT:

0

OPTIONS:

n $n \geq 0$ Number of frozen virtual orbitals.

RECOMMENDATION:

None.

RASSF_GUESS

Determines which initial set of guess vectors to use for Davidson.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 Random orthonormal guess (default for CAS)

1 Identity guess

2 CAS guess (default for RAS)

RECOMMENDATION:

Starting from a CAS guess is recommended for larger molecules. If Davidson encounters issues with linearly dependent eigenvectors, consider using identity. The random orthonormal guess requires building a large NxN matrix and is therefore only recommended for calculations with fewer determinants.

RASSF_WRITE_EVALS

Determines whether to write eigenvalues to an output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not write eigenvalues to an output file

1 Write eigenvalues to an output file

RECOMMENDATION:

None.

RASSF_WRITE_EVECS

Determines whether to write eigenvectors to an output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not write eigenvectors to an output file

1 Write eigenvectors to an output file

RECOMMENDATION:

None.

CAS_LOCAL

Determines whether to do localization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No localization

1 Boys localization

2 Pipek-Mezey localization

RECOMMENDATION:

None.

CAS_LOCAL_ALGO

Passed into localizer. Set to 1 if doing Boys localization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No localization

1 Boys localization

2 Pipek-Mezey localization

RECOMMENDATION:

None.

RASSF_DO_BLOCH

Determines whether to do effective Hamiltonian analysis.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Skip analysis

1 Do effective Hamiltonian analysis

RECOMMENDATION:

None.

7.12.15 Examples

Example 7.176 Input for a RAS-2SF-CI calculation of three states of the DDMX tetraradical using RASCI1. The active space (RAS2) contains 4 electrons in the 4 singly occupied orbitals in the ROHF quintet reference. Natural orbital occupancies are requested.

```
$molecule
   0 5
   C      0.000000    0.000000    1.092150
   C     -1.222482    0.000000    0.303960
   C     -2.390248    0.000000    1.015958
   H     -2.344570    0.000000    2.095067
   H     -3.363161    0.000000    0.537932
   C     -1.215393    0.000000   -1.155471
   H     -2.150471    0.000000   -1.702536
   C      0.000000    0.000000   -1.769131
   C      1.215393    0.000000   -1.155471
   H      2.150471    0.000000   -1.702536
   C      1.222482    0.000000    0.303960
   C      2.390248    0.000000    1.015958
   H      2.344570    0.000000    2.095067
   H      3.363161    0.000000    0.537932
$end

$rem
   EXCHANGE        hf
   CORRELATION     rasci
   BASIS           6-31g
   UNRESTRICTED    false
   MEM_TOTAL       4000
   MEM_STATIC      100
   RAS_ROOTS       3
   RAS_ACT         4
   RAS_ELEC        4
   RAS_OCC         25
   RAS_SPIN_MULT   0
   RAS_NATORB      true
   SYMMETRY        true
$end
```

Example 7.177 Input for a RAS-2IP-CI calculation of triplet states of F_2 molecule using the dianion closed shell F_2^{2-} as the reference determinant. RASCI1 code is used

```
$molecule
   -2 1
   F
   F  1  1.4136
$end

$rem
   EXCHANGE        hf
   CORRELATION     rasci
   BASIS           cc-pVTZ
   MEM_TOTAL       4000
   MEM_STATIC      100
   RAS_ROOTS       2
   RAS_ACT         6
   RAS_ELEC        10
   RAS_OCC         4
   RAS_SPIN_MULT   3
$end
```

Example 7.178 Input for a FCI/STO-3G calculation of water molecule expanding the RAS2 space to the entire molecular orbital set. RAS-CI code is used.

```
$molecule
   0 1
   O   0.000  0.000   0.120
   H  -0.762  0.000  -0.479
   H   0.762  0.000  -0.479
$end

$rem
   EXCHANGE        hf
   CORRELATION     rasci
   BASIS           sto-3g
   MEM_TOTAL       4000
   MEM_STATIC      100
   RAS_ROOTS       1
   RAS_ACT         7
   RAS_ELEC        10
   RAS_OCC         0
   RAS_SPIN_MULT   1
   RAS_DO_HOLE     false
   RAS_DO_PART     false
$end
```

Example 7.179 Methylene single spin-flip calculation using RASCI2

```
$molecule
   0 3
   C     0.0000000    0.0000000    0.0000000
   H    -0.8611113    0.0000000    0.6986839
   H     0.8611113    0.0000000    0.6986839
$end

$rem
   EXCHANGE       HF
   CORRELATION    RASCI2
   BASIS          cc-pVDZ
   AUX_BASIS      rimp2-cc-pVDZ
   UNRESTRICTED   false
   RAS_ACT_OCC    1    ! # alpha electrons
   RAS_ACT_VIR    1    ! # virtuals in active space
   RAS_ACT_DIFF   0    ! # set to 1 for odd # of e-s
   RAS_N_ROOTS    4
   MAX_CIS_CYCLES 25   ! number of iterations in RASCI
$end
```

Example 7.180 Two methylene separated by 10 Å; double spin-flip calculation using RASCI2. Note that the $\langle \hat{S}^2 \rangle$ values for this case will not be uniquely defined at the triply-degenerate ground state.

```
$molecule
   0 5
   C     0.0000000     0.0000000    0.0000000
   H    -0.8611113     0.0000000    0.6986839
   H     0.8611113     0.0000000    0.6986839
   C     0.0000000    10.0000000    0.0000000
   H    -0.8611113    10.0000000    0.6986839
   H     0.8611113    10.0000000    0.6986839
$end

$rem
   EXCHANGE       HF
   CORRELATION    RASCI2
   BASIS          cc-pVDZ
   AUX_BASIS      rimp2-cc-pVDZ
   RAS_ACT_OCC    2    ! # alpha electrons
   RAS_ACT_VIR    2    ! # virtuals in active space
   RAS_ACT_DIFF   0    ! # set to 1 for odd # of e-s
   UNRESTRICTED   false
   RAS_N_ROOTS    8
   MAX_CIS_CYCLES 25
$end
```

Example 7.181 RASCI2 calculation of the nitrogen cation using double spin-flip.

```
$molecule
   1 6
   N
   N   1  4.5
$end

$rem
   EXCHANGE           HF
   CORRELATION        RASCI2
   BASIS              6-31G*
   AUX_BASIS          rimp2-VDZ
   RAS_ACT_OCC        3   ! # alpha electrons
   RAS_ACT_VIR        3   ! # virtuals in active space
   RAS_ACT_DIFF       1   ! # for odd # e-s, cation
   UNRESTRICTED       false
   N_FROZEN_CORE      2
   N_FROZEN_VIRTUAL   2
   RAS_N_ROOTS        8
   MAX_CIS_CYCLES     25
$end
```

Example 7.182 CAS-1SF calculation of the nitrogen dimer using CASMAN.

```
$rem
 exchange                hf
 correlation             sf-cas
 cas_level_shift         0
 basis                   6-31g*
 aux_basis               rimp2-vdz
 unrestricted            0
 scf_guess               core
 integral_symmetry       0
 point_group_symmetry    false
$end

$molecule
0 7
N
N,1,R1

R1 2.5
$end
```

Example 7.183 RAS(3x)+D-1SF calculation of the nitrogen dimer using CASMAN.

```
$rem
 exchange                hf
 correlation             sf-cas(3x)+D
 ras_roots               10
 basis                   6-31g*
 aux_basis               rimp2-vdz
 unrestricted            0
 scf_guess               read
 integral_symmetry       0
 point_group_symmetry    false
$end

$molecule
0 7
N
N,1,R1

R1 2.5
$end
```

Example 7.184 Custom SF-CAS(2x)2 calculation of the nitrogen dimer cation using CASMAN.

```
$rem
 exchange                hf
 correlation             sf-cas
 cas_do_2x               1
 cas_qdpt_order          2
 cas_level_shift         107
 ras_roots               10
 basis                   6-31g*
 aux_basis               rimp2-vdz
 unrestricted            0
 scf_guess               core
 integral_symmetry       0
 point_group_symmetry    false
$end

$molecule
0 5
O
H,1,R1
H,1,R1,2,109.5

R1 2.0
$end
```

Example 7.185 RAS(h,p)-1SF calculation of the quartet state of the nitrogen dimer using LIBRASSF.

```
$molecule
   0 7
   N
   N   1   1.75
$end

$rem
   EXCHANGE            HF
   CORRELATION         RASCI2
   BASIS               cc-pvdz
   LIBRASSF            1
   RASSF_GUESS         1
   RASSF_DELTA_ALPHA   1
   RASSF_DELTA_BETA    1
   RAS_N_ROOTS         4
   RAS_DO_HOLE         1
   RAS_DO_PART         1
   RAS_NATORB          0
$end
```

Example 7.186 RAS(h)-1SF-IP calculation of the nitrogen cation using LIBRASSF.

```
$molecule
   0 7
   N
   N   1   1.75
$end

$rem
   EXCHANGE            HF
   CORRELATION         RASCI2
   BASIS               cc-pvdz
   LIBRASSF            1
   RASSF_GUESS         1
   RASSF_DELTA_ALPHA   1
   RASSF_DELTA_BETA    1
   RAS_N_ROOTS         4
   RAS_DO_HOLE         1
   RAS_DO_PART         1
   RAS_NATORB          0
$end
```

Example 7.187 SOC calculation of HSiCl molecule using RASCI2 method with double spin-flip. No NTO analysis is requested within this example.

```
$molecule
   0 1
   Si    0.00000    0.00000    0.00000
   H     1.56100    0.00000    0.00000
   Cl   -0.45728    2.01271    0.00000
$end

$rem
   EXCHANGE          HF
   CORRELATION       RASCI2
   BASIS             cc-pVDZ
   AUX_BASIS         rimp2-cc-pVDZ
   UNRESTRICTED      false
   RAS_ACT_OCC       2    ! # alpha electrons
   RAS_ACT_VIR       2    ! # virtuals in active space
   RAS_ACT_DIFF      0    ! # set to 1 for odd # of e-s
   RAS_N_ROOTS       8
   MAX_CIS_CYCLES    25
   N_FROZEN_CORE     fc
   N_FROZEN_VIRTUAL  0
   RAS_CALC_SOC      true
$end
```

Example 7.188 SOC calculation followed by NTO analysis of acetophenone molecule using RASCI2 method with double spin-flip.

```
$molecule
   0 1
   C   -2.517371    0.513126    0.001828
   C   -1.316451   -0.371636    0.001013
   C   -1.553894   -1.827349    0.000191
   C   -2.806559   -2.314044   -0.001563
   C   -3.962422   -1.432985   -0.002120
   C   -3.757473   -0.015730    0.000047
   C   -0.030094    0.202242    0.001108
   C    1.208765   -0.658368   -0.000434
   O    0.079290    1.457055    0.003022
   H    2.079440   -0.009753   -0.004191
   H   -0.705807   -2.494690    0.000993
   H   -2.970202   -3.383004   -0.002372
   H   -4.958256   -1.847303   -0.003648
   H   -4.620034    0.636483    0.000372
   H   -2.340078    1.575881    0.003550
   H    1.252514   -1.299975    0.880472
   H    1.247817   -1.304378   -0.878269
$end

$rem
   EXCHANGE          HF
   CORRELATION       RASCI2
   BASIS             cc-pVDZ
   AUX_BASIS         rimp2-cc-pVDZ
   UNRESTRICTED      false
   RAS_ACT_OCC       2    ! # alpha electrons
   RAS_ACT_VIR       2    ! # virtuals in active space
   RAS_ACT_DIFF      0    ! # set to 1 for odd # of e-s
   RAS_N_ROOTS       4
   MAX_CIS_CYCLES    25
   N_FROZEN_CORE     fc
   N_FROZEN_VIRTUAL  0
   RAS_CALC_SOC      true
   MOLDEN_FORMAT     true
   NTO_PAIRS         2
$end
```

7.13 Core Ionization Energies and Core-Excited States

In experiments using high-energy radiation (such as x-ray spectroscopy, EXAFS, NEXAFS, XAS, XES, RIXS, REXS, *etc.*) core electrons can be ionized or excited to low-lying virtual orbitals. There are several ways to compute ionization or excitation energies of core electrons in Q-CHEM. Standard approaches for excited and ionized states need to be modified to tackle core-level states, because these states have very high energies and are embedded in the ionization continuum, *i.e.*, they are Feshbach resonances.[219]

7.13.1 Many-Body Methods for Core-Excited States

A highly robust and accurate strategy is to invoke many-body methods, such as EOM or ADC, together with the core-valence separation (CVS) scheme.[47] In this approach, the excitations involving core electrons are decoupled from the

rest of the configurational space. This allows one to reduce computational costs and decouple the highly excited core states from the continuum. These methods are described in Sections 7.10.8 and 7.11.6; CVS can also be deployed within TDDFT as described in Section 7.13.2. Error introduced by the CVS approximation is negligible for K-edge (1s $\rightarrow$ virtual) excitations,[112] though its accuracy for other types of x-ray excitations is less certain.

An alternative highly accurate approach for finding core-excitation energies of closed-shell molecules is to use the restricted open-shell Kohn-Sham (ROKS) approach that is described in Section 7.8.3. ROKS is not systematically improvable like EOM or ADC methods, but is nonetheless quite accurate and modern density functionals are capable of predicting excitation energies to < 0.5 eV error.[90,106] The great strength of the ROKS approach is its computational efficiency: highly accurate results can be obtained for the same $\mathcal{O}(N^3)$ scaling as ground-state meta-GGAs, versus the $\mathcal{O}(N^6)$ scaling of EOM-CCSD or $\mathcal{O}(N^5)$ scaling of ADC(2). The basis set requirements of ROKS are also much more modest than wave function theories, with a mixed basis strategy being highly effective in practice. Details about using ROKS for core-level excitations can be found in Section 7.13.4.

Within EOM-CC formalism, one can also use an approximate EOM-EE/IP methods in which the target states are described by single excitations and double excitations are treated perturbatively; these methods are described in Section 7.10.15.2. While being moderately useful, these methods are less accurate than the CVS-EOM variants.[219]

In addition, one can use the ΔE approach, which amounts to a simple energy difference calculation in which core ionization is computed from energy differences computed for the neutral and core-ionized state. This procedure is

illustrated by Example 7.189 below.

Example 7.189 Q-CHEM input for calculating chemical shift for 1s-level of methane (CH_4). The first job is just an SCF calculation to obtain the orbitals and CCSD energy of the neutral. The second job solves the HF and CCSD equations for the core-ionized state.

```
$molecule
   0,1
   C      0.000000    0.000000    0.000000
   H      0.631339    0.631339    0.631339
   H     -0.631339   -0.631339    0.631339
   H     -0.631339    0.631339   -0.631339
   H      0.631339   -0.631339   -0.631339
$end

$rem
   EXCHANGE        = HF
   CORRELATION     = CCSD
   BASIS           = 6-31G*
   MAX_CIS_CYCLES  = 100
$end

@@@

$molecule
   +1,2
   C      0.000000    0.000000    0.000000
   H      0.631339    0.631339    0.631339
   H     -0.631339   -0.631339    0.631339
   H     -0.631339    0.631339   -0.631339
   H      0.631339   -0.631339   -0.631339
$end

$rem
   UNRESTRICTED    = TRUE
   EXCHANGE        = HF
   BASIS           = 6-31G*
   MAX_CIS_CYCLES  = 100
   SCF_GUESS       = read  Read MOs from previous job and use occupied as specified below
   CORRELATION     = CCSD
   MOM_START       = 1 Do not reorder orbitals in SCF procedure!
$end

$occupied
   1 2 3 4 5
   2 3 4 5
$end
```

In this job, we first compute the HF and CCSD energies of neutral CH_4: $E_{\text{SCF}} = -40.1949062375\ E_h$ and $E_{\text{CCSD}} = -40.35748087\ E_h$ (HF orbital energy of the neutral gives the Koopmans IE, which is 11.210 hartree = 305.03 eV). In the second job, we do the same for core-ionized CH_4. To obtain the desired SCF solution, MOM_START option and *$occupied* keyword are used. The resulting energies are $E_{\text{SCF}} = -29.4656758483\ E_h$ ($\langle \hat{S}^2 \rangle$ = 0.7730) and $E_{\text{CCSD}} = -29.64793957\ E_h$. Thus, $\Delta E_{\text{CCSD}} = 10.709\ E_h = 291.42$ eV.

This approach can be further extended to obtain multiple excited states involving core electrons by performing CIS, TDDFT, or EOM-EE calculations.

Note: This approach often leads to convergence problems in correlated calculations.

One can also use the following trick illustrated by Example 7.190.

Example 7.190 Q-CHEM input for calculating chemical shift for 1s-level of methane using EOM-IP. Here we solve SCF as usual, then reorder the MOs such that the core orbital becomes the "HOMO", then solve the CCSD and EOM-IP equations with all valence orbitals frozen and the core orbital being active.

```
$molecule
   0,1
   C      0.000000    0.000000    0.000000
   H      0.631339    0.631339    0.631339
   H     -0.631339   -0.631339    0.631339
   H     -0.631339    0.631339   -0.631339
   H      0.631339   -0.631339   -0.631339
$end

$rem
   EXCHANGE         = HF
   BASIS            = 6-31G*
   MAX_CIS_CYCLES   = 100
   CORRELATION      = CCSD
   CCMAN2           = false
   N_FROZEN_CORE    = 4 Freeze all valence orbitals
   IP_STATES        = [1,0,0,0] Find one EOM_IP state
$end

$reorder_mo
   5 2 3 4 1
   5 2 3 4 1
$end
```

Here we use EOM-IP to compute core-ionized states. Since core states are very high in energy, we use "frozen core" trick to eliminate valence ionized states from the calculation. That is, we reorder MOs such that our core is the last occupied orbital and then freeze all the rest. The so computed EOM-IP energy is 245.57 eV. From the EOM-IP amplitude, we note that this state of a Koopmans character (dominated by single core ionization); thus, canonical HF MOs provide good representation of the correlated Dyson orbital. The same strategy can be used to compute core-excited states.

Note: The accuracy of this approach is rather poor and is similar to Koopmans' approximation.

7.13.2 Calculations of X-Ray Spectroscopy with TDDFT

x-ray absorption spectroscopy can be calculated using TDDFT by restricting the excitation space to include excitations from a set of core orbitals. This is achieved by setting TRNSS = TRUE in the *$rem* section, which triggers the use of TDDFT with a truncated excitation space as described in Section 7.3.3. The occupied core orbitals that the user desires to be active in such a calculation should be listed individually in the *$alist* input section, and the number of such orbitals must be specified using N_SOL in the *$rem* section. This invokes the CVS approximation, which for TDDFT amounts to freezing all of the occupied orbitals except for the ones that are listed in *$alist*, while using the full virtual space.[106] Such calculations are not suited to describe the extended x-ray absorption fine structure (EXAFS) region, which corresponds to the scattering of the ionized electron by the neighboring atoms.

Standard exchange-correlation functionals (including hybrids) tend to severely underestimate core excitation energies,[25] although chemical shifts (from one compound to the next) may still be valid.[75] Q-CHEM has short-range corrected (SRC) functionals available that are designed to predict K-edge core excitation energies accurately.[25] These functionals are a modification of the more familiar long-range corrected functionals (discussed in Section 5.6). However, in SRC-DFT the short-range component of the Coulomb operator is predominantly Hartree-Fock exchange, while

the mid to long-range component is primarily treated with standard DFT exchange. Job control variables for SRC functionals are provided below and their use in conjunction with the CVS approximation is illustrated in Example 7.191. See Section 7.3.5.1 for basic TDDFT job control.

HF_SR

Sets the fraction of Hartree-Fock exchange at $r_{12} = 0$.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to HF_SR = $n/1000$

RECOMMENDATION:

None

HF_LR

Sets the fraction of Hartree-Fock exchange at $r_{12} = \infty$.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to HF_LR = $n/1000$

RECOMMENDATION:

None

OMEGA

Sets the Coulomb attenuation parameter for the short-range component.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

OMEGA2

Sets the Coulomb attenuation parameter for the long-range component.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega_2 = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

SRC_DFT

Selects form of the short-range corrected functional.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

1 SRC1 functional.

2 SRC2 functional.

RECOMMENDATION:

None

Example 7.191 Calculation of carbon K-edge [C(1s) → virtual excitations] using SRC-TDDFT within the CVS approximation.

```
$comment
Carbon K-edge excitations for acetone.
The only active occupied orbitals are three C(1s).
The SRC1-R1 is parameterized for "first row" (C, N, O, ...)
$end

$molecule
0 1
  C   -3.0219081    1.0061477    0.0000001
  O   -2.9337180    2.2246186    0.0000001
  C   -1.7817549    0.1662163   -0.0000003
  C   -4.3700966    0.3535647    0.0000004
  H   -0.8735407    0.8061311   -0.0000005
  H   -1.7663727   -0.4765415   -0.9049102
  H   -1.7663723   -0.4765416    0.9049094
  H   -5.1766925    1.1175964    0.0000006
  H   -4.4778785   -0.2802782    0.9049091
  H   -4.4778790   -0.2802781   -0.9049084
$end

$rem
exchange           src1-r1  ! r1 = "first row"
basis              6-31++G*
cis_n_roots        25
cis_triplets       false
trnss              true
trtype             3
n_sol              3 ! no. of active orbs
$end

$alist
2 3 4
$end
```

Relativistic effects become increasingly significant for calculation of x-ray absorption spectra at the K-edge of heavier elements. The REL_SHIFT keyword introduces a correction to the calculated excitation energies to account for these effects, as illustrated in Example 7.192 below. The element-specific correction that is added is equal to the difference between relativistic (Douglas-Kroll-Hess) and non-relativistic Hartree-Fock eigenvalues for the $1s$ orbital of the isolated atom.[24,260] This probably suffices to account for relativistic effects for K-edge excitations, although in the case of L- or M-edge excitations the situation is complicated by spin-orbit coupling.

REL_SHIFT

Corrects the calculated TDDFT excitation energy for scalar relativistic effects.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

Z Corresponding to the atomic number of the core-ionized element.

RECOMMENDATION:

The relativistic correction is equal to the difference of relativistic (Douglas-Kroll-Hess) and non-relativistic Hartree-Fock/cc-pCVTZ eigenvalues for the $1s$ orbital of the isolated atom.

Example 7.192 Calculation of core-excited states at the phosphorus K-edge, including a scalar relativistic shift.

```
$molecule
   0 1
   H     1.196206    0.000000   -0.469131
   P     0.000000    0.000000    0.303157
   H    -0.598103   -1.035945   -0.469131
   H    -0.598103    1.035945   -0.469131
$end

$rem
   EXCHANGE        SRC2-R2  ! R2 = "second row" (Al, S, P, ...)
   BASIS           6-311(2+,2+)G**
   CIS_N_ROOTS     6
   CIS_TRIPLETS    false
   TRNSS           true
   TRTYPE          3
   N_SOL           1
   REL_SHIFT       15
$end

$alist
   1
$end
```

Despite the relatively low computational cost of TDDFT, it can become challenging to calculate x-ray absorption spectra for large systems. The high density of core-excited states makes simulating spectra more computationally expensive than comparable calculations of the UV/vis spectra. This is particularly the case when excitations from many core-orbitals are required, which is often the situation when studying the carbon K-edge of organic molecules. There are two aspects to the computational cost, firstly the CPU time required and secondly the memory required. An implementation of TDDFT called "fTDDFTs" combines aggressive integral screening and a coarse DFT quadrature grid, which is especially efficient for the calculation of x-ray absorption spectra.[21,22] This approach may be fine-tuned using the *$rem* variables FAST_XAS, XAS_SCREEN_LEVEL and XAS_EDGE. The memory required for these calculations can be reduced further through the TDDFT_NVIRT keyword that reduces the number of virtual orbitals included in the TDDFT calculation. Job control variables for fTDDFTs are listed below and are illustrated in Example 7.193.

Note: Currently fTDDFTs works only for restricted CIS/TDA calculations and is not parallelized. (Multiple threads can be used for the initial SCF calculation but the subsequent CIS/TDA calculation is performed in serial.)

FAST_XAS

Controls whether fast TDDFT for core excitations is used.

TYPE:

LOGICAL

DEFAULT:

FALSE Normal TDDFT calculation.

OPTIONS:

TRUE Use fast TDDFT.

RECOMMENDATION:

None

XAS_SCREEN_LEVEL

Sets the integral screening procedure for fast TDDFT.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 only evaluate integrals that include the inner core basis function on relevant atom(s).

2 only evaluate integrals that include basis functions on relevant atom(s).

RECOMMENDATION:

1

XAS_EDGE

Specifies the nuclear charge of element being excited.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Corresponding to the nuclear charge of element being excited.

RECOMMENDATION:

None

TDDFT_NVIRT

Specifies the number of virtual orbitals included in the XAS TDDFT calculation.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Corresponding to the lowest energy n virtual orbitals.

RECOMMENDATION:

None

Example 7.193 Fast, low-memory calculation of core-excited states at the oxygen K-edge of CO using fTDDFTs.

```
$molecule
   0 1
   C      0.000000    0.000000   -0.648906
   O      0.000000    0.000000    0.486357
$end

$rem
   EXCHANGE            SRC1-R1
   BASIS               6-311G*
   CIS_N_ROOTS         6
   CIS_TRIPLETS        false
   TRNSS               true
   TRTYPE              3
   N_SOL               1
   FAST_XAS            true
   XAS_EDGE            6
   XAS_SCREEN_LEVEL    1
$end

$alist
   1
$end
```

It is also possible to compute x-ray emission spectroscopy using TDDFT. This is achieved by using a reference determinant with a core-hole.[74,260] The calculated excitation energies can be quite sensitive to the choice of basis set, and for the K-edge of heavier elements it can be necessary to use large or specially adapted basis sets to provide a good description of the core region.[73,97]

Example 7.194 This example shows a calculation of the XES spectrum of water using TDDFT + MOM.

```
$molecule
   0 1
   O    0.0000    0.0000    0.1168
   H    0.0000    0.7629   -0.4672
   H    0.0000   -0.7629   -0.4672
$end

$rem
   method       cam-b3lyp
   basis        cc-pvdz
$end

@@@

$molecule
   +1 2
   O    0.0000    0.0000    0.1168
   H    0.0000    0.7629   -0.4672
   H    0.0000   -0.7629   -0.4672
$end

$rem
   method       cam-b3lyp
   basis        cc-pvdz
   scf_guess    read
   mom_start    1
   cis_n_roots  5
   cis_triplets false
$end

$occupied
   1:5
   2:5
$end
```

7.13.3 Methods Based on Kohn-Sham Eigenvalues

7.13.3.1 Koopmans' Approach

A simpler alternative to TDDFT for x-ray emission is to use Kohn-Sham eigenvalue differences,

$$\Delta E = \epsilon_v - \epsilon_c \ , \tag{7.160}$$

where ϵ_v and ϵ_c are valence and core energy levels, respectively. Oscillator strengths are obtained from the corresponding transition dipole matrix elements,

$$f_{c\to v} \propto \|\langle\phi_c|\hat{\boldsymbol{\mu}}|\phi_v\rangle\|^2 \ . \tag{7.161}$$

This Koopmans' theorem-type approach is somewhat crude, as there is no account for orbital relaxation in the core-excited state, but it has the benefit that only a ground-state calculation is required and therefore this approach is applicable to large systems,[96] and in conjunction with SRC functionals even this simple procedure can afford reasonable estimates of the transition energies.[95] The method is controlled by *$rem* variables NCORE_XES and NVAL_XES that specify the number of core (c) and valence (v) levels to consider, and an example is given in Example 7.195. Extension of this approach to resonant x-ray emission spectroscopy (involving an excited electronic state) is possible by modeling that state as a non-*aufbau* solution of the SCF equations, *e.g.*, using algorithms such as (Section 4.5.14), SGM (Section 4.5.15), or STEP (4.5.16).

NCORE_XES

Specifies how many core levels to use in a Koopmans-type XES calculation.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Compute transition dipoles corresponding to the first (lowest energy) n core orbitals, ϕ_c.

RECOMMENDATION:

None

NVAL_XES

Specifies how many valence virtual levels to use in a Koopmans-type XES calculation.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Compute transition dipoles corresponding to the highest n occupied orbitals, ϕ_v.

RECOMMENDATION:

Setting $n = 1$ will include the HOMO in the occupied space, $n = 2$ will include HOMO and HOMO $- 1$, etc.

Example 7.195 The calculation of the XES spectrum of water using the Koopmans approach with a short-range corrected functional.

```
$molecule
   0 1
   C          0.0000000000    0.0000000000    0.5121520001
   O          0.0000000000    0.0000000000   -0.6942567610
   H          0.9377642813    0.0000000000    1.1074358558
   H         -0.9377642813    0.0000000000    1.1074358558
$end

$rem
   METHOD           src1r1
   BASIS            6-311G**
   NCORE_XES        2
   NVAL_XES         4
$end
```

7.13.3.2 Transition-Potential DFT

The *transition potential* (TP-)DFT method [125,239,248] is an alternative approach that accounts for some orbital relaxation yet retains a framework based on Kohn-Sham eigenvalues, requiring only a ground-state calculation. This approach is based on Slater's transition concept, [106,124,232] in which an SCF calculation with a fractional electron (originally $n_i = 1/2$) is removed from occupied orbital ϕ_i, then the ionization energy for that MO is approximated as

$$\mathrm{IE}_i \approx -\epsilon_i(n_i = 1/2) \; . \tag{7.162}$$

This can be justified based on a Taylor expansion in terms of the orbital occupations. [106,124,232] Excitation energies are approximated as eigenvalue differences $\epsilon_a - \epsilon_i$ obtained from a fractional-electron SCF calculation in which $n_i = 1/2$

electron is promoted from ϕ_i into the LUMO:

$$\Delta E_{i\to a} = \epsilon_a(n_i = 1/2, n_{\text{LUMO}} = 1/2) - \epsilon_i(n_i = 1/2, n_{\text{LUMO}} = 1/2)\ , \tag{7.163}$$

with oscillator strengths $f_{i\to a} \propto |\langle\phi_i|\hat{\boldsymbol{\mu}}|\phi_a\rangle|^2$, as in Eq. (7.161).

TP-DFT calculations in Q-CHEM are setup to remove 1/2 electron from the lowest core orbital of a given atom that is specified using TPDFT_ATOM. (For more flexible and general fractional-electron SCF schemes, see Section 7.13.3.3.) Optionally, one may use TPDFT_LUMO to occupy the LUMO, corresponding to an excitation energy calculation [Eq. (7.163)], or omit this variable to compute the core-level electron binding energy [Eq. (7.162)].

TPDFT_ATOM

Activate TP-DFT by specifying the atom from which to remove an electron.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Remove an electron from the lowest-energy orbital on the atom whose index is n.

RECOMMENDATION:

Be sure to set UNRESTRICTED = TRUE for TP-DFT calculations.

TPDFT_FRAC

Specify the fractional value of n_i to be removed.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Remove $n/100$ of an electron from the orbital specified using TPDFT_ATOM.

RECOMMENDATION:

None

TPDFT_LUMO

Specify the fractional value of n_{LUMO} to be added.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Add $n/100$ of an electron to the LUMO.

RECOMMENDATION:

Leave TPDFT_LUMO = 0 for core-level binding energy calculations [Eq. (7.162)] or use TPDFT_LUMO = 50 to implement Eq. (7.163).

Example 7.196 TP-DFT excitation energy calculation, removing $n_i = 1/2$ from O(1s) and placing it in the LUMO.

```
$molecule
   0 1
   O
   H 1 0.95
   H 2 0.95 2 104.5
$end

$rem
   METHOD       b3lyp
   BASIS        aug-cc-pCVQZ
   UNRESTRICTED true
   TPDFT_ATOM   1
   TPDFT_FRAC   50
   TPDFT_LUMO   50 ! set to 0 for IE calculation
$end
```

7.13.3.3 Slater-Type Fractional-Electron Methods

A more general set of fractional-electron methods for both core-level ionization (*i.e.*, XPS) and core-level excitation (XAS and also XES) has been explored by Jana and Herbert.[124,125] These methods generalize Slater's transition concept,[106,124,232] and allow an arbitrary fraction of an electron to be removed from a core MO and (optionally) placed into a virtual MO, as specified by the user. The use of these generalized Slater-type methods is controlled by the *$rem* variables that are described below and illustrated in examples that follow. For XAS, oscillator strengths are computed according to Eq. (7.161), and the (occupied, virtual) orbital pairs (ϕ_i, ϕ_a) for which the transition dipole moment is computed are specified using NCORE_XAS and NVAL_XAS, as described in Section 7.13.3.1. To converge the fractional-electron state, it may be necessary to use an algorithm such as MOM (Section 4.5.14), SGM (Section 4.5.15), or STEP (4.5.16). Consult Refs. 124 and 125 for best practices regarding which generalized Slater-type method to use.

FRACTIONAL_ELECTRON

Specify the fraction of an electron to be removed from the occupied space.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

$-n$ Remove $n/1000$ of an electron.

RECOMMENDATION:

The original Slater method ($n_i = 1/2$) corresponds to FRACTIONAL_ELECTRON = -500 but there can be other choices.

FRAC_VIR_ELEC

Specify the fraction of an electron to place into the occupied space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Add $n/1000$ of an electron.

RECOMMENDATION:

A value > 0 should be used for excitation (XAS or XES), whereas the default is appropriate for ionization (XPS).

FRAC_ELEC_ORB

Specify the occupied orbital from which the fractional electron should be removed.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Remove from ϕ_n.

RECOMMENDATION:

None

FRAC_VIR_ELEC_ORB

Specify the virtual orbital to which the fractional electron should be added.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Add to ϕ_n.

RECOMMENDATION:

Use this only if FRAC_VIR_ELEC > 0.

Example 7.197 Original Slater transition method (STM) for oxygen K-shell XPS, removing $n_i = 1/2$ electron from O(1s) core orbital. A normal SCF calculation is used to obtain the initial set of orbitals, followed by a fractional-electron calculation that is converged using the IMOM algorithm.

```
$molecule
0 1
C  -0.00000000  0.00000000  0.07378202
O   0.00000000  0.00000000  1.20921798
$end

$rem
 UNRESTRICTED  TRUE
 BASIS         def2-QZVP
 METHOD        B3LYP
 POINT_GROUP_SYMMETRY FALSE
 INTEGRAL_SYMMETRY    FALSE
$end

@@@

$molecule
read
$end

$rem
 SCF_GUESS     READ
 METHOD        B3LYP
 BASIS         def2-QZVP
 UNRESTRICTED  TRUE
 FRAC_ELEC_ORB        1  ! lowest-energy orbital is O(1s)
 FRAC_VIR_ELEC_ORB    1
 FRACTIONAL_ELECTRON -500
 FRAC_VIR_ELEC        0
 MOM_START            1
 MOM_METHOD           IMOM
 POINT_GROUP_SYMMETRY FALSE
 INTEGRAL_SYMMETRY    FALSE
$end
```

Example 7.198 XAS at carbon K-edge, using Slater's transition method with $n_i = 1/2$ electrons removed from C(1s) and $n_a = 1/2$ electron placed in the LUMO.

```
$molecule
 0 1
C  -0.00000000  0.00000000  0.07378202
O   0.00000000  0.00000000  1.20921798
$end

$rem
 UNRESTRICTED  TRUE
 BASIS         def2-QZVP
 METHOD        B3LYP
 POINT_GROUP_SYMMETRY FALSE
 INTEGRAL_SYMMETRY    FALSE
$end

@@@

$molecule
read
$end

$rem
 SCF_GUESS     READ
 METHOD        B3LYP
 BASIS         def2-QZVP
 UNRESTRICTED  TRUE
 FRAC_ELEC_ORB        2   ! C(1s)
 FRAC_VIR_ELEC_ORB    1
 FRACTIONAL_ELECTRON -500
 FRAC_VIR_ELEC        500
 MOM_START            1
 MOM_METHOD           IMOM
 NCORE_XAS            2
 NVAL_XAS             2
 POINT_GROUP_SYMMETRY FALSE
 INTEGRAL_SYMMETRY    FALSE
$end
```

7.13.4 Calculations of Core Excitations with ROKS

The restricted open-shell Kohn-Sham (ROKS) approach is a highly accurate method for estimating core-excitation energies of closed-shell molecules,[90,106] as described in Section 7.8.3. Here, we briefly recapitulate the key aspects and refer the reader to Ref. 90 for details ROKS with the SCAN functional is found to reproduce 40 experimental core excitation energies (from the 1s orbital, *i.e.*, K-edge) of second-period elements (C, N, O, and F) to an RMS error of 0.2 eV and a maximum absolute error of 0.5 eV. The ωB97X-V functional provides similar (if a little worse) accuracy as well. Similar behavior is observed for the $L_{2,3}$ edges of third-period elements Si, P, S and Cl. Other widely used functionals like PBE fare somewhat worse, but still predict much lower error as compared to TDDFT using the same functionals. Recently, we extended the applicability for ROKS for core excited states of heavier elements by including scalar relativistic effects as described in Section 4.9.6. Accurate modeling of K-edge of elements up to $Z = 24$ can be achieved with the SCAN functional.[55]

That said, the ROKS approach is state-specific in that it can only predict a single state at a time and needs to be told which state to target (via the *$reorder_mo* section, as shown in Example 7.53). This makes it less black-box than

TDDFT as the final orbital needs to be identified *a priori*, perhaps via a pilot TDDFT job if no other information is available. (For core-level excitations, the initial orbital is usually intuitively obvious.) ROKS can also be used for two-site doubly core-ionized states, or other systems with one broken electron pair in total.

The accuracy of ROKS stems from three factors: choice of density functional (SCAN or ωB97X-V), excited state orbital optimization (only available via SGM for core excitations, as described in Section 4.5.15) and a sufficiently flexible basis set. The last is key, as the split-core functions (as provided by basis sets like cc-pCVnZ) are needed instead of standard basis sets like cc-pVnZ that only have split valence functions. Indeed, a basis of triple-ζ quality like cc-pCVTZ is necessary to fully account for the core-hole relaxation and smaller basis sets lead to systematic overestimation of excitation energies. However, the highly local nature of the core-hole ensures that a large basis is only needed for the target atom of the ROKS calculation, and a smaller basis (of double-ζ quality, like cc-pVDZ) is adequate for all other atoms. An example of this mixed basis strategy is given below in Example 7.199. Details about using mixed basis sets in general can be found in Section 8.5.

The number of cycles needed for ROKS calculations can also be considerably reduced by decoupling the core hole relaxation from the rest of the orbital optimization. This entails a restricted open-shell ΔSCF calculation of the core-ionized state first, and use of those orbitals as guess for ROKS. Example 7.53 is a representative case for how such calculations should proceed.

The conjunction of high accuracy and low computational cost (due to the affordability of the SCAN meta-GGA and the mixed basis strategy) makes ROKS a very attractive approach for computing core spectra of large, closed-shell systems where more expensive wave function theories are unaffordable. Users are requested to cite Ref. 90 when using ROKS

for core excitations and Ref. 55 when performing calculations that include scalar relativistic effects.

Example 7.199 RO-ΔSCF core-ionization at C for CO, using SGM. The mixed basis strategy is used as the core-hole is local to C.

```
$molecule
   0 1
   C 0.0000   0.0000   0.0000
   O 0.0000   0.0000   1.1282
$end

$rem
   METHOD     scan
   BASIS      gen
   BASIS2     aug-cc-pVDZ
   INTEGRAL_SYMMETRY false
$end

$basis
C
aug-cc-pCVTZ
****
O
aug-cc-pVDZ
****
$end

@@@@

$molecule
   1 2
   C 0.0000   0.0000   0.0000
   O 0.0000   0.0000   1.1282
$end

$rem
   METHOD          scan
   BASIS           gen
   UNRESTRICTED    false
   SCF_GUESS       read
   INTEGRAL_SYMMETRY false
   SCF_ALGORITHM   sgm
$end

$reorder_mo
   1 3 4 5 6 7 2
   1 3 4 5 6 7 2
$end

$basis
C
aug-cc-pCVTZ
****
O
aug-cc-pVDZ
****
$end
```

Example 7.7.200 Combined RO-ΔSCF core-ionization and $1s \rightarrow$ LUMO ROKS core-excited state for HCl including scalar relativistic effects. The uncontracted aug-pcX-2 basis has to be supplied by the user

View input online

7.14 Visualization of Excited States

7.14.1 Introduction

As methods for *ab initio* calculations of excited states are becoming increasingly more routine, questions arise concerning how best to extract chemical meaning from such calculations. There are several approaches for analyzing molecular excited states; they are based on reduced one-particle density matrices (OPDMs). The two objects exploited in this analysis are: (i) the difference between the ground- and excited-state OPDMs and (ii) the transition OPDM connecting the ground and excited state. In the case of CIS and TDDFT/TDA wave functions, both quantities are identical and can be directly mapped into the CIS amplitudes; however, for correlated wave functions the two objects are not the same. The most basic analysis includes calculation of attachment and detachment densities[100] (equivalent within TDDFT to particle and hole densities, respectively),[106] and natural transition orbitals.[106,169] These quantities allow one to arrive to a most compact description of an excited state. More detailed analysis allows one to derive additional insight about the nature of the excited state. Detailed description and illustrative examples can be found elsewhere.[207,208]

This section describes the theoretical background behind attachment/detachment analysis and natural transition orbitals, while details of the input for creating data suitable for plotting these quantities is described separately in Chapter 10, which also describes additional excited-state analysis tools. For historical reasons, there are duplicate implementations of some features. For example, CIS and TDDFT wave functions can be analyzed using an original built-in code and by using a more recent module, LIBWFA.

7.14.2 Attachment/Detachment Density Analysis

Consider the one-particle density matrices of the initial and final states of interest, $\mathbf{P}_1$ and $\mathbf{P}_2$ respectively. Assuming that each state is represented in a finite basis of spin-orbitals, such as the molecular orbital basis, and each state is at the same geometry. Subtracting these matrices yields the difference density

$$\mathbf{\Delta} = \mathbf{P}_1 - \mathbf{P}_2 \tag{7.164}$$

Now, the eigenvectors of the one-particle density matrix $\mathbf{P}$ describing a single state are termed the natural orbitals, and provide the best orbital description that is possible for the state, in that a CI expansion using the natural orbitals as the single-particle basis is the most compact. The basis of the attachment/detachment analysis is to consider what could be termed natural orbitals of the electronic transition and their occupation numbers (associated eigenvalues). These are defined as the eigenvectors $\mathbf{U}$ defined by

$$\mathbf{U}^\dagger \mathbf{\Delta} \mathbf{U} = \delta \tag{7.165}$$

The sum of the occupation numbers δ_p of these orbitals is then

$$\mathrm{tr}(\mathbf{\Delta}) = \sum_{p=1}^{N} \delta_p = n \tag{7.166}$$

where n is the net gain or loss of electrons in the transition. The net gain in an electronic transition which does not involve ionization or electron attachment will obviously be zero.

The detachment density

$$\mathbf{D} = \mathbf{U}\mathbf{d}\mathbf{U}^\dagger \tag{7.167}$$

is defined as the sum of all natural orbitals of the difference density with negative occupation numbers, weighted by the absolute value of their occupations where $\mathbf{d}$ is a diagonal matrix with elements

$$d_p = -\min(\delta_p, 0) \tag{7.168}$$

The detachment density corresponds to the electron density associated with single particle levels vacated in an electronic transition or hole density.

The attachment density

$$\mathbf{A} = \mathbf{U}\mathbf{a}\mathbf{U}^{\dagger} \tag{7.169}$$

is defined as the sum of all natural orbitals of the difference density with positive occupation numbers where $\mathbf{a}$ is a diagonal matrix with elements

$$a_p = \max(\delta_p, 0) \; . \tag{7.170}$$

The attachment density corresponds to the electron density associated with the single particle levels occupied in the transition or particle density. The difference between the attachment and detachment densities yields the original difference density matrix

$$\mathbf{\Delta} = \mathbf{A} - \mathbf{D} \; . \tag{7.171}$$

Within a CIS or TDDFT calculation, where the transitions are strictly one-electron in nature, the matrices $\mathbf{A}$ and $\mathbf{D}$ are the particle (virtual-virtual) and hole (occupied-occupied) components of the unrelaxed difference density matrix.[106]

7.14.3 Natural Transition Orbitals

In certain situations, even the attachment/detachment densities may be difficult to analyze. An important class of examples are systems with multiple chromophores, which may support exciton states consisting of linear combinations of localized excitations. For such states, both the attachment and the detachment density are highly delocalized and occupy basically the same region of space.[107,147] Lack of phase information makes the attachment/detachment densities difficult to analyze, while strong mixing of the canonical MOs means that excitonic states are also difficult to characterize in terms of MOs.[107]

Analysis of these and other excited states is greatly simplified by constructing Natural Transition Orbitals (NTOs) for the excited states.[106,107] (The basic idea behind NTOs is rather old,[158] and has been rediscovered several times.[169,177] These orbitals were later shown to be equivalent to CIS natural orbitals.[242]) Let $\mathbf{T}$ denote the transition density matrix from an excited-state calculation. The dimension of this matrix is $O \times V$, where O and V denote the number of occupied and virtual MOs, respectively. The NTOs are defined by transformations $\mathbf{U}$ and $\mathbf{V}$ obtained by singular value decomposition (SVD) of the matrix $\mathbf{T}$, *i.e.*,[177]

$$\mathbf{U}\mathbf{T}\mathbf{V}^{\dagger} = \mathbf{\Lambda} \tag{7.172}$$

The matrices $\mathbf{U}$ and $\mathbf{V}$ are unitary and $\mathbf{\Lambda}$ is diagonal, with the latter containing at most O non-zero elements. The matrix $\mathbf{U}$ is a unitary transformation from the canonical occupied MOs to a set of NTOs that together represent the "hole" orbital that is left by the excited electron, while $\mathbf{V}$ transforms the canonical virtual MOs into a set of NTOs representing the excited electron. (Equivalently, the "holes" are the eigenvectors of the $O \times O$ matrix $\mathbf{T}\mathbf{T}^{\dagger}$ and the particles are eigenvectors of the $V \times V$ matrix $\mathbf{T}^{\dagger}\mathbf{T}$.[169]) These "hole" and "particle" NTOs come in pairs, and their relative importance in describing the excitation is governed by the diagonal elements of $\mathbf{\Lambda}$, which are excitation amplitudes in the NTO basis. By virtue of the SVD in Eq. (7.172), any excited state may be represented using at most O excitation amplitudes and corresponding hole/particle NTO pairs. (The‘ discussion here assumes that $V \geq O$, which is typically the case except possibly in minimal basis sets. Although it is possible to use the transpose of Eq. (7.172) to obtain NTOs when $V < O$, this has not been implemented in Q-CHEM due to its limited domain of applicability.)

The SVD generalizes the concept of matrix diagonalization to the case of rectangular matrices, and therefore reduces as much as possible the number of non-zero outer products needed for an exact representation of $\mathbf{T}$. In this sense, the NTOs represent the best possible particle/hole picture of an excited state. The detachment density is recovered as the sum of the squares of the "hole" NTOs, while the attachment density is precisely the sum of the squares of the "particle" NTOs. Unlike the attachment/detachment densities, however, NTOs preserve phase information, which can

be very helpful in characterizing the diabatic character (*e.g.*, $\pi\pi^*$ or $n\pi^*$) of excited states in complex systems. In the limit that there is only one significant pair of NTOs, the squares of these two orbitals ($|\psi_{\text{hole}}(\mathbf{r})|^2$ and $|\psi_{\text{particle}}(\mathbf{r})|^2$) are precisely equivalent to the detachment and attachment densities that were introduced in Section 7.14.2.[106] Even when there is more than one significant NTO amplitude, the NTOs still represent a significant compression of information, as compared to the canonical MO basis.

NTOs are available within Q-CHEM for CIS, RPA, TDDFT, ADC, and EOM-CC methods. For the correlated wave functions (EOM-CC and ADC) and for SF-DFT, they can be computed using LIBWFA module. The simplest way to visualize the NTOs is to generate them in a format suitable for viewing with the freely-available MOLDEN or MACMOLPLT programs, as described in Chapter 10.

References and Further Reading

[1] The VMD program may be downloaded from http://www.ks.uiuc.edu/Research/vmd.

[2] Ground-State Methods (Chapters 4 and 6).

[3] Basis Sets and Effective Core Potentials (Chapter 8).

[4] H. Agren, V. Carravetta, O. Vahtras, and L. G. M. Pettersson. *Theor. Chem. Acc.*, 97:14, 1997. DOI: 10.1007/s002140050234.

[5] B. Alam, H. Jiang, P. M. Zimmerman, and J. M. Herbert. *J. Chem. Phys.*, 156:194110, 2022. DOI: 10.1063/5.0091636.

[6] J. A. Andersen, K. D. Nanda, A. I. Krylov, and S. Coriani. *J. Chem. Theory Comput.*, 18:1748, 2022. DOI: 10.1021/acs.jctc.1c00937.

[7] J. E. Arias-Martinez, H. Wu, and M. Head-Gordon. *J. Chem. Theory Comput.*, 20:752, 2024. DOI: 10.1021/acs.jctc.3c01139.

[8] V. Athavale, H.-H. Teh, and J. E. Subotnik. *J. Chem. Phys.*, 155:154105, 2021. DOI: 10.1063/5.0064269.

[9] V. Athavale, H.-H. Teh, Y. Shao, and J. E. Subotnik. *J. Chem. Phys.*, 157:244110, 2022. DOI: 10.1063/5.0130404.

[10] G. M. J. Barca, A. T. B. Gilbert, and P. M. W. Gill. *J. Chem. Theory Comput.*, 14:1501, 2018. DOI: 10.1021/acs.jctc.7b00994.

[11] A. Barth and L. S. Cederbaum. *Phys. Rev. A*, 12223:1038, 1981. DOI: 10.1103/PhysRevA.23.1038.

[12] J. E. Bates and F. Furche. *J. Chem. Phys.*, 137:164105, 2012. DOI: 10.1063/1.4759080.

[13] F. Bedurke, T. Klamroth, P. Krause, and P. Saalfrank. *J. Chem. Phys.*, 150:234114, 2019. DOI: 10.1063/1.5096473.

[14] F. Bell, D. Casanova, and M. Head-Gordon. *J. Am. Chem. Soc.*, 132:11314, 2010. DOI: 10.1021/ja104772w.

[15] F. Bell, P. M. Zimmerman, D. Casanova, M. Goldey, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 15:358, 2013. DOI: 10.1039/C2CP43293E.

[16] N. Bellonzi, E. Alguire, S. Fatehi, Y. Shao, and J. E. Subotnik. *J. Chem. Phys.*, 152:044112, 2020. DOI: 10.1063/1.5126440.

[17] Z. Benda and T.-C. Jagau. *J. Chem. Phys.*, 146:031101, 2017. DOI: 10.1063/1.4974094.

[18] Y. A. Bernard, Y. Shao, and A. I. Krylov. *J. Chem. Phys.*, 136:204103, 2012. DOI: 10.1063/1.4714499.

[19] N. A. Besley. *Chem. Phys. Lett.*, 390:124, 2004. DOI: 10.1016/j.cplett.2004.04.004.

[20] N. A. Besley. *J. Chem. Phys.*, 122:184706, 2005. DOI: 10.1063/1.1891687.

[21] N. A. Besley. *J. Chem. Theory Comput.*, 12:5018, 2016. DOI: 10.1021/acs.jctc.6b00656.

[22] N. A. Besley. *Acc. Chem. Res.*, 53:1306, 2020. DOI: 10.1021/acs.accounts.0c00171.

[23] N. A. Besley, M. T. Oakley, A. J. Cowan, and J. D. Hirst. *J. Am. Chem. Soc.*, 126:13502, 2004. DOI: 10.1021/ja047603l.

[24] N. A. Besley, A. T. B. Gilbert, and P. M. W. Gill. *J. Chem. Phys.*, 130:124308, 2009. DOI: 10.1063/1.3092928.

[25] N. A. Besley, M. J. G. Peach, and D. J. Tozer. *Phys. Chem. Chem. Phys.*, 11:10350, 2009. DOI: 10.1039/b912718f.

[26] T. D. Bouman and A. E. Hansen. *Int. J. Quantum Chem. Symp.*, 23:381, 1989. DOI: 10.1002/qua.560360842.

[27] K. B. Bravaya, D. Zuev, E. Epifanovsky, and A. I. Krylov. *J. Chem. Phys.*, 138:124106, 2013. DOI: 10.1063/1.4795750.

[28] R. Cammi and B. Mennucci. *J. Chem. Phys.*, 110:9877, 1999. DOI: 10.1063/1.478861.

[29] R. Cammi, S. Corni, B. Mennucci, and J. Tomasi. *J. Chem. Phys.*, 122:104513, 2005. DOI: 10.1063/1.1867373.

[30] M. Caricato, B. Mennucci, J. Tomasi, F. Ingrosso, R. Cammi, S. Corni, and G. Scalmani. *J. Chem. Phys.*, 124: 124520, 2006. DOI: 10.1063/1.2183309.

[31] A. Carreras, H. Jiang, P. Pokhilko, A. I. Krylov, P. M. Zimmerman, and D. Casanova. *J. Chem. Phys.*, 153: 214107, 2020. DOI: 10.1063/5.0029146.

[32] K. Carter-Fenk and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 24:26170, 2022. DOI: 10.1039/D2CP04077H.

[33] K. Carter-Fenk and J. M. Herbert. *J. Chem. Theory Comput.*, 16:5067, 2020. DOI: 10.1021/acs.jctc.0c00502.

[34] K. Carter-Fenk, L. A. Cunha, J. E. Arias-Martinez, and M. Head-Gordon. *J. Phys. Chem. Lett.*, 13:9664, 2022. DOI: 10.1021/acs.jpclett.2c02564.

[35] D. Casanova. *J. Chem. Phys.*, 137:084105, 2012. DOI: 10.1063/1.4747341.

[36] D. Casanova. *J. Comput. Chem.*, 34:720, 2013. DOI: 10.1002/jcc.23188.

[37] D. Casanova. *J. Chem. Phys.*, 140:144111, 2014. DOI: 10.1063/1.4870638.

[38] D. Casanova. *J. Chem. Phys.*, 148:124118, 2018. DOI: 10.1063/1.5018895.

[39] D. Casanova and M. Head-Gordon. *J. Chem. Phys.*, 129:064104, 2008. DOI: 10.1063/1.2965131.

[40] D. Casanova and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 11:9779, 2009. DOI: 10.1039/b911513g.

[41] D. Casanova and A. I. Krylov. *J. Chem. Phys.*, 144:014102, 2016. DOI: 10.1063/1.4939222.

[42] D. Casanova and A. I. Krylov. *Phys. Chem. Chem. Phys.*, 22:4326, 2020. DOI: 10.1039/C9CP06507E.

[43] D. Casanova, Y. M. Rhee, and M. Head-Gordon. *J. Chem. Phys.*, 128:164106, 2008. DOI: 10.1063/1.2907724.

[44] D. Casanova, L. V. Slipchenko, A. I. Krylov, and M. Head-Gordon. *J. Chem. Phys.*, 130:044103, 2009. DOI: 10.1063/1.3066652.

[45] M. E. Casida. Time-dependent density functional response theory for molecules. In D. P. Chong, editor, *Recent Advances in Density Functional Methods, Part I*, pages 155–192. World Scientific, Singapore, 1995. DOI: 10.1142/2914.

[46] M. E. Casida, C. Jamorski, K. C. Casida, and D. R. Salahub. *J. Chem. Phys.*, 108:4439, 1998. DOI: 10.1063/1.475855.

[47] L. S. Cederbaum, W. Domcke, and J. Schirmer. *Phys. Rev. A*, 22:206, 1980. DOI: 10.1103/PhysRevA.22.206.

[48] J.-D. Chai. *J. Chem. Phys.*, 136:154104, 2012. DOI: 10.1063/1.3703894.

[49] O. Christiansen, H. Koch, and P. Jørgensen. *Chem. Phys. Lett.*, 243:409, 1995. DOI: 10.1016/0009-2614(95)00841-Q.

[50] O. Christiansen, A. Halkier, H. Koch, P. Jørgensen, and T. Helgaker. *J. Chem. Phys.*, 108:2801, 1998. DOI: 10.1063/1.475671.

[51] D. Claudino and N. J. Mayhall. *J. Chem. Theory Comput.*, 15:1053, 2019. DOI: 10.1021/acs.jctc.8b01112.

[52] E. Coccia, B. Mussard, M. Labeye, J. Caillat, R. Taïeb, J. Toulouse, and E. Luppi. *Int. J. Quantum Chem.*, 116: 1120, 2016. DOI: 10.1002/qua.25146.

[53] F. Cordova, L. J. Doriol, A. Ipatov, M. E. Casida, C. Filippi, and A. Vela. *J. Chem. Phys.*, 127:164111, 2007. DOI: 10.1063/1.2786997.

[54] S. Coriani and H. Koch. *J. Chem. Phys.*, 143:181103, 2015. DOI: 10.1063/1.4935712.

[55] L. A. Cunha, D. Hait, R. Kang, Y. Mao, and M. Head-Gordon. *J. Phys. Chem. Lett.*, 13:3438, 2022. DOI: 10.1021/acs.jpclett.2c00578.

[56] C. Daul. *Int. J. Quantum Chem.*, 52:867, 1994. DOI: 10.1002/qua.560520414.

[57] J. E. Del Bene, R. Ditchfield, and J. A. Pople. *J. Chem. Phys.*, 55:2236, 1971. DOI: 10.1063/1.1676398.

[58] A. L. Dempwolff, M. Schneider, M. Hodecker, and A. Dreuw. *J. Chem. Phys.*, 150:064108, 2019. DOI: 10.1063/1.5081674.

[59] A. L. Dempwolff, A. C. Paul, A. M. Belogolova, A. B. Trofimov, and A. Dreuw. *J. Chem. Phys.*, 152:024113, 2020. DOI: 10.1063/1.5137792.

[60] A. L. Dempwolff, A. C. Paul, A. M. Belogolova, A. B. Trofimov, and A. Dreuw. *J. Chem. Phys.*, 152:024125, 2020. DOI: 10.1063/1.5137794.

[61] A. L. Dempwolff, A. M. Belogolova, T. Sommerfeld, A. B. Trofimov, and A. Dreuw. *J. Chem. Phys.*, 155: 054103, 2021. DOI: 10.1063/5.0057737.

[62] A. L. Dempwolff, A. M. Belogolova, A. B. Trofimov, and A. Dreuw. *J. Chem. Phys.*, 154:104117, 2021. DOI: 10.1063/5.0043337.

[63] A. Dreuw and M. Head-Gordon. *Chem. Rev.*, 105:4009, 2005. DOI: 10.1021/cr0505627.

[64] E. Epifanovsky, D. Zuev, X. Feng, K. Khistyaev, Y. Shao, and A. I. Krylov. *J. Chem. Phys.*, 139:134105, 2013. DOI: 10.1063/1.4820484.

[65] E. Epifanovsky, K. Klein, S. Stopkowicz, J. Gauss, and A. I. Krylov. *J. Chem. Phys.*, 143:064102, 2015. DOI: 10.1063/1.4927785.

[66] U. Fano. *Phys. Rev.*, 124:1866, 1961. DOI: 10.1103/physrev.124.1866.

[67] S. Faraji, S. Matsika, and A. I. Krylov. *J. Chem. Phys.*, 148:044103, 2018. DOI: 10.1063/1.5009433.

[68] S. Fatehi, E. Alguire, Y. Shao, and J. E. Subotnik. *J. Chem. Phys.*, 135:234105, 2011. DOI: 10.1063/1.3665031.

[69] D. G. Fedorov, S. Koseki, M. W. Schmidt, and M. S. Gordon. *Int. Rev. Phys. Chem.*, 22:551, 2003. DOI: 10.1080/0144235032000101743.

[70] H. Feshbach. *Ann. Phys.*, 19:287, 1962. DOI: 10.1016/0003-4916(62)90221-X.

[71] M. Filatov and S. Shaik. *Chem. Phys. Lett.*, 304:429, 1999. DOI: 10.1016/S0009-2614(99)00336-X.

[72] J. B. Foresman, M. Head-Gordon, J. A. Pople, and M. J. Frisch. *J. Phys. Chem.*, 96:135, 1992. DOI: 10.1021/j100180a030.

[73] A. E. A. Fouda and N. A. Besley. *Theor. Chem. Acc.*, 137:6, 2018. DOI: 10.1007/s00214-017-2181-0.

[74] A. E. A. Fouda and N. A. Besley. *J. Comput. Chem.*, 41:1081, 2020. DOI: 10.1002/jcc.26153.

[75] T. Fransson, I. E. Brumboiu, M. L. Vidal, P. Norman, S. Coriani, and A. Dreuw. *J. Chem. Theory Comput.*, 17: 1618, 2021. DOI: 10.1021/acs.jctc.0c01082.

[76] J. Friedrichs, K. Damianos, and I. Frank. *Chem. Phys.*, 347:17, 2008. DOI: 10.1016/j.chemphys.2007.09.035.

[77] T. Froitzheim, L. Kunze, S. Grimme, J. M. Herbert, and J.-M. Mewes. *J. Phys. Chem. A*, 128:6324, 2024. DOI: 10.1021/acs.jpca.4c03273.

[78] F. Furche. *J. Chem. Phys.*, 114:5982, 2001. DOI: 10.1063/1.1353585.

[79] A. T. B. Gilbert, N. A. Besley, and P. M. W. Gill. *J. Phys. Chem. A*, 112:13164, 2008. DOI: 10.1021/jp801738f.

[80] C. M. Gittins, E. A. Rohlfing, and C. M. Rohlfing. *J. Chem. Phys.*, 105:7323, 1996. DOI: 10.1063/1.472591.

[81] A. A. Golubeva, P. A. Pieniazek, and A. I. Krylov. *J. Chem. Phys.*, 130:124113, 2009. DOI: 10.1063/1.3098949.

[82] S. Gozem and A. I. Krylov. ezDyson User Manual, 2015. http://iopenshell.usc.edu/downloads.

[83] S. Gozem and A. I. Krylov. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 12:e1546, 2022. DOI: 10.1002/wcms.1546.

[84] S. Grimme. *Chem. Phys. Lett.*, 259:128, 1996. DOI: 10.1016/0009-2614(96)00722-1.

[85] E. K. U. Gross and W. Kohn. *Adv. Quantum Chem.*, 21:255, 1990. DOI: 10.1016/S0065-3276(08)60600-0.

[86] E. K. U. Gross and N. T. Maitra. Introduction to TDDFT. In M. A. L. Marques, N. T. Maitra, F. M. S. Nogueira, E. K. U. Gross, and A. Rubio, editors, *Fundamentals of Time-Dependent Density Functional Theory*, volume 837 of *Lecture Notes in Physics*, chapter 1, pages 53–97. Springer-Verlag, 2012. DOI: 10.1007/978-3-642-23518-4_4.

[87] E. K. U. Gross, C. A. Ullrich, and U. J. Gossmann. Density functional theory of time-dependent systems. In E. K. U. Gross and R. M. Dreizler, editors, *Density Functional Theory*, pages 149–171. Plenum Press, New York, 1995. DOI: 10.1007/978-1-4757-9975-0_7.

[88] C. A. Guido, P. Cortona, B. Mennucci, and C. Adamo. *J. Chem. Theory Comput.*, 9:3118, 2013. DOI: 10.1021/ct400337e.

[89] C. A. Guido, P. Cortona, and C. Adamo. *J. Chem. Phys.*, 140:104101, 2014. DOI: 10.1063/1.4867007.

[90] D. Hait and M. Head-Gordon. *J. Phys. Chem. Lett.*, 11:775, 2020. DOI: 10.1021/acs.jpclett.9b03661.

[91] D. Hait and M. Head-Gordon. *J. Chem. Theory Comput.*, 16:1699, 2020. DOI: 10.1021/acs.jctc.9b01127.

[92] D. Hait, T. Zhu, D. P. McMahon, and T. Van Voorhis. *J. Chem. Theory Comput.*, 12:3353, 2016. DOI: 10.1021/acs.jctc.6b00426.

[93] A. E. Hansen, B. Voight, and S. Rettrup. *Int. J. Quantum Chem.*, 23:595, 1983. DOI: 10.1002/qua.560230230.

[94] M. W. D. Hanson-Heine, M. W. George, and N. A. Besley. *J. Chem. Phys.*, 138:064101, 2013. DOI: 10.1063/1.4789813.

[95] M. W. D. Hanson-Heine, M. W. George, and N. A. Besley. *J. Chem. Phys.*, 146:094106, 2017. DOI: 10.1063/1.4977178.

[96] M. W. D. Hanson-Heine, M. W. George, and N. A. Besley. *Chem. Phys. Lett.*, 696:119, 2018. DOI: 10.1016/j.cplett.2018.02.028.

[97] M. W. D. Hanson-Heine, M. W. George, and N. A. Besley. *Chem. Phys. Lett.*, 699:279, 2018. DOI: 10.1016/j.cplett.2018.03.066.

[98] P. H. Harbach, M. Wormit, and A. Dreuw. *J. Chem. Phys.*, 141:064113, 2014. DOI: 10.1007/BF01113068.

[99] M. Head-Gordon, R. J. Rico, M. Oumi, and T. J. Lee. *Chem. Phys. Lett.*, 219:21, 1994. DOI: 10.1016/0009-2614(94)00070-0.

[100] M. Head-Gordon, A. M. Graña, D. Maurice, and C. A. White. *J. Phys. Chem.*, 99:14261, 1995. DOI: 10.1021/j100039a012.

[101] M. Head-Gordon, D. Maurice, and M. Oumi. *Chem. Phys. Lett.*, 246:114, 1995. DOI: 10.1016/0009-2614(95)01111-L.

[102] M. Head-Gordon, M. Oumi, and D. Maurice. *Mol. Phys.*, 96:593, 1999. DOI: 10.1080/00268979909482996.

[103] A. Hellweg, S. A. Grün, and C. Hättig. *Phys. Chem. Chem. Phys.*, 10:4119, 2008. DOI: 10.1039/b803727b.

[104] J. M. Herbert. The quantum chemistry of loosely-bound electrons. In A. L. Parill and K. Lipkowitz, editors, *Reviews in Computational Chemistry*, volume 28, chapter 8, pages 391–517. Wiley, 2015. DOI: 10.1002/9781118889886.ch8.

[105] J. M. Herbert. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 11:e1519, 2021. DOI: 10.1002/wcms.1519.

[106] J. M. Herbert. Density-functional theory for electronic excited states. In C. García-Iriepa and M. Marazzi, editors, *Theoretical and Computational Photochemistry: Fundamentals, Methods, Applications and Synergy with Experimental Approaches*, chapter 3, pages 69–118. Elsevier, 2023. DOI: 10.1016/B978-0-323-91738-4.00005-1.

[107] J. M. Herbert. *Phys. Chem. Chem. Phys.*, 26:3755, 2024. DOI: 10.1039/D3CP04226J.

[108] J. M. Herbert and A. Mandal. Spin-flip TDDFT for photochemistry. In C. Zhu, editor, *Time-Dependent Density Functional Theory: Nonadiabatic Dynamics*, chapter 10, pages 361–404. Jenny Sanford, 2023. DOI: 10.1201/9781003319214-10.

[109] J. M. Herbert and A. Mandal. *J. Chem. Theory Comput.*, 20:9446, 2024. DOI: 10.1021/acs.jctc.4c01085.

[110] J. M. Herbert, X. Zhang, A. F. Morrison, and J. Liu. *Acc. Chem. Res.*, 49:931, 2016. DOI: 10.1021/acs.accounts.6b00047.

[111] J. M. Herbert, Y. Zhu, B. Alam, and A. K. Ojha. *J. Chem. Theory Comput.*, 19:6745, 2023. DOI: 10.1021/acs.jctc.3c00673.

[112] M. F. Herbst and T. Fransson. *J. Chem. Phys.*, 153:054114, 2020. DOI: 10.1063/5.0013538.

[113] K. Hirao and H. Nakatsuji. *J. Chem. Phys.*, 59:1457, 1973. DOI: 10.1063/1.1680203.

[114] S. Hirata and M. Head-Gordon. *Chem. Phys. Lett.*, 314:291, 1999. DOI: 10.1016/S0009-2614(99)01149-5.

[115] S. Hirata, M. Nooijen, and R. J. Bartlett. *Chem. Phys. Lett.*, 326:255, 2000. DOI: 10.1016/S0009-2614(00)00772-7.

[116] C. E. Hoyer, X. Xu, D. Ma, L. Gagliardi, and D. G. Truhlar. *J. Chem. Phys.*, 141:114104, 2014. DOI: 10.1039/C6CP03784D.

[117] M. Huix-Rotllant, B. Natarajan, A. Ipatov, C. M. Wawire, T. Deutsch, and M. E. Casida. *Phys. Chem. Chem. Phys.*, 12:12811, 2010. DOI: 10.1039/c0cp00273a.

[118] W. Humphrey, A. Dalke, and K. Schulten. *J. Molec. Graphics*, 14:33, 1996. DOI: 10.1016/0263-7855(96)00018-5.

[119] W. J. Hunt and W. A. Goddard III. *Chem. Phys. Lett.*, 3:414, 1969. DOI: 10.1016/0009-2614(69)80154-5.

[120] R. Improta, G. Scalmani, M. J. Frisch, and V. Barone. *J. Chem. Phys.*, 127:074504, 2007. DOI: 10.1063/1.2757168.

[121] J. R. McClean et. al. *Quantum Science and Technology*, 5:034014, 2020. DOI: 10.1088/2058-9565/ab8ebc.

[122] T.-C. Jagau, D. Zuev, K. B. Bravaya, E. Epifanovsky, and A. I. Krylov. *J. Phys. Chem. Lett.*, 5:310, 2014. DOI: 10.1021/jz402482a.

[123] T.-C. Jagau, K. B. Bravaya, and A. I. Krylov. *Annu. Rev. Phys. Chem.*, 68:525, 2017. DOI: 10.1146/annurev-physchem-052516-050622.

[124] S. Jana and J. M. Herbert. *J. Chem. Phys.*, 158:094111, 2023. DOI: 10.1063/5.0134459.

[125] S. Jana and J. M. Herbert. *J. Chem. Theory Comput.*, 19:4100, 2023. DOI: 10.1021/acs.jctc.3c00202.

[126] N. K. Jayadev, W. Skomorowski, and A. I. Krylov. *J. Phys. Chem. Lett.*, 14:8612, 2023. DOI: 10.1021/acs.jpclett.3c01966.

[127] S. Kähler, A. Cebreiro, P. Pokhilko, D. Casanova, and A. I. Krylov. *J. Phys. Chem. A*, 127:8459, 2023. DOI: 10.1021/acs.jpca.3c04134.

[128] J. Kauczor, P. Norman, O. Christiansen, and S. Coriani. *J. Chem. Phys.*, 139:211102, 2013. DOI: 10.1063/1.4840275.

[129] K. Kaufmann, W. Baumeister, and M. Jungen. *J. Phys. B: At. Mol. Opt. Phys.*, 22:2223, 1989. DOI: 10.1088/0953-4075/22/14/007.

[130] H. Koch and P. Jørgensen. *J. Chem. Phys.*, 93:3333, 1990. DOI: 10.1063/1.458814.

[131] H. Koch, H. J. A. Jensen, P. Jørgensen, and T. Helgaker. *J. Chem. Phys.*, 93:3345, 1990. DOI: 10.1063/1.458815.

[132] F. Kossoski and P-F. Loos. *J. Chem. Theory Comput.*, 19:8654, 2023. DOI: 10.1021/acs.jctc.3c00946.

[133] S. Kotaru, P. Pokhilko, and A. I. Krylov. *J. Chem. Phys.*, 157:224110, 2022. DOI: 10.1063/5.0130868.

[134] T. Kowalczyk, T. Tsuchimochi, L. Top, P.-T. Chen, and T. Van Voorhis. *J. Chem. Phys.*, 138:164101, 2013. DOI: 10.1063/1.4801790.

[135] P. Krause and H. B. Schlegel. *J. Chem. Phys.*, 141:174104, 2014. DOI: 10.1063/1.4900576.

[136] C. M. Krauter, M. Pernpointner, and A. Dreuw. *J. Chem. Phys.*, 138:044107, 2013. DOI: 10.1063/1.4776675.

[137] A. I. Krylov. *Chem. Phys. Lett.*, 338:375, 2001. DOI: 10.1016/S0009-2614(01)00287-1.

[138] A. I. Krylov. *Chem. Phys. Lett.*, 350:522, 2002. DOI: 10.1016/S0009-2614(01)01316-1.

[139] A. I. Krylov. *Acc. Chem. Res.*, 39:83, 2006. DOI: 10.1021/ar0402006.

[140] A. I. Krylov. *Annu. Rev. Phys. Chem.*, 59:433, 2008. DOI: 10.1146/annurev.physchem.59.032607.093602.

[141] A. I. Krylov. The quantum chemistry of open-shell species. In A. L. Parrill and K. B. Lipkowitz, editors, *Reviews in Computational Chemistry*, volume 30, chapter 4, pages 151–224. John Wiley & Sons, Inc, 2017. DOI: 10.1002/9781119356059.ch4.

[142] A. I. Krylov, C. D. Sherrill, and M. Head-Gordon. *J. Chem. Phys.*, 113:6509, 2000. DOI: 10.1063/1.1311292.

[143] T. Kuś and A. I. Krylov. *J. Chem. Phys.*, 135:084109, 2011. DOI: 10.1063/1.3626149.

[144] T. Kuś and A. I. Krylov. *J. Chem. Phys.*, 136:244109, 2012. DOI: 10.1063/1.4730296.

[145] A. Landau, K. Khistyaev, S. Dolgikh, and A. I. Krylov. *J. Chem. Phys.*, 132:014109, 2010. DOI: 10.1063/1.3276630.

[146] A. Lange and J. M. Herbert. *J. Chem. Theory Comput.*, 3:1680, 2007. DOI: 10.1021/ct700125v.

[147] A. W. Lange and J. M. Herbert. *J. Am. Chem. Soc.*, 131:124115, 2009. DOI: 10.1021/ja808998q.

[148] A. D. Laurent and D. Jacquemin. *Int. J. Quantum Chem.*, 113:2019, 2013. DOI: 10.1002/qua.24438.

[149] D. Lefrancois, M. Wormit, and A. Dreuw. *J. Chem. Phys.*, 143:124107, 2015. DOI: 10.1063/1.4931653.

[150] S. V. Levchenko and A. I. Krylov. *J. Chem. Phys.*, 120:175, 2004. DOI: 10.1063/1.1630018.

[151] S. V. Levchenko, T. Wang, and A. I. Krylov. *J. Chem. Phys.*, 122:224106, 2005. DOI: 10.1063/1.1877072.

[152] X. Li, S. M. Smith, A. N. Markevitch, D. A. Romanov, R. J. Lewis, and H. B. Schlegel. *Phys. Chem. Chem. Phys.*, 7:233, 2005. DOI: 10.1039/B415849K.

[153] X. Li, N. Govind, C. Isborn, A. E. DePrince III, and K. Lopata. *Chem. Rev.*, 120:9951, 2020. DOI: 10.1021/acs.chemrev.0c00223.

[154] F. Liu, Z. Gan, Y. Shao, C.-P. Hsu, A. Dreuw, M. Head-Gordon, B. T. Miller, B. R. Brooks, J.-G. Yu, T. R. Furlani, and J. Kong. *Mol. Phys.*, 108:2791, 2010. DOI: 10.1080/00268976.2010.526642.

[155] J. Liu and W. Liang. *J. Chem. Phys.*, 135:014113, 2011. DOI: 10.1063/1.3605504.

[156] J. Liu and W. Liang. *J. Chem. Phys.*, 135:184111, 2011. DOI: 10.1063/1.3659312.

[157] J. Liu and W. Liang. *J. Chem. Phys.*, 138:024101, 2013. DOI: 10.1063/1.4773397.

[158] A. V. Luzanov, A. A. Sukhorukov, and V. E. Umanskii. *Theor. Exp. Chem.*, 10:354, 1976. DOI: 10.1007/BF00526670.

[159] A. V. Luzanov, D. Casanova, X. Feng, and A. I. Krylov. *J. Chem. Phys.*, 142:224104, 2015. DOI: 10.1063/1.4921635.

[160] F. R. Manby, M. Stella, J. D. Goodpaster, and T. F. Miller III. *J. Chem. Theory Comput.*, 8:2564, 2012. DOI: 10.1021/ct300544e.

[161] A. Mandal. pySETSOC: a code for TDDFT + SOC calculations based on Q-CHEM output, 2025. https://gitlab.com/john-herbert-group/pysetsoc.

[162] A. Mandal and J. M. Herbert. Computing L- and M-edge spectra using the DFT/CIS method with spin-orbit coupling. *ChemRxiv*, 2025. DOI: 10.26434/chemrxiv-2025-zfvnh.

[163] A. Mandal, E. J. Berquist, and J. M. Herbert. *J. Chem. Phys.*, 161:044114, 2024. DOI: 10.1063/5.0220535.

[164] Manisha and P. U. Manohar. *Phys. Chem. Chem. Phys.*, 26:21204, 2024. DOI: 10.1039/D4CP02265C.

[165] P. U. M. Manisha and A. I. Krylov. Equation-of-motion coupled-cluster singles, doubles and (full) triples for doubly ionized and two-electron-attached states: A computational implementation. *arRxiv*, 2025. DOI: 10.48550/arXiv.2501.07159.

[166] P. U. Manohar and A. I. Krylov. *J. Chem. Phys.*, 129:194105, 2008. DOI: 10.1063/1.3013087.

[167] P. U. Manohar, J. F. Stanton, and A. I. Krylov. *J. Chem. Phys.*, 131:114112, 2009. DOI: 10.1063/1.3231133.

[168] A. V. Marenich, C. J. Cramer, D. G. Truhlar, C. A. Guido, B. Mennucci, G. Scalmani, and M. J. Frisch. *Chem. Sci.*, 2:2143, 2011. DOI: 10.1039/c1sc00313e.

[169] R. L. Martin. *J. Chem. Phys.*, 118:4775, 2003. DOI: 10.1063/1.1558471.

[170] S. Matsika and P. Krause. *Annu. Rev. Phys. Chem.*, 62:621, 2011. DOI: 10.1146/annurev-physchem-032210-103450.

[171] F. Matz and T.-C. Jagau. *J. Chem. Phys.*, 156:114117, 2022. DOI: 10.1063/5.0075646.

[172] F. Matz and T.-C. Jagau. *Mol. Phys.*, 121:e2105270, 2023. DOI: 10.1080/00268976.2022.2105270.

[173] D. Maurice. *Single Electron Theories of Excited States*. PhD thesis, University of California, Berkeley, CA, 1998.

[174] D. Maurice and M. Head-Gordon. *Int. J. Quantum Chem.*, 29:361, 1995. DOI: 10.1002/qua.560560840.

[175] D. Maurice and M. Head-Gordon. *J. Phys. Chem.*, 100:6131, 1996. DOI: 10.1021/jp952754j.

[176] D. Maurice and M. Head-Gordon. *Mol. Phys.*, 96:1533, 1999. DOI: 10.1080/00268979909483096.

[177] I. Mayer. *Chem. Phys. Lett.*, 437:284, 2007. DOI: 10.1016/j.cplett.2007.02.038.

[178] I. Mayer and P.-O Löwdin. *Chem. Phys. Lett.*, 202:1, 1993. DOI: 10.1016/0009-2614(93)85341-K.

[179] N. J. Mayhall and M. Head-Gordon. *J. Chem. Phys.*, 141:044112, 2014. DOI: 10.1063/1.4889918.

[180] N. J. Mayhall and M. Head-Gordon. *J. Phys. Chem. Lett.*, 6:1982, 2015. DOI: 10.1021/acs.jpclett.5b00733.

[181] J. L. McHale. *Molecular Spectroscopy*. Prentice Hall, New York, 1999.

[182] J.-M. Mewes, Z.-Q. You, M. Wormit, T. Kriesche, J. M. Herbert, and A. Dreuw. *J. Phys. Chem. A*, 119:5446, 2015. DOI: 10.1021/jp511163y.

[183] J.-M. Mewes, J. M. Herbert, and A. Dreuw. *Phys. Chem. Chem. Phys.*, 19:1644, 2017. DOI: 10.1039/C6CP05986D.

[184] A. F. Morrison and J. M. Herbert. *J. Phys. Chem. Lett.*, 6:4390, 2015. DOI: 10.1021/acs.jpclett.5b02109.

[185] K. Nanda and A. I. Krylov. *J. Chem. Phys.*, 145:204116, 2016. DOI: 10.1063/1.4967860.

[186] K. D. Nanda and A. I. Krylov. *J. Chem. Phys.*, 142:064118, 2015. DOI: 10.1063/1.4907715.

[187] K. D. Nanda and A. I. Krylov. *J. Phys. Chem. Lett.*, 8:3256, 2017. DOI: 10.1021/acs.jpclett.7b01422.

[188] K. D. Nanda and A. I. Krylov. *J. Chem. Phys.*, 152:244118, 2020. DOI: 10.1063/5.0010295.

[189] K. D. Nanda and A. I. Krylov. *J. Chem. Phys.*, 154:184109, 2021. DOI: 10.1063/5.0049184.

[190] K. D. Nanda, A. I. Krylov, and J. Gauss. *J. Chem. Phys.*, 149:141101, 2018. DOI: 10.1063/1.5053727.

[191] K. D. Nanda, M. L. Vidal, R. Faber, S. Coriani, and A. I. Krylov. *Phys. Chem. Chem. Phys.*, 22:2629, 2019. DOI: 10.1039/C9CP03688A.

[192] M. Nooijen and R. J. Bartlett. *J. Chem. Phys.*, 102:3629, 1995. DOI: 10.1063/1.468592.

[193] C. M. Oana and A. I. Krylov. *J. Chem. Phys.*, 127:234106, 2007. DOI: 10.1063/1.2805393.

[194] C. M. Oana and A. I. Krylov. *J. Chem. Phys.*, 131:124114, 2009. DOI: 10.1063/1.3231143.

[195] K. J. Oosterbaan, A. F. White, and M. Head-Gordon. *J. Chem. Phys.*, 149:044116, 2018. DOI: 10.1063/1.5023051.

[196] K. J. Oosterbaan, A. F. White, and M. Head-Gordon. *J. Chem. Theory Comput.*, 15:2966, 2019. DOI: 10.1021/acs.jctc.8b01259.

[197] K. J. Oosterbaan, A. F. White, D. Hait, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 22:8182, 2020. DOI: 10.1039/C9CP06592J.

[198] Q. Ou, G. D. Bellchambers, F. Furche, and J. E. Subotnik. *J. Chem. Phys.*, 142:064114, 2015. DOI: 10.1063/1.4906941.

[199] M. Oumi, D. Maurice, T. J. Lee, and M. Head-Gordon. *Chem. Phys. Lett.*, 279:151, 1997. DOI: 10.1016/S0009-2614(97)01028-2.

[200] V. Parravicini and T. C. Jagau. *Mol. Phys.*, 119:e1943029, 2021. DOI: 10.1080/00268976.2021.1943029.

[201] M. J. G. Peach, P. Benfield, T. Helgaker, and D. J. Tozer. *J. Chem. Phys.*, 128:044118, 2008. DOI: 10.1063/1.2831900.

[202] M. Pecul and K. Ruud. *Adv. Quantum Chem.*, 50:185, 2005. DOI: 10.1016/S0065-3276(05)50009-1.

[203] T. B. Pedersen, H. Koch, L. Boman, and A. M. J. Sánchez de Merás. *Chem. Phys. Lett.*, 393:319, 2004. DOI: 10.1016/j.cplett.2004.06.065.

[204] A. Perera, R. W. Molt, V. F. Lotrich, and R. J. Bartlet. *Theor. Chem. Acc.*, 133:1514, 2014. DOI: 10.1007/s00214-014-1514-5.

[205] P. Piecuch and M. Włoch. *J. Chem. Phys.*, 123:224105, 2005. DOI: 10.1063/1.2137318.

[206] P. A. Pieniazek, S. E. Bradforth, and A. I. Krylov. *J. Chem. Phys.*, 129:074104, 2008. DOI: 10.1063/1.2969107.

[207] F. Plasser, S. A. Bäppler, M. Wormit, and A. Dreuw. *J. Chem. Phys.*, 141:024107, 2014. DOI: 10.1063/1.4885820.

[208] F. Plasser, M. Wormit, and A. Dreuw. *J. Chem. Phys.*, 141:024106, 2014. DOI: 10.1063/1.4885819.

[209] P. Pokhilko and A. I. Krylov. *J. Phys. Chem. Lett.*, 10:4857–4862, 2019. DOI: 10.1021/acs.jpclett.9b02120.

[210] P. Pokhilko, E. Epifanovsky, and A. I. Krylov. *J. Chem. Theory Comput.*, 14:4088, 2018. DOI: 10.1021/acs.jctc.8b00321.

[211] P. Pokhilko, E. Epifanovsky, and A. I. Krylov. *J. Chem. Phys.*, 151:034106, 2019. DOI: 10.1063/1.5108762.

[212] P. Pokhilko, R. Shannon, D. Glowacki, H. Wang, and A. I. Krylov. *J. Phys. Chem. A*, 123:482, 2019. DOI: 10.1021/acs.jpca.8b10225.

[213] P. Pokhilko, D. Izmodenov, and A. I. Krylov. *J. Chem. Phys.*, 152:034105, 2020. DOI: 10.1063/1.5138643.

[214] S. Prager, A. Zech, F. Aquilante, A. Dreuw, and T. A. Wesolowski. *J. Chem. Phys.*, 144:204103, 2016. DOI: 10.1063/1.4948741.

[215] M. R. Provorse and C. M. Isborn. *Int. J. Quantum Chem.*, 116:739, 2016. DOI: 10.1002/qua.25096.

[216] Y. M. Rhee and M. Head-Gordon. *J. Phys. Chem. A*, 111:5314, 2007. DOI: 10.1021/jp068409j.

[217] U. V. Riss and H.-D. Meyer. *J. Phys. B: At. Mol. Opt. Phys.*, 26:4503, 1993. DOI: 10.1088/0953-4075/26/23/021.

[218] D. M. Rogers, N. A. Besley, P. O'Shea, and J. D. Hirst. *J. Phys. Chem. B*, 109:23061, 2005. DOI: 10.1021/jp053309j.

[219] A. Sadybekov and A. I. Krylov. *J. Chem. Phys.*, 147:014107, 2017. DOI: 10.1063/1.4990564.

[220] R. Santra and L. S. Cederbaum. *J. Chem. Phys.*, 117:5511, 2002. DOI: 10.1063/1.1501903.

[221] R. Santra and L. S. Cederbaum. *Phys. Rep.*, 368:1, 2002. DOI: 10.1016/S0370-1573(02)00143-6.

[222] M. Savarese, C. A. Guido, E. Brémond, I. Ciofini, and C. Adamo. *J. Phys. Chem. A*, 121:7543, 2017. DOI: 10.1021/acs.jpca.7b07080.

[223] J. Schirmer. *Phys. Rev. A*, 26:2395, 1982. DOI: 10.1103/PhysRevA.26.2395.

[224] J. Schirmer and A. B. Trofimov. *J. Chem. Phys.*, 120:11449, 2004. DOI: 10.1063/1.1752875.

[225] J. Schirmer, A. B. Trofimov, and G. Stelter. *J. Chem. Phys.*, 109:4734, 1998. DOI: 10.1063/1.477085.

[226] H. Sekino and R. J. Bartlett. *Int. J. Quantum Chem. Symp.*, 18:255, 1984. DOI: 10.1002/qua.560260826.

[227] M. Seth, G. Mazur, and T. Ziegler. *Theor. Chem. Acc.*, 129:331, 2011. DOI: 10.1007/s00214-010-0819-2.

[228] Y. Shao, M. Head-Gordon, and A. I. Krylov. *J. Chem. Phys.*, 118:4807, 2003. DOI: 10.1063/1.1545679.

[229] D. Sinha, D. Mukhopadhya, R. Chaudhuri, and D. Mukherjee. *Chem. Phys. Lett.*, 154:544, 1989. DOI: 10.1016/0009-2614(89)87149-0.

[230] W. Skomorowski and A. I. Krylov. *J. Chem. Phys.*, 154:084124, 2021. DOI: 10.1063/5.0036976.

[231] W. Skomorowski and A. I. Krylov. *J. Chem. Phys.*, 154:084125, 2021. DOI: 10.1063/5.0036977.

[232] J. C. Slater. *Adv. Quantum Chem.*, 6:1, 1972. DOI: 10.1016/S0065-3276(08)60541-9.

[233] T. Sommerfeld and M. Ehara. *J. Chem. Theory Comput.*, 11:4627, 2015. DOI: 10.1021/acs.jctc.5b00465.

[234] T. Sommerfeld and R. Santra. *Int. J. Quantum Chem.*, 82:218–226, 2001. DOI: 10.1002/qua.1042.

[235] J. F. Stanton and R. J. Bartlett. *J. Chem. Phys.*, 98:7029, 1993. DOI: 10.1063/1.464746.

[236] J. F. Stanton and J. Gauss. *J. Chem. Phys.*, 101:8938, 1994. DOI: 10.1063/1.468022.

[237] J. F. Stanton and J. Gauss. *J. Chem. Phys.*, 103:1064, 1995. DOI: 10.1063/1.469817.

[238] J. F. Stanton, J. Gauss, N. Ishikawa, and M. Head-Gordon. *J. Chem. Phys.*, 103:4160, 1995. DOI: 10.1063/1.469601.

[239] M. Stener, A. Lisini, and P. Decleva. *Chem. Phys.*, 191:141, 1995. DOI: 10.1016/0301-0104(94)00347-D.

[240] J. E. Subotnik, S. Yeganeh, R. J. Cave, and M. A. Ratner. *J. Chem. Phys.*, 129:244101, 2008. DOI: 10.1063/1.3042233.

[241] J. E. Subotnik, R. J. Cave, R. P. Steele, and N. Shenvi. *J. Chem. Phys.*, 130:234102, 2009. DOI: 10.1063/1.3148777.

[242] P. R. Surján. *Chem. Phys. Lett.*, 439:393, 2007. DOI: 10.1016/j.cplett.2007.03.094.

[243] A. Tajti and P. G. Szalay. *J. Chem. Phys.*, 131:124104, 2009. DOI: 10.1063/1.3232011.

[244] H.-H. Teh and J. E. Subotnik. *J. Phys. Chem. Lett.*, 10:3426, 2019. DOI: 10.1021/acs.jpclett.9b00981.

[245] H.-H. Teh and J. E. Subotnik. *J. Chem. Phys.*, 153, 2020. DOI: 10.1063/5.0018441.

[246] A. J. W. Thom and M. Head-Gordon. *J. Chem. Phys.*, 131:124113, 2009. DOI: 10.1063/1.3236841.

[247] D. J. Tozer and N. C. Handy. *J. Chem. Phys.*, 109:10180, 1998. DOI: 10.1063/1.477711.

[248] L. Triguero, L. G. M. Pettersson, and H. Ågren. *Phys. Rev. B*, 58:8097, 1998. DOI: 10.1103/PhysRevB.58.8097.

[249] A. B. Trofimov and J. Schirmer. *J. Phys. B: At. Mol. Opt. Phys.*, 28:2299, 1995. DOI: 10.1088/0953-4075/28/12/003.

[250] A. B. Trofimov and J. Schirmer. *J. Chem. Phys.*, 123:144115, 2005. DOI: 10.1063/1.2047550.

[251] A. B. Trofimov, G. Stelter, and J. Schirmer. *J. Chem. Phys.*, 111:9982, 1999. DOI: 10.1063/1.480352.

[252] A. B. Trofimov, G. Stelter, and J. Schirmer. *J. Chem. Phys.*, 117:6402, 2002. DOI: 10.1063/1.1504708.

[253] H-Y. Tsai and J.-D. Chai. *Molecules*, 28:7247, 2023. DOI: 10.3390/molecules28217247.

[254] C. A. Ullrich. Oxford University Press, New York, 2011. DOI: 10.1093/acprof:oso/9780199563029.003.0005.

[255] M. Hernandez Vera and T.-C. Jagau. *J. Chem. Phys.*, 151:111101, 2019. DOI: 10.1063/1.5119695.

[256] M. L. Vidal. L-edge program for post-processing q-chem output, 2021. https://gitlab.com/malop/l-edge.

[257] M. L. Vidal, X. Feng, E. Epifanovsky, A. I. Krylov, and S. Coriani. *J. Chem. Theory Comput.*, 15:3117, 2019. DOI: 10.1021/acs.jctc.9b00039.

[258] M. L. Vidal, A. I. Krylov, and S. Coriani. *Phys. Chem. Chem. Phys.*, 22:2693, 2019. DOI: 10.1039/C9CP03695D.

[259] M. L. Vidal, P. Pokhilko, A. I. Krylov, and S. Coriani. *J. Phys. Chem. Lett.*, 11:8314, 2020. DOI: 10.1021/acs.jpclett.0c02027.

[260] J. D. Wadey and N. A. Besley. *J. Chem. Theory Comput.*, 10:4557, 2014. DOI: 10.1021/ct500566k.

[261] F. Wang and T. Ziegler. *J. Chem. Phys.*, 121:12191, 2004. DOI: 10.1063/1.1821494.

[262] J. Wenzel, M. Wormit, and A. Dreuw. *J. Comput. Chem.*, 35:1900, 2014. DOI: 10.1002/jcc.23703.

[263] J. Wenzel, M. Wormit, and A. Dreuw. *J. Chem. Theory Comput.*, 10:4583, 2014. DOI: 10.1021/ct5006888.

[264] J. Wenzel, A. Holzer, M. Wormit, and A. Dreuw. *J. Chem. Phys.*, 142:214104, 2015. DOI: 10.1063/1.4921841.

[265] T. A. Wesolowski. *Phys. Rev. A*, 77:012504, 2008. DOI: 10.1103/PhysRevA.77.012504.

[266] T. A. Wesolowski and A. Warshel. *J. Phys. Chem.*, 97:8050, 1993. DOI: 10.1021/j100132a040.

[267] N. O. Winter and C. Hättig. *J. Chem. Phys.*, 134:184101, 2011. DOI: 10.1063/1.3584177.

[268] M. Wladyslawski and M. Nooijen. The photoelectron spectrum of the NO_3 radical revisited: A theoretical investigation of potential energy surfaces and conical intersections. In *Low-Lying Potential Energy Surfaces*, volume 828 of *ACS Symposium Series*, pages 65–92. American Chemical Society, Washington, D.C., 2002. DOI: 10.1021/bk-2002-0828.

[269] M. Wormit, D. R. Rehn, P. H. P. Harbach, J. Wenzel, C. M. Krauter, E. Epifanovsky, and A. Dreuw. *Mol. Phys.*, 112:774, 2014. DOI: 10.1080/00268976.2013.859313.

[270] M. Yang, A. Sissay, M. Chen, and K. Lopata. *J. Chem. Theory Comput.*, 18:992, 2022. DOI: 10.1021/acs.jctc.1c00079.

[271] Z.-Q. You, J.-M. Mewes, A. Dreuw, and J. M. Herbert. *J. Chem. Phys.*, 143:204104, 2015. DOI: 10.1063/1.4936357.

[272] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 141:064104, 2014. DOI: 10.1063/1.4891984.

[273] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 142:064109, 2015. DOI: 10.1063/1.4907376.

[274] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 143:234107, 2015. DOI: 10.1063/1.4937571.

[275] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 155:124111, 2021. DOI: 10.1063/5.0062757.

[276] Y. Zhu and J. M. Herbert. *J. Chem. Phys.*, 148:044117, 2018. DOI: 10.1063/1.5004675.

[277] Y. Zhu and J. M. Herbert. *J. Chem. Phys.*, 156:204123, 2022. DOI: 10.1063/5.0079910.

[278] T. Ziegler, A. Rauk, and E. J. Baerends. *Theor. Chem. Acc.*, 43:261, 1977. DOI: 10.1007/BF00551551.

[279] S. Zilberg and Y. Haas. *J. Chem. Phys.*, 103:20, 1995. DOI: 10.1063/1.469633.

[280] P. M. Zimmerman, F. Bell, D. Casanova, and M. Head-Gordon. *J. Am. Chem. Soc.*, 133:19944, 2011. DOI: 10.1021/ja208431r.

[281] P. M. Zimmerman, F. Bell, M. Goldey, A. T. Bell, and M. Head-Gordon. *J. Chem. Phys.*, 137:164110, 2012. DOI: 10.1063/1.4759076.

Chapter 8

Basis Sets and Effective Core Potentials

8.1 Introduction to Basis Sets

A basis set is a set of functions combined linearly to model molecular orbitals. Basis functions can be considered as representing the atomic orbitals of the atoms and are introduced in quantum chemical calculations because the equations defining the molecular orbitals are otherwise very difficult to solve.

Many standard basis sets have been carefully optimized and tested over the years. In principle, a user would employ the largest basis set available in order to model molecular orbitals as accurately as possible. In practice, the computational cost grows rapidly with the size of the basis set so a compromise must be sought between accuracy and cost. If this is systematically pursued, it leads to a "theoretical model chemistry",[39] that is, a well-defined energy procedure (*e.g.*, Hartree-Fock) in combination with a well-defined basis set.

Basis sets have been constructed from Slater, Gaussian, plane wave and delta functions. Slater functions were initially employed because they are considered "natural" and have the correct behavior at the origin and in the asymptotic regions. However, the two-electron repulsion integrals (ERIs) encountered when using Slater basis functions are expensive and difficult to evaluate. Delta functions are used in several quantum chemistry programs. However, while codes incorporating delta functions are simple, thousands of functions are required to achieve accurate results, even for small molecules. Plane waves are widely used and highly efficient for calculations on periodic systems, but are not so convenient or natural for molecular calculations.

The most important basis sets are contracted sets of atom-centered Gaussian functions. The number of basis functions used depends on the number of core and valence atomic orbitals, and whether the atom is light (H or He) or heavy (everything else). Contracted basis sets have been shown to be computationally efficient and to have the ability to yield chemical accuracy (see Appendix A). The Q-CHEM program has been optimized to exploit basis sets of the contracted Gaussian function type and has a large number of built-in standard basis sets (developed by Dunning and Pople, among others) which the user can access quickly and easily.

The selection of a basis set for quantum chemical calculations is very important. It is sometimes possible to use small basis sets to obtain good chemical accuracy, but calculations can often be significantly improved by the addition of diffuse and polarization functions. Consult the literature, including pertinent review articles,[12,24,39,42,43] in order to aid your selection. See also the "Further Reading" section at the end of this chapter.

8.2 Built-In Basis Sets

Q-CHEM is equipped with many standard basis sets,[1,64] and allows the user to specify the required basis set by its standard symbolic representation. The available built-in basis sets include the following types:

- Pople basis sets[6,14–16,31,33]
- Dunning basis sets[21]
- Correlation-consistent (cc) Dunning basis sets, including:
  - Standard cc-pVXZ and aug-cc-pVXZ (X = D, T, Q, 5, and 6)[4,22,79,80]
  - Partially-augmented "calendar" versions,[60,81] may-, jun-, and jul-cc-pVXZ (X = D, T, Q)
  - Versions with core–valence polarization functions,[63] cc-pCVXZ and cc-pwCVXZ (X = D, T, Q)
  - Partially-augmented core–valence polarization basis sets jun-, jul-, and aug-cc-pCVXZ (X = D, T, Q)
  - Pseudopotential (PP) basis sets for heavy elements, cc-pVXZ-PP and aug-cc-pVXZ-PP (X = D, T, Q)[40,61,62]
- Ahlrichs basis sets[70]
- Karlsruhe "def2" basis sets,[76] including
  - Augmented versions,[66] such as def2-SVPD
  - Partially-augmented versions,[34] such as def2-ha-SVP and def2-ha-SVP
- Jensen polarization consistent basis sets[44–46]
- Universal Gaussian basis set (UGBS)[13]

In addition, Q-CHEM supports the following features:

- Extra diffuse functions available for high quality excited-state calculations.
- Standard polarization functions.
- s, p, sp, d, f, g and h angular momentum types of basis functions (for energy calculations, up to k are supported).
- Pure and Cartesian basis functions.
- Mixed basis sets (see Section 8.5).
- Basis set superposition error (BSSE) corrections.
- Automatic, on-the-fly generation of a superposition of atomic densities (SAD) guess for any basis set (including general and mixed basis sets) and any SCF level of theory (see Section 4.4.2).

The following *$rem* keyword controls the basis set:

Basis Name[a]	j	m[b]	n[c]
STO-j(k+,l+)G(m,n)	2,3,6	d	p
j-21(k+,l+)G(m,n)	3	2d	2p
j-31(k+,l+)G(m,n)	4,6	3d	3p
j-311(k+,l+)G(m,n)	6	df, 2df, 3df	pd, 2pd, 3pd

[a] k and l denote the number of sets of diffuse functions on heavy atoms and on hydrogen atoms, respectively.
[b] m denotes the number of sets of polarization functions on the heavy atoms.
[c] n denotes the number of sets of polarization functions on the hydrogen atoms.

Table 8.1: Summary of Pople-type basis sets available in Q-CHEM.

BASIS

Sets the basis set to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen	User-defined. See section below
Symbol	Use standard basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and reviews to aid your selection.

8.3 Basis Set Symbolic Representation

8.3.1 Symbolic Representation Overview

Examples are given in the tables below and follow the standard format generally adopted for specifying basis sets. The single exception applies to additional diffuse functions. These are best inserted in a similar manner to the polarization functions; in parentheses with the light atom designation following heavy atom designation: (*heavy*, *light*), using a period as a placeholder in the unusual case that diffuse functions are to be added to hydrogen atoms but not to heavy atoms. See Table 8.1 for the general form. This convention can be applied, for example, to the named Pople-style basis sets listed in Table 8.2, resulting in specific examples given in Table 8.3.

Although not widely used in modern quantum chemistry, Dunning[21] introduced an early set of basis sets denoted SV, DZ, and TZ; see Table 8.4. (These are not to be confused with the widely-used "correlation-consistent" basis sets, which are also associated with Dunning's name.) The original Dunning basis sets can be extended with diffuse and polarization functions using a nomenclature similar to that used for Pople basis sets: *name*(k+,l+)(md,np), where k is the number of additional heavy atom diffuse functions, l is the number of additional light atom diffuse functions, m is the number of additional d polarization functions on heavy atoms, and n is the number of additional p polarization functions on light atoms.

The much more widely-used basis sets that are associated with Dunning are the correlation-consistent ("cc") ones.[22,80] The basic ones and their augmented counterparts are listed in Table 8.5. Those appended with "-PP" are pseudopotential basis sets, defined for heavy elements only and intended to be used in conjunction with effective core potentials (ECPs),

Symbolic Name	**Atoms Supported**
STO-2G	H, He, Li→Ne, Na→Ar, K, Ca, Sr
STO-3G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→I
STO-6G	H, He, Li→Ne, Na→Ar, K→Kr
3-21G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→Xe, Cs
4-31G	H, He, Li→Ne, P→Cl
6-31G	H, He, Li→Ne, Na→Ar, K→Kr
6-311G	H, He, Li→Ne, Na→Ar, Ga→I
G3LARGE	H, He, Li→Ne, Na→Ar, K→Kr
G3MP2LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr

Table 8.2: Atoms supported for Pople basis sets available in Q-CHEM.

Symbolic Name	**Atoms Supported**
3-21G	H, He, Li → Ne, Na → Ar, K →Kr, Rb → Xe, Cs
3-21+G	H, He, Na → Cl, Na → Ar, K, Ca, Ga → Kr
3-21G*	Na → Ar
6-31G	H, He, Li → Ne, Na → Ar, K → Zn, Ga → Kr
6-31+G	H, He, Li → Ne, Na → Ar, Ga → Kr
6-31G*	H, He, Li → Ne, Na → Ar, K → Zn, Ga → Kr
6-31G(d,p)	H, He, Li → Ne, Na → Ar, K → Zn, Ga → Kr
6-31G(.,+)G	H, He, Li → Ne, Na → Ar, Ga → Kr
6-31+G*	H, He, Li → Ne, Na → Ar, Ga → Kr
6-311G	H, He, Li → Ne, Na → Ar, Ga → I
6-311+G	H, He, Li → Ne, Na → Ar
6-311G*	H, He, Li → Ne, Na → Ar, Ga → I
6-311G(d,p)	H, He, Li → Ne, Na → Ar, Ga → I
G3LARGE	H, He, Li → Ne, Na → Ar, K → Kr
G3MP2LARGE	H, He, Li → Ne, Na → Ar, Ga → Kr

Table 8.3: Examples of extended Pople basis sets.

Symbolic Name	**Atoms Supported**
SV	H, Li → Ne
SV*	H, B → Ne
SV(d,p)	H, B → Ne
SV(2+,+)(2d,p)	H, B → Ne
DZ	H, Li → Ne, Al → Cl
DZ+	H, B → Ne
DZ++	H, B → Ne
DZ*	H, Li → Ne
DZ**	H, Li → Ne
DZ(d,p)	H, Li → Ne
DZ(2+,+)(2d,p)	H, B → Ne
TZ	H, Li → Ne
TZ+	H, Li → Ne
TZ++	H, Li → Ne
TZ*	H, Li → Ne
TZ**	H, Li → Ne
TZ(d,p)	H, Li → Ne

Table 8.4: Examples of extended Dunning basis sets.

which are discussed in Section 8.10. Each correlation-consistent basis set (cc-*name* has an "augmented" counterpart (aug-cc-*name*) that includes diffuse functions.

The correlation-consistent paradigm adds additional diffuse functions for each angular momentum class, meaning that for a second-row atom such as carbon, the aug-cc-pVDZ basis set contains diffuse s, p, and d functions (10 diffuse functions per atom), while hydrogen contains diffuse s and p functions. The aug-cc-pVTZ basis set also includes diffuse f functions for carbon (for a total of 20 diffuse functions per atom) and diffuse d functions for hydrogen. As compared to functions with tighter exponents, inclusion of diffuse functions is relatively expensive and prone to incur linear dependencies that hamper SCF convergence, as discussed in Section 8.3.2. At the same time, diffuse functions are often crucial to the description of anions, excited states, and noncovalent interactions but the high angular momentum diffuse functions included in aug-cc-pVXZ are not always necessary. In recognition of this fact. "calendar" versions of the correlation-consistent basis sets have been introduced (jul-, jun-, and may-*name*),[60,81] which systemically remove diffuse basis functions starting from aug-cc-*name*. The jul-cc-pVXZ basis set removes all diffuse functions from hydrogen, and is equivalent to using cc-pVXZ for hydrogen and aug-cc-pVXZ for heavy atoms. The jun-cc-pVXZ basis set additionally removes the highest angular momentum diffuse functions from each heavy atom, *e.g.*, for a carbon atom the diffuse d functions are removed to make jun-cc-pVDZ and the diffuse f functions are removed to make jun-cc-pVTZ. The may-cc-pVXZ basis sets then remove the highest angular momentum diffuse functions that remain in jun-cc-pVXZ, so that for a carbon atom, may-cc-pVDZ is minimally augmented with only a single diffuse s function. Q-CHEM includes may-, jun-, and jul-cc-pVXZ and similarly may-, jun-, and jul-cc-pCVXZ (for X = D, T, and Q in both cases). Also available are the jun-cc-pVXZ-PP parings of aug-cc-pVXZ-PP and the jun-cc-pwCVXZ(-PP) parings of aug-cc-pwCVXZ(-PP), again for X = D, T, or Q. If the user has questions as to what functions are included in any of these basis sets, simply set PRINT_GENERAL_BASIS = TRUE in the *$rem* section (as described in Section 8.3.2) to get a printout of the basis function information.

The name Ahlrichs is also associated with two different collections of basis sets. The older set (TZV, VDZ, and VTZ) is listed in Table 8.6;[70] these basis sets are available but are no longer in common use. Much more widely used are the second-generation "def2" basis sets that are listed in Table 8.7,[66,76] which are sometimes called "Karlsruhe" basis

Symbolic Name	**Atoms Supported**
cc-pVDZ	H → Ar, Ca, Ga → Kr
cc-pVDZ-full	H → Ar, Ca → Kr
cc-pVDZ-PP	Cu → Rn
cc-pVTZ	H → Ar, Ca, Ga → Kr
cc-pVTZ-full	H → Ar, Ca → Kr
cc-pVTZ-PP	Cu → Rn
cc-pVQZ	H → Ar, Ca, Ga → Kr
cc-pVQZ-full	H → Ar, Ca → Kr
cc-pVQZ-PP	Cu → Rn
cc-pV5Z	H → Ar, Ca → Kr
cc-pV6Z	H → Ar except Li, Na, Mg
cc-pCVDZ	H → Ar, Ca (H and He use cc-pVDZ)
cc-pCVTZ	H → Ar, Ca (H and He use cc-pVTZ)
cc-pCVQZ	H → Ar, Ca (H and He use cc-pVQZ)
cc-pCV5Z	H, He, B → Ar, Ca (H and He use cc-pV5Z)
cc-pwCVDZ	B → Ne, Al → Ar
cc-pwCVTZ	B → Ne, Al → Ar, Sc → Zn
cc-pwCVQZ	B → Ne, Al → Ar, Sc → Zn, Br
cc-pwCVDZ-PP	Cu → Rn
cc-pwCVTZ-PP	Cu → Rn
cc-pwCVQZ-PP	Cu → Rn
aug-cc-pVDZ[a]	H → Kr
aug-cc-pVDZ-PP[b]	Cu → Rn
aug-cc-pVTZ[a]	H → Kr
aug-cc-pVTZ-PP[b]	Cu → Rn
aug-cc-pVQZ[a]	H → Kr
aug-cc-pVQZ-PP[b]	Cu → Rn
aug-cc-pV5Z	H → Ar, Sc → Kr
aug-cc-pV6Z	H → Ar except Li, Be, Na, Mg
aug-cc-pCVDZ[a]	H → Ar (H and He use aug-cc-pVDZ)
aug-cc-pCVTZ[a]	H → Ar (H and He use aug-cc-pVTZ)
aug-cc-pCVQZ[a]	H → Ar (H and He use aug-cc-pVQZ)
aug-cc-pCV5Z	H, He, B → Ar (H and He use aug-cc-pV5Z)
aug-cc-pwCVDZ[c]	B → Ne, Al → Ar
aug-cc-pwCVTZ[c]	B → Ne, Al → Ar, Sc → Zn
aug-cc-pwCVQZ[c]	B → Ne, Al → Ar, Sc → Zn, Br
aug-cc-pwCVDZ-PP[c]	Cu → Rn
aug-cc-pwCVTZ-PP[c]	Cu → Rn
aug-cc-pwCVQZ-PP[c]	Cu → Rn

[a] may-, jun-, and jul-cc-p(C)VXZ variants are also available
[b] jun-cc-pVXZ-PP variant is also available
[c] jun-cc-p(w)VXZ(-PP) variant is also available

Table 8.5: Atoms supported in Q-CHEM for correlation-consistent basis sets. For cc-pVXZ (X = D, T, Q), those names which do not end in "-full" correspond to the definitions with a segmented contraction scheme,[11] and those that do end in "-full" correspond to the original optimized generally-contracted definitions. For all other basis sets, where there is no distinction, the only definition is from optimized general contraction. For the augmented basis sets, footnotes indicate the availability of "calendar" variants.[60,81]

Symbolic Name	Atoms Supported
TZV	H → Kr
VDZ	H → Kr
VTZ	H → Kr

Table 8.6: Atoms supported in Q-CHEM for the original Ahlrichs basis sets.[70] (Note that these are different from the more modern Karlsruhe "def2" basis sets, which are described in Table 8.7.)

Symbolic Name	Atoms Supported
def2-mSVP	H–Kr,[a] Rb–Rn (with def2-ECP)
def2-SV(P), def2-SVP	H–Kr; Rb–Rn (with def2-ECP)
def2-ma-SVP, def2-ha-SVP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-SVPD	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-TZVP, def2-TZVPP	H–Kr; Rb–Rn (with def2-ECP)
def2-ma-TZVP, def2-ma-TZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-ha-TZVP, def2-ha-TZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-TZVPD, def2-TZVPPD	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-QZVP, def2-QZVPP	H–Kr; Rb–Rn (with def2-ECP)
def2-ma-QZVP, def2-ma-QZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-ha-QZVP, def2-ha-QZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-QZVPD, def2-QZVPPD	H–Kr; Rb–La, Hf–Rn (with def2-ECP)

[a] Na–Kr are identical to def2-SV(P)

Table 8.7: Atoms supported in Q-CHEM for the Karlsruhe "def2" basis sets.[34,66,76]

sets to distinguish them from the older basis sets developed by Ahlrichs and co-workers at the University of Karlsruhe. These basis sets were originally designed for SCF calculations although more recently they have seen some use in correlated wave function calculations. Diffuse functions were added later,[66] and are stipulated with a name ending in "D", *e.g.*, def2-SVP does not contain diffuse functions but def2-SVPD does. The def2-ha and def2-ma variants (*e.g.*, def2-ha-SVP) include partial augmentation.[34] The def2-ha basis sets are "heavy-augmented", eliminating all diffuse functions on the hydrogen atoms, so that def2-ha-SVP consists of def2-SVP for hydrogen and def2-SVPD for other atoms. The def2-ma basis sets are "minimally-augmented", and are constructed from def2-ha-SVP by removing the highest angular moment diffuse function on each heavy atom, similar to the jun-cc-pVXZ prescription.

Finally, there is a set of basis sets associated with the name of Jensen[44,45] (see Table 8.8), which were developed primarily for NMR calculations. There is also a "universal" Gaussian basis set,[13] which is supported for elements H–Lr except for Pa–Np and Cm–Bk.

8.3.2 Customization

Q-CHEM offers a number of standard and special customization features. One of the most important is that of supplying additional diffuse functions. Diffuse functions are often important for studying anions and excited states of molecules, and for the latter several sets of additional diffuse functions may be required. These extra diffuse functions can be generated from the standard diffuse functions by applying a scaling factor to the exponent of the original diffuse function. This yields a geometric series of exponents for the diffuse functions which includes the original standard functions along with more diffuse functions.

Symbolic Name[a]	**Atoms Supported**
pcseg-n	H $\rightarrow$ Kr
pc-n	H $\rightarrow$ Kr
pcJ-n	H $\rightarrow$ Ar
psS-n	H $\rightarrow$ Ar
aug-pcseg-n	H $\rightarrow$ Kr
aug-pc-n	H $\rightarrow$ Kr
aug-pcJ-n	H $\rightarrow$ Ar
aug-psS-n	H $\rightarrow$ Ar

[a]For $n = 0, 1, 2, 3, 4$ in each case

Table 8.8: Atoms supported for Jensen polarization consistent basis sets available in Q-CHEM. The pcseg-n sets should be preferred instead of pc-n, as they are more efficient in Q-CHEM. The pcJ-n[45] and pcS-n[44] basis sets are optimized for NMR spin-spin couplings and chemical shieldings, respectively.

When using very large basis sets, especially those that include many diffuse functions, or if the system being studied is very large, linear dependence in the basis set may arise. This results in an over-complete description of the space spanned by the basis functions, and can cause a loss of uniqueness in the molecular orbital coefficients. Consequently, the SCF may be slow to converge or behave erratically. Q-CHEM will automatically check for linear dependence in the basis set, and will project out the near-degeneracies, if they exist. This will result in there being slightly fewer molecular orbitals than there are basis functions. Q-CHEM checks for linear dependence by considering the eigenvalues of the overlap matrix. Very small eigenvalues are an indication that the basis set is close to being linearly dependent. The size at which the eigenvalues are considered to be too small is governed by the *$rem* variable BASIS_LIN_DEP_THRESH. By default this is set to 6, corresponding to a threshold of 10^{-6}. This has been found to give reliable results, however SCF convergence failure (especially for large molecules or those with highly diffuse basis sets) may be a symptom of linear dependencies. The smallest overlap matrix eigenvalue is printed in the Q-CHEM output file, and usually when this number goes below 10^{-5}, numerical issues caused by basis function linear dependence may occur and the SCF calculation may not give reasonable solutions. If the smallest overlap matrix eigenvalue is less than the square root of the integral threshold, a warning message urging to tighten the integral threshold (*e.g.*, setting THRESH = 14) will be printed out. In any case, when a linear dependence issue is suspected, tightening the integral threshold should be tried first. Especially for larger molecules in basis sets that contain diffuse functions, tightening the integral threshold sometimes has the nonintuitive effect of *decreasing* the time-to-solution, by significantly reducing the number of SCF cycles at only a modest per-cycle increase in cost.

PRINT_GENERAL_BASIS

Controls print out of built in basis sets in input format

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print out standard basis set information

FALSE Do not print out standard basis set information

RECOMMENDATION:

Useful for modification of standard basis sets.

BASIS_LIN_DEP_THRESH

Sets the threshold for determining linear dependence in the basis set

TYPE:

INTEGER

DEFAULT:

6 Corresponding to a threshold of 10^{-6}

OPTIONS:

n Sets the threshold to 10^{-n}

RECOMMENDATION:

Set to 5 or smaller if you have a poorly behaved SCF and you suspect linear dependence in you basis set. Lower values (larger thresholds) may affect the accuracy of the calculation.

8.4 User-Defined Basis Sets (*$basis*)

8.4.1 Introduction

Users may, on occasion, prefer to use non-standard basis, and it is possible to declare user-defined basis sets in Q-CHEM input (see Chapter 3 on Q-CHEM inputs). The format for inserting a non-standard user-defined basis set is both logical and flexible, and is described in detail in the job control section below.

Note that the SAD guess is not currently supported with non-standard or user-defined basis sets. The current default is to use SCF_GUESS = AUTOSAD, unless a mixed basis is specified. When using a mixed basis, the simplest alternative is to specify the GWH or CORE options for SCF_GUESS, but these are relatively ineffective other than for small basis sets. The recommended alternative is to employ basis set projection by specifying a standard basis set for the BASIS2 keyword. See Section 4.4 on initial guesses for more information.

8.4.2 Job Control

In order to use a user-defined basis set, the BASIS *$rem* must be set to GENERAL or GEN.

When using a non-standard basis set which incorporates d or higher angular momentum basis functions, the *$rem* variable PURECART needs to be set. This *$rem* variable indicates to the Q-CHEM program how to handle the angular form of the basis functions. As indicated above, each integer represents an angular momentum type which can be defined as either pure (1) or Cartesian (2). For example, 1111 specifies that d, f, g and h basis functions should have pure form whereas 1121 indicates that d- g- and h-functions are pure but f functions are Cartesian. These four-digit codes can be used even if the basis does not contain g or h functions; PURECART = 1111 (or 2222) specifies to use pure (or Cartesian) functions for all angular momentum types.

PURECART

INTEGER

TYPE:

Controls the use of pure (spherical harmonic) or Cartesian angular forms

DEFAULT:

1111 Pure h, g, f, d functions

OPTIONS:

$hgfd$ Use 1 for pure and 2 for Cartesian.

RECOMMENDATION:

This is pre-defined for all standard basis sets

In standard basis sets all functions are pure, except for the d functions in n-21G–type bases (*e.g.*, 3-21G) and n-31G bases (*e.g.*, 6-31G, 6-31G*,6-31+G*, ...). In particular, the 6-311G series uses pure functions for both d and f.[27,50] Note that other electronic structure codes may use Cartesian functions for basis sets of the same name, in which case absolute energies will differ. Set PURECART = 2222 to request Cartesian Gaussian functions, if desired.

8.4.3 Format for User-Defined Basis Sets

The format for the user-defined basis section is as follows:

```
$basis
```

$$
\begin{array}{lllll}
X & 0 & & & \\
L & K & scale & & \\
\alpha_1 & C_1^{L_{\min}} & C_1^{L_{\min}+1} & \ldots & C_1^{L_{\max}} \\
\alpha_2 & C_2^{L_{\min}} & C_2^{L_{\min}+1} & \ldots & C_2^{L_{\max}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\alpha_K & C_K^{L_{\min}} & C_K^{L_{\min}+1} & \ldots & C_K^{L_{\max}}
\end{array}
$$

```
****
$end
```

where

X	Atomic symbol of the atom (atomic number not accepted)
L	Angular momentum symbol (S, P, SP, D, F, G)
K	Degree of contraction of the shell (integer)
$scale$	Scaling to be applied to exponents (default is 1.00)
α_i	Gaussian primitive exponent (positive real number)
C_i^L	Contraction coefficient for each angular momentum (non-zero real numbers).

Atoms are terminated with $\star\star\star\star$ and the complete basis set is terminated with the *$end* keyword terminator. No blank lines can be incorporated within the general basis set input. Note that more than one contraction coefficient per line is

required for compound shells like SP. As with all Q-CHEM input deck information, all input is case-insensitive.

Example 8.1 Example of adding a user-defined non-standard basis set. Note that since d, f and g functions are incorporated, the *$rem* variable PURECART must be set. Note the use of BASIS2 for the initial guess.

```
$molecule
   0  1
   O
   H  O  oh
   H  O  oh  2  hoh

   oh  =   1.2
   hoh = 110.0
$end

$rem
   EXCHANGE   hf
   BASIS      gen      user-defined general basis
   BASIS2     sto-3g   sto-3g orbitals as initial guess
   PURECART   112      Cartesian d functions, pure f and g
$end

$basis
   H  0
   S  2  1.00
      1.30976    0.430129
      0.233136   0.678914
****
   O  0
   S  2  1.00
      49.9810    0.430129
      8.89659    0.678914
   SP 2  1.00
      1.94524    0.049472    0.511541
      0.49336    0.963782    0.612820
   D  1  1.00
      0.39000    1.000000
   F  1  1.00
      4.10000    1.000000
   G  1  1.00
      3.35000    1.000000
****
$end
```

8.5 Mixed Basis Sets

In addition to defining a custom basis set, it is also possible to specify different standard basis sets for different atoms. For example, in a large alkene molecule the hydrogen atoms could be modeled by the STO-3G basis, while the carbon atoms have the larger 6-31G(d) basis. This can be specified within the *$basis* block using the more familiar basis set labels.

Note: (1) It is not possible to augment a standard basis set in this way; the whole basis needs to be inserted as for a user-defined basis (angular momentum, exponents, contraction coefficients) and additional functions added. Standard basis set exponents and coefficients can be easily obtained by setting the PRINT_GENERAL_BASIS *$rem* variable to TRUE.

(2) The PURECART flag must be set for *all* general basis input containing d angular momentum or higher functions, regardless of whether standard basis sets are entered in this non-standard manner.

The user can also specify different basis sets for atoms of the same type, but in different parts of the molecule. This allows a larger basis set to be used for the active region of a system, and a smaller basis set to be used in the less important regions. To enable this the BASIS keyword must be set to MIXED and a *$basis* section included in the input deck that gives a complete specification of the basis sets to be used. The format is exactly the same as for the user-defined basis, except that the atom number (as ordered in the *$molecule* section) must be specified in the field after the atomic symbol. A basis set must be specified for every atom in the input, even if the same basis set is to be used for all atoms of a particular element. Custom basis sets can be entered, and the shorthand labeling of basis sets is also supported.

The use of different basis sets for a particular element means the global potential energy surface is no longer unique. The user should exercise caution when using this feature of mixed basis sets, especially during geometry optimizations and transition state searches.

Example 8.2 Example of adding a user defined non-standard basis set. The user is able to specify different standard basis sets for different atoms.

```
$molecule
   0  1
   O
   H  O  oh
   H  O  oh  2  hoh

   oh  =   1.2
   hoh = 110.0
$end

$rem
   EXCHANGE   hf
   BASIS      General   user-defined general basis
   PURECART   2         Cartesian D functions
   BASIS2     sto-3g    use STO-3G as initial guess
$end

$basis
   H 0
   6-31G
   ****
   O 0
   6-311G(d)
   ****
$end
```

Example 8.3 Example of using a mixed basis set for methanol. The user is able to specify different standard basis sets for some atoms and supply user-defined exponents and contraction coefficients for others. This might be particularly useful in cases where the user has constructed exponents and contraction coefficients for atoms not defined in a standard basis set so that only the non-defined atoms need have the exponents and contraction coefficients entered. Note that a basis set has to be specified for every atom in the molecule, even if the same basis is to be used on an atom type.

```
$molecule
  0  1
  C    0.0000000    0.0148306    0.7155831
  H    0.9153226    0.5361067    1.0707116
  H    0.0000000   -1.0112551    1.1374379
  H   -0.9153226    0.5361067    1.0707116
  O    0.0000000   -0.0695490   -0.6801243
  H    0.0000000    0.8662925   -1.0101622
$end

$rem
   EXCHANGE   hf
   BASIS      mixed          ! user-defined mixed basis
   PRINT_GENERAL_BASIS true  ! confirm what basis functions are used
$end

$basis
   C  1
   3-21G
   ****
   H  2
   sto-3g
   ****
   H  3
   sto-3g
   ****
   H  4
   sto-3g
   ****
   O  5
   S  3  1.00
      3.22037000E+02   5.92394000E-02
      4.84308000E+01   3.51500000E-01
      1.04206000E+01   7.07658000E-01
   SP 2 1.00
      7.40294000E+00  -4.04453000E-01   2.44586000E-01
      1.57620000E+00   1.22156000E+00   8.53955000E-01
   SP 1 1.00
      3.73684000E-01   1.00000000E+00   1.00000000E+00
   SP 1 1.00
      8.45000000E-02   1.00000000E+00   1.00000000E+00
   ****
   H  6
   6-31(+,+)G(d,p)
   ****
$end
```

8.6 Dual Basis Sets

There are several types of calculation that can be performed within Q-CHEM using *two* atomic orbital basis sets instead of just one as we have been assuming in this chapter so far. Such calculations are said to involve *dual* basis sets. Typi-

cally iterations are performed in a smaller, primary, basis, which is specified by the *$rem* keyword BASIS2. Examples of calculations that can be performed using dual basis sets include:

- An improved initial guess for an SCF calculation in the large basis. See Section 4.4.5.
- Dual basis self-consistent field calculations (Hartree-Fock and density functional theory). See discussion in Section 4.7.
- Density functional perturbative corrections by "triple jumping". See Section 4.8.
- Dual basis MP2 calculations. See discussion in Section 6.6.2.

BASIS2

Defines the (small) second basis set.

TYPE:

STRING

DEFAULT:

No default for the second basis set.

OPTIONS:

Symbol	Use standard basis sets as for BASIS.
BASIS2_GEN	General BASIS2
BASIS2_MIXED	Mixed BASIS2

RECOMMENDATION:

BASIS2 should be smaller than BASIS. There is little advantage to using a basis larger than a minimal basis when BASIS2 is used for initial guess purposes. Larger, standardized BASIS2 options are available for dual-basis calculations as discussed in Section 4.7 and summarized in Table 4.7.4.

In addition to built-in basis sets for BASIS2, it is also possible to enter user-defined second basis sets using an additional *$basis2* input section, whose syntax generally follows the *$basis* input section documented above in Section 8.4.

8.7 Complex Basis Sets

Complex basis function (CBF) methods for describing meta-stable electronic states (see Section 4.9.5) require the user to specify a basis set in which the most diffuse functions have a complex-scaled exponent. The real-valued basis functions are specified using the standard *$rem* keyword BASIS. The complex basis set is specified using the *$rem* keyword COMPLEX_BASIS, and it should be a superset of the real basis. If the basis is specified as general or mixed, the user-defined basis should be entered in an an additional *$zbasis* input section in the same format as the *$basis* section.

Q-CHEM will automatically determine the additional basis functions in the complex basis and scales the exponents of those functions. For example, if BASIS is specified as "cc-pVTZ" and COMPLEX_BASIS is specified as "aug-cc-pVTZ", the augmenting functions will have complex-scaled exponents. An alternative way to specify the scaled functions is the scaling variable in the *$complex_ccman* section. If a scaling independent of the BASIS is desired, this variable accepts a comma-separated list of the indices of the shells to be scaled. This requires the COMPLEX_CCMAN variable to be set to 1 so that the scaling string is recognized. The indices are referring to the order in which they appear in the basis set file or respectively, in the *$zbasis* section. Furthermore, in the case of multiple centers, basis functions on different centers are enumerated separately, in the order as the centers are listed in the *$molecule* section, which allows different scaling of basis functions even when centered at atoms of the same type. As an example, in a calculation with two atoms which each possess 10 shells, the shells of the first atom would be given the indices 1–10 and the shells of the

second atom 11–20. The following *$complex_ccman* section is an example for this, in which the scaling of shells 5, 6, 15 and 16 is requested.

```
$rem
...
complex_ccman 1
...
$end

$complex_ccman
   CS_THETA  0
   CS_ALPHA  1000
   scaling 5,6,15,16
$end
```

The scaling is done according to

$$\alpha \rightarrow \alpha e^{-2i\theta} \tag{8.1}$$

where α is the exponent of a particular basis function and θ is specified by the *$rem* keyword COMPLEX_THETA.

COMPLEX_BASIS

Defines the complex basis.

TYPE:

STRING

DEFAULT:

No default complex basis set

OPTIONS:

Symbol	Use a standard basis set
ZBASIS_GENERAL, ZBASIS_GEN	User-defined. As for BASIS
ZBASIS_MIXED	User-defined mixed basis

RECOMMENDATION:

Consult Ref. 78 and the Basis Set Exchange.

COMPLEX_THETA

Sets the value of θ in degrees for a calculation with complex basis functions.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n $\theta = n/10$ (degrees)

RECOMMENDATION:

Consult Ref. 78. Usually calculations at several different values of θ (a "θ-trajectory") should be performed.

8.8 Auxiliary Basis Sets for RI (Density Fitting)

While atomic orbital standard basis sets are used to expand one-electron functions such as molecular orbitals, *auxiliary basis sets* are also used in many Q-CHEM jobs to efficiently approximate products of one-electron functions, such as arise in electron correlation methods.

For a molecule of fixed size, increasing the number of basis functions *per atom*, n, leads to $\mathcal{O}(n^4)$ growth in the number of significant four-center two-electron integrals, since the number of non-negligible product charge distributions, $|\mu\nu\rangle$, grows as $\mathcal{O}(n^2)$. As a result, the use of large (high-quality) basis expansions is computationally costly. Perhaps the most practical way around this "basis set quality" bottleneck is the use of auxiliary basis expansions.[20,25,47] The ability to use auxiliary basis sets to accelerate a variety of electron correlation methods, including both energies and analytical gradients, is a major feature of Q-CHEM.

The auxiliary basis $\{|K\rangle\}$ is used to approximate products of Gaussian basis functions:

$$|\mu\nu\rangle \approx |\widetilde{\mu\nu}\rangle = \sum_K |K\rangle C_{\mu\nu}^K \tag{8.2}$$

Auxiliary basis expansions were introduced long ago, and are now widely recognized as an effective and powerful approach, which is sometimes synonymously called resolution of the identity (RI) or density fitting (DF). When using auxiliary basis expansions, the rate of growth of computational cost of large-scale electronic structure calculations with n is reduced to approximately n^3.

If n is fixed and molecule size increases, auxiliary basis expansions reduce the pre-factor associated with the computation, while not altering the scaling. The important point is that the pre-factor can be reduced by 5 or 10 times or more. Such large speedups are possible because the number of auxiliary functions required to obtain reasonable accuracy, X, has been shown to be only about 3 or 4 times larger than N.

The auxiliary basis expansion coefficients, $\mathbf{C}$, are determined by minimizing the deviation between the fitted distribution and the actual distribution, $\langle \mu\nu - \widetilde{\mu\nu} | \mu\nu - \widetilde{\mu\nu} \rangle$, which leads to the following set of linear equations:

$$\sum_L \langle K\,|L\rangle C_{\mu\nu}^L = \langle K\,|\mu\nu\,\rangle \tag{8.3}$$

Evidently solution of the fit equations requires only two- and three-center integrals, and as a result the (four-center) two-electron integrals can be approximated as the following optimal expression for a given choice of auxiliary basis set:

$$\langle \mu\nu|\lambda\sigma\rangle \approx \langle \widetilde{\mu\nu}|\widetilde{\lambda\sigma}\rangle = \sum_{K,L} C_{\mu\nu}^L \langle L|K\rangle C_{\lambda\sigma}^K \tag{8.4}$$

In the limit where the auxiliary basis is complete (*i.e.* all products of AOs are included), the fitting procedure described above will be exact. However, the auxiliary basis is invariably incomplete (as mentioned above, $X \approx 3N$) because this is essential for obtaining increased computational efficiency.

More details on Q-CHEM's use of RI methods is given in Section 6.6 on RI-MP2 and related methods, Section 6.19 on pairing methods, Section 6.12.9 on coupled cluster methods, Section 4.6.6 on DFT methods, and Section 7.12 on restricted active space methods. In the remainder of this section we focus on documenting the input associated with the auxiliary basis itself.

Q-CHEM contains a variety of built-in auxiliary basis sets, that can be specified by the *$rem* keyword AUX_BASIS. These auxiliary basis sets are listed in Table 8.9 and specified in the *$rem* input section by means of the job control variables that are listed below. Note that in addition to the built-in auxiliary basis sets (Table 8.9), it is also possible to enter user-defined auxiliary basis sets using an *$aux_basis* (or *$aux_basis_j*, *$aux_basis_k*, *$aux_basis_corr*) input section, whose syntax generally follows the *$basis* input section documented above in Section 8.4.

AUX_BASIS

Sets the auxiliary basis set to be used

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

AUX_BASIS_J

Sets the auxiliary basis set for RI-J to be used or invokes RI-J

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

AUX_BASIS_K

Sets the auxiliary basis set for RI-K or occ-RI-K to be used or invokes occ-RI-K

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

AUX_BASIS_CORR

Sets the auxiliary basis set for RI-MP2 to be used or invokes RI-MP2 in case of double-hybrid DFT or MP2

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

Symbolic Name	**Atoms Supported**
RIJ-def2-XX[a]	H, He, Li → Ne, Na → Ar, K → Br, Rb → Xe, Cs → La, Hf → Rn
RIJK-def2-XX[a]	H, He, Li → Ne, Na → Ar, K → Br, Rb → Xe, Cs → La, Hf → Rn
RIMP2-def2-XX[b]	H, He, Li → Ne, Na → Ar, K → Br, Rb → Xe, Cs → La, Hf → Rn
RIMP2-def2-ma-XX[c]	H, He, Li → Ne, Na → Ar, K → Br, Rb → Xe, Cs → La, Hf → Rn
RIMP2-def2-ha-XX[c]	H, He, Li → Ne, Na → Ar, K → Br, Rb → Xe, Cs → La, Hf → Rn
RIMP2-VDZ[d]	H, He, Li → F, Na → Cl, K → Br
RIMP2-TZVPP	H, He, Li → Ne, Na → Ar, Ga → Kr
RIMP2-cc-pVDZ	H, He, Li → Ne, Na → Ar, Ga → Kr
RIMP2-cc-pVTZ	H, He, Li → Ne, Na → Ar, Ga → Kr
RIMP2-cc-pVQZ	H, He, Li → Ne, Na → Ar, Ga → Kr
RIMP2-*cal*-cc-pVDZ[e]	H, He, B → Ne, Al → Ar, Ga → Kr
RIMP2-*cal*-cc-pVTZ[e]	H, He, B → Ne, Al → Ar, Ga → Kr
RIMP2-*cal*-cc-pVQZ[e]	H, He, B → Ne, Al → Ar, Ga → Kr
RIMP2-aug-cc-pVDZ	H, He, B → Ne, Al → Ar, Ga → Kr
RIMP2-aug-cc-pVTZ	H, He, B → Ne, Al → Ar, Ga → Kr
RIMP2-aug-cc-pVQZ	H, He, B → Ne, Al → Ar, Ga → Kr

[a] XX includes SV(P), SVP, TZVP, TZVPP, QZVP, and QZVPP. [b] XX includes SV(P), SVP, SVPD, TZVP, TZVPD, TZVPP, TZVPPD, QZVP, and QZVPPD. [c] XX includes SVP, TZVP, TZVPP, and QZVPP, obtained by systematic pruning of diffuse functions. For example, RIMP2-def2-ma-SVP is obtained from RIMP2-def2-SVPD by pruning in the same manner that def2-ma-SVP is obtained from def2-SVPD. [d] This is the original SVP fitting set from Ref. 77. [e] Where *cal* = may, jun, or jul. These are constructed from the corresponding RIMP2-aug-cc-pVXZ basis set by systematic deletion of diffuse functions analogous to how the "calendar" basis set *cal*-cc-pVXZ is constructed from aug-cc-pVXZ.

Table 8.9: Built-in auxiliary basis sets available in Q-CHEM for electron correlation.

8.9 Ghost Atoms and Basis Set Superposition Error

When calculating intermolecular interaction energies, a naïve calculation of the energy difference

$$\Delta E_{AB} = E_{AB} - E_A - E_B \tag{8.5}$$

usually results in severe overestimation of the interaction energy, even if all three energies in Eq. (8.5) are computed at a good level of theory. This phenomenon, known as *basis set superposition error* (BSSE), is an artifact of an

unbalanced approximation, namely, that the dimer energy E_{AB} is computed in a more flexible basis set as compared to the two monomer energies. Although BSSE disappears in the complete basis-set limit, it does so extremely slowly: in $(H_2O)_6$, for example, an MP2/aug-cc-pVQZ calculation of the interaction energy is still a bit more than 1 kcal/mol away from the MP2 complete-basis limit.[67] Short of computing all energies in very large basis sets and extrapolating to the complete-basis limit, the conventional solution to the BSSE problem is the *counterpoise correction*, originally proposed by Boys and Bernardi.[7] Here, one corrects for BSSE by computing the monomer energies E_A and E_B in the dimer basis set, with the idea being that this results in a more balanced treatment of ΔE_{AB}.

In truth the average of the counterpoise-corrected and uncorrected results is often a better approximation than either of them individually, but in any case one needs the counterpoise-corrected result. This requires basis functions to be placed at arbitrary points in space, not just those defined by the nuclear centers; these are usually termed "floating centers" or "ghost atoms". Ghost atoms have zero nuclear charge but can support a user-defined basis set. Their positions are specified in the *$molecule* section alongside all the other atoms (atomic symbol: Gh), and their intended basis functions are specified in one of two ways:

1. Via a user-defined *$basis* section, using BASIS = MIXED.

2. Placing "@" next to an atomic symbol in the *$molecule* section designates it as a ghost atom supporting the same basis functions as the corresponding atom, so that a *$basis* section is not required.

Examples of either procedure appear below.

The calculation of ΔE_{AB} in Eq. (8.5) requires three separate electronic structure calculations but this process can be performed automatically using Q-CHEM's BSSE job type based on calculations using fragment jobs. This machinery is much more versatile and is described in detail later so we will not discuss the automatic procedure here; see Section 12.4

for that.

Example 8.4 A calculation on a water monomer in the presence of the full dimer basis set. The energy will be slightly lower than that without the ghost atom functions due to the greater flexibility of the basis set.

```
$molecule
   0  1
   O    1.68668  -0.00318   0.000000
   H    1.09686   0.01288  -0.741096
   H    1.09686   0.01288   0.741096
   Gh  -1.45451   0.01190   0.000000
   Gh  -2.02544  -0.04298  -0.754494
   Gh  -2.02544  -0.04298   0.754494
$end

$rem
   METHOD        mp2
   BASIS         mixed
$end

$basis
   O 1
   6-31G*
   ****
   H 2
   6-31G*
   ****
   H 3
   6-31G*
   ****
   O 4
   6-31G*
   ****
   H 5
   6-31G*
   ****
   H 6
   6-31G*
   ****
$end
```

Example 8.5 A calculation on ammonia in the presence of the basis set of ammonia borane.

```
$molecule
   0 1
    N     0.0000   0.0000   0.7288
    H     0.9507   0.0001   1.0947
    H    -0.4752  -0.8234   1.0947
    H    -0.4755   0.8233   1.0947
   @B     0.0000   0.0000  -0.9379
   @H     0.5859   1.0146  -1.2474
   @H     0.5857  -1.0147  -1.2474
   @H    -1.1716   0.0001  -1.2474
$end

$rem
   METHOD          B3LYP
   BASIS           6-31G(d,p)
   PURECART        1112
$end
```

8.10 Effective Core Potentials (ECPs)

8.10.1 Introduction & Overview

The application of quantum chemical methods to elements in the lower half of the Periodic Table is more difficult than for the lighter atoms. There are two key reasons for this:

- the number of electrons in heavy atoms is large
- relativistic effects in heavy atoms are often non-negligible

Both of these problems stem from the presence of large numbers of core electrons and, given that such electrons do not play a significant *direct* role in chemical behavior, it is natural to ask whether it is possible to model their effects in some simpler way. Such enquiries led to the invention of Effective Core Potentials (ECPs) or pseudopotentials. Many review articles discussing relativistic effects in chemistry are available.[2,8,10,26,32,65].

If we seek to replace the core electrons around a given nucleus by a pseudopotential, while affecting the chemistry as little as possible, the pseudopotential should have the same effect on nearby valence electrons as the core electrons. The most obvious effect is the simple electrostatic repulsion between the core and valence regions but the requirement that valence orbitals must be orthogonal to core orbitals introduces additional subtler effects that cannot be neglected.

One of the key issues in the development of ECPs is the definition of the "core". So-called "large-core" ECPs include all shells except the outermost one, but "small-core" ECPs include all except the outermost *two* shells. Although the small-core ECPs are more expensive to use (because more electrons are treated explicitly), it is often found that their enhanced accuracy justifies their use.

When an ECP is constructed, it is usually based on either non-relativistic or quasi-relativistic all-electron calculations. As one might expect, the quasi-relativistic ECPs tend to yield better results than their non-relativistic brethren, especially for atoms beyond the $3d$ block

Q-CHEM's ECP package is integrated with its electron correlation and DFT packages. Of course, no correlation or exchange-correlation energy due to the core electrons is included when using an ECP in a DFT or correlated method, respectively.

The most widely used ECPs today are of the form first proposed by Kahn *et al.* in the 1970s.[48] These model the effects of the core by a one-electron operator $U(r)$ whose matrix elements are simply added to the one-electron Hamiltonian matrix. The ECP operator is given by

$$U(r) = U_L(r) + \sum_{\ell=0}^{L-1} \sum_{m=-l}^{+l} |Y_{\ell m}\rangle\, U_l(r)\, \langle Y_{\ell m}| \tag{8.6}$$

where the radial potentials have the form

$$U_\ell(r) = \sum_{k=1}^{K_\ell} D_{\ell_k}\, r^{n_{\ell_k}}\, e^{-\eta_{\ell_k} r^2} \tag{8.7}$$

and $\sum_m |Y_{\ell m}\rangle\, \langle Y_{\ell m}|$ is the spherical harmonic projector of angular momentum ℓ. In practice, $n_{\ell_k} = -2$, -1 or 0 and L rarely exceeds 5. In addition, $U_L(r)$ contains a Coulombic term N_c/r, where N_c is the number of core electrons.

8.10.2 ECP Fitting

The ECP matrix elements are arguably the most difficult one-electron integrals in existence. Indeed, using current methods, the time taken to compute the ECP integrals can exceed the time taken to compute the far more numerous

electron repulsion integrals. Q-CHEM 5.0 implements a state-of-the-art ECP implementation[56] based on efficient recursion relations and upper bounds. This method relies on a restricted radial potential $U_\ell(r)$, where the radial power is only ever zero, *i.e.* $n = 0$. Whilst true for some ECPs, such as the Stuttgart-Bonn sets, many other ECPs have radial potentials containing $n = -2$ and $n = -1$ terms. To overcome this challenge, we fit these ECP radial potentials using only $n = 0$ terms. Each $n = -2$ and $n = -1$ term is expanded as a sum of three $n = 0$ terms, each with independent contraction coefficient D_{ℓ_k} and Gaussian exponent η_{ℓ_k}. The Gaussian exponents are given by a predetermined recipe and the contraction coefficients are computed in a least squares fitting procedure. The errors introduced by the ECP fitting are insignificant and of the same order as those introduced by numerical integration present in other ECP methods. For the built-in ECPs, fitted variants of each are now provided in the `$QCAUX` directory, *e.g.*, `fit-LANL2DZ`. For user-defined ECPs with $n = -2$ or $n = -1$ terms, Q-CHEM will perform a fit at run time with the additional rem keyword ECP_FIT = TRUE.

8.10.3 Built-In ECPs

8.10.3.1 Introduction

Q-CHEM is equipped with several standard ECP sets which are specified using the ECP keyword within the *$rem* block. The built-in ECPs, which are described in some detail at the end of this Chapter, fall into four families:

- The Hay-Wadt (or Los Alamos) sets (fit-HWMB and fit-LANL2DZ)
- The Stevens-Basch-Krauss-Jansien-Cundari set (fit-SBKJC)
- The Christiansen-Ross-Ermler-Nash-Bursten sets (fit-CRENBS and fit-CRENBL)
- The Stuttgart-Bonn sets (SRLC and SRSC)
- Karlsruhe def2-ECPs, for use with the def2 basis sets

References and information about the definition and characteristics of most of these sets can be found at the Basis Set Exchange:[64]

https://www.basissetexchange.org

Each of the built-in ECPs comes with a matching orbital basis set for the valence electrons. In general, it is advisable to use these together and, if you select a basis set other than the matching one, Q-CHEM will print a warning message in the output file. If you omit the BASIS *$rem* keyword entirely, Q-CHEM will automatically provide the matching one.

The following *$rem* variable controls which ECP is used:

ECP

Defines the effective core potential and associated basis set to be used

TYPE:

STRING

DEFAULT:

No ECP

OPTIONS:

General, Gen — User defined. (*$ecp* keyword required)

Symbol — Use standard ECPs discussed above.

RECOMMENDATION:

ECPs are recommended for first row transition metals and heavier elements. Consult the reviews for more details.

8.10.3.2 Combining ECPs

If you wish, you can use different ECP sets for different elements in the system. This is especially useful if you would like to use a particular ECP but find that it is not available for all of the elements in your molecule. To combine different ECP sets, you set the ECP and BASIS keywords to "GEN" or (equivalently) "GENERAL", and then add a *$ecp* block and a *$basis* block to your input file. In each of these blocks, you must name the ECP and the orbital basis set that you wish to use, separating each element by "****". There is also a built-in combination that can be invoked specifying ECP = fit-LACVP. It automatically assigns 6-31G for atoms H–Ar and fit-LANL2DZ for heavier atoms.

8.10.3.3 Examples

Example 8.6 Computing the HF/fit-LANL2DZ energy of AgCl at a bond length of 2.4 Å.

```
$molecule
   0  1
   Ag
   Cl  Ag  r

   r  =  2.4
$end

$rem
   METHOD   hf           Hartree-Fock calculation
   ECP      fit-lanl2dz  Using the Hay-Wadt ECP
   BASIS    lanl2dz      And the matching basis set
$end
```

Example 8.7 Computing the single point energy of HI with B3LYP/def2-SV(P) (using def2-ECP for I).

```
$molecule
   0 1
   H    0.0    0.0    0.0
   I    0.0    0.0    1.5
$end

$rem
   METHOD           b3lyp
   BASIS            def2-sv(p)
   ECP              def2-ecp
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 8.8 Optimization of the structure of Se_8 using HF/fit-LANL2DZ, followed by a single-point energy calculation at the MP2/fit-LANL2DZ level.

```
$molecule
   0  1
   x1
   x2   x1  xx
   Se1  x1  sx  x2  90.
   Se2  x1  sx  x2  90.  Se1  90.
   Se3  x1  sx  x2  90.  Se2  90.
   Se4  x1  sx  x2  90.  Se3  90.
   Se5  x2  sx  x1  90.  Se1  45.
   Se6  x2  sx  x1  90.  Se5  90.
   Se7  x2  sx  x1  90.  Se6  90.
   Se8  x2  sx  x1  90.  Se7  90.

   xx = 1.2
   sx = 2.8
$end

$rem
   JOBTYPE    opt
   METHOD     hf
   ECP        fit-lanl2dz
   BASIS         lanl2dz
$end

@@@

$molecule
   read
$end

$rem
   METHOD        mp2       MP2 correlation energy
   ECP           fit-lanl2dz   Hay-Wadt ECP and basis
   BASIS         lanl2dz
   SCF_GUESS     read      Read in the MOs
$end
```

Example 8.9 Computing the HF geometry of $CdBr_2$ using the Stuttgart relativistic ECPs. The small-core ECP and basis are employed on the Cd atom and the large-core ECP and basis on the Br atoms.

```
$molecule
   0  1
   Cd
   Br1  Cd  r
   Br2  Cd  r  Br1  180.0

   r = 2.4
$end

$rem
   JOBTYPE    opt   Geometry optimization
   METHOD     hf    Hartree-Fock theory
   ECP        gen   Combine ECPs
   BASIS      gen   Combine basis sets
   PURECART   1     Use pure d functions
$end

$ecp
   Cd
   srsc
   ****
   Br
   srlc
   ****
$end

$basis
   Cd
   srsc
   ****
   Br
   srlc
   ****
$end
```

8.10.4 A Brief Guide to Q-CHEM's Built-In ECPs

8.10.4.1 Introduction

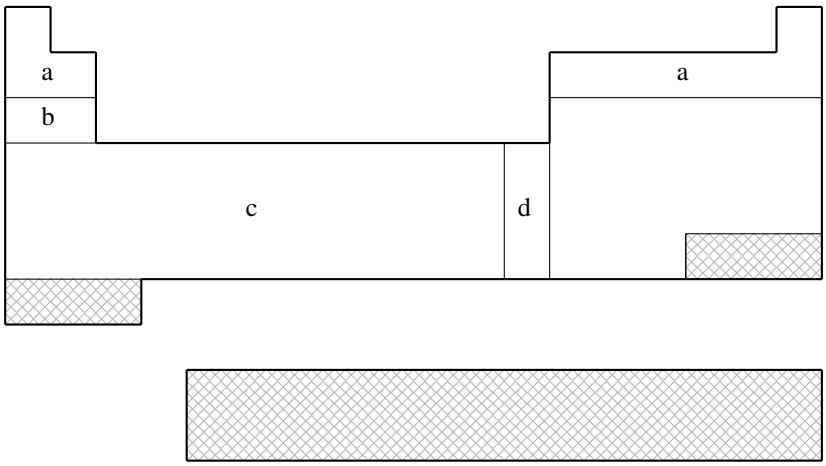
The remainder of this Chapter consists of a brief reference guide to Q-CHEM's built-in ECPs. The ECPs vary in their complexity and their accuracy and the purpose of the guide is to enable the user quickly and easily to decide which ECP to use in a planned calculation.

The following information is provided for each ECP:

- The elements for which the ECP is available in Q-CHEM. This is shown on a schematic Periodic Table by shading all the elements that are *not* supported.
- The literature reference for each element for which the ECP is available in Q-CHEM.
- The matching orbital basis set that Q-CHEM will use for light (*i.e.*. non-ECP atoms). For example, if the user requests SRSC ECPs—which are defined only for atoms beyond argon—Q-CHEM will use the 6-311G basis set for all atoms up to Ar.

- The core electrons that are replaced by the ECP. For example, in the fit-LANL2DZ ECP for the Fe atom, the core is [Ne], indicating that the $1s$, $2s$ and $2p$ electrons are removed.
- The maximum spherical harmonic projection operator that is used for each element. This often, but not always, corresponds to the maximum orbital angular momentum of the core electrons that have been replaced by the ECP. For example, in the fit-LANL2DZ ECP for the Fe atom, the maximum projector is of P-type.
- The number of valence basis functions of each angular momentum type that are present in the matching orbital basis set. For example, in the matching basis for the fit-LANL2DZ ECP for the Fe atom, there the three s shells, three p shells and two d shells. This basis is therefore almost of triple-split valence quality.

8.10.4.2 The fit-HWMB ECP



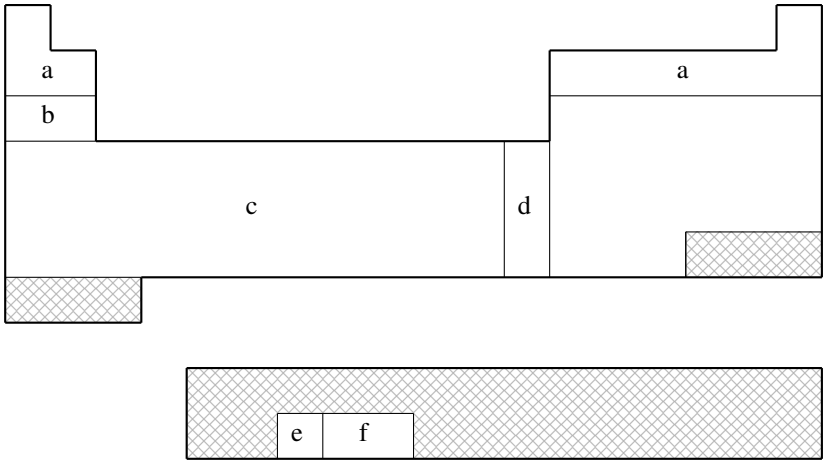
*fit-HWMB is **not** available for shaded elements*

(a) No ECP; Pople STO-3G basis used
(b) Wadt & Hay (Ref. 75)
(c) Hay & Wadt (Ref. 38)
(d) Hay & Wadt (Ref. 37)

Element	**Core**	**Max Projector**	**Valence**
H–He	*none*	*none*	(1s)
Li–Ne	*none*	*none*	(2s,1p)
Na–Ar	[Ne]	*P*	(1s,1p)
K–Ca	[Ne]	*P*	(2s,1p)
Sc–Cu	[Ne]	*P*	(2s,1p,1d)
Zn	[Ar]	*D*	(1s,1p,1d)
Ga–Kr	[Ar]+3d	*D*	(1s,1p)
Rb–Sr	[Ar]+3d	*D*	(2s,1p)
Y–Ag	[Ar]+3d	*D*	(2s,1p,1d)
Cd	[Kr]	*D*	(1s,1p,1d)
In–Xe	[Kr]+4d	*D*	(1s,1p)
Cs–Ba	[Kr]+4d	*D*	(2s,1p)
La	[Kr]+4d	*D*	(2s,1p,1d)
Hf–Au	[Kr]+4d+4f	*F*	(2s,1p,1d)
Hg	[Xe]+4f	*F*	(1s,1p,1d)
Tl–Bi	[Xe]+4f+5d	*F*	(1s,1p)

Table 8.10: Supported elements for the fit-HWMB ECP.

8.10.4.3 The fit-LANL2DZ ECP



*fit-LANL2DZ is **not** available for shaded elements*

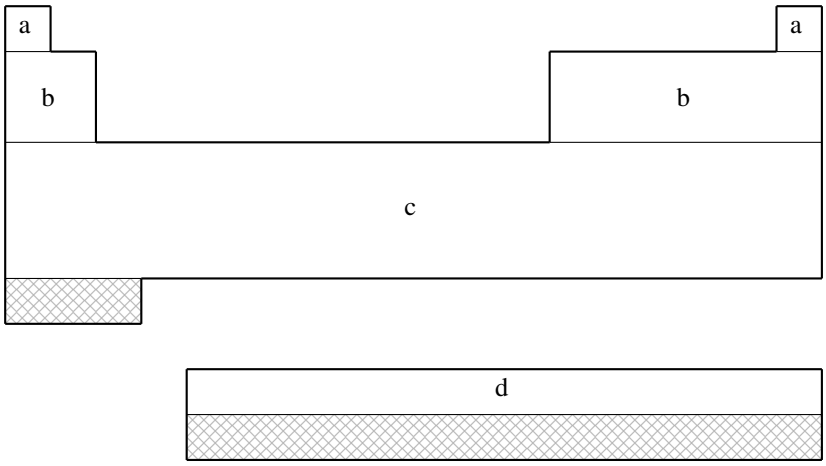
(a) No ECP; Pople 6-31G basis used
(b) Wadt & Hay (Ref. 75)
(c) Hay & Wadt (Ref. 38)
(d) Hay & Wadt (Ref. 37)
(e) Hay (Ref. 35)
(f) Hay & Martin (Ref. 36)

Element	Core	Max Projector	Valence
H–He	*none*	*none*	(2s)
Li–Ne	*none*	*none*	(3s,2p)
Na–Ar	[Ne]	*P*	(2s,2p)
K–Ca	[Ne]	*P*	(3s,3p)
Sc–Cu	[Ne]	*P*	(3s,3p,2d)
Zn	[Ar]	*D*	(2s,2p,2d)
Ga–Kr	[Ar]+3d	*D*	(2s,2p)
Rb–Sr	[Ar]+3d	*D*	(3s,3p)
Y–Ag	[Ar]+3d	*D*	(3s,3p,2d)
Cd	[Kr]	*D*	(2s,2p,2d)
In–Xe	[Kr]+4d	*D*	(2s,2p)
Cs–Ba	[Kr]+4d	*D*	(3s,3p)
La	[Kr]+4d	*D*	(3s,3p,2d)
Hf–Au	[Kr]+4d+4f	*F*	(3s,3p,2d)
Hg	[Xe]+4f	*F*	(2s,2p,2d)
Tl	[Xe]+4f+5d	*F*	(2s,2p,2d)
Pb–Bi	[Xe]+4f+5d	*F*	(2s,2p)
U–Pu	[Xe]+4f+5d	*F*	(3s,3p,2d,2f)

Table 8.11: Supported elements for the fit-LANL2DZ ECP.

Note that Q-CHEM 4.2.2 and later versions also support the LANL2DZ-SV basis, which employs SV basis functions (instead of 6-31G) on H, Li-Ne elements (like some other quantum chemistry packages).

8.10.4.4 The fit-SBKJC ECP



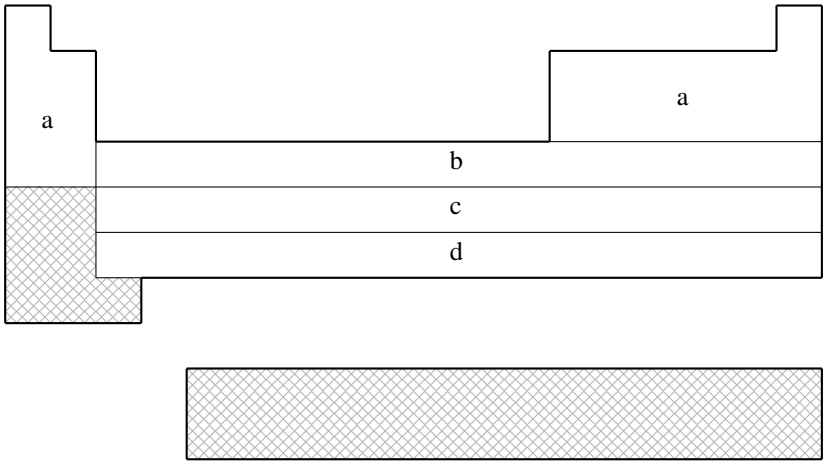
*fit-SBKJC is **not** available for shaded elements*

(a) No ECP; Pople 3-21G basis used
(b) Stevens, Basch, & M. Krauss (Ref. 72)
(c) Stevens, Krauss, Basch, & Jasien (Ref. 73)
(d) Cundari & Stevens (Ref. 9)

Element	Core	Max Projector	Valence
H–He	*none*	*none*	(2s)
Li–Ne	[He]	*S*	(2s,2p)
Na–Ar	[Ne]	*P*	(2s,2p)
K–Ca	[Ar]	*P*	(2s,2p)
Sc–Ga	[Ne]	*P*	(4s,4p,3d)
Ge–Kr	[Ar]+3d	*D*	(2s,2p)
Rb–Sr	[Kr]	*D*	(2s,2p)
Y–In	[Ar]+3d	*D*	(4s,4p,3d)
Sn–Xe	[Kr]+4d	*D*	(2s,2p)
Cs–Ba	[Xe]	*D*	(2s,2p)
La	[Kr]+4d	*F*	(4s,4p,3d)
Ce–Lu	[Kr]+4d	*D*	(4s,4p,1d,1f)
Hf–Tl	[Kr]+4d+4f	*F*	(4s,4p,3d)
Pb–Rn	[Xe]+4f+5d	*F*	(2s,2p)

Table 8.12: Supported elements for the fit-SBKJC ECP.

8.10.4.5 The fit-CRENBS ECP



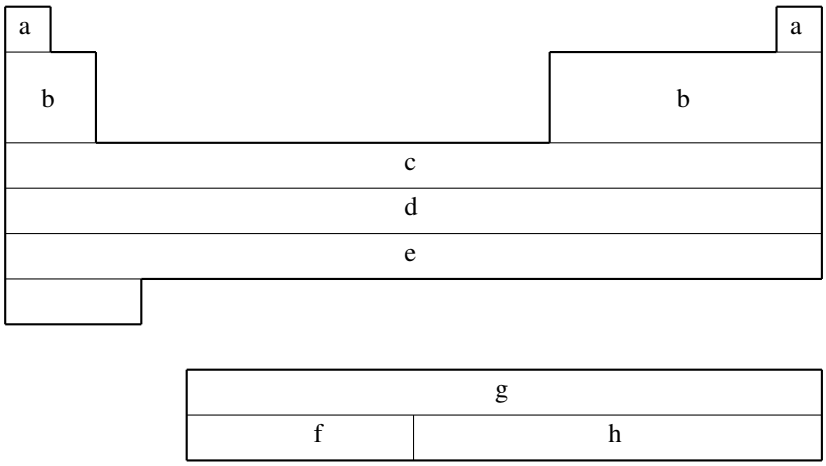
*fit-CRENBS is **not** available for shaded elements*

(a) No ECP; Pople STO-3G basis used
(b) Hurley, Pacios, Christiansen, Ross, & Ermler (Ref. 41)
(c) LaJohn, Christiansen, Ross, Atashroo & Ermler (Ref. 54)
(d) Ross, Powers, Atashroo, Ermler, LaJohn & Christiansen (Ref. 68)
(e) Nash, Bursten, & Ermler (Ref. 57)

Element	Core	Max Projector	Valence
H–He	*none*	*none*	(1s)
Li–Ne	*none*	*none*	(2s,1p)
Na–Ar	*none*	*none*	(3s,2p)
K–Ca	*none*	*none*	(4s,3p)
Sc–Zn	[Ar]	*P*	(1s,0p,1d)
Ga–Kr	[Ar]+3d	*D*	(1s,1p)
Y–Cd	[Kr]	*D*	(1s,1p,1d)
In–Xe	[Kr]+4d	*D*	(1s,1p)
La	[Xe]	*D*	(1s,1p,1d)
Hf–Hg	[Xe]+4f	*F*	(1s,1p,1d)
Tl–Rn	[Xe]+4f+5d	*F*	(1s,1p)

Table 8.13: Supported elements for the fit-CRENBS ECP.

8.10.4.6 The fit-CRENBL ECP

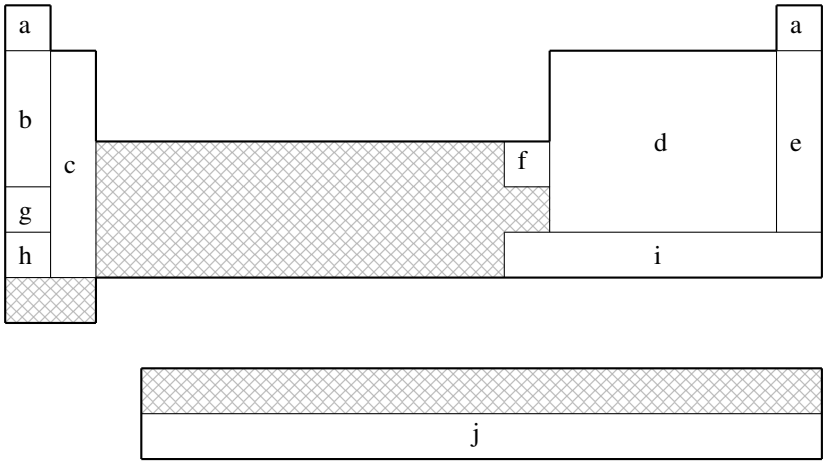


(a) No ECP; Pople 6-311G basis used
(b) Pacios & Christiansen (Ref. 59)
(c) Hurley, Pacios, Christiansen, Ross, & Ermler (Ref. 41)
(d) LaJohn, Christiansen, Ross, Atashroo, & Ermler (Ref. 54)
(e) Ross, Powers, Atashroo, Ermler, LaJohn, & Christiansen (Ref. 68)
(f) Ermler, Ross, & Christiansen (Ref. 23)
(g) Ross, Gayen, & Ermler (Ref. 69)
(h) Nash, Bursten, & Ermler (Ref. 57)

Element	Core	Max Projector	Valence
H–He	*none*	*none*	(3s)
Li–Ne	[He]	S	(4s,4p)
Na–Mg	[He]	S	(6s,4p)
Al–Ar	[Ne]	P	(4s,4p)
K–Ca	[Ne]	P	(5s,4p)
Sc–Zn	[Ne]	P	(7s,6p,6d)
Ga–Kr	[Ar]	P	(3s,3p,4d)
Rb–Sr	[Ar]+3d	D	(5s,5p)
Y–Cd	[Ar]+3d	D	(5s,5p,4d)
In–Xe	[Kr]	D	(3s,3p,4d)
Cs–La	[Kr]+4d	D	(5s,5p,4d)
Ce–Lu	[Xe]	D	(6s,6p,6d,6f)
Hf–Hg	[Kr]+4d+4f	F	(5s,5p,4d)
Tl–Rn	[Xe]+4f	F	(3s,3p,4d)
Fr–Ra	[Xe]+4f+5d	F	(5s,5p,4d)
Ac–Pu	[Xe]+4f+5d	F	(5s,5p,4d,4f)
Am–Lr	[Xe]+4f+5d	F	(0s,2p,6d,5f)

Table 8.14: Supported elements for the fit-CRENBL ECP.

8.10.4.7 The SRLC ECP



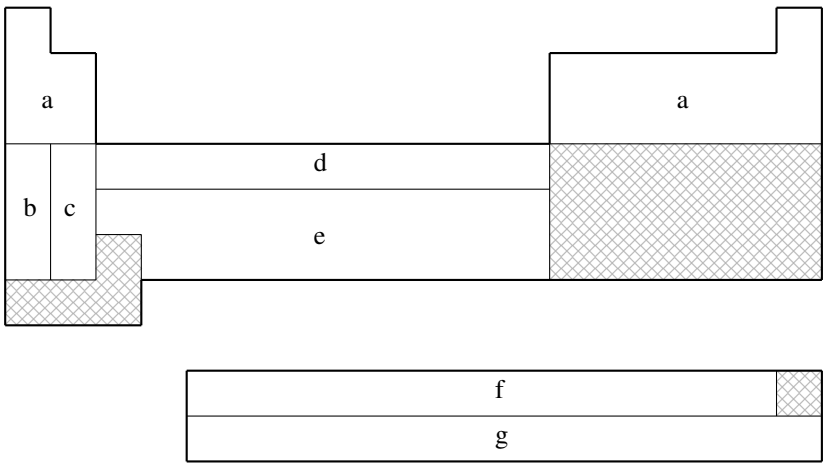
*SRLC is **not** available for shaded elements*

(a) No ECP; Pople 6-31G basis used
(b) Fuentealba, Preuss, Stoll, & von Szentpály (Ref. 28)
(c) Fuentealba, von Szentpály, Preuss, & Stoll (Ref. 30)
(d) Bergner, Dolg, Küchle, Stoll, & Preuss (Ref. 5)
(e) Nicklass, Dolg, Stoll, & Preuss, (Ref. 58)
(f) Schautz, Flad, & Dolg (Ref. 71)
(g) Fuentealba, Stoll, von Szentpály, Schwerdtfeger, & Preuss (Ref. 29)
(h) von Szentpály, Fuentealba, Preuss, & Stoll (Ref. 74)
(i) Küchle, Dolg, Stoll, & Preuss (Ref. 51)
(j) Küchle (Ref. 53)

Element	Core	Max Projector	Valence
H–He	*none*	*none*	(2s)
Li–Be	[He]	P	(2s,2p)
B–N	[He]	D	(2s,2p)
O–F	[He]	D	(2s,3p)
Ne	[He]	D	(4s,4p,3d,1f)
Na–P	[Ne]	D	(2s,2p)
S–Cl	[Ne]	D	(2s,3p)
Ar	[Ne]	F	(4s,4p,3d,1f)
K–Ca	[Ar]	D	(2s,2p)
Zn	[Ar]+3d	D	(3s,2p)
Ga–As	[Ar]+3d	F	(2s,2p)
Se–Br	[Ar]+3d	F	(2s,3p)
Kr	[Ar]+3d	G	(4s,4p,3d,1f)
Rb–Sr	[Kr]	D	(2s,2p)
In–Sb	[Kr]+4d	F	(2s,2p)
Te–I	[Kr]+4d	F	(2s,3p)
Xe	[Kr]+4d	G	(4s,4p,3d,1f)
Cs–Ba	[Xe]	D	(2s,2p)
Hg–Bi	[Xe]+4f+5d	G	(2s,2p,1d)
Po–At	[Xe]+4f+5d	G	(2s,3p,1d)
Rn	[Xe]+4f+5d	G	(2s,2p,1d)
Ac–Lr	[Xe]+4f+5d	G	(5s,5p,4d,3f,2g)

Table 8.15: Supported elements for the SRLC ECP.

8.10.4.8 The SRSC ECP



*SRSC is **not** available for shaded elements*

(a) No ECP; Pople 6-311G basis used
(b) Leininger, Nicklass, Küchle, Stoll, Dolg, & Bergner (Ref. 55)
(c) Kaupp, Schleyer, Stoll, & Preuss (Ref. 49)
(d) Dolg, Wedig, Stoll, & Preuss (Ref. 18)
(e) Andrae, Häußermann, Dolg, Stoll, & Preuss (Ref. 3)
(f) Dolg, Stoll, & Preuss (Ref. 17)
(g) Küchle, Dolg, Stoll, & Preuss (Ref. 52)
(h) Dolg, Stoll, Preuss, & Pitzer (Ref. 19)

Element	**Core**	**Max Projector**	**Valence**
H–He	*none*	*none*	(3s)
Li–Ne	*none*	*none*	(4s,3p,1d)
Na–Ar	*none*	*none*	(6s,5p,1d)
K	[Ne]	*F*	(5s,4p)
Ca	[Ne]	*F*	(4s,4p,2d)
Sc–Zn	[Ne]	*D*	(6s,5p,3d)
Rb	[Ar]+3d	*F*	(5s,4p)
Sr	[Ar]+3d	*F*	(4s,4p,2d)
Y–Cd	[Ar]+3d	*F*	(6s,5p,3d)
Cs	[Kr]+4d	*F*	(5s,4p)
Ba	[Kr]+4d	*F*	(3s,3p,2d,1f)
Ce–Yb	[Ar]+3d	*G*	(5s,5p,4d,3f)
Hf–Pt	[Kr]+4d+4f	*G*	(6s,5p,3d)
Au	[Kr]+4d+4f	*F*	(7s,3p,4d)
Hg	[Kr]+4d+4f	*G*	(6s,6p,4d)
Ac–Lr	[Kr]+4d+4f	*G*	(8s,7p,6d,4f)

Table 8.16: Supported elements for the SRSC ECP.

8.10.4.9 The Karlsruhe "def2" ECPs

For elements Rb–Rn, all the Karlsruhe "def2" basis sets are paired with a common set of ECPs.[76] It is briefly summarized in the table below (the number of valence basis functions depend on the basis set in use, so it is not presented):

Element	**Core**	**Max Projector**
H–Kr	*none*	*none*
Rb–Xe	[Ar]+3d	*D*
Cs–La	[Kr]+4d	*D*
Hf–Rn	[Kr]+4d+4f	*D*

Table 8.17: Supported elements for the def2 ECP.

8.10.5 User-Defined ECPs

Many users will find that the library of built-in ECPs is adequate for their needs. However, if you need to use an ECP that is not built into Q-CHEM, you can enter it in much the same way as you can enter a user-defined orbital basis set; see Chapter 8.

To apply a user-defined ECP, you must set the ECP and BASIS keywords in *$rem* to GEN. You then add a *$ecp* block that defines your ECP, element by element, and a *$basis* block that defines your orbital basis set, separating elements by asterisks.

The syntax within the *$basis* block is described in Chapter 8. The syntax for each record within the *$ecp* block is as follows:.

$ecp
For each atom that will bear an ECP:
    Chemical symbol for the atom
    ECP name; the L value for the ECP; number of core electrons removed
    For each ECP component (in the order unprojected, $\hat{P}_0$, $\hat{P}_1$, ..., $\hat{P}_{L-1}$):
        The component name
        The number of Gaussians in the component
        For each Gaussian in the component
            The power of r ; the exponent ; the contraction coefficient
A sequence of four asterisks (*i.e.*, $\star\star\star\star$)
$end

Note:

1. All of the information in the *$ecp* block is case-insensitive.
2. The power of r (which includes the Jacobian r^2 factor) must be 0, 1, or 2.
3. If an r^0 or r^1 term is included you must include the rem keyword "ECP_FIT = TRUE".

Example 8.10 Optimizing the HF geometry of AlH_3 using a user-defined ECP and basis set on Al and the 3-21G basis on H.

```
$molecule
   0  1
   Al
   H1  Al  r
   H2  Al  r  H1  120.0
   H3  Al  r  H1  120.0  H2  180.0

   r = 1.6
$end

$rem
   JOBTYPE    opt   Geometry optimization
   METHOD     hf    Hartree-Fock theory
   ECP        gen   User-defined ECP
   BASIS      gen   User-defined basis
   ECP_FIT    true
   geom_opt_driver optimize
$end

$ecp
   Al
   Stevens_ECP  2  10
   d potential
     1
     1   1.95559  -3.03055
   s-d potential
     2
     0   7.78858   6.04650
     2   1.99025  18.87509
   p-d potential
     2
     0   2.83146   3.29465
     2   1.38479   6.87029
****
$end

$basis
   Al
   SP 3  1.00
      0.90110  -0.30377  -0.07929
      0.44950   0.13382   0.16540
      0.14050   0.76037   0.53015
   SP 1  1.00
      0.04874   0.32232   0.47724
****
   H
   3-21G
****
$end
```

8.10.6 ECPs and Electron Correlation

The ECP package is integrated with the electron correlation package and it is therefore possible to apply any of Q-CHEM's post-Hartree-Fock methods to systems in which some of the atoms may bear pseudopotentials. Of course, the correlation energy contribution arising from core electrons that have been replaced by an ECP is *not* included. In this sense, correlation energies with ECPs are comparable to correlation energies from frozen-core calculations. However, the use of ECPs effectively removes both core electrons *and* the corresponding virtual (unoccupied) orbitals.

In a DFT calculation with ECPs, the exchange-correlation energy is obtained entirely from the non-core electrons. This will be satisfactory if there are no chemically important cores/valence effects but may introduce significant errors if not, particularly if you are using a "large-core" ECP. Any of the local, gradient-corrected and hybrid functionals discussed in Chapter 5 may be used and you may also perform ECP calculations with user-defined hybrid functionals.

Example 8.11 Optimization of the structure of Se_8 using HF/fit-LANL2DZ, followed by a single-point energy calculation at the MP2/fit-LANL2DZ level.

```
$molecule
   0  1
   x1
   x2   x1  xx
   Se1  x1  sx  x2  90.
   Se2  x1  sx  x2  90.  Se1  90.
   Se3  x1  sx  x2  90.  Se2  90.
   Se4  x1  sx  x2  90.  Se3  90.
   Se5  x2  sx  x1  90.  Se1  45.
   Se6  x2  sx  x1  90.  Se5  90.
   Se7  x2  sx  x1  90.  Se6  90.
   Se8  x2  sx  x1  90.  Se7  90.

   xx = 1.2
   sx = 2.8
$end

$rem
   JOBTYPE     opt
   METHOD      hf
   ECP         fit-lanl2dz
   BASIS       lanl2dz
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE       sp        Single-point energy
   METHOD        mp2       MP2 correlation energy
   ECP           fit-lanl2dz   Hay-Wadt ECP and basis
   BASIS         lanl2dz
   SCF_GUESS     read      Read in the MOs
$end
```

8.10.7 Forces and Vibrational Frequencies with ECPs

It is important to be able to optimize geometries using pseudopotentials and for this purpose Q-CHEM contains analytical first derivatives of the nuclear potential energy term for ECPs.

The ECP package is also integrated with the vibrational analysis package and it is therefore possible to compute the vibrational frequencies (and hence the infrared and Raman spectra) of systems in which some of the atoms may bear ECPs. Starting with Q-CHEM version 5.0, fully analytical second derivatives of ECPs are available.

Example 8.12 Structure and vibrational frequencies of TeO_2 using Hartree-Fock theory and the Stuttgart relativistic large-core ECPs. Note that the vibrational frequency job reads both the optimized structure and the molecular orbitals from the geometry optimization job that precedes it.

```
$molecule
   0  1
   Te
   O1  Te  r
   O2  Te  r  O1  a

   r = 1.8
   a = 108
$end

$rem
   JOBTYPE    opt
   METHOD     hf
   ECP        srlc
   BASIS      srlc
   geom_opt_driver optimize
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE     freq
   METHOD      hf
   ECP         srlc
   BASIS      srlc
   SCF_GUESS   read
$end
```

References and Further Reading

[1] Basis sets were obtained from the Extensible Computational Chemistry Environment Basis Set Database, Version 1.0, as developed and distributed by the Molecular Science Computing Facility, Environmental and Molecular Sciences Laboratory which is part of the Pacific Northwest Laboratory, P.O. Box 999, Richland, Washington 99352, USA, and funded by the U.S. Department of Energy. The Pacific Northwest Laboratory is a multiprogram laboratory operated by Battelle Memorial Institute for the U.S. Department of Energy under contract DE-AC06-76RLO 1830. Contact David Feller, Karen Schuchardt or Don Jones for further information.

[2] J. Almlöf and O. Gropen. Relativistic effects in chemistry. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 8, chapter 4, pages 203–244. Wiley-VCH, New York, 1996. DOI: 10.1002/9780470125854.ch4.

[3] D. Andrae, U. Häußermann, M. Dolg, H. Stoll, and H. Preuß. *Theor. Chem. Acc.*, 77:123, 1990. DOI: 10.1007/BF01114537.

[4] N. B. Balabanov and K. A. Peterson. *J. Chem. Phys.*, 123:064107, 2005. DOI: 10.1063/1.1998907.

[5] A. Bergner, M. Dolg, W. Küchle, H. Stoll, and H. Preuß. *Mol. Phys.*, 80:1431, 1993. DOI: 10.1080/00268979300103121.

[6] J. S. Binkley, J. A. Pople, and W. J. Hehre. *J. Am. Chem. Soc.*, 102:939, 1980. DOI: 10.1021/ja00523a008.

[7] S. F. Boys and F. Bernardi. *Mol. Phys.*, 19:553, 1970. DOI: 10.1080/00268977000101561.

[8] P. A. Christiansen, W. C. Ermler, and K. S. Pitzer. *Annu. Rev. Phys. Chem.*, 36:407, 1985. DOI: 10.1146/annurev.pc.36.100185.002203.

[9] T. R. Cundari and W. J. Stevens. *J. Chem. Phys.*, 98:5555, 1993. DOI: 10.1063/1.464902.

[10] T. R. Cundari, M. T. Benson, M. L. Lutz, and S. O. Sommerer. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 8, chapter 3, pages 145–202. Wiley-VCH, New York, 1996. DOI: 10.1002/9780470125854.ch3.

[11] E. R. Davidson. *Chem. Phys. Lett.*, 260:514, 1996. DOI: 10.1016/0009-2614(96)00917-7.

[12] E. R. Davidson and D. Feller. *Chem. Rev.*, 86:681, 1986. DOI: 10.1021/cr00074a002.

[13] E. V. R. de Castro and F. E. Jorge. *J. Chem. Phys.*, 108:5225, 1998. DOI: 10.1063/1.475959.

[14] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 7:359, 1986. DOI: 10.1002/jcc.540070216.

[15] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 8:861, 1987. DOI: 10.1002/jcc.540080614.

[16] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 8:880, 1987. DOI: 10.1002/jcc.540080615.

[17] M. Dolg and H. Preuss. *J. Chem. Phys.*, 90:1730, 1989. DOI: 10.1063/1.456066.

[18] M. Dolg, U. Wedig, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 86:866, 1987. DOI: 10.1063/1.452288.

[19] M. Dolg, H. Stoll, H. Preuss, and R. M. Pitzer. *J. Phys. Chem.*, 97:5852, 1993. DOI: 10.1021/j100124a012.

[20] B. I. Dunlap. *Phys. Chem. Chem. Phys.*, 2:2113, 2000. DOI: 10.1039/b000027m.

[21] T. H. Dunning Jr. *J. Chem. Phys.*, 55:716, 1971. DOI: 10.1063/1.1676139.

[22] T. H. Dunning Jr. *J. Chem. Phys.*, 90:1007, 1989. DOI: 10.1063/1.456153.

[23] W. C. Ermler, R. B. Ross, and P. A. Christiansen. *Int. J. Quantum Chem.*, 40:829, 1991. DOI: 10.1002/qua.560400611.

[24] D. Feller and E. R. Davidson. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 1, chapter 1, pages 1–43. Wiley-VCH, New York, 1990. DOI: 10.1002/9780470125786.ch1.

[25] M. Feyereisen, G. Fitzgerald, and A. Komornicki. *Chem. Phys. Lett.*, 208:359, 1993. DOI: 10.1016/0009-2614(93)87156-W.

[26] G. Frenking, I. Antes, M. Boehme, S. Dapprich, A. W. Ehlers, V. Jonas, A. Neuhaus, M. Otto, R. Stegmann, A. Veldkamp, and S. F. Vyboishchikov. Pseudopotential calculations of transition metal compounds: Scope and limitations. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 8, chapter 2, pages 63–144. Wiley-VCH, New York, 1996. DOI: 10.1002/9780470125854.ch2.

[27] M. J. Frisch, J. A. Pople, and J. S. Binkley. *J. Chem. Phys.*, 80:3265, 1984. DOI: 10.1063/1.447079.

[28] P. Fuentealba, H. Preuss, H. Stoll, and L. von Szentpály. *Chem. Phys. Lett.*, 89:418, 1982. DOI: 10.1016/0009-2614(82)80012-2.

[29] P. Fuentealba, H. Stoll, L. von Szentpály, P. Schwerdtfeger, and H. Preuss. *J. Phys. B: At. Mol. Opt. Phys.*, 16: L323, 1983. DOI: 10.1088/0022-3700/16/11/001.

[30] P. Fuentealba, L. von Szentpály, H. Preuss, and H. Stoll. *J. Phys. B: At. Mol. Opt. Phys.*, 18:1287, 1985. DOI: 10.1088/0022-3700/18/7/010.

[31] E. D. Glendening and D. Feller. *J. Phys. Chem.*, 99:3060, 1995. DOI: 10.1021/j100010a015.

[32] M. S. Gordon and T. R. Cundari. *Coord. Chem. Rev.*, 147:87, 1996. DOI: 10.1016/0010-8545(95)01133-1.

[33] M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre. *J. Am. Chem. Soc.*, 104:2797, 1982. DOI: 10.1021/ja00374a017.

[34] M. Gray and J. M. Herbert. *J. Chem. Theory Comput.*, 18:2308, 2022. DOI: 10.1021/acs.jctc.1c01302.

[35] P. J. Hay. *J. Chem. Phys.*, 79:5469, 1983. DOI: 10.1063/1.445665.

[36] P. J. Hay and R. L. Martin. *J. Chem. Phys.*, 109:3875, 1998. DOI: 10.1063/1.476988.

[37] P. J. Hay and W. R. Wadt. *J. Chem. Phys.*, 82:270, 1985. DOI: 10.1063/1.448799.

[38] P. J. Hay and W. R. Wadt. *J. Chem. Phys.*, 82:299, 1985. DOI: 10.1063/1.448975.

[39] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople. *Ab Initio Molecular Orbital Theory*. Wiley, New York, 1986.

[40] J. G. Hill and K. A. Peterson. *J. Chem. Phys.*, 147:244106, 2017. DOI: 10.1063/1.5010587.

[41] M. M. Hurley, L. F. Pacios, and P. A. Christiansen. *J. Chem. Phys.*, 84:6840, 1986. DOI: 10.1063/1.450689.

[42] S. Huzinaga. *Comp. Phys. Rep.*, 2:281, 1985. DOI: 10.1016/0167-7977(85)90003-6.

[43] F. Jensen. *Introduction to Computational Chemistry*. Wiley, New York, 1994.

[44] F. Jensen. *J. Chem. Theory Comput.*, 4:719, 2008. DOI: 10.1021/ct800013z.

[45] F. Jensen. *Theor. Chem. Acc.*, 126:371, 2010. DOI: 10.1007/s00214-009-0699-5.

[46] F. Jensen. *J. Chem. Theory Comput.*, 10:1074, 2014. DOI: 10.1021/ct401026a.

[47] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon. *Proc. Natl. Acad. Sci. USA*, 102:6692, 2005. DOI: 0.1073/pnas.0408475102.

[48] L. R. Kahn and W. A. Goddard III. *J. Chem. Phys.*, 56:2685, 1972. DOI: 10.1063/1.1677597.

[49] M. Kaupp, P. v. R. Schleyer, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 94:1360, 1991. DOI: 10.1063/1.459993.

[50] R. Krishnan, J. S. Binkley, R. Seeger, and J. A. Pople. *J. Chem. Phys.*, 72:650, 1980. DOI: 10.1063/1.438955.

[51] W. Küchle, M. Dolg, H. Stoll, and H. Preuss. *Mol. Phys.*, 74:1245, 1991. DOI: 10.1080/00268979100102941.

[52] W. Küchle, M. Dolg, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 100:7535, 1994. DOI: 10.1063/1.466847.

[53] Wolfgang Küchle. Pseudopotential für Aktinoide. Diplomarbeit, Universität Stuttgart, Stuttgart, Germany, 1993.

[54] L. A. LaJohn, P. A. Christiansen, R. B. Ross, T. Atashroo, and W. C. Ermler. *J. Chem. Phys.*, 87:2812, 1987. DOI: 10.1063/1.453069.

[55] T. Leininger, A. Nicklass, W. Küchle, H. Stoll, M. Dolg, and A. Bergner. *Chem. Phys. Lett.*, 255:274, 1996. DOI: 10.1016/0009-2614(96)00382-X.

[56] S. C. McKenzie, E. Epifanovsky, G. M. J. Barca, A. T. B. Gilbert, and P. M. W. Gill. *J. Phys. Chem. A*, 122:3066, 2018. DOI: 10.1021/acs.jpca.7b12679.

[57] C. S. Nash, B. E. Bursten, and W. C. Ermler. *J. Chem. Phys.*, 106:5133, 1997. DOI: 10.1063/1.473992.

[58] A. Nicklass, M. Dolg, H. Stoll, and H. Preuss. *J. Chem. Phys.*, 102:8942, 1995. DOI: 10.1063/1.468948.

[59] L. F. Pacios and P. A. Christiansen. *J. Chem. Phys.*, 82:2664, 1985. DOI: 10.1063/1.448263.

[60] E. Papajak and D. G. Truhlar. *J. Chem. Theory Comput.*, 7:10, 2011. DOI: 10.1021/ct1005533.

[61] K. A. Peterson and C. Puzzarini. *Theor. Chem. Acc.*, 114:283, 2005. DOI: 10.1007/s00214-005-0681-9.

[62] K. A. Peterson, D. Figgen, E. Goll, H. Stoll, and M. Dolg. *J. Chem. Phys.*, 119:11113, 2003. DOI: 10.1063/1.1622924.

[63] B. P. Prascher, D. E. Woon, K. A. Peterson, T. H. Dunning Jr., and A. K. Wilson. *Theor. Chem. Acc.*, 128:69, 2011. DOI: 10.1007/s00214-010-0764-0.

[64] B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibson, and T. L. Windus. *J. Chem. Inf. Model*, 59:4814, 2019. DOI: 10.1021/acs.jcim.9b00725.

[65] P. Pyykko. *Chem. Rev.*, 88:563, 1988. DOI: 10.1021/cr00085a006.

[66] D. Rappoport and F. Furche. *J. Chem. Phys.*, 133:134105, 2010. DOI: 10.1063/1.3484283.

[67] R. M. Richard, K. U. Lao, and J. M. Herbert. *J. Phys. Chem. Lett.*, 4:2674, 2013. DOI: 10.1021/jz401368u.

[68] R. B. Ross, J. M. Powers, T. Atashroo, W. C. Ermler, L. A. LaJohn, and P. A. Christiansen. *J. Chem. Phys.*, 93: 6654, 1990. DOI: 10.1063/1.458934.

[69] R. B. Ross, S. Gayen, and W. C. Ermler. *J. Chem. Phys.*, 100:8145, 1994. DOI: 10.1063/1.466809.

[70] A. Schäfer, H. Horn, and R. Ahlrichs. *J. Chem. Phys.*, 97:2571, 1992. DOI: 10.1063/1.463096.

[71] F. Schautz, H.-J. Flad, and M. Dolg. *Theor. Chem. Acc.*, 99:231, 1998. DOI: 10.1007/s002140050331.

[72] W. J. Stevens, H. Basch, and M. Krauss. *J. Chem. Phys.*, 81:6026, 1984. DOI: 10.1063/1.447604.

[73] W. J. Stevens, M. Krauss, H. Basch, and P. G. Jasien. *Can. J. Chem.*, 70:612, 1992. DOI: 10.1139/v92-085.

[74] L. von Szentpály, P. Fuentealba, H. Preuss, and H. Stoll. *Chem. Phys. Lett.*, 93:555, 1982. DOI: 10.1016/0009-2614(82)83728-7.

[75] W. R. Wadt and P. J. Hay. *J. Chem. Phys.*, 82:284, 1985. DOI: 10.1063/1.448800.

[76] F. Weigend and R. Ahlrichs. *Phys. Chem. Chem. Phys.*, 7:3297, 2005. DOI: 10.1039/b508541a.

[77] F. Weigend, M. Häser, H. Patzelt, and R. Ahlrichs. *Chem. Phys. Lett.*, 294:143, 1998. DOI: 10.1016/S0009-2614(98)00862-8.

[78] A. F. White, M. Head-Gordon, and C. W. McCurdy. *J. Chem. Phys.*, 142:054103, 2015. DOI: 10.1063/1.4906940.

[79] A. K. Wilson, D. E. Woon, K. A. Peterson, and T. H. Dunning Jr. *J. Chem. Phys.*, 110:7667, 1999. DOI: 10.1063/1.478678.

[80] D. E. Woon and T. H. Dunning Jr. *J. Chem. Phys.*, 98:1358, 1993. DOI: 10.1063/1.464303.

[81] E. Papajak J. Zheng, X. Xu, H. R. Leverentz, and D. G. Truhlar. *J. Chem. Theory Comput.*, 7:3027, 2011. DOI: 10.1021/ct200106a.

Chapter 9

Exploring Potential Energy Surfaces: Searches for Critical Points and Molecular Dynamics

9.1 Equilibrium Geometries and Transition-State Structures with Q-CHEM

9.1.1 Introduction

Molecular potential energy surfaces rely on the Born-Oppenheimer separation of nuclear and electronic motion. Of particular interest are the critical points on these surfaces, *i.e.* where the gradient of the energy vanishes. Characterization of a critical point requires consideration of the eigenvalues of the Hessian (second derivative matrix) calculated at that point. An equilibrium geometry corresponds to a critical point where the eigenvalues of the Hessian are all positive, whereas a transition-state structure is defined as a first-order saddle point, and therefore has a Hessian with precisely one negative eigenvalue. The latter is a local maximum along the reaction path between the minima corresponding to the reactants and products, and a minimum in all directions perpendicular to this reaction path.

The quality of a geometry optimization algorithm is of major importance; even the fastest integral code in the world will be useless if combined with an inefficient optimization algorithm that requires excessive numbers of steps to converge. Q-CHEM is currently transitioning to a new geometry optimization driver, LIBOPT3, which improves on the older OPTIMIZE driver. Details on the difference in capabilities between the two drivers are provided in Table 9.1.

The key to optimizing a molecular geometry successfully is to proceed from the starting geometry to the final geometry in as few steps as possible. Four factors influence the path and number of steps:

- starting geometry
- coordinate system
- optimization algorithm
- quality of the Hessian (and gradient)

Q-CHEM controls the last three of these, but the starting geometry is solely determined by the user, and the closer it is to the converged geometry, the fewer optimization steps will be required. Decisions regarding the optimization algorithm

Feature	LIBOPT3	OPTIMIZE
Optimization Type:		
Unconstrained	✓	✓
Constrained	✗	✓
Structure Search:		
Equilibrium	✓	✓
Transition	✓	✓
Coordinates:		
Cartesian	✓	✓
Z-matrix	✓	✓
Redundant Internal	✓	–
Delocalized Internal	✓	✓
Optimization Algorithms:		
Gradient Based	✓	–
Quasi-Newton	✓	✓
Newton	✓	–
Step Generation Algorithms:		
Line Search (LS)	✓	–
Eigenvector Following (EF)	✓	✓
QM/MM	✗	✓
GDIIS algorithms	✗	✓
Hessian-Free Characterization	✗	✓
Intrinsic Reaction Coordinates (IRC)	✗	✓
Minimum-Energy Crossing Points (MECPs)	✓	✓
Harmonic Confiner	✓	✓

Table 9.1: Differences of available features within the two optimization drivers, as controlled by GEOM_OPT_DRIVER in the *$rem* section. For unconstrained optimization, the default is GEOM_OPT_DRIVER = LIBOPT3, whereas Q-CHEM's original driver is selected using GEOM_OPT_DRIVER = OPTIMIZE.

and the coordinate system are generally made by the LIBOPT3 and OPTIMIZE drivers (*i.e.*, internally, within Q-CHEM) to maximize the rate of convergence. Although users may override these choices, caution should be exercised when doing so as changes may significantly impact the computational cost.

Q-CHEM provides the capability to optimize a molecule using Cartesian, *Z*-matrix, redundant internal, or delocalized internal coordinates. The last two of these are generated automatically from the Cartesian coordinates, and delocalized internal coordinates are usually the best choice. These coordinates were developed by Baker *et al.*,[10] and can be considered as an extension of the natural internal coordinates developed by Pulay *et al.*[39,84] and the redundant optimization method of Pulay and Fogarasi.[82]

The heart of the geometry optimization in Q-CHEM (for both minima and transition states) is Baker's eigenvector-following (EF) algorithm.[5] This was developed following the work of Cerjan and Miller,[21] and of Simons and coworkers.[12,95] The Hessian mode-following option incorporated into this algorithm is capable of locating a transition state by walking uphill from the associated minimum. By following the lowest Hessian mode, the EF algorithm can locate a transition state starting from any reasonable input geometry and Hessian.

An additional option available for minimization is Pulay's GDIIS algorithm,[27] which is based on the well known DIIS technique for accelerating SCF convergence.[81] GDIIS must be specifically requested, as the EF algorithm is the default.

Level of Theory (Algorithm)	**Analytical Gradients**	**Maximum Angular Momentum Type**	**Analytical Hessian**	**Maximum Angular Momentum Type**
HF/DFT	✓	k	✓	k
ROHF	✓	h	✗	
canonical ROHF	✗		✗	
RI-MP2	✓	h	✗	
CCSD	✓	h	✗	
CIS/TDDFT (except RO)	✓	h	✓	g
EOM-CCSD	✓	h	✗	
ADC(n)	✗		✗	

Table 9.2: Gradients and Hessians currently available for geometry optimizations with maximum angular momentum types for analytical derivative calculations. Higher angular momentum are supported when GEN_SCFMAN = TRUE.

Q-CHEM incorporates a very accurate and efficient Lagrange multiplier algorithm for constrained optimization. This was originally developed for use with Cartesian coordinates[6,8] and can handle constraints that are not necessarily satisfied by the starting geometry. The Lagrange multiplier approach has been modified for use with delocalized internal coordinates,[7] which is much more efficient and is now the default within Q-CHEM. The Lagrange multiplier code can locate constrained transition states as well as minima.

Another consideration when trying to minimize the total optimization time concerns the quality of the gradient and Hessian. A higher-quality Hessian (*i.e.*, analytical versus approximate) will in many cases lead to faster convergence, in the sense of requiring fewer optimization steps. This is why by default for transition-state optimization with LIBOPT3 the exact Hessian will be calculated. However, the construction of an analytical Hessian requires significant computational effort and may outweigh the advantage of fewer optimization cycles. Currently available analytical gradients and Hessians are summarized in Table 9.2.

Features of Q-CHEM's geometry and transition-state optimization capabilities include:

- Cartesian, *Z*-matrix or internal coordinate systems
- Eigenvector Following (EF) or GDIIS algorithms
- Constrained optimizations
- Equilibrium structure searches
- Transition structure searches
- Hessian-free characterization of stationary points
- Initial Hessian and Hessian update options
- Reaction pathways using intrinsic reaction coordinates (IRC)
- Optimization of minimum-energy crossing points (MECPs) along conical seams

9.1.2 Theoretical Background

Geometry optimization refers to the determination of stationary points, principally minima and transition states, on molecular potential energy surfaces. It is an iterative process, requiring the calculation of the energy, gradient and (possibly) Hessian at each optimization cycle. The energy, gradient and Hessian information is used to compute an

optimization step, $\mathbf{h}$, which displaces the geometry to be closer to the target stationary point. This process is repeated until convergence is attained. The art of geometry optimization lies in calculating the step, $\mathbf{h}$, so as to converge in as few cycles as possible.

Consider the energy, $E(\mathbf{x}_0)$ at some point $\mathbf{x}_0$ on a potential energy surface. We can express the energy at a nearby point $\mathbf{x} = \mathbf{x}_0 + \mathbf{h}$ by means of the Taylor series

$$E(\mathbf{x}_0 + \mathbf{h}) = E(\mathbf{x}_0) + \mathbf{h}^t \left(\frac{dE(\mathbf{x}_0)}{d\mathbf{x}} \right) + \frac{1}{2}\mathbf{h}^t \left(\frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2} \right) \mathbf{h} + \cdots \tag{9.1}$$

If we knew the exact form of the energy functional $E(\mathbf{x})$ and all its derivatives, we could move from the current point $\mathbf{x}_0$ directly to a stationary point, (*i.e.*, we would know exactly what the step $\mathbf{h}$ ought to be). Since we typically know only the lower derivatives of $E(\mathbf{x})$ at best, then we can estimate the step $\mathbf{h}$ by differentiating the Taylor series with respect to $\mathbf{h}$, keeping only the first few terms on the right hand side, and setting the left hand side, $dE(\mathbf{x}_0 + \mathbf{h})/d\mathbf{h}$, to zero, which is the value it would have at a genuine stationary point. Thus

$$\frac{dE(\mathbf{x}_0 + \mathbf{h})}{d\mathbf{h}} = \frac{dE(\mathbf{x}_0)}{d\mathbf{x}} + \left(\frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2} \right) \mathbf{h} + \text{higher terms (ignored)} \tag{9.2}$$

from which

$$\mathbf{h} = -\mathbf{H}^{-1}\mathbf{g} \tag{9.3}$$

where

$$\frac{dE}{d\mathbf{x}} \equiv \mathbf{g} \text{ (gradient vector)}, \qquad \frac{d^2E}{d\mathbf{x}_1 d\mathbf{x}_2} \equiv \mathbf{H} \text{ (Hessian matrix)} \tag{9.4}$$

Equation (9.3) is known as the Newton-Raphson step. It is the major component of almost all geometry optimization algorithms in quantum chemistry.

The above derivation assumed exact first (gradient) and second (Hessian) derivative information. Analytical gradients are available for most methodologies supported in Q-CHEM, however, analytical second derivatives are less likely to be available, see Table 9.2. Furthermore, even if they were available, it would not necessarily be advantageous to use them as their evaluation is computationally demanding, and efficient optimizations can be performed without an exact Hessian. An excellent compromise in practice is to begin with an approximate Hessian matrix, and update this using gradient and displacement information generated as the optimization progresses. In this way the starting Hessian can be "improved" at essentially no cost. Using Eq. (9.3) with an approximate Hessian is called the quasi Newton-Raphson step.

The nature of the Hessian matrix (in particular its eigenvalue structure) plays a crucial role in a successful optimization. All stationary points on a potential energy surface have a zero gradient vector, however, the character of the stationary point (*i.e.*, what type of structure it corresponds to) is determined by the Hessian matrix. Diagonalization of the Hessian yields a set of mutually orthogonal directions on the energy surface (the eigenvectors) together with the curvature along these directions (the eigenvalues). At a local minimum (corresponding to a well in the potential energy surface) the curvature along all of these directions must be positive, reflecting the fact that a small displacement along any of these directions causes the energy to rise. At a transition state, the curvature is negative (*i.e.*, the energy is a maximum) along one direction, but positive along all other directions. Thus, for a stationary point to be a transition state, the Hessian matrix at that point must have one, and only one, negative eigenvalue, while for a minimum the Hessian must have all positive eigenvalues. In the latter case the Hessian is called *positive definite*. If searching for a minimum it is important that the Hessian matrix be positive definite; in fact, unless the Hessian is positive definite there is no guarantee that the step predicted by Eq. (9.3) is will actually lower the energy. Similarly, for a transition state search, the Hessian must have one negative eigenvalue. Maintaining the correct Hessian eigenvalue structure is not difficult for minimization, but can be problematic when searching for a transition state.

In a diagonal Hessian representation the Newton-Raphson step can be written

$$\mathbf{h} = -\sum_i \left(\frac{F_i}{b_i} \right) \mathbf{u}_i \tag{9.5}$$

where $\mathbf{u}_i$ and b_i are the eigenvectors and eigenvalues of the Hessian matrix $\mathbf{H}$ and $F_i = \mathbf{u}_i^t\mathbf{g}$ is the component of $\mathbf{g}$ along the local direction (eigenmode) $\mathbf{u}_i$. As discussed by Simons *et al.*,[95] the Newton-Raphson step can be considered as minimizing along directions $\mathbf{u}_i$, which have positive eigenvalues, and maximizing along directions with negative eigenvalues. Thus, if the user is searching for a minimum and the Hessian matrix is positive definite, then the Newton-Raphson step is appropriate since it is attempting to minimize along all directions simultaneously. However, if the Hessian has one or more negative eigenvalues, then the basic Newton-Raphson step is not appropriate for a minimum search, since it will be maximizing and not minimizing along one or more directions. Exactly the same arguments apply during a transition state search except that the Hessian must have one negative eigenvalue, because the user has to maximize along one direction. However, there must be *only* one negative eigenvalue. A positive definite Hessian is a disaster for a transition state search because the Newton-Raphson step will head towards a minimum.

If firmly in a region of the potential energy surface with the right Hessian character, then a careful search (based on the Newton-Raphson step) will almost always lead to a stationary point of the correct type. However, this is only true if the Hessian is exact. If the Hessian is being updated approximately, then there is no guarantee that the Hessian eigenvalue structure will be preserved from one cycle to the next unless one is very careful during the update. Updating procedures that "guarantee" conservation of a positive definite Hessian do exist (or at least warn the user if the update is likely to introduce negative eigenvalues). This can be very useful during a minimum search; but there are no such guarantees for preserving the Hessian character required for a transition state.

In addition to the difficulties in retaining the correct Hessian character, there is the matter of obtaining a "correct" Hessian in the first instance. This is particularly acute for a transition state search. For a minimum search it is possible to "guess" a reasonable, positive-definite starting Hessian (for example, by carrying out a molecular mechanics minimization and calculating the Hessian, also using molecular mechanics, at the minimum) but this option is usually not available for transition states. Even if the user calculates the Hessian exactly at the starting geometry, the guess for the structure may not be sufficiently accurate, and the expensive, exact Hessian may not have the desired eigenvalue structure.

Consequently, particularly for a transition state search, an alternative to the basic Newton-Raphson step is clearly needed, especially when the Hessian matrix is inappropriate for the stationary point being sought.

One of the first algorithms that was capable of taking corrective action during a transition state search if the Hessian had the wrong eigenvalue structure, was developed by Poppinger,[78] who suggested that, instead of taking the Newton-Raphson step, if the Hessian had all positive eigenvalues, the lowest Hessian mode be followed uphill; whereas, if there were two or more negative eigenvalues, the mode corresponding to the least negative eigenvalue be followed downhill. While this step should lead the user back into the right region of the energy surface, it has the disadvantage that the user is maximizing or minimizing along one mode only, unlike the Newton-Raphson step which maximizes/minimizes along all modes simultaneously. Another drawback is that successive such steps tend to become linearly dependent, which degrades most of the commonly used Hessian updates.

9.1.3 Eigenvector-Following (EF) Algorithm

The work of Cerjan and Miller,[21] and later Simons and co-workers,[12,95] showed that there was a better step than simply directly following one of the Hessian eigenvectors. A simple modification to the Newton-Raphson step is capable of guiding the search away from the current region towards a stationary point with the required characteristics. This is

$$\mathbf{h} = -\sum_i \left(\frac{F_i}{b_i - \lambda}\right)\mathbf{u}_i \tag{9.6}$$

in which λ can be regarded as a shift parameter on the Hessian eigenvalue b_i. Scaling the Newton-Raphson step in this manner effectively directs the step to lie primarily, but not exclusively (unlike Poppinger's algorithm[78]), along one of

the local eigenmodes, depending on the value chosen for λ. References 12,21,95 all use the same basic approach of Eq. (9.6) but differ in the means of determining the value of λ.

The EF algorithm[5] uses the rational function approach presented in Refs. 12, yielding an eigenvalue equation of the form

$$\begin{pmatrix} \mathbf{H} & \mathbf{g} \\ \mathbf{g}^t & 0 \end{pmatrix} \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} \tag{9.7}$$

from which a suitable λ can be obtained. Expanding Eq. (9.7) yields

$$(\mathbf{H} - \lambda)\mathbf{h} + \mathbf{g} = 0 \tag{9.8}$$

and

$$\mathbf{g}^t\mathbf{h} = \lambda \tag{9.9}$$

In terms of a diagonal Hessian representation, Eq. (9.8) rearranges to Eq. (9.6), and substitution of Eq. (9.6) into the diagonal form of Eq. (9.9) gives

$$\lambda = -\sum_i \left(\frac{-F_i^2}{b_i - \lambda} \right) \tag{9.10}$$

which can be used to evaluate λ iteratively.

The eigenvalues, λ, of the RFO equation Eq. (9.7) have the following important properties:[12]

- The $(n+1)$ values of λ bracket the n eigenvalues of the Hessian matrix $\lambda_i < b_i < \lambda_{i+1}$.
- At a stationary point, one of the eigenvalues, λ, of Eq. (9.7) is zero and the other n eigenvalues are those of the Hessian at the stationary point.
- For a saddle point of order m, the zero eigenvalue separates the m negative and the $(n-m)$ positive Hessian eigenvalues.

This last property, the separability of the positive and negative Hessian eigenvalues, enables two shift parameters to be used, one for modes along which the energy is to be maximized and the other for which it is minimized. For a transition state (a first-order saddle point), in terms of the Hessian eigenmodes, we have the two matrix equations

$$\begin{pmatrix} b_1 & F_1 \\ F_1 & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ 1 \end{pmatrix} \tag{9.11}$$

$$\begin{pmatrix} b_2 & & & F_2 \\ & \ddots & \mathbf{0} & \vdots \\ & \mathbf{0} & b_n & F_n \\ F_2 & \cdots & F_n & 0 \end{pmatrix} \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} \tag{9.12}$$

where it is assumed that we are maximizing along the lowest Hessian mode $\mathbf{u}_1$. Note that λ_p is the highest eigenvalue of Eq. (9.11), which is always positive and approaches zero at convergence, and λ_n is the lowest eigenvalue of Eq. (9.12), which it is always negative and again approaches zero at convergence.

Choosing these values of λ gives a step that attempts to maximize along the lowest Hessian mode, while at the same time minimizing along all the other modes. It does this regardless of the Hessian eigenvalue structure (unlike the Newton-Raphson step). The two shift parameters are then used in Eq. (9.6) to give the final step

$$\mathbf{h} = -\left(\frac{F_1}{b_1 - \lambda_p} \right) \mathbf{u}_1 + \sum_{i=2}^{n} \left(\frac{F_i}{b_i - \lambda_n} \right) \mathbf{u}_i \tag{9.13}$$

If this step is greater than the maximum allowed, it is scaled down. For minimization only one shift parameter, λ_n, is used which acts on all modes.

In Eq. (9.11)) and Eq. (9.12) it was assumed that the step would maximize along the lowest Hessian mode, b_1, and minimize along all the higher modes. However, it is possible to maximize along modes other than the lowest, and in this way potentially locate transition states for alternative rearrangements/dissociations from the same initial starting point. For maximization along the kth mode (instead of the lowest mode), Eq. (9.11) is replaced by

$$\begin{pmatrix} b_k & F_k \\ F_k & 0 \end{pmatrix} \begin{pmatrix} h_k \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_k \\ 1 \end{pmatrix} \tag{9.14}$$

and Eq. (9.12) now excludes the kth mode, but includes the lowest mode. Since what was originally the kth mode is the mode along which the negative eigenvalue is required, then this mode will eventually become the lowest mode at some stage of the optimization. To ensure that the original mode is being followed smoothly from one cycle to the next, the mode that is actually followed is the one with the greatest overlap with the mode followed on the previous cycle. This procedure is known as *mode following*. For more details and some examples, see Ref. 5.

9.1.4 Delocalized Internal Coordinates

The choice of coordinate system can have a major influence on the rate of convergence during a geometry optimization. For complex potential energy surfaces with many stationary points, a different choice of coordinates can even result in convergence to a different final structure.

The key attribute of a good set of coordinates for geometry optimization is the degree of coupling between the individual coordinates. In general, the less coupling the better, as variation of one particular coordinate will then have minimal impact on the other coordinates. Coupling manifests itself primarily as relatively large partial derivative terms between different coordinates. For example, a strong harmonic coupling between two different coordinates, i and j, results in a large off-diagonal element, H_{ij}, in the Hessian matrix. Normally this is the only type of coupling that can be directly "observed" during an optimization, as third and higher derivatives are ignored in almost all optimization algorithms.

In the early days of computational quantum chemistry geometry optimizations were carried out in Cartesian coordinates. They are an obvious choice as they can be defined for all systems and gradients and second derivatives are calculated directly in Cartesian coordinates. Unfortunately, Cartesian coordinates are often heavily coupled, making them a poor choice for optimizations. Despite this, Cartesians have recently returned to favor because of their generality, and because it has been clearly demonstrated that if reliable second derivative information is available (*i.e.*, a good starting Hessian) and the initial geometry is reasonable, then Cartesians can be as efficient as any other coordinate set for small to medium-sized molecules.[8,9] Without good Hessian data, however, Cartesians are inefficient, especially for long chain acyclic systems.

In the 1970s Cartesians were replaced by *Z*-matrix coordinates. Initially the *Z*-matrix was used simply as a means of geometry input; it is far easier to describe a molecule in terms of bond lengths, bond angles and dihedral angles (the natural coordinates for a chemist) than to develop a suitable set of Cartesian coordinates. It was subsequently found that optimization was generally more efficient in *Z*-matrix coordinates than in Cartesians, especially for acyclic systems. This is not always the case, and care must be taken in constructing a suitable *Z*-matrix. A good general rule is ensure that each variable is defined in such a way that changing its value will not change the values of any of the other variables. A brief discussion concerning good *Z*-matrix construction strategy is given by Schlegel.[88]

In 1979 Pulay *et al.* published a key paper introducing what were termed natural internal coordinates into geometry optimization.[84] These coordinates involve the use of individual bond displacements as stretching coordinates, but linear combinations of bond angles and torsions as deformational coordinates. Suitable linear combinations of bends and torsions (the two are considered separately) are selected using group theoretical arguments based on local

pseudo-symmetry. For example, bond angles around an sp^3 hybridized carbon atom are all approximately tetrahedral, regardless of the groups attached, and idealized tetrahedral symmetry can be used to generate deformational coordinates around the central carbon atom.

The major advantage of natural internal coordinates in geometry optimization is their ability to significantly reduce the coupling, both harmonic and anharmonic, between the various coordinates. Compared to natural internals, *Z*-matrix coordinates arbitrarily omit some angles and torsions (to prevent redundancy), and this can induce strong anharmonic coupling between the coordinates, especially with a poorly constructed *Z*-matrix. Another advantage of the reduced coupling is that successful minimizations can be carried out in natural internals with only an approximate (*e.g.*, diagonal) Hessian provided at the starting geometry. A good starting Hessian is still needed for a transition state search.

Despite their clear advantages, natural internals have only become widely used more recently. This is because, when used in the early programs, it was necessary for the user to define them. This situation changed in 1992 with the development of computational algorithms capable of automatically generating natural internals from input Cartesians.[39] For minimization, natural internals have become the coordinates of first choice.[8,39]

There are some disadvantages to natural internal coordinates as they are commonly constructed and used:

- Algorithms for the automatic construction of natural internals are complicated. There are a large number of structural possibilities, and to adequately handle even the most common of them can take several thousand lines of code.
- For the more complex molecular topologies, most assigning algorithms generate more natural internal coordinates than are required to characterize all possible motions of the system (*i.e.*, the generated coordinate set contains redundancies).
- In cases with a very complex molecular topology (*e.g.*, multiply fused rings and cage compounds) the assigning algorithm may be unable to generate a suitable set of coordinates.

The redundancy problem has been addressed in an excellent paper by Pulay and Fogarasi,[82] who have developed a scheme for carrying out geometry optimization directly in the redundant coordinate space.

Baker *et al.*[10] developed a set of delocalized internal coordinates that eliminate all of the above-mentioned difficulties. Building on some of the ideas in the redundant optimization scheme of Pulay and Fogarasi,[82] delocalized internals form a complete, non-redundant set of coordinates which are as good as, if not superior to, natural internals, and which can be generated in a simple and straightforward manner for essentially any molecular topology, no matter how complex.

Consider a set of n internal coordinates $\mathbf{q} = (q_1, q_2, \ldots q_n)^t$ Displacements $\Delta\mathbf{q}$ in $\mathbf{q}$ are related to the corresponding Cartesian displacements $\Delta\mathbf{x}$ by means of the usual Wilson $\mathbf{B}$-matrix,[107]

$$\Delta\mathbf{q} = \mathbf{B}\Delta\mathbf{x} \tag{9.15}$$

If any of the internal coordinates $\mathbf{q}$ are redundant, then the rows of the $\mathbf{B}$-matrix will be linearly dependent.

Delocalized internal coordinates are obtained by constructing and diagonalizing the matrix $\mathbf{G} = \mathbf{B}\mathbf{B}^t$. Diagonalization of $\mathbf{G}$ results in two sets of eigenvectors; a set of m (typically $3N - 6$, where N is the number of atoms) eigenvectors with eigenvalues $\lambda > 0$, and a set of nm eigenvectors with eigenvalues $\lambda = 0$ (to numerical precision). In this way, any redundancies present in the original coordinate set $\mathbf{q}$ are isolated (they correspond to those eigenvectors with zero eigenvalues). The eigenvalue equation of $\mathbf{G}$ can thus be written

$$\mathbf{G}(\mathbf{UR}) = (\mathbf{UR}) \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} \tag{9.16}$$

where $\mathbf{U}$ is the set of non-redundant eigenvectors of $\mathbf{G}$ (those with $\lambda > 0$) and $\mathbf{R}$ is the corresponding redundant set.

The nature of the original set of coordinates $\mathbf{q}$ is unimportant, as long as it spans all the degrees of freedom of the system under consideration. We include in $\mathbf{q}$, all bond stretches, all planar bends and all proper torsions that can be generated based on the atomic connectivity. These individual internal coordinates are termed *primitives*. This blanket approach generates far more primitives than are necessary, and the set $\mathbf{q}$ contains much redundancy. This is of little concern, as solution of Eq. (9.16) takes care of all redundancies.

Note that eigenvectors in both $\mathbf{U}$ and $\mathbf{R}$ will each be linear combinations of potentially all the original primitives. Despite this apparent complexity, we take the set of non-redundant vectors $\mathbf{U}$ as our working coordinate set. Internal coordinates so defined are much more delocalized than natural internal coordinates (which are combinations of a relatively small number of bends or torsions) hence, the term delocalized internal coordinates.

It may appear that because delocalized internals are such a complicated mixing of the original primitive internals, they are a poor choice for use in an actual optimization. On the contrary, arguments can be made that delocalized internals are, in fact, the "best" possible choice, certainly at the starting geometry. The interested reader is referred to the original literature for more details.[10]

The situation for geometry optimization, comparing Cartesian, Z-matrix and delocalized internal coordinates, and assuming a "reasonable" starting geometry, is as follows:

- For small or very rigid medium-sized systems (up to about 15 atoms), optimizations in Cartesian and internal coordinates ("good" Z-matrix or delocalized internals) should perform similarly.
- For medium-sized systems (say 15–30 atoms) optimizations in Cartesians should perform as well as optimizations in internal coordinates, provided a reliable starting Hessian is available.
- For large systems (30+ atoms), unless these are very rigid, neither Cartesian nor Z-matrix coordinates can compete with delocalized internals, even with good quality Hessian information. As the system increases, and with less reliable starting geometries, the advantage of delocalized internals only increases.

There is one particular situation in which Cartesian coordinates may be the best choice. Natural internal coordinates (and by extension delocalized internals) show a tendency to converge to low energy structures.[8] This is because steps taken in internal coordinate space tend to be much larger when translated into Cartesian space As a result, higher-energy local minima tend to be "jumped over", especially if there is no reliable Hessian information available (which is generally not needed for a successful optimization). Consequently, if the user is looking for a local minimum (*i.e.*, a meta-stable structure) and has both a good starting geometry and a decent Hessian, the user should carry out the optimization in Cartesian coordinates.

9.1.5 Constrained Optimization

Constrained optimization refers to the optimization of molecular structures in which certain parameters (*e.g.*, bond lengths, bond angles or dihedral angles) are fixed. In quantum chemistry calculations, this has traditionally been accomplished using Z-matrix coordinates, with the desired parameter set in the Z-matrix and simply omitted from the optimization space. In 1992, Baker presented an algorithm for constrained optimization directly in Cartesian coordinates.[6] Baker's algorithm used both penalty functions and the classical method of Lagrange multipliers,[38] and was developed in order to impose constraints on a molecule obtained from a graphical model builder as a set of Cartesian coordinates. Some improvements widening the range of constraints that could be handled were made in 1993.[8] Q-CHEM includes the latest version of this algorithm, which has been modified to handle constraints directly in delocalized internal coordinates.[7]

The essential problem in constrained optimization is to minimize a function of n variables $F(\mathbf{x})$ subject to a series of m constraints of the form $C_i(\mathbf{x}) = 0$ for $i = \ell, \ldots, m$. Assuming $m < n$, then perhaps the best way to proceed (if

this were possible in practice) would be to use the m constraint equations to eliminate m variables, and then solve the resulting unconstrained problem in terms of the $n-m$ independent variables. This is exactly what occurs in a *Z*-matrix optimization. Such an approach cannot be used in Cartesian coordinates as standard distance and angle constraints are non-linear functions of the appropriate coordinates. For example, a distance constraint between atoms i and j is, in Cartesians, given by $(R_{ij} - R_0) = 0$, with

$$R_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2\right]^{1/2} \tag{9.17}$$

and R_0 is the constrained distance. This obviously cannot be satisfied by elimination. What can be eliminated in Cartesians are the individual x, y and z coordinates themselves and in this way individual atoms can be totally or partially frozen.

Internal constraints can be handled in Cartesian coordinates by introducing the Lagrangian function

$$L(\mathbf{x}, \lambda) = F(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i C_i(\mathbf{x}) \tag{9.18}$$

which replaces the function $F(\mathbf{x})$ in the unconstrained case. Here, the λ_i are the so-called Lagrange multipliers, one for each constraint $C_i(\mathbf{x})$. Differentiating Eq. (9.18) with respect to $\mathbf{x}$ and λ affords

$$\frac{dL(\mathbf{x}, \lambda)}{dx_j} = \frac{dF(\mathbf{x})}{dx_j} - \sum_{i=1}^{m} \lambda_i \left(\frac{dC_i(\mathbf{x})}{dx_j}\right) \tag{9.19}$$

$$\frac{dL(\mathbf{x}, \lambda)}{d\lambda_i} = -C_i(\mathbf{x}) \tag{9.20}$$

At a stationary point of the Lagrangian we have $\hat{\boldsymbol{\nabla}}\mathbf{L} = \mathbf{0}$, *i.e.*, all $dL/dx_j = 0$ and all $dL/d\lambda_i = 0$. This latter condition means that all $C_i(\mathbf{x}) = 0$ and thus all constraints are satisfied. Hence, finding a set of values $(\mathbf{x}, \lambda)$ for which $\hat{\boldsymbol{\nabla}}\mathbf{L} = \mathbf{0}$ will give a possible solution to the constrained optimization problem in exactly the same way as finding an $\mathbf{x}$ for which $\mathbf{g} = \hat{\boldsymbol{\nabla}}\mathbf{F} = \mathbf{0}$ gives a solution to the corresponding unconstrained problem.

The Lagrangian second derivative matrix, which is the analogue of the Hessian matrix in an unconstrained optimization, is given by

$$\hat{\nabla}^2\mathbf{L} = \begin{pmatrix} \dfrac{d^2L(\mathbf{x}, \lambda)}{dx_j dx_k} & \dfrac{d^2L(\mathbf{x}, \lambda)}{dx_j d\lambda_i} \\ \dfrac{d^2L(\mathbf{x}, \lambda)}{dx_j d\lambda_i} & \dfrac{d^2L(\mathbf{x}, \lambda)}{d\lambda_j d\lambda_i} \end{pmatrix} \tag{9.21}$$

where

$$\frac{d^2L(\mathbf{x}, \lambda)}{dx_j dx_k} = \frac{d^2F(\mathbf{x})}{dx_j dx_k} - \sum_{i=1}^{m} \lambda_i \left(\frac{d^2C_i(\mathbf{x})}{dx_j dx_k}\right) \tag{9.22}$$

$$\frac{d^2L(\mathbf{x}, \lambda)}{dx_j d\lambda_i} = -\left(\frac{dC_i(\mathbf{x})}{dx_j}\right) \tag{9.23}$$

$$\frac{d^2L(\mathbf{x}, \lambda)}{d\lambda_j d\lambda_i} = 0\ . \tag{9.24}$$

Thus, in addition to the standard gradient vector and Hessian matrix for the unconstrained function $F(\mathbf{x})$, we need both the first and second derivatives (with respect to coordinate displacement) of the constraint functions. Once these quantities are available, the corresponding Lagrangian gradient, (Eq. (9.19)), and Lagrangian second derivative matrix, (Eq. (9.21)), can be formed, and the optimization step calculated in a similar manner to that for a standard unconstrained optimization.[6]

In the Lagrange multiplier method, the unknown multipliers, λ_i, are an integral part of the parameter set. This means that the optimization space consists of all n variables $\mathbf{x}$ plus all m Lagrange multipliers λ, one for each constraint.

The total dimension of the constrained optimization problem, $n + m$, has thus increased by m compared to the corresponding unconstrained case. The Lagrangian Hessian matrix, $\hat{\nabla}^2\mathbf{L}$, has m extra modes compared to the standard (unconstrained) Hessian matrix, $\hat{\nabla}^2\mathbf{F}$. What normally happens is that these additional modes are dominated by the constraints (*i.e.*, their largest components correspond to the constraint Lagrange multipliers) and they have negative curvature (a negative Hessian eigenvalue). This is perhaps not surprising when one realizes that any motion in the parameter space that breaks the constraints is likely to lower the energy.

Compared to a standard unconstrained minimization, where a stationary point is sought at which the Hessian matrix has all positive eigenvalues, in the constrained problem we are looking for a stationary point of the Lagrangian function at which the Lagrangian Hessian matrix has as many negative eigenvalues as there are constraints (*i.e.*, we are looking for an mth-order saddle point). For further details and practical applications of constrained optimization using Lagrange multipliers in Cartesian coordinates; see Ref. 6.

Eigenvector following can be implemented in a constrained optimization in a similar way to the unconstrained case. Considering a constrained minimization with m constraints, then Eq. (9.11) is replaced by

$$\begin{pmatrix} b_1 & & & F_1 \\ & \ddots & \mathbf{0} & \vdots \\ & \mathbf{0} & b_m & F_m \\ F_1 & \cdots & F_m & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} \tag{9.25}$$

and Eq. (9.12) by

$$\begin{pmatrix} b_{m+1} & & & F_{m+1} \\ & \ddots & \mathbf{0} & \vdots \\ & \mathbf{0} & b_{m+n} & F_{m+n} \\ F_{m+1} & \cdots & F_{m+n} & 0 \end{pmatrix} \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} \tag{9.26}$$

where b_i are now the eigenvalues of $\hat{\nabla}^2\mathbf{L}$, with corresponding eigenvectors $\mathbf{u}_i$, and $F_i = \mathbf{u}_i^t\hat{\nabla}\mathbf{L}$. Here Eq. (9.25) includes the m constraint modes along which a negative Lagrangian Hessian eigenvalue is required, and Eq. (9.26) includes all the other modes.

Equations (9.25) and (9.26) implement eigenvector following for a constrained minimization. Constrained transition state searches can be carried out by selecting one extra mode to be maximized in addition to the m constraint modes, *i.e.*, by searching for a saddle point of the Lagrangian function of order $m + \ell$.

It should be realized that, in the Lagrange multiplier method, the desired constraints are only satisfied at convergence, and not necessarily at intermediate geometries. The Lagrange multipliers are part of the optimization space; they vary just as any other geometrical parameter and, consequently the degree to which the constraints are satisfied changes from cycle to cycle, approaching 100% satisfied near convergence. One advantage this brings is that, unlike in a standard *Z*-matrix approach, constraints do not have to be satisfied in the starting geometry.

Imposed constraints can normally be satisfied to very high accuracy, 10^{-6} or better. However, problems can arise for both bond and dihedral angle constraints near 0° and 180° and, instead of attempting to impose a single constraint, it is better to split angle constraints near these limiting values into two by using a dummy atom,[8] exactly analogous to splitting a 180° bond angle into two 90° angles in a *Z*-matrix.

Note: Exact 0° and 180° single angle constraints cannot be imposed, as the corresponding constraint normals, $\hat{\nabla}\mathbf{C}_i$, are zero, and result in rows and columns of zeros in the Lagrangian Hessian matrix.

9.1.6 Constrained Optimization in Delocalized Internal Coordinates

Imposing constraints in delocalized internal coordinates is far simpler than in Cartesian, so we will give no further details on the latter.

At first glance, imposing any constraints in delocalized coordinates appears problematic, given that each coordinate is potentially a linear combination of all possible primitives. However, this is deceptive, and in fact all standard constraints can be imposed by a relatively simple Schmidt orthogonalization procedure. Consider a unit vector with only one non-zero component corresponding to the primitive internal (stretch, bend or torsion) that one wishes to keep constant. This vector is projected on to the full set, $\mathbf{U}$, of active delocalized coordinates and normalized. All n delocalized coordinates are then Schmidt orthogonalized with respect to this normalized, projected constraint vector. The last coordinate orthogonalized in this way will be linearly dependent on the constraint vector and previous $n-1$ active vectors, and can be removed from the active space.

In more detail, the procedure is as follows (taken directly from Ref. 10). The initial (usually unit) constraint vector $\mathbf{C}$ is projected on to the set $\mathbf{U}$ of delocalized internal coordinates according to

$$\mathbf{C}^{\text{proj}} = \sum_{k=1}^{n} \big\langle \mathbf{C} \big| \mathbf{C}\mathbf{U}_k\mathbf{U}_k \big\rangle \mathbf{U}_k \ , \tag{9.27}$$

where the summation is over all active coordinates $\mathbf{U}_k$. The projected vector $\mathbf{C}^{\text{proj}}$ is then normalized and an $n+l$ dimensional vector space $\mathbf{V}$ is formed, comprising the normalized, projected constraint vector together with all active delocalized coordinates

$$\mathbf{V} = \big\{ \mathbf{C}^{\text{proj}},\ \mathbf{U}_k (k = 1, \ldots, n) \big\} \ . \tag{9.28}$$

This set of vectors is Schmidt orthogonalized according to the standard procedure,

$$\widetilde{\mathbf{V}}_k = \alpha_k \left(\mathbf{V}_k - \sum_{\ell=1}^{k-1} \big\langle \mathbf{V}_k \big| \mathbf{V}_k \widetilde{\mathbf{V}}_\ell \widetilde{\mathbf{V}}_\ell \big\rangle \widetilde{\mathbf{V}}_\ell \right) \ , \tag{9.29}$$

where the first vector is $\mathbf{V}_1$, is taken to be $\mathbf{C}^{\text{proj}}$. The coefficient α_k is a normalization factor. As noted above, the last vector taken, $\mathbf{V}_{n+1} \equiv \mathbf{U}_k$, will drop out, leaving a fully orthonormal set of $n-1$ active vectors and one constraint vector.

After the Schmidt orthogonalization the constraint vector will contain all the weight in the active space of the primitive to be fixed, which will have a zero component in all of the other $n-1$ vectors. The fixed primitive has thus been isolated entirely in the constraint vector which can now be removed from the active subspace for the geometry optimization step.

Extension of the above procedure to multiple constraints is straightforward. In addition to constraints on individual primitives, it is also possible to impose combinatorial constraints. For example, if, instead of a unit vector, one started the constraint procedure with a vector in which two components were set to unity, then this would impose a constraint in which the sum of the two relevant primitives were always constant. In theory any desired linear combination of any primitives could be constrained.

Note further that imposed constraints are not confined to those primitive internals generated from the initial atomic connectivity. If we wish to constrain a distance, angle or torsion between atoms that are not formally connected, then all we need to do is add that particular coordinate to our primitive set. It can then be isolated and constrained in exactly the same way as a formal connectivity constraint.

Everything discussed thus far regarding the imposition of constraints in delocalized internal coordinates has involved isolating each constraint in one vector which is then eliminated from the optimization space. This is very similar in effect to a *Z*-matrix optimization, in which constraints are imposed by elimination. This, of course, can only be done if the desired constraint is satisfied in the starting geometry. We have already seen that the Lagrange multiplier algorithm, used to impose distance, angle and torsion constraints in Cartesian coordinates, can be used even when the constraint is not satisfied initially. The Lagrange multiplier method can also be used with delocalized internals, and its implementation with internal coordinates brings several simplifications and advantages.

As already noted, standard internal constraints (bond distances, angles and torsions) are non-linear functions of the Cartesian coordinates of the atoms involved. A torsion, for example, which involves four atoms, is a function of twelve

different coordinates. In internals each constraint is a coordinate in its own right and is therefore a simple linear function of a single coordinate.

If we denote a general internal coordinate by R, then the constraint function $C_i(\mathbf{R})$ is a function of one coordinate, R_i, and it and its derivatives can be written

$$C_i(R_i) = R_i - R_0 \tag{9.30}$$

$$dC_i(R_i)/dR_i = 1; \quad dC_i(R_i)/dR_j = 0 \tag{9.31}$$

$$d^2C_i(R_i)/dR_i dR_j = 0 \tag{9.32}$$

where R_0 is the desired value of the constrained coordinate, and R_i is its current value. From Eq. (9.31) we see that the constraint normals, $dC_i(\mathbf{R})/dR_i$, are simply unit vectors and the Lagrangian Hessian matrix, Eq. (9.21), can be obtained from the normal Hessian matrix by adding m columns (and m rows) of unit vectors.

A further advantage, in addition to the considerable simplification, is the handling of 0° and 180° dihedral angle constraints. In Cartesian coordinates it is not possible to formally constrain bond angles and torsions to exactly 0° or 180° because the corresponding constraint normal is a zero vector. Similar difficulties do not arise in internal coordinates, at least for torsions, because the constraint normals are unit vectors regardless of the value of the constraint; thus 0° and 180° dihedral angle constraints can be imposed just as easily as any other value. 180° bond angles still cause difficulties, but near-linear arrangements of atoms require special treatment even in unconstrained optimizations. A typical solution involves replacing a near 180° bond angle by two special linear co-planar and perpendicular bends,[20] and modifying the torsions where necessary. A linear arrangement can be enforced by constraining the co-planar and perpendicular bends.

One other advantage over Cartesians is that in internals the constraint coordinate can be eliminated once the constraint is satisfied to the desired accuracy (the default tolerance is 10^{-6} in atomic units: bohr and radians). This is not possible in Cartesians due to the functional form of the constraint. In Cartesians, therefore, the Lagrange multiplier algorithm must be used throughout the entire optimization, whereas in delocalized internal coordinates it need only be used until all desired constraints are satisfied. As constraints become satisfied they can be simply eliminated from the optimization space and, once all constraint coordinates have been eliminated, standard algorithms can be used in the space of the remaining unconstrained coordinates. Unless the starting geometry is particularly poor, constraints are usually satisfied early in the optimization and at more or less the same time for multiple constraints. Therefore, Lagrange multipliers only need to be used in the first half-dozen or so cycles of a constrained optimization in internal coordinates.

9.1.7 GDIIS Algorithm

Direct inversion in the iterative subspace (DIIS) was originally developed by Pulay for accelerating SCF convergence.[81] Subsequently, Csaszar and Pulay used a similar scheme for geometry optimization, which they termed GDIIS.[27] The method is somewhat different from the usual quasi-Newton type approach and is included in OPTIMIZE as an alternative to the EF algorithm. Tests indicate that its performance is similar to EF, at least for small systems, however, there is rarely an advantage in using GDIIS in preference to EF.

In GDIIS geometries generated in previous optimization cycles, $\mathbf{x}_i$, are linearly combined to find the "best" geometry for the current cycle

$$\mathbf{x}_n = \sum_{i=1}^{m} c_i \mathbf{x}_i \tag{9.33}$$

where the problem is to find the best values for the coefficients c_i.

If we express each geometry by its deviation from the sought-after final geometry, $\mathbf{x}_f$, *i.e.*, $\mathbf{x}_f = \mathbf{x}_i + \mathbf{e}_i$, where $\mathbf{e}_i$ is an error vector, then it is obvious that if the conditions

$$\mathbf{r} = \sum c_i \mathbf{e}_i \tag{9.34}$$

and

$$\sum c_i = 1 \tag{9.35}$$

are satisfied, then the relation

$$\sum c_i \mathbf{x}_i = \mathbf{x}_f \tag{9.36}$$

also holds.

The true error vectors $\mathbf{e}_i$ are, of course, unknown. However, in the case of a nearly quadratic energy function they can be approximated by

$$\mathbf{e}_i = -\mathbf{H}^{-1}\mathbf{g}_i \tag{9.37}$$

where $\mathbf{g}_i$ is the gradient vector corresponding to the geometry $\mathbf{x}_i$ and $\mathbf{H}$ is an approximation to the Hessian matrix. Minimization of the norm of the residuum vector $\mathbf{r}$, Eq. (9.34), together with the constraint equation, Eq. (9.35), leads to a system of $m + l$ linear equations

$$\begin{pmatrix} B_{11} & \cdots & B_{1m} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ B_{m1} & \cdots & B_{mm} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \\ -\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{9.38}$$

where $B_{ij} = \langle \mathbf{e}_i | \mathbf{e}_j \rangle$ is the scalar product of the error vectors $\mathbf{e}_i$ and $\mathbf{e}_j$, and λ is a Lagrange multiplier.

The coefficients c_i determined from Eq. (9.38) are used to calculate an intermediate interpolated geometry

$$\mathbf{x}'_{m+1} = \sum c_i \mathbf{x}_i \tag{9.39}$$

and its corresponding interpolated gradient

$$\mathbf{g}'_{m+1} = \sum c_i \mathbf{g}_i \tag{9.40}$$

A new, independent geometry is generated from the interpolated geometry and gradient according to

$$\mathbf{x}_{m+1} = \mathbf{x}'_{m+1} - \mathbf{H}^{-1}\mathbf{g}'_{m+1} \; . \tag{9.41}$$

Note: Convergence is theoretically guaranteed regardless of the quality of the Hessian matrix, as long as it is positive definite. The original GDIIS algorithm used a static Hessian (*i.e.* the initial guess Hessian) which was often a simple identity matrix. Updating the Hessian at each cycle generally results in more rapid convergence, and this is the default in OPTIMIZE.

Other improvements to the original method include limiting the number of previous geometries used in Eq. (9.33) by neglecting earlier geometries, and eliminating any geometries more than a certain distance from the current geometry (default = 0.3 a.u.).

9.2 Geometry Optimization Job Controls

9.2.1 Job Control Overview

Obviously a level of theory, basis set, and starting molecular geometry must be specified to begin a geometry optimization or transition-structure search. These aspects are described elsewhere in this manual, and this section describes job-control variables specific to optimizations.

The job controls for geometry optimization have the general rem variable pattern, GEOM_OPT. These are the original rem variables used within the OPTIMIZE driver, Section 9.2.2. The job controls for LIBOPT3 are contained within *$geom_opt* section, which provide even greater job control to the users than the original rem variables, Section 9.2.5.

To allow our users with existing workflows to not experience any disruptions, the GEOM_OPT variables can be read within the LIBOPT3 driver if they have similar matching variables/controls. The decreasing order of precedence for setting of rem variables are as follows: 1) *$geom_opt* variables, 2) GEOM_OPT variables, 3) Default.

JOBTYPE

Specifies the calculation.

TYPE:

STRING

DEFAULT:

Default is single-point, which should be changed to one of the following options.

OPTIONS:

OPT	Equilibrium structure optimization.
TS	Transition structure optimization.
RPATH	Intrinsic reaction path following.

RECOMMENDATION:

Application-dependent.

GEOM_OPT_DRIVER

Controls the geometry optimization driver.

TYPE:

STRING

DEFAULT:

LIBOPT3

OPTIONS:

OPTIMIZE	Use the original optimization original driver.
LIBOPT3	Use the new driver.

RECOMMENDATION:

Note that the new driver is still under active development.

9.2.2 OPTIMIZE Job Control

GEOM_OPT_HESSIAN

Determines the initial Hessian status.

TYPE:

STRING/INTEGER

DEFAULT:

-1

OPTIONS:

-1	Approximate Hessian based on the force constant matrix (same as the "Model" option in LIBOPT3)
READ	Have exact or initial Hessian. Use as is if Cartesian, or transform if internals.

RECOMMENDATION:

An accurate initial Hessian will improve the performance of the optimizer, but it may be expensive to compute. Note that more options for the initial Hessian are available in LIBOPT3, controlled by INITIAL_HESSIAN under the *$geom_opt* section (see Sec. 9.2.5).

GEOM_OPT_COORDS

Controls the type of optimization coordinates.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

0	Optimize in Cartesian coordinates.
1	Generate and optimize in internal coordinates, if this fails abort.
-1	Generate and optimize in internal coordinates, if this fails at any stage of the optimization, switch to Cartesian and continue.
2	Optimize in Z-matrix coordinates, if this fails abort.
-2	Optimize in Z-matrix coordinates, if this fails during any stage of the optimization switch to Cartesians and continue.

RECOMMENDATION:

Use the default, as delocalized internals are more efficient. Note that optimization in Z-matrix coordinates requires that the input be specified in Z-matrix format.

GEOM_OPT_TOL_GRADIENT

Convergence on maximum gradient component.

TYPE:

INTEGER

DEFAULT:

300 $\equiv 300 \times 10^{-6}$ tolerance on maximum gradient component.

OPTIONS:

n Integer value (tolerance = $n \times 10^{-6}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_TOL_DISPLACEMENT

Convergence on maximum atomic displacement.

TYPE:

INTEGER

DEFAULT:

1200 $\equiv 1200 \times 10^{-6}$ tolerance on maximum atomic displacement.

OPTIONS:

n Integer value (tolerance = $n \times 10^{-6}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_TOL_ENERGY

Convergence on energy change of successive optimization cycles.

TYPE:

INTEGER

DEFAULT:

$100 \equiv 100 \times 10^{-8}$ tolerance on maximum (absolute) energy change.

OPTIONS:

n Integer value (tolerance = value $n \times 10^{-8}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_MAX_CYCLES

Maximum number of optimization cycles.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n User defined positive integer.

RECOMMENDATION:

The default should be sufficient for most cases. Increase if the initial guess geometry is poor, or for systems with shallow potential wells.

GEOM_OPT_PRINT

Controls the amount of OPTIMIZE print output.

TYPE:

INTEGER

DEFAULT:

3 Error messages, summary, warning, standard information and gradient print out.

OPTIONS:

0 Error messages only.

1 Level 0 plus summary and warning print out.

2 Level 1 plus standard information.

3 Level 2 plus gradient print out.

4 Level 3 plus Hessian print out.

5 Level 4 plus iterative print out.

6 Level 5 plus internal generation print out.

7 Debug print out.

RECOMMENDATION:

Use the default.

GEOM_OPT_SYMFLAG

Controls the use of symmetry in OPTIMIZE.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Make use of point group symmetry.

FALSE Do not make use of point group symmetry.

RECOMMENDATION:

Use the default.

GEOM_OPT_MODE

Determines Hessian mode followed during a transition state search.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Mode following off.

n Maximize along mode n.

RECOMMENDATION:

Use the default, for geometry optimizations.

GEOM_OPT_MAX_DIIS

Controls maximum size of subspace for GDIIS.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use GDIIS.

-1 Default size = min(NDEG, NATOMS, 4) NDEG = number of molecular degrees of freedom.

n Size specified by user.

RECOMMENDATION:

Use the default or do not set n too large.

GEOM_OPT_DMAX

Maximum allowed step size. Value supplied is multiplied by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

300 = 0.3

OPTIONS:

n User-defined cutoff.

RECOMMENDATION:

Use the default.

GEOM_OPT_UPDATE

Controls the Hessian update algorithm.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

-1 Use the default update algorithm.
0 Do not update the Hessian (not recommended).
1 Murtagh-Sargent update.
2 Powell update.
3 Powell/Murtagh-Sargent update (TS default).
4 BFGS update (OPT default).
5 BFGS with safeguards to ensure retention of positive definiteness (GDIIS default).

RECOMMENDATION:

Use the default.

GEOM_OPT_LINEAR_ANGLE

Threshold for near linear bond angles (degrees).

TYPE:

INTEGER

DEFAULT:

165 degrees.

OPTIONS:

n User-defined level.

RECOMMENDATION:

Use the default.

FDIFF_STEPSIZE

Displacement used for calculating derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 0.001 Å. For calculating second derivatives.

OPTIONS:

n Use a step size of $n \times 10^{-5}$.

RECOMMENDATION:

Use the default except in cases where the potential surface is very flat, in which case a larger value should be used. See FDIFF_STEPSIZE_QFF for third and fourth derivatives.

9.2.3 Hessian-Free Characterization of Stationary Points

Q-CHEM allows the user to characterize the stationary point found by a geometry optimization or transition state search without performing a full analytical Hessian calculation, which is sometimes unavailable or computationally unaffordable. This is achieved via a finite difference Davidson procedure developed by Sharada *et al.*[93] For a geometry optimization, it solves for the lowest eigenvalue of the Hessian (λ_1) and checks if $\lambda_1 > 0$ (a negative λ_1 indicates a

saddle point); for a TS search, it solves for the lowest two eigenvalues, and $\lambda_1 < 0$ and $\lambda_2 > 0$ indicate a transition state. The lowest eigenvectors of the updated P-RFO (approximate) Hessian at convergence are used as the initial guess for the Davidson solver.

The cost of this Hessian-free characterization method depends on the rate of convergence of the Davidson solver. For example, to characterize an energy minimum, it requires $2 \times N_{\mathrm{iter}}$ total energy + gradient calculations, where N_{iter} is the number of iterations that the Davidson algorithm needs to converge, and "2" is for forward and backward displacements on each iteration. According to Ref. 93, this method can be much more efficient than exact Hessian calculation for substantially large systems.

Note: At the moment, this method does not support QM/MM or systems with fixed atoms.

GEOM_OPT_CHARAC

Use the finite difference Davidson method to characterize the resulting energy minimum/transition state.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE do not characterize the resulting stationary point.

TRUE perform a characterization of the stationary point.

RECOMMENDATION:

Set it to TRUE when the character of a stationary point needs to be verified, especially for a transition structure.

GEOM_OPT_CHARAC_CONV

Overide the built-in convergence criterion for the Davidson solver.

TYPE:

INTEGER

DEFAULT:

0 (use the built-in default value 10^{-5})

OPTIONS:

n Set the convergence criterion to 10^{-n}.

RECOMMENDATION:

Use the default. If it fails to converge, consider loosening the criterion with caution.

9.2.4 OPTIMIZE Job Examples

Example 9.1 Optimization in Z-matrix coordinates. The input must be specified in Z-matrix format with coordinates specified. In the example below there are two coordinates representing the bond length and bond angle of a water molecule.

```
$molecule
   0 1
   O
   H 1 r
   H 1 r 2 ang

   r   0.95
   ang 104.5
$end

$rem
   JOBTYPE           OPT
   METHOD            HF
   BASIS             STO-3G
   GEOM_OPT_COORDS   2
$end
```

Example 9.2 Geometry optimization of a triflate anion that converges to an eclipsed conformation, which is a first order saddle point. This is verified via the finite difference Davidson method by setting GEOM_OPT_CHARAC to TRUE.

```
$molecule
   -1 1
   C  0.00000 -0.00078  0.98436
   F -1.09414 -0.63166  1.47859
   S  0.00000  0.00008 -0.94745
   O  1.25831 -0.72597 -1.28972
   O -1.25831 -0.72597 -1.28972
   O  0.00000  1.45286 -1.28958
   F  1.09414 -0.63166  1.47859
   F  0.00000  1.26313  1.47663
$end

$rem
   JOBTYPE                    opt
   METHOD                     BP86
   GEOM_OPT_DMAX              50
   BASIS                      6-311+G*
   SCF_CONVERGENCE            8
   THRESH                     14
   GEOM_OPT_TOL_DISPLACEMENT  10
   GEOM_OPT_TOL_ENERGY        10
   GEOM_OPT_TOL_GRADIENT      10
   GEOM_OPT_CHARAC            TRUE
   INTEGRAL_SYMMETRY          FALSE
   POINT_GROUP_SYMMETRY       FALSE
$end
```

Example 9.3 TS search for alanine dipeptide rearrangement reaction beginning with a guess structure converges correctly. The resulting TS structure is verified using the finite difference Davidson method.

```
$molecule
   0    1
   C          3.21659       -1.41022       -0.26053
   C          2.16708       -0.35258       -0.59607
   N          1.21359       -0.16703        0.41640
   C          0.11616        0.82394        0.50964
   C         -1.19613        0.03585        0.74226
   N         -2.18193       -0.02502       -0.18081
   C         -3.43891       -0.74663        0.01614
   O          2.19596        0.25708       -1.63440
   C          0.11486        1.96253       -0.53088
   O         -1.29658       -0.59392        1.85462
   H          3.25195       -2.14283       -1.08721
   H          3.06369       -1.95423        0.67666
   H          4.20892       -0.93714       -0.22851
   H          1.24786       -0.78278        1.21013
   H          0.25990        1.31404        1.47973
   H         -2.02230        0.38818       -1.10143
   H         -3.60706       -1.48647       -0.76756
   H         -4.29549       -0.06423        0.04327
   H         -3.36801       -1.25875        0.98106
   H         -0.68664        2.66864       -0.27269
   H          0.01029        1.65112       -1.56461
   H          1.06461        2.50818       -0.45885
$end

$rem
   JOBTYPE              freq
   EXCHANGE             B3LYP
   BASIS                6-31G
   SCF_MAX_CYCLES       250
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE              ts
   SCF_GUESS            read
   GEOM_OPT_DMAX        100
   GEOM_OPT_MAX_CYCLES  1500
   EXCHANGE             B3LYP
   BASIS                6-31G
   MAX_SCF_CYCLES       250
   GEOM_OPT_HESSIAN     read
   GEOM_OPT_CHARAC      true
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

9.2.5 LIBOPT3 Job Control

Within LIBOPT3 there is a separate *$geom_opt* section for geometry optimization job controls, which provide greater control then the GEOM_OPT variables. This new section contains all the following input variables for the LIBOPT3 driver.

ALGORITHM

Specifies which type of minimization algorithm to use.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

BFGS (OPT) / BOFILL (TS)

OPTIONS:

SD	Steepest Descent[70]
CG	Conjugate Gradient[70]
Newton	Exact Newton's Method[70]
BFGS	Broyden-Fletcher-Goldfarb-Shanno[18,37,41,90]
LBFGS	Limited-memory BFGS[70]
SR1	Symmetric-Rank One (Murtagh-Sargent)[69]
PSB	Powell symmetric Broyden[30]
BOFILL	Bofill combination of PSB and SR1[16]
FS	Farkas and Schlegel combination of SR1 and BFGS[35]

RECOMMENDATION:

Steepest descent and conjugate gradient methods are slow to converge in general but are useful when the near the minimum. Best to start with other algorithms and finalize with these two methods if a tighter converged minimum is needed. Newton's will be efficient but requires a Hessian evaluation at each step, so the cost of the Hessian calculation must be accounted for when using exact Newton's method. BFGS is a default algorithm for jobtype = OPT for its speed and efficiency for finding the minimum. L-BFGS is recommended when dealing with very large systems when memory is of concern. The default algorithm for jobtype = TS is Bofill.

COORDINATES

Specifies which type of coordinate system to use for optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Delocalized

OPTIONS:

Cartesian	Cartesian Coordiantes
Redundant	Redundant Internal Coordinates
Delocalized	Delocalized Natural Internal Coordinates

RECOMMENDATION:

Cartesian can be more stable than internal coordinates but are generally slower than internal coordinates. If there are problems with internal coordinate optimization restart with Cartesian coordinates at the last known internal coordinate geometry can be controlled with OPTIMIZATION_RESTART.

MAXITER

Maximum number of geometry optimization cycles.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

Integer Any positive integer

RECOMMENDATION:

None

INITIAL_HESSIAN

Specifies the type of initial Hessian to use.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Model (OPT) / EXACT (TS)

OPTIONS:

Identity Identity Matrix

Simple Simple Approximate Guess Hessian

Model Model Approximate Guess Hessian

Exact Analytical Hessian

Read Read-in Hessian

RECOMMENDATION:

The initial guess Hessian for Cartesian coordinates is a unit matrix where for internal coordinates it is an approximate Hessian based on the internal coordinates.[11] The model approximate Hessian is based on a force constant matrix.[88,108] The exact Hessian is always a quality initial Hessian but could be costly but can be calculated at the start of the optimization or read in from scratch. The default for OPT is the model approximate guess Hessian but for TS the Exact analytical Hessian will be used.

CONVERGENCE_CHECK

Specifies the type of convergence check during geometry optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

DEFAULT

OPTIONS:

Default Check max absolute gradient component and (maximum absolute displacement or change in energy)

Energy Change in energy

Gradient Check norm of gradient

RECOMMENDATION:

None.

GRADIENT_CONVERGENCE

The value of maximum absolute gradient or norm of gradient for convergence check.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

3e-4

OPTIONS:

Float Any positive float

RECOMMENDATION:

This variable is used as the comparison value for the gradient checked with CONVERGENCE_CHECK.

DISPLACEMENT_CONVERGENCE

The value of maximum absolute displacement for convergence check.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

1.2e-3

OPTIONS:

Float Any positive float

RECOMMENDATION:

None.

ENERGY_CONVERGENCE

The value of maximum absolute energy difference for convergence check.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

1e-6

OPTIONS:

Float Any positive float

RECOMMENDATION:

None.

STEP_LIMITER

Specifies the type of limiter to use for adjustment of the step during geometry optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

RMS (OPT) / NORM (TS)

OPTIONS:

RMS Root mean square of the step

NORM Norm of the step

RECOMMENDATION:

Adjust the type of condition used for MAX_DISPLACEMENT.

MAX_DISPLACEMENT

The value of maximum for the STEP_LIMITER of the Eigenvector following algorithm step.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

3e-1

OPTIONS:

Float Any positive float

RECOMMENDATION:

If this value is too large there may be trouble with the optimization and a small value could require additional optimization cycles.

RECOMPUTE_HESSIAN

Recompute the exact Hessian during optimization algorithms

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

False

OPTIONS:

None Do not recompute of Hessian

Recompute Compute Hessian during optimization

RECOMMENDATION:

Recompute the exact Hessian during optimization during BFGS, SR1, PSB, BOFILL, and FS Quasi-Newton Hessian update algorithms.

RECOMPUTE_HESSIAN_CYCLES

The number of cycles before recomputing the Hessian during optimization algorithms.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

Integer Any positive integer

RECOMMENDATION:

None.

STEP_SEARCH_ALGORITHM

Specifies the type of algorithm for geometry step generation.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

EFA

OPTIONS:

EFA Eigenvector Following Algorithm

LS Simple Line Search

RECOMMENDATION:

For Quasi-Newton methods the default step generation is Eigenvector following algorithm, but line search can be used if desired.

EIGENVECTOR_ALGORITHM

Specifies the type of Eigenvector algorithm.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

RFO

OPTIONS:

RFO Rational Function Optimization

PRFO Partitioned Rational Function Optimization

RECOMMENDATION:

This allows the switching of the Eigenvector following algorithm for state specific searches with PRFO.

PRINT_LEVEL

Specifies the printing verbosity of the optimizer.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 General Print

10 Verbose

RECOMMENDATION:

None.

MAX_LBGFGS_HISTORY

Specifies the number of cycles to retain for L-BFGS history.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

Integer Any positive integer

RECOMMENDATION:

It is recommended to keep this number small.

LS_PARAM

Specifies the type of line search algorithm to use.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Quasi-Newton

OPTIONS:

Quasi-Newton Quasi-Newton

Strict Strict line search parameters

Very_Strict Very strict line search parameters

RECOMMENDATION:

None

LS_MAXITER

Specifies the number of maximum iterations to perform during line search step calculation.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

Integer Any positive integer

RECOMMENDATION:

None

LS_PRINT

Specifies the verbosity of printing for the line search algorithm.

INPUT SECTION: *$geom_opt*

TYPE:

String

DEFAULT:

Minimal

OPTIONS:

Minimal Minimal printing

Verbose Verbose printing

RECOMMENDATION:

None

HESSIAN_VERIFY

Specifies the type of verification with the Hessian after geometry optimization

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

RESULT

OPTIONS:

No_Verification Do no verification of optimization

Result Verify with final Hessian obtained during optimization

Without Verify without a Hessian (Only convergence criteria)

Recomputed Verify with recomputed exact Hessian

RECOMMENDATION:

None.

OPTIMIZATION_RESTART

Specifies if optimization should restart in Cartesian coordinates after back-transformation failure with internal coordinate optimizations.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

TRUE

OPTIONS:

True Restart with Cartesian Coordinates

False Do not restart

RECOMMENDATION:

Restart a failed back-transformation internal coordinate optimization job in Cartesian coordinates. This will use the current retained updated Internal coordinate Hessian transform it to Cartesian coordinates and continue the optimization from the last known position.

FINAL_ZMAT_PRINT

Controls if a *Z*-matrix is printed at the end of the job.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

TRUE

OPTIONS:

True Construct *Z*-matrix and print

False Do calculate or print *Z*-matrix

RECOMMENDATION:

After optimization the final structure can be used to compute and print the final *Z*-matrix. This can be turned off if molecule contains many atoms and Z-Matrix is not needed.

FINAL_VIBRATIONAL_ANALYSIS

Run Vibrational Analysis after geometry optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

False

OPTIONS:

True Perform Vibrational Analysis

False Do not compute vibrational analysis

RECOMMENDATION:

Vibrational analysis can be performed only if the final Hessian for verification was recomputed, HESSIAN_VERIFY

PRINT_TOPOLOGY

Print the topology for optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

FALSE

OPTIONS:

True	Print the topology
False	Do not print the topology
Terminate	Print the topology and terminate job

RECOMMENDATION:

Print the topology used in optimization. In addition the topology can be printed then the job terminated.

USER_TOPOLOGY

Specifies if a user provided topology is to be read.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Generated

OPTIONS:

Generated	Generate the topology
Read	Read a user provided topology.

RECOMMENDATION:

A user can provide a topology for a given molecule in the *$geom_opt_topology* section in the input.

A user defined topology can be read in using the USER_TOPOLOGY keyword and providing the topology in *$geom_opt_topology* section. The available topology definitions that can be used are:

1 (Bond) A-B

2 (Angle) A-C-B (Apex is B)

4 (Torsion) Torsion A-B-C-D

5 (Co-Linear Angle) use with 6 D A-B-C (ABC is Linear)

6 (Co-Linear Angle) use with 5 D A-B-C (ABC is Linear)

Note: The first line is the total number of coordinates to read. Then the following lines are the coordinate definition to be used, following the definition above: coordinate type, atom number 1, ...

```
$geom_opt_topology
Total Number of Coordinates
1   atomA  atomB !Bonds
...
2   atomA  atomC  atomB !Angles
...
4   atomA  atomB  atomC atomD !Torsions
...
5   atomD  atomA  atomB atomC !Co-linear Torsion
...
6   atomD  atomA  atomB atomC !Co-linear Torsion
$end
```

9.2.6 LIBOPT3 Job Examples

Example 9.4 As outlined, the rate of convergence of the iterative optimization process is dependent on a number of factors, one of which is the use of an initial analytic Hessian. This is easily achieved by instructing Q-CHEM to calculate an analytic Hessian and proceed then to determine the required critical point. This job runs within LIBOPT3 driver which retains the reading of analytical Hessian feature.

```
$molecule
   0  1
   O
   H  1  oh
   H  1  oh 2 hoh

   oh  = 1.1
   hoh = 104
$end

$rem
   JOBTYPE    freq   Calculate an analytic Hessian
   METHOD     hf
   BASIS      6-31g(d)
$end

$comment
Now proceed with the optimization making sure to read in the analytic
Hessian (use other available information too).
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE            opt
   METHOD             hf
   BASIS              6-31g(d)
   SCF_GUESS          read
   GEOM_OPT_HESSIAN   read   Have the initial Hessian
$end

@@@

$comment
LIBOPT3 driver does not require the need for two jobs, as the exact analytic
Hessian can be computed from the optimization job.  LIBOPT3 still retains the
reading of Hessian for users who wish to split the jobs.
$end

$molecule
   0  1
   O
   H  1  oh
   H  1  oh 2 hoh

   oh  = 1.1
   hoh = 104
$end

$rem
   JOBTYPE    opt
   METHOD     hf
   BASIS      6-31g(d)
$end

$geom_opt
initial_hessian exact !Start with analytic Hessian
```

9.3 Improved Algorithms for Transition-Structure Optimization

9.3.1 Introduction

Transition-structure searches tend to be more difficult (meaning, more likely to be unsuccessful) as compared to minimum-energy (equilibrium) geometry optimizations. Odds of success can be enhanced via an initial guess structure that is determined in an automated way, rather than simply "guessed" by the user. Several such automated algorithms are available in Q-CHEM, and are described in this section.

9.3.2 Freezing String Method

Perhaps the most significant difficulty in locating transition states is to obtain a good initial guess of the geometry to feed into a surface-walking algorithm. This difficulty becomes especially relevant for large systems, for which the dimensionality of the search space is large. Interpolation algorithms are promising for locating good guesses of the minimum-energy pathway connecting reactant and product states as well as approximate saddle-point geometries. For example, the *nudged elastic band method*[45,68] and the *string method*[34] start from a certain initial reaction pathway connecting the reactant and the product state, and then optimize in discretized path space towards the minimum-energy pathway. The highest-energy point on the approximate minimum-energy pathway becomes a good initial guess for the saddle-point configuration that can subsequently be used with any local surface-walking algorithm.

Inevitably, the performance of any interpolation method heavily relies on the choice of the initial reaction pathway, and a poorly-chosen initial pathway can cause slow convergence, or possibly convergence to an incorrect pathway. The growing-string method[76] and freezing-string method[14,92] offer solutions to this problem, in which two string fragments (one representing the reactant state and the other representing the product state) are "grown" (*i.e.*, increasingly-finely defined) until the two fragments join. The freezing-string method offers a choice between Cartesian interpolation and linear synchronous transit (LST) interpolation. It also allows the user to choose between conjugate gradient and quasi-Newton optimization techniques.

Freezing-string calculations are requested by setting JOBTYPE = FSM in the *$rem* section. Additional job-control keywords are described below, along with examples. Consult Refs. 14 and 92 for a guide to a typical use of this method.

FSM_NNODE

Specifies the number of nodes along the string

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

N number of nodes in FSM calculation

RECOMMENDATION:

$N = 15$. Use 10 to 20 nodes for a typical calculation. Reaction paths that connect multiple elementary steps should be separated into individual elementary steps, and one FSM job run for each pair of intermediates. Use a higher number when the FSM is followed by an approximate-Hessian based transition state search (Section 9.3.3).

FSM_NGRAD

Specifies the number of perpendicular gradient steps used to optimize each node

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

N Number of perpendicular gradients per node

RECOMMENDATION:

Anything between 2 and 6 should work, where increasing the number is only needed for difficult reaction paths.

FSM_MODE

Specifies the method of interpolation

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Cartesian

2 LST

RECOMMENDATION:

In most cases, LST is superior to Cartesian interpolation.

FSM_OPT_MODE

Specifies the method of optimization

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Conjugate gradients

2 Quasi-Newton method with BFGS Hessian update

RECOMMENDATION:

The quasi-Newton method is more efficient when the number of nodes is high.

An example input appears below. Note that the *$molecule* section includes geometries for two optimized intermediates, separated by `****`. The order of the atoms is important, as Q-CHEM assumes that the nth atom in the reactant moves toward the nth atom in the product. The FSM string is printed out in the file `stringfile.txt`, which contains Cartesian coordinates of the structures that connect reactant to product. Each node along the path is labeled in this file, and its energy is provided. The geometries and energies are also printed at the end of the Q-CHEM output file, where they are labeled:

```
---------------------------------------
   STRING
---------------------------------------
```

Finally, if MOLDEN_FORMAT is set to TRUE, then geometries along the string are printed in a MOLDEN-readable format at the end of the Q-CHEM output file. The highest-energy node can be taken from this file and used to run a

transition structure search as described in Section 9.1.1. If the string returns a pathway that is unreasonable, check whether the atoms in the two input geometries are in the correct order.

Example 9.9 Example of the freezing-string method.

```
$molecule
   0  1
   Si    1.028032  -0.131573  -0.779689
   H     0.923921  -1.301934   0.201724
   H     1.294874   0.900609   0.318888
   H    -1.713989   0.300876  -0.226231
   H    -1.532839   0.232021   0.485307
****
   Si    0.000228  -0.000484  -0.000023
   H     0.644754  -1.336958  -0.064865
   H     1.047648   1.052717   0.062991
   H    -0.837028   0.205648  -1.211126
   H    -0.855603   0.079077   1.213023
$end

$rem
   JOBTYPE         fsm
   FSM_NGRAD       3
   FSM_NNODE       12
   FSM_MODE        2
   FSM_OPT_MODE    2
   METHOD          b3lyp
   BASIS           6-31G
$end
```

9.3.3 Hessian-Free Transition-State Search

Once a guess structure to the transition state is obtained, standard eigenvector-following methods such as Baker's partitioned rational-function optimization (P-RFO) algorithm[5] can be employed to refine the guess to the exact transition state. The reliability of P-RFO depends on the quality of the Hessian input, which enables the method to distinguish between the reaction coordinate (characterized by a negative eigenvalue) and the remaining degrees of freedom. In routine calculations therefore, an exact Hessian is determined via frequency calculation prior to the P-RFO search. Since the cost of evaluating an exact Hessian typically scales one power of system size higher than the energy or the gradient, this step becomes impractical for systems containing large number of atoms.

The exact Hessian calculation can be avoided by constructing an approximate Hessian based on the output of FSM.[93] The tangent direction at the transition state guess on the FSM string is a good approximation to the Hessian eigenvector corresponding to the reaction coordinate. The tangent is therefore used to calculate the correct eigenvalue and corresponding eigenvector by variationally minimizing the Rayleigh-Ritz ratio.[58] The reaction coordinate information is then incorporated into a guess matrix which, in turn, is obtained by transforming a diagonal matrix in delocalized internal coordinates[10,39] to Cartesian coordinates. The resulting approximate Hessian, by design, has a single negative eigenvalue corresponding to the reaction coordinate. This matrix is then used in place of the exact Hessian as input to

the P-RFO method.

Example 9.10 An example one-shot, Hessian-free approach that combines the FSM and P-RFO methods in order to determine the exact transition state from reactant and product structures.

```
$molecule
   0  1
   Si    1.028032  -0.131573  -0.779689
   H     0.923921  -1.301934   0.201724
   H     1.294874   0.900609   0.318888
   H    -1.713989   0.300876  -0.226231
   H    -1.532839   0.232021   0.485307
****
   Si    0.000228  -0.000484  -0.000023
   H     0.644754  -1.336958  -0.064865
   H     1.047648   1.052717   0.062991
   H    -0.837028   0.205648  -1.211126
   H    -0.855603   0.079077   1.213023
$end

$rem
   JOBTYPE               fsm
   METHOD                b3lyp
   BASIS                 6-31g
   FSM_NGRAD             3
   FSM_NNODE             18
   FSM_MODE              2
   FSM_OPT_MODE          2
   INTEGRAL_SYMMETRY     false
   POINT_GROUP_SYMMETRY  false
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE               ts
   METHOD                b3lyp
   BASIS                 6-31g
   SCF_GUESS             read
   GEOM_OPT_HESSIAN      read
   MAX_SCF_CYCLES        250
   GEOM_OPT_DMAX         50
   GEOM_OPT_MAX_CYCLES   100
   INTEGRAL_SYMMETRY     false
   POINT_GROUP_SYMMETRY  false
$end
```

9.4 Constrained Optimization

9.4.1 Introduction

Constrained optimization refers to the optimization of molecular structures (transition state or minimum-energy) in which certain parameters such as bond lengths, bond angles or dihedral angles are fixed. Q-CHEM can handle con-

straints directly in delocalized internal coordinates using the method of Lagrange multipliers (see Section 9.1.5). Features of constrained optimization in Q-CHEM are:

- Starting geometries need not satisfy the requested constraints.
- Constrained optimization is performed in delocalized internal coordinates, which is typically the most efficient coordinate system for optimization of large molecules.
- Q-CHEM's free-format *$opt* section allows the user to apply constraints with ease.

Constraints are imposed via the *$opt* input section, whose format is shown below, and the various parts of this input section are described below.

Note: As with the rest of the Q-CHEM input file, the *$opt* section is case-insensitive, but there should be no blank space at the beginning of a line.

```
$opt
CONSTRAINT
stre  atom1  atom2  value
...
bend  atom1  atom2  atom3  value
...
outp  atom1  atom2  atom3  atom4  value
...
tors  atom1  atom2  atom3  atom4  value
...
linc  atom1  atom2  atom3  atom4  value
...
linp  atom1  atom2  atom3  atom4  value
...
ENDCONSTRAINT

FIXED
atom   coordinate_reference
...
ENDFIXED

DUMMY
idum   type   list_length   defining_list
...
ENDDUMMY

CONNECT
atom   list_length   list
...
ENDCONNECT
$end
```

9.4.2 Geometry Optimization with General Constraints

CONSTRAINT and ENDCONSTRAINT define the beginning and end, respectively, of the constraint section of *$opt* within which users may specify up to six different types of constraints:

interatomic distances

Values in Ångstroms; *value* > 0:

```
stre   atom1   atom2   value
```

angles

Values in degrees, $0 \leq value \leq 180$; *atom2* is the middle atom of the bend:

```
bend   atom1   atom2   atom3   value
```

out-of-plane-bends

Values in degrees, $-180 \leq value \leq 180$ *atom2*; angle between *atom4* and the *atom1–atom2–atom3* plane:

```
outp   atom1   atom2   atom3   atom4   value
```

dihedral angles

Values in degrees, $-180 \leq value \leq 180$; angle the plane *atom1–atom2–atom3* makes with the plane *atom2–atom3–atom4*:

```
tors   atom1   atom2   atom3   atom4   value
```

coplanar bends

Values in degrees, $-180 \leq value \leq 180$; bending of *atom1–atom2–atom3* in the plane *atom2–atom3–atom4*:

```
linc   atom1   atom2   atom3   atom4   value
```

perpendicular bends

Values in degrees, $-180 \leq value \leq 180$; bending of *atom1–atom2–atom3* perpendicular to the plane *atom2–atom3–atom4*:

```
linp   atom1   atom2   atom3   atom4   value
```

Example 9.11 Methanol geometry optimization with constraints in bond length and bond angles.

```
$molecule
   0  1
   C    0.14192    0.33268    0.00000
   O    0.14192   -1.08832    0.00000
   H    1.18699    0.65619    0.00000
   H   -0.34843    0.74268    0.88786
   H   -0.34843    0.74268   -0.88786
   H   -0.77395   -1.38590    0.00000
$end

$rem
   JOBTYPE           opt
   METHOD            hf
   BASIS             3-21g
   GEOM_OPT_PRINT    6
$end

$opt
   CONSTRAINT
   stre  1  6  1.8
   bend  2  1  4  110.0
   bend  2  1  5  110.0
   ENDCONSTRAINT
$end
```

9.4.3 Frozen Atoms

Absolute atom positions can be frozen with the *FIXED* section. The section starts with the *FIXED* keyword as the first line and ends with the *ENDFIXED* keyword on the last. The format to fix a coordinate or coordinates of an atom is:

```
atom   coordinate_reference
```

coordinate_reference can be any combination of up to three characters X, Y and Z to specify the coordinate(s) to be fixed: X, Y, Z, *XY*, *XZ*, *YZ*, *XYZ*. The fixing characters must be next to each other. *e.g.*,

```
FIXED
2 XY
ENDFIXED
```

means the x-coordinate and y-coordinate of atom 2 are fixed, whereas

```
FIXED
2 X Y
ENDFIXED
```

will yield erroneous results.

Note: When the *FIXED* section is specified within *$opt*, the optimization will proceed in Cartesian coordinates.

Note that *frequency* calculations for a system with frozen atom constraints will generally produce imaginary frequencies. These arise because the Hessian computed for the frequency calculation has different physics compared to the gradient computed during the fixed-atom optimization. One possible workaround is simply to zero out the elements of the Hessian associated with the frozen atoms. This removes the contribution of the frozen atoms to the normal modes and reduces the number of frequencies computed. For a system with N atoms, where n atoms are frozen, there will be $3(N-n)-6$ normal modes, or $3(N-n)-5$ if the system is linear. This "zeroing out" approach can be requested with a pair of *$rem* variables, FRZN_OPT and FRZ_ATOMS, as described below. A more elegant approach is to replace the strictly fixed atomic positions with soft harmonic confining potentials, which achieves effectively the same result but allows for relaxation (*i.e.*, optimization) of all atomic coordinates, so that a proper vibrational frequency calculation can be performed. The use of harmonic confining potentials as a replacement for fixed atom constraints is described in Section 9.4.7.

FRZN_OPT

Controls whether the job uses zeroed Hessian technique in the frequency calculations

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use the zeroed out Hessian

True Use the zeroed out Hessian

RECOMMENDATION:

False

FRZ_ATOMS

Controls the number of frozen atoms

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

Note: The above two *$rem* variables are only relevant to frequency calculations; they are not required in geometry optimization with fixed atoms.

Example 9.12 Optimization of the water dimer structure with the structure of the first water frozen

```
$molecule
 0 1
 O   -1.551007  -0.114520   0.000000
 H   -1.934259   0.762503   0.000000
 H   -0.599677   0.040712   0.000000
 O    1.350625   0.111469   0.000000
 H    1.680398  -0.373741  -0.758561
 H    1.680398  -0.373741   0.758561
$end

$rem
 JOBTYPE          OPT
 METHOD           B3LYP
 BASIS            6-31G(D)
 SCF_CONVERGENCE  8
$end

$opt
 FIXED
 1:3  XYZ
 ENDFIXED
$end
```

Example 9.13 Frequency calculations using zeroed out Hessian technique. Note the frozen atom indices are specified in the *$frozen_opt* section.

```
$molecule
   0 1
   C        -0.0014570824    1.4001468208    0.0000000000
   H        -0.0030029737    2.4867609686    0.0000000000
   C        -1.2093554750    0.6986986619    0.0000000000
   H        -2.1525287416    1.2392252926    0.0000000000
   C        -1.2094237515   -0.6985901650    0.0000000000
   H        -2.1508487680   -1.2427180000    0.0000000000
   C        -0.0003285453   -1.3965954363    0.0000000000
   H        -0.0006460438   -2.4839940415    0.0000000000
   C         1.2098250013   -0.6978846084    0.0000000000
   H         2.1493830451   -1.2446769146    0.0000000000
   C         1.2133329176    0.6987868208    0.0000000000
   H         2.1550504173    1.2408406009    0.0000000000
$end

$rem
   JOBTYPE      freq
   METHOD       b3lyp
   BASIS        6-31G*
   FRZN_OPT     1  ! Turns on the freq calculation with frozen Hessians
   FRZ_ATOMS    2  ! No. of frozen atoms
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

$frozen_opt
1 11
$end
```

9.4.4 Dummy Atoms

DUMMY defines the beginning of the dummy atom section and ENDDUMMY its conclusion. Dummy atoms are used to help define constraints during constrained optimizations in Cartesian coordinates. They cannot be used with delocalized internals.

All dummy atoms are defined with reference to a list of real atoms, that is, dummy atom coordinates are generated from the coordinates of the real atoms from the dummy atoms defining list (see below). There are three types of dummy atom:

1. Positioned at the arithmetic mean of up to seven real atoms in the defining list.
2. Positioned a unit distance along the normal to a plane defined by three atoms, centered on the middle atom of the three.
3. Positioned a unit distance along the bisector of a given angle.

The format for declaring dummy atoms is:

```
DUMMY
idum    type    list_length    defining_list
ENDDUMMY
```

idum	Center number of defining atom (must be one greater than the total number of real atoms for the first dummy atom, two greater for second *etc.*).
type	Type of dummy atom (either 1, 2 or 3; see above).
list_length	Number of atoms in the defining list.
defining_list	List of up to seven atoms defining the position of the dummy atom.

Once defined, dummy atoms can be used to define standard internal (distance, angle) constraints as per the constraints section, above.

Note: The use of dummy atoms of type 1 has never progressed beyond the experimental stage.

9.4.5 Dummy Atom Placement in Dihedral Constraints

Bond and dihedral angles cannot be constrained in Cartesian optimizations to exactly 0° or $\pm 180^\circ$. This is because the corresponding constraint normals are zero vectors. Also, dihedral constraints near these two limiting values (within, say 20°) tend to oscillate and are difficult to converge.

These difficulties can be overcome by defining dummy atoms and redefining the constraints with respect to the dummy atoms. For example, a dihedral constraint of 180° can be redefined to two constraints of 90° with respect to a suitably positioned dummy atom. The same thing can be done with a 180° bond angle (long a familiar use in *Z*-matrix construction).

Typical usage is as shown in Table 9.3. Note that the order of atoms is important to obtain the correct signature on the dihedral angles. For a 0° dihedral constraint, atoms J and K should be switched in the definition of the second torsion constraint in Cartesian coordinates.

Note: In almost all cases the above discussion is somewhat academic, as internal coordinates are now best imposed using delocalized internal coordinates and there is no restriction on the constraint values.

Internal Coordinates	Cartesian Coordinates
$opt	$opt
CONSTRAINT	DUMMY
tors I J K L 180.0	M 2 I J K
ENDCONSTRAINT	ENDDUMMY
$end	CONSTRAINT
	tors I J K M 90
	tors M J K L 90
	ENDCONSTRAINT
	$end

Table 9.3: Comparison of dihedral angle constraint method for adopted coordinates.

9.4.6 Additional Atom Connectivity

Normally delocalized internal coordinates are generated automatically from the input Cartesian coordinates. This is accomplished by first determining the atomic connectivity list (*i.e.*, which atoms are formally bonded) and then constructing a set of individual primitive internal coordinates comprising all bond stretches, all planar bends and all proper torsions that can be generated based on the atomic connectivity. The delocalized internal are in turn constructed from this set of primitives.

The atomic connectivity depends simply on distance and there are default bond lengths between all pairs of atoms in the code. In order for delocalized internals to be generated successfully, all atoms in the molecule must be formally bonded so as to form a closed system. In molecular complexes with long, weak bonds or in certain transition states where parts of the molecule are rearranging or dissociating, distances between atoms may be too great for the atoms to be regarded as formally bonded, and the standard atomic connectivity will separate the system into two or more distinct parts. In this event, the generation of delocalized internal coordinates will fail. Additional atomic connectivity can be included for the system to overcome this difficulty.

CONNECT defines the beginning of the additional connectivity section and ENDCONNECT the end. The format of the CONNECT section is:

```
CONNECT
atom   list_length   list
ENDCONNECT
```

atom	Atom for which additional connectivity is being defined.
list_length	Number of atoms in the list of bonded atoms.
list	List of up to 8 atoms considered as being bonded to the given atom.

9.4.7 Atomic Confining Potentials as Alternatives to Constrained Optimization

In principle, the same effect of constrained optimization using fixed atoms can be achieved instead using soft harmonic confining potentials of the form

$$V_{\text{conf}}(\mathbf{r}_1, \mathbf{r}_2, \ldots) = \frac{1}{2} \sum_i k \|\mathbf{r}_i - \mathbf{r}_i^0\|^2 \, . \tag{9.42}$$

This represents an external potential that confines the ith atom (having coordinates $\mathbf{r}_i$) around the position $\mathbf{r}_i^0$. In applications to cluster models of enzymes (as a low-cost alternative to QM/MM simulations), it is necessary to lock

certain atoms at their crystallographic positions in order to relax the geometry (in the gas phase or in continuum solvent) without collapsing the active-site model.[17,29,94]

Use of a confining potential allows this optimization to proceed in an *unconstrained* manner, using delocalized internal coordinates (rather than Cartesian coordinates) for efficiency, yet achieves the same effect as the traditional fixed-atom approach that is widely used in cluster models of enzymatic reactions.[94] Moreover, the use of harmonic confining potentials does not result in imaginary frequencies that can plague fixed-atom optimizations, making it straightforward to compute zero-point vibrational corrections.[17,29]

Harmonic confining potentials are activated by setting the *$rem* variable HARM_OPT to true, listing the indices of the confined atoms in the *$harmonic_opt* section and their corresponding equilibrium positions ($\mathbf{r}_i^0$) in the *$coords* section.

HARM_OPT

Controls whether the job uses confining potentials

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use the potential

True Use the potential

RECOMMENDATION:

False

HOATOMS

Controls the number of confined atom

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

HARM_FORCE

Sets the force constant for harmonic confiner, in units of N/m.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

Example 9.14 Optimization using soft harmonic confining potentials

```
$molecule
   0 1
   C          2.2847229688   -0.3069830925   -0.2968221397
   C          0.9156471557    0.1503924513    0.1693932675
   N         -0.0576877706   -0.7876400788    0.0645249649
   H          2.9837678662    0.5043669375   -0.1693203557
   H          2.2497378474   -0.5929607607   -1.3422589452
   H          2.6126794028   -1.1626691284    0.2825927880
   O          0.6966207559    1.2669942030    0.6077661092
   C         -1.4350712383   -0.4874947903    0.4670886412
   H          0.1463602169   -1.6783001309   -0.3307859180
   C         -2.1768099264    0.3412632672   -0.5936684676
   H         -1.3995705380    0.0636682083    1.3955334557
   H         -1.9421824240   -1.4270154508    0.6422037013
   H         -1.6624625664    1.2829541077   -0.7297597438
   H         -3.1943263155    0.5415731987   -0.2762358302
   H         -2.2051967614   -0.1880845317   -1.5391034623
$end

$rem
   JOBTYPE       OPT
   METHOD        HF
   BASIS         3-21G
   NO_REORIENT   true
   HARM_OPT      1    ! Turn on harmonic confining potential
   HOATOMS       2    ! No. of confined atoms
   HARM_FORCE    450  ! Force constant of the potential
   POINT_GROUP_SYMMETRY False
$end

$harmonic_opt
   1  10 ! indices of the confined atoms
$end

$coords !coordinates of confined atoms
   C1     2.2847229688   -0.3069830925   -0.2968221397
   C10   -2.1768099264    0.3412632672   -0.5936684676
$end
```

9.5 Application of Pressure and Forces

9.5.1 Application of External Forces

In 2009, three methods for optimizing the geometry of a molecule under a constant external force were introduced, which were called *Force-Modified Potential Energy Surface* (FMPES),[73] *External Force is Explicitly Included* (EFEI),[86] and *Enforced Geometry Optimization* (EGO).[109] These methods are closely related, and the interested reader is referred to Ref. 98 for a detailed discussion of the similarities and differences between them. For simplicity, we will stick to the term EFEI here. An EFEI calculation is a geometry optimization in which a constant that is equal to the external force is added to the nuclear gradient of two atoms specified by the user. The external force is applied along the vector connecting the two atoms, thus driving them apart. The geometry optimization converges when the restoring force of the molecule is equal to the external force. The EFEI method can also be used in AIMD simulations (Section 9.9), in which case the force is added in every time step.

Q-CHEM 5.4 is the first version that uses a new syntax for specifying EFEI calculations, which requires DISTORT = TRUE in the *$rem* section (see Section 9.5):

```
$distort
   model efei
   force [atom1 atom2 force1]
   force [atom3 atom4 force2]
   ...
$end
```

Here, *atom1* and *atom2* are the indices of the atoms to which a force is applied. *force1* is the sum of the force values that acts on *atom1* and *atom2* in nanoNewtons (nN). If this value is positive, a mechanical force of magnitude *force1*/2 acts on each of these atoms, thus driving them apart. If it is negative, an attractive force acts between the atoms. Optionally, additional pairs of atoms that are subject to a force can be specified by adding lines in the *$distort* section.

9.5.2 Application of Pressure

Q-CHEM features a number of methods to apply pressure to a chemical system during a geometry optimization or an AIMD simulation.[97] The following methods are implemented:

Hydrostatic Compression Force Field (HCFF)[99]	Section 9.5.2.1
eXtended Hydrostatic Compression Force Field (X-HCFF)[96]	Section 9.5.2.2
Gaussians On Surface Tesserae Simulate HYdrostatic Pressure (GOSTSHYP)[87]	Section 9.5.2.4

To invoke one of these methods, set DISTORT = TRUE in the *$rem* section. By setting the *$rem* variable scf_final_print = 1, the energy contribution due to pressure is printed in the output.

DISTORT

Specifies whether to apply pressure or external force to a chemical system

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use pressure or force

True Use pressure or force

RECOMMENDATION:

Set to true to apply pressure or force.

The parameters of the jobs are set *via* options specified in the *$distort* input section. The format of the *$distort* section is analogous to the *$rem* section:

```
$distort
  <Keyword>  <parameter/option>
$end
```

Note: The following job control variables belong *only* in the *$distort* section. Do not place them in the *$rem* section.

Model

Specifies which model is used to distort the molecule.

INPUT SECTION: *$distort*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

HCFF	Hydrostatic Compression Force Field[99]
XHCFF	eXtended Hydrostatic Compression Force Field[96]
PV	Add PV term by computing vDW volume[111]
GOSTSHYP	Gaussians On Surface Tesserae Simulate HYdrostatic Pressure[87]
EFEI	External Force is Explicitly Included (Section 9.5.1)

RECOMMENDATION:

Please refer to the following subsections for recommendations on which model to use.

Pressure

Specifies the pressure (in MPa) used to compress the molecule.

INPUT SECTION: *$distort*

TYPE:

DOUBLE

DEFAULT:

None

OPTIONS:

User defined

RECOMMENDATION:

None

NPoints_Heavy

Specifies the number of tessellation points per non-hydrogen atom.

INPUT SECTION: *$distort*

TYPE:

INTEGER

DEFAULT:

110

OPTIONS:

User defined

RECOMMENDATION:

Use the default.

NPoints_Hydrogen

Specifies the number of tessellation points per hydrogen atom.

INPUT SECTION: *$distort*

TYPE:

INTEGER

DEFAULT:

110

OPTIONS:

User defined

RECOMMENDATION:

Use the default.

Scaling

Specifies the scaling factor of the atomic van der Waals radii used in the tessellation of the molecular surface, which is used in the pressure models.

INPUT SECTION: *$distort*

TYPE:

DOUBLE

DEFAULT:

1.2

OPTIONS:

User defined

RECOMMENDATION:

Increase when modeling a chemical complex to make sure that the complex is placed inside a single cavity.[87] A value of 1.0 was suggested to be used in the X-HCFF model.[96]

Screener

Enables/disables Integral screening for GOSTSHYP calculations.

INPUT SECTION: *$distort*

TYPE:

BOOL

DEFAULT:

True

OPTIONS:

True Enable integral screening for GOSTSHYP

False Disable integral screening for GOSTSHYP

RECOMMENDATION:

Use default. Disabling integral screening will lead to much higher memory usage and severe performance drops.

9.5.2.1 Hydrostatic Compression Force Field (HCFF)

The Hydrostatic Compression Force Field (HCFF) model was introduced by Stauch, Chakraborty and Head-Gordon.[99] In HCFF, mechanical forces that point towards the non-mass-weighted molecular centroid are used to compress a molecule. Care must be exercised when modeling extended molecules due to the tendency of HCFF to generate spherical geometries under very high pressure.[97] Also, the pressure input by the user is only a guess for the pressure that is applied to the molecule. The latter is calculated *a posteriori* based on the generated geometry and the molecular surface and is output as *HCFF Macroscopic Pressure*. Typically, the applied pressure is lower than the input pressure. It should be noted that the dependence on the nuclear gradient precludes the application of pressure to single atoms in

HCFF. Moreover, the increase in electronic energy when compressing a molecule is typically underestimated by HCFF, since the pressure acts only on the nuclei, whereas the compression of electron density is not modeled directly. HCFF works with any electronic structure method for which a nuclear gradient is available.

Example 9.15 Geometry optimization of diborane under pressure using the HCFF model with an input pressure of 3808 MPa

```
$molecule
   0 1
   B     0.0000000000    0.0000000000    0.8917854534
   B     0.0000000000    0.0000000000   -0.8917854534
   H    -0.5244343500    0.9105724300    1.4720415209
   H     0.5244343500   -0.9105724300    1.4720415209
   H    -0.5244343500    0.9105724300   -1.4720415209
   H     0.5244343500   -0.9105724300   -1.4720415209
   H     0.8561835151    0.4929549655    0.0000000000
   H    -0.8561835151   -0.4929549655    0.0000000000
$end

$rem
   JOBTYPE            opt
   METHOD             m06-2x
   BASIS              6-311++G(d,p)
   DISTORT            true
$end

$distort
   model              hcff
   pressure           3808
   scaling            1.0
   npoints_heavy      590
   npoints_hydrogen   590
$end
```

9.5.2.2 eXtended Hydrostatic Compression Force Field (X-HCFF)

The eXtended Hydrostatic Compression Force Field (X-HCFF) approach was introduced by Stauch to solve the problems associated with HCFF.[96] In X-HCFF, mechanical forces are used to compress the molecule as well, but, in contrast to HCFF, these forces are strictly perpendicular to the tessellated molecular surface, thus simulating truly hydrostatic conditions. As a result, chemically feasible geometries are retained even at high pressures. In addition, the user is able to input the precise pressure that is applied to the molecule during the simulation. It was suggested to use the unscaled atomic van der Waals radii in the tessellation routine.[96] X-HCFF works with any electronic structure method for which a nuclear gradient is available. It is also possible to perform an analytic Hessian calculation with X-HCFF using JOBTYPE = FREQ, to obtain related structural properties such as IR and Raman frequencies.[106]

It should be noted that the forces formulated for the X-HCFF model are not derived as the derivative of an energy term, but direcctly formulated and added to gradient. Thus no corresponding energy term is added to the electronic energy and consequently pressure induced increases in electronic energy are typically low. The X-HCFF forces can be

understood as a approximation to the gradient of the vDW volume.[111]

Example 9.16 Geometry optimization of the CO_2 dimer under a pressure of 100 GPa using the X-HCFF model

```
$molecule
   0 1
   O     2.6192991230   -0.0571311942    0.0000000000
   C     1.6782610262    0.6502025480    0.0000000000
   O     0.7413912820    1.3674070371    0.0000000000
   C    -1.6782610262   -0.6502025480    0.0000000000
   O    -2.6192991230    0.0571311942    0.0000000000
   O    -0.7413912820   -1.3674070371    0.0000000000
$end

$rem
   JOBTYPE            opt
   METHOD             pbe
   BASIS              cc-pvdz
   DISTORT            true
$end

$distort
   model              xhcff
   pressure           100000
   scaling            1.0
   npoints_heavy      302
   npoints_hydrogen   302
$end
```

9.5.2.3 PV term

The enthalpy H of a molecular system in an arbitrary pressure exerting medium may be expressed as the sum of the electronic energy E_{el} and the pressure P times the volume V occupied by the molecule.

$$H = E_{el} + PV \tag{9.43}$$

The probably most simple approximation for the molecular Volume is the vDW Volume, which is used here. It can be calculated by numerical integration of the molecular vDW surface (see 11.2.3.2) and leads to a well defined contribution to the molecular gradient.[111] This approach leads to pressure induced deformations similar to the X-HCFF model, however it provides a far more realistic correction to the energy.

A Hessian is available, to obtain IR and Raman frequencies using frequency calculations.

9.5.2.4 Gaussians On Surface Tesserae Simulate HYdrostatic Pressure (GOSTSHYP)

The Gaussians On Surface Tesserae Simulate HYdrostatic Pressure (GOSTSHYP) method, which was introduced by Scheurer and co-workers,[87] overcomes the problems associated with the mechanochemical models of pressure, *i.e.* HCFF and X-HCFF. GOSTSHYP uses a uniform field of Gaussian potentials that is placed on the tessellated molecular surface and that compresses the electron density. Each Gaussian potential G_j has the form

$$G_j = p_j \cdot \exp\left(-w_j(\mathbf{r} - \mathbf{r}_0)^2\right) \tag{9.44}$$

During the GOSTSHYP routine, the parameters of the Gaussian potentials, p_j and w_j, are adjusted such that a user-defined pressure is applied. Atoms and molecules can be treated, and the pressure-induced increase in the electronic energy is physically sound. During the SCF, the energy expression takes the form

$$\begin{aligned} E_{\text{GOSTSHYP}} &= \sum_j E_j = \sum_j \int G_j(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} \\ &= \sum_j \sum_{\mu,\nu} \sum_a \langle \chi_\mu | G_j | \chi_\nu \rangle \, c^*_{\mu a} c_{\nu a} \end{aligned} \tag{9.45}$$

Due to the availability of nuclear gradients, geometry optimizations under pressure using the GOSTSHYP model are possible. At present, GOSTSHYP is implemented at the SCF level, allowing calculations with Hartree-Fock and Density Functional Theory.

For good performance GOSTSHYP needs relatively large amounts of available RAM. If not enough available RAM is detected, GOSTSHYP will switch to a memory efficient algorithm at the cost of performance, a warning containing the required amount of memory for better performance will be printed in the output.

We found, that at the edges between the tessellation spheres of two atoms "negative amplitudes" p_j may appear. Since those would lead to nonphysical attractive pressure potentials they are generally blacklisted in GOSTSHYP calculations. This however leads to instabilities within SCF calculations. We found that negative amplitudes appear very rarely for VDW-scaling factors larger than 1.5 but become more likely to appear for smaller scaling factors. Thus we

recommend to use a scaling factor of at least 1.5 in GOSTSHYP calculations.

Example 9.17 Geometry optimization of cyclopentadiene and ethylene under a pressure of 40 GPa using the GOSTSHYP model

```
$molecule
   0 1
   C    1.1148422354   -0.6418674001    0.7279292386
   C    1.1148422354   -0.6418674001   -0.7279292386
   C    0.5936432126    0.5363396649    1.1772168767
   C   -2.0464511598   -0.6129291257    0.6711240568
   C   -2.0464511598   -0.6129291257   -0.6711240568
   C    0.5936432126    0.5363396649   -1.1772168767
   C    0.2915208637    1.4128825196    0.0000000000
   H    0.9756522868    2.2894492537    0.0000000000
   H   -0.7374232239    1.8214336422    0.0000000000
   H    1.4681344173   -1.4690333337   -1.3527755131
   H    1.4681344173   -1.4690333337    1.3527755131
   H   -2.3879086093    0.2541525765    1.2531118994
   H   -1.7231567891   -1.4887031107    1.2461940178
   H   -1.7231567891   -1.4887031107   -1.2461940178
   H   -2.3879086093    0.2541525765   -1.2531118994
   H    0.4773764265    0.8454441265    2.2200767812
   H    0.4773764265    0.8454441265   -2.2200767812
$end

$rem
   JOBTYPE               opt
   METHOD                pbe
   BASIS                 cc-pvdz
   GEOM_OPT_MAX_CYCLES   150
   SCF_ALGORITHM         diis_gdm
   MAX_SCF_CYCLES        150
   USE_LIBQINTS          1
   GEN_SCFMAN            1
   DISTORT               1
$end

$distort
   model                 gostshyp
   pressure              40000
   npoints_heavy         302
   npoints_hydrogen      302
   scaling               1.8
$end
```

9.6 Potential Energy Scans

It is often useful to scan the potential energy surface (PES), optimizing all other degrees of freedom for each particular value of the scanned variable(s). Such a "relaxed" scan may provide a rough estimate of a pathway between reactant and product—assuming the coordinate(s) for the scan has been chosen wisely—and is often used in development of classical force fields to optimize dihedral angle parameters. Ramachandran plots, for example, are key tools for studying conformational changes of peptides and proteins, and are essentially two-dimensional torsional scans.

In certain cases, relaxed scans might encounter some difficulties on optimizations. A "frozen" scan can be easier to perform because of no geometry optimizations although it provides less information of real dynamics.

Q-CHEM supports one- and two-dimensional PES scans, by setting JOBTYPE = PES_SCAN in the *$rem* section. In addition, a *$scan* input section with the following format should be specified, in the format below but with no more than two bond-length, bond-angle, or torsional variables specified.

```
$scan
   stre  atom1  atom2  value1 value2 incr
   ...
   bend  atom1  atom2  atom3  value1 value2 incr
   ...
   tors  atom1  atom2  atom3  atom4  value1 value2 incr
   ...
$end
```

Note: Potential scans with JOBTYPE = PES_SCAN are available for ground-state potential energy surfaces only, although one may compute excitation energies along the scanned coordinate(s) by including appropriate keywords for vertical excitation in the *$rem* section, *e.g.*, CIS_N_ROOTS for CIS/TDDFT excited states.

The first example below demonstrates how to scan the torsional potential of butane, which is a sequence of constrained optimizations with the C1–C2–C3–C4 dihedral angle fixed at $-180°$, $-165°$, $-150°$, . . ., $165°$, $180°$.

Example 9.18 One-dimensional torsional scan of butane

```
$molecule
   0 1
   C       1.934574    -0.128781    -0.000151
   C       0.556601     0.526657     0.000200
   C      -0.556627    -0.526735     0.000173
   C      -1.934557     0.128837    -0.000138
   H       2.720125     0.655980    -0.000236
   H       2.061880    -0.759501    -0.905731
   H       2.062283    -0.759765     0.905211
   H       0.464285     1.168064    -0.903444
   H       0.464481     1.167909     0.903924
   H      -0.464539    -1.167976     0.903964
   H      -0.464346    -1.168166    -0.903402
   H      -2.062154     0.759848     0.905185
   H      -2.720189    -0.655832    -0.000229
   H      -2.061778     0.759577    -0.905748
$end

$rem
   JOBTYPE  pes_scan
   METHOD   hf
   BASIS    sto-3g
$end

$scan
   tors 1 2 3 4 -180 180 15
$end
```

The next example is a two-dimension potential scan. The first dimension is a scan of the C1–C2–C3–C4 dihedral angle from $-180°$ to $180°$ degree in $30°$ intervals; the second dimension is a scan of the C2–C3 bond length from 1.5 Å to

1.6 Å in 0.05 Å increments.

Example 9.19 Two-dimensional torsional scan of butane

```
$molecule
   0 1
   C       1.934574    -0.128781    -0.000151
   C       0.556601     0.526657     0.000200
   C      -0.556627    -0.526735     0.000173
   C      -1.934557     0.128837    -0.000138
   H       2.720125     0.655980    -0.000236
   H       2.061880    -0.759501    -0.905731
   H       2.062283    -0.759765     0.905211
   H       0.464285     1.168064    -0.903444
   H       0.464481     1.167909     0.903924
   H      -0.464539    -1.167976     0.903964
   H      -0.464346    -1.168166    -0.903402
   H      -2.062154     0.759848     0.905185
   H      -2.720189    -0.655832    -0.000229
   H      -2.061778     0.759577    -0.905748
$end

$rem
   JOBTYPE  pes_scan
   METHOD   hf
   BASIS    sto-3g
$end

$scan
   tors 1 2 3 4 -180 180 30
   stre 2 3 1.5 1.6 0.05
$end
```

To perform a frozen PES scan, set FROZEN_SCAN to be TRUE and use input geometry in *Z*-matrix format. The example

demonstrates a frozen PES of the C1–C2 bond stretching from 1.0 Å to 2.0 Å for methanol.

Example 9.20 One-dimensional frozen PES scan of methanol

```
$molecule
  0 1
   C
   O   C  RCO
   H1  C  RCH1  O  H1CO
   X   C  1.00  O  XCO   H1  180.0
   H2  C  RCH2  X  H2CX  H1   90.0
   H3  C  RCH2  X  H2CX  H1  -90.0
   H4  O  ROH   C  HOC   H1  180.0

   RCO  = 1.421
   RCH1 = 1.094
   RCH2 = 1.094
   ROH  = 0.963
   H1CO = 107.2
   XCO  = 129.9
   H2CX = 54.25
   HOC  = 108.0
$end

$rem
   JOBTYPE        pes_scan
   EXCHANGE       s
   CORRELATION    vwn
   BASIS          3-21g
   FROZEN_SCAN    true
$end

$scan
   stre 1 2 1.0 2.0 0.5
$end
```

Q-CHEM supports one-dimensional restrained PES scans including bond-length, bond-angle, torsional, atom to center of mass of two atoms, and atom to a set of atoms. Note that the set of atoms is defined as the atoms with indices from `atom2` to `atom3`, where the atoms indexed at `atom2` and `atom3` are included in the set. These restrained scans apply a harmonic potential to bias geometry optimization. Q-CHEM also supports restrained PES scans for transition state search of typical S_N2 reactions where the geometry restraints have the form

$$f(R) = k\left(R_{12} \pm R_{34} - R\right)^2 \ . \tag{9.46}$$

Here, R_{12} and R_{34} are two bond lengths in the reaction coordinate and k is a force constant for restraining R in the range of $R_{12} \pm R_{34}$. To perform a restrained PES scan, the following format should be specified.

```
$scan
 r12      atom1  atom2  Rmax   Rmin   incr  force_constant
 ...
 rbend    atom1  atom2  atom3  Rmax   Rmin  incr  force_constant
 ...
 rtors    atom1  atom2  atom3  atom4  Rmax  Rmin  incr  force_constant
 ...
 r1to23   atom1  atom2  atom3  Rmax   Rmin  incr  force_constant
 ...
 r1toset  atom1  atom2  atom3  Rmax   Rmin  incr  force_constant
 ...
 r12mr34  atom1  atom2  atom3  atom4  Rmin  Rmax  incr  force_constant
 ...
 r12pr34  atom1  atom2  atom3  atom4  Rmin  Rmax  incr  force_constant
 ...
$end
```

Example 9.21 One-dimensional restrained PES scan of chloromethane S_N2 reaction

```
$molecule
  -1 1
   C     0.418808   -1.240869    0.249048
   Cl   -0.775224   -1.495584    1.586668
   H     1.408172   -1.490565    0.631227
   H     0.147593   -1.907736   -0.568952
   H     0.413296   -0.199000   -0.092071
   Cl    1.947359    1.619163   -1.747832
$end

$rem
   JOBTYPE   pes_scan
   METHOD    b3lyp
   BASIS     6-31G*
$end

$scan
   r12mr34 1 2 1 6 -2.0 2.0 0.2 1000.0
$end
```

Example 9.22 One-dimensional restrained torsional PES scan of butane

```
$molecule
 0 1
  C       1.934574    -0.128781    -0.000151
  C       0.556601     0.526657     0.000200
  C      -0.556627    -0.526735     0.000173
  C      -1.934557     0.128837    -0.000138
  H       2.720125     0.655980    -0.000236
  H       2.061880    -0.759501    -0.905731
  H       2.062283    -0.759765     0.905211
  H       0.464285     1.168064    -0.903444
  H       0.464481     1.167909     0.903924
  H      -0.464539    -1.167976     0.903964
  H      -0.464346    -1.168166    -0.903402
  H      -2.062154     0.759848     0.905185
  H      -2.720189    -0.655832    -0.000229
  H      -2.061778     0.759577    -0.905748
$end

$rem
   JOBTYPE PES_SCAN
   METHOD HF
   BASIS STO-3G
$end

$scan
   rtors 1 2 3 4 -180 180 15 1000.0
$end
```

Example 9.23 One-dimensional restrained PES scan pulling Na+ to benzene

```
$molecule
 1 1
  Na        0.0000000000    0.0000000000   -1.8358304117
  C         1.4040705495    0.0000000000    0.5608720717
  C         0.7020352748   -1.2159607646    0.5608720717
  C        -0.7020352748   -1.2159607646    0.5608720717
  C        -1.4040705495    0.0000000000    0.5608720717
  C        -0.7020352748    1.2159607646    0.5608720717
  C         0.7020352748    1.2159607646    0.5608720717
  H         2.4900504384    0.0000000000    0.5941562674
  H         1.2450252192   -2.1564469363    0.5941562674
  H        -1.2450252192   -2.1564469363    0.5941562674
  H        -2.4900504384    0.0000000000    0.5941562674
  H        -1.2450252192    2.1564469363    0.5941562674
  H         1.2450252192    2.1564469363    0.5941562674
$end

$rem
   JOBTYPE PES_SCAN
   BASIS STO-3G
   METHOD B3LYP
   DFT_D D3_BJ
$end

$scan
R1toSet 1 2 13 2.3 1.8 -0.1 1000.0
$end
```

9.7 Intrinsic Reaction Coordinate

The concept of a reaction path is chemically intuitive (a pathway from reactants to products) yet somewhat theoretically ambiguous because most mathematical definitions depend upon the chosen coordinate system. Stationary points on a potential energy surface are independent of this choice, but the path connecting them is not, and there exist various mathematical definitions of a "reaction path". Q-CHEM uses the intrinsic reaction coordinate (IRC) definition, as originally defined by Fukui,[40] which has come to be a fairly standard choice in quantum chemistry. The IRC is essentially sequence of small, steepest-descent paths going downhill from the transition state.

The reaction path is most unlikely to be a straight line and so by taking a finite step length along the direction of the gradient you will leave the "true" reaction path. A series of small steepest descent steps will zig-zag along the actual reaction path (a behavior known as "stitching"). Ishida *et al.*[54] developed a predictor-corrector algorithm, involving a second gradient calculation after the initial steepest-descent step, followed by a line search along the gradient bisector to get back on the path, and this algorithm was subsequently improved by Schmidt *et al.*.[89] This is the method that Q-CHEM adopts. It cannot be used for the first downhill step from the transition state, since the gradient is zero, so instead a step is taken along the Hessian mode whose frequency is imaginary.

The reaction path can be defined and followed in Z-matrix coordinates, Cartesian coordinates or mass-weighted Cartesian coordinates. The latter represents the "true" IRC as defined by Fukui.[40] If the rationale for following the reaction path is simply to determine which local minima are connected by a given transition state, which, is arguably the major use of IRC algorithms, then the choice of coordinates is irrelevant. In order to use the IRC code, the transition state geometry and the exact Hessian must be available. These must be computed via two prior calculations, with JOBTYPE = TS (transition structure search) and JOBTYPE = FREQ (Hessian calculation), respectively. Job control variables and examples appear below.

An IRC calculation is invoked by setting JOBTYPE = RPATH in the *$rem* section, and additional *$rem* variables are described below. IRC calculations may benefit from the methods discussed in Section 9.3 for obtaining good initial guesses for transition-state structures. The final pathway (in both forward and backward directions) is printed at the end of the Q-CHEM output file as a sequence of Cartesian-coordinate geometries representing steps along the pathway. The format for this output is a standard one that is readable, for example, by the Visual Molecular Dynamics (VMD) program.[1,53]

RPATH_COORDS

Determines which coordinate system to use in the IRC search.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Use mass-weighted coordinates.

1 Use Cartesian coordinates.

2 Use Z-matrix coordinates.

RECOMMENDATION:

Use the default. Note that use of Z-matrix coordinates requires that geometries be input in Z-matrix format.

RPATH_DIRECTION

Determines the first direction of the eigenmode to follow. This will not usually be known prior to the Hessian diagonalization.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Descend in the positive direction of the eigenmode, then restart in the negative direction.

-1 Descend in the negative direction of the eigenmode, then restart in the positive direction.

RECOMMENDATION:

It is usually not possible to determine in which direction to go *a priori*, so both directions are automatically considered. A job that reads in the final geometry from the reaction path job will use the final step from the second direction.

RPATH_MAX_CYCLES

Specifies the maximum number of points to find on the reaction path.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

n User-defined number of cycles.

RECOMMENDATION:

Use more points if the minimum is desired, but not reached using the default.

RPATH_MAX_STEPSIZE

Specifies the maximum step size to be taken (in 0.001 a.u.).

TYPE:

INTEGER

DEFAULT:

150 corresponding to a step size of 0.15 a.u..

OPTIONS:

n Step size = n/1000 a.u.

RECOMMENDATION:

None.

RPATH_TOL_DISPLACEMENT

Specifies the convergence threshold for the step. If a step size is chosen by the algorithm that is smaller than this, the path is deemed to have reached the minimum.

TYPE:

INTEGER

DEFAULT:

5000 Corresponding to 0.005 a.u.

OPTIONS:

n User-defined. Tolerance = n/1000000 a.u.

RECOMMENDATION:

Use the default. Note that this option *only* controls the threshold for ending the RPATH job and does nothing to the intermediate steps of the calculation. A smaller value will provide reaction paths that end closer to the true minimum. Use of smaller values without adjusting RPATH_MAX_STEPSIZE, however, can lead to oscillations about the minimum.

RPATH_PRINT

Specifies the print output level.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n

RECOMMENDATION:

Use the default, as little additional information is printed at higher levels. Most of the output arises from the multiple single point calculations that are performed along the reaction pathway.

Example 9.24 Reaction path search. Note that there are three required jobs: a TS search, followed by a frequency (Hessian) calculation, and finally the IRC calculation.

```
$molecule
   0 1
   C
   H 1 1.20191
   N 1 1.22178 2 72.76337
$end

$rem
   JOBTYPE            ts
   BASIS              sto-3g
   METHOD             hf
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE            freq
   METHOD             hf
   BASIS              sto-3g
   SCF_GUESS          read
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE            rpath
   BASIS              sto-3g
   METHOD             hf
   SCF_GUESS          read
   RPATH_MAX_CYCLES   50
$end
```

9.8 Nonadiabatic Couplings and Optimization of Minimum-Energy Crossing Points

9.8.1 Nonadiabatic Couplings

Conical intersections are degeneracies between Born-Oppenheimer potential energy surfaces that facilitate nonadiabatic transitions between excited states, *i.e.*, internal conversion and intersystem crossing processes, both of which represent a breakdown of the Born-Oppenheimer approximation.[49,50,66] Although simultaneous intersections between more than two electronic states are possible,[66] consider for convenience the two-state case, and let

$$\mathbf{H} = \begin{pmatrix} H_{JJ}(\mathbf{R}) & H_{JK}(\mathbf{R}) \\ H^{*}_{JK}(\mathbf{R}) & H_{KK}(\mathbf{R}) \end{pmatrix} . \tag{9.47}$$

denote the matrix representation of the vibronic (vibrational + electronic) Hamiltonian [Eq. (4.2)] in a basis of two electronic states, J and K. (Electronic degrees of freedom have been integrated out of this expression, and $\mathbf{R}$ represents the remaining, nuclear coordinates.) By definition, the Born-Oppenheimer states are the ones that diagonalize $\mathbf{H}$ at a particular molecular geometry $\mathbf{R}$, and thus two conditions must be satisfied in order to obtain degeneracy in the Born-Oppenheimer representation: $H_{JJ} = H_{KK}$ and $H_{JK} = 0$. As such, degeneracies between two Born-Oppenheimer potential energy surfaces exist in subspaces of dimension $N_{\text{int}} - 2$, where $N_{\text{int}} = 3N_{\text{atoms}} - 6$ is the number of internal (vibrational) degrees of freedom (assuming the molecule is non-linear). This $(N_{\text{int}} - 2)$-dimensional subspace is known as the *seam space* because the two states are degenerate everywhere within this space. In the remaining two degrees of freedom, known as the *branching space*, the degeneracy between Born-Oppenheimer surfaces is lifted by an infinitesimal displacement, which in a three-dimensional plot resembles a double cone about the point of intersection, hence the name *conical intersection*.

The branching space is defined by the span of a pair of vectors $\mathbf{g}_{JK}$ and $\mathbf{h}_{JK}$. The former is simply the difference in the gradient vectors of the two states in question,

$$\mathbf{g}_{JK} = \frac{\partial E_J}{\partial \mathbf{R}} - \frac{\partial E_K}{\partial \mathbf{R}} \ , \tag{9.48}$$

and is readily evaluated at any level of theory for which analytic energy gradients are available (or less-readily, via finite difference, if they are not!). The definition of the *nonadiabatic coupling vector* $\mathbf{h}_{JK}$, on the other hand, is more involved and not directly amenable to finite-difference calculations:

$$\mathbf{h}_{JK} = \left\langle \Psi_J \left| \left(\partial \hat{H} / \partial \mathbf{R} \right) \right| \Psi_K \right\rangle \ . \tag{9.49}$$

This is closely related to the *derivative coupling* vector

$$\mathbf{d}_{JK} = \left\langle \Psi_J \middle| (\partial / \partial \mathbf{R}) \middle| \Psi_K \right\rangle = \frac{\mathbf{h}_{JK}}{E_J - E_K} \ . \tag{9.50}$$

The latter expression for $\mathbf{d}_{JK}$ demonstrates that the coupling between states becomes large in regions of the potential surface where the two states are nearly degenerate. The relative orientation and magnitudes of the vectors $\mathbf{g}_{JK}$ and $\mathbf{h}_{JK}$ determined the topography around the intersection, *i.e.*, whether the intersection is "peaked" or "sloped";[4,110] see Ref. 50 for a pedagogical overview.

Algorithms to compute the nonadiabatic couplings $\mathbf{d}_{JK}$ are not widely available in quantum chemistry codes, but thanks to the efforts of the Herbert and Subotnik groups, they are available in Q-CHEM when the wave functions Ψ_J and Ψ_K, and corresponding electronic energies E_J and E_K, are computed at the CIS or TDDFT level,[36,75,113,114] or at the corresponding spin-flip (SF) levels of theory (SF-CIS or SF-TDDFT). The spin-flip implementation[113] is particularly significant, because only that approach—and not traditional spin-conserving CIS or TDDFT—affords correct topology around conical intersections that involve the ground state.[49]

To understand why, suppose that J in Eq. (9.47) represents the ground state; call it $J = 0$ for definiteness. In linear response theory (TDDFT) or in CIS (by virtue of Brillouin's theorem), the coupling matrix elements between the reference (ground) state and all of the excited states vanish identically, hence $H_{0K}(\mathbf{R}) \equiv 0$. This means that there is only one condition to satisfy in order to obtain degeneracy, hence the branching space is one- rather than two-dimensional, for any conical intersection that involves the ground state.[61] (For intersections between two excited states, the topology should be correct.) In the spin-flip approach, however, the reference state has a different spin multiplicity than the target states; if the latter have spin quantum number S, then the reference state has spin $S + 1$. This has the effect that the ground state of interest (spin S) is treated as an excitation, and thus on a more equal footing with excited states of the same spin, and it rigorously fixes the topology problem around conical intersections.[49,50,113] This can have consequences for simulation of internal conversion to the ground state, where for example S_1 lifetimes may be significantly affected by warping of the potential energy surfaces around the S_1/S_0 intersection in conventional TDDFT.[116]

Nonadiabatic (derivative) couplings are available for both CIS and TDDFT. The CIS nonadiabatic couplings can be obtained from direct differentiation of the wave functions with respect to nuclear positions.[36,113] For TDDFT, the same procedure can be carried out to calculate the approximate nonadiabatic couplings, in what has been termed the "pseudo-wave function" approach.[74,113] Formally more rigorous TDDFT nonadiabatic couplings derived from quadratic response theory are also available, although they are subject to certain undesirable, accidental singularities if for the two states J and K in Eq. (9.49), the energy difference $|E_J - E_K|$ is quasi-degenerate with the excitation energy $\omega_I = E_I - E_0$ for some third state, I.[75,114] As such, the pseudo-wave function method is the recommended approach for computing nonadiabatic couplings with TDDFT, although in the spin flip case the pseudo-wave function approach is rigorously equivalent to the pseudo-wave function approach, and is free of singularities.[114]

Finally, we note that there is mounting evidence that SF-TDDFT calculations are most accurate when used with functionals containing $\sim$50% Hartree-Fock exchange,[49,52,91] and many studies with this method have used the BH&HLYP functional; see Refs. 50 and 49 for reviews. The BH&H-LYP functional combines LYP correlation is combined with Becke's "half and half" exchange functional.[42] In Q-CHEM, it is selected by setting METHOD = BHHLYP.

9.8.2 Job Control and Examples

In order to perform nonadiabatic coupling calculations, the *$derivative_coupling* section must be given:

```
$derivative_coupling
<one line comment>
i, j, k, ...
$end
```

Nonadiabatic couplings will then be computed between all pairs of the states $i, j, k, \ldots$; use "0" to request the HF or DFT reference state, "1" for the first excited state, etc. Note that the derivative couplings are have dimensions of inverse length and are printed in atomic units, *i.e.*, bohr^{-1}.

CALC_NAC

Determines whether we are calculating nonadiabatic couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate nonadiabatic couplings.

FALSE Do not calculate nonadiabatic couplings.

RECOMMENDATION:

None.

CIS_DER_NUMSTATE

Determines among how many states we calculate nonadiabatic couplings. These states must be specified in the *$derivative_coupling* section.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not calculate nonadiabatic couplings.

n Calculate $n(n-1)/2$ pairs of nonadiabatic couplings.

RECOMMENDATION:

None.

SET_QUADRATIC

Determines whether to include full quadratic response contributions for TDDFT.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Include full quadratic response contributions for TDDFT.

FALSE Use pseudo-wave function approach.

RECOMMENDATION:

The pseudo-wave function approach is usually accurate enough and is free of accidental singularities. Consult Refs. 114 and 75 for additional guidance.

Example 9.25 Nonadiabatic couplings among the lowest five singlet states of ethylene, computed at the TD-B3LYP level using the pseudo-wave function approach.

```
$molecule
   0 1
   C     1.85082356   -1.78953123    0.00000000
   H     2.38603593   -2.71605577    0.00000000
   H     0.78082359   -1.78977646    0.00000000
   C     2.52815456   -0.61573833    0.00000000
   H     1.99294220    0.31078621    0.00000000
   H     3.59815453   -0.61549310    0.00000000
$end

$rem
   EXCHANGE           b3lyp
   BASIS              6-31G*
   CIS_N_ROOTS        4
   CIS_TRIPLETS       false
   MAX_CIS_CYCLES     50
   CIS_DER_NUMSTATE   5
   CALC_NAC           true
$end

$derivative_coupling
   0 is the reference state
   0 1 2 3 4
$end
```

Example 9.26 Nonadiabatic couplings between S_1 and S_3 states of ethylene using BH&HLYP and spin-flip TDDFT.

```
$molecule
   0 3
   C    1.85082356   -1.78953123    0.00000000
   H    2.38603593   -2.71605577    0.00000000
   H    0.78082359   -1.78977646    0.00000000
   C    2.52815456   -0.61573833    0.00000000
   H    1.99294220    0.31078621    0.00000000
   H    3.59815453   -0.61549310    0.00000000
$end

$rem
   EXCHANGE           bhhlyp
   BASIS              6-31G*
   SPIN_FLIP          true
   UNRESTRICTED       true
   CIS_N_ROOTS        4
   CIS_TRIPLETS       false
   MAX_CIS_CYCLES     50
   CIS_DER_NUMSTATE   2
   CALC_NAC           true
$end

$derivative_coupling
   comment
   1 3
$end
```

Example 9.27 Nonadiabatic couplings between S_1 and S_2 states of ethylene computed via quadratic response theory at the TD-B3LYP level.

```
$molecule
   0 1
   C     1.85082356   -1.78953123    0.00000000
   H     2.38603593   -2.71605577    0.00000000
   H     0.78082359   -1.78977646    0.00000000
   C     2.52815456   -0.61573833    0.00000000
   H     1.99294220    0.31078621    0.00000000
   H     3.59815453   -0.61549310    0.00000000
$end

$rem
   EXCHANGE          b3lyp
   BASIS             6-31G*
   CIS_N_ROOTS       4
   CIS_TRIPLETS      false
   RPA               true
   MAX_CIS_CYCLES    50
   CIS_DER_NUMSTATE  2
   CALC_NAC          true
   SET_QUADRATIC     true ! include full quadratic response
$end

$derivative_coupling
   comment
   1 2
$end
```

9.8.3 Minimum-Energy Crossing Points

The seam space of a conical intersection is really a (hyper)surface of dimension $N_{\text{int}} - 2$, and while the two electronic states in question are degenerate at every point within this space, the electronic energy varies from one point to the next. To provide a simple picture of photochemical reaction pathways, it is often convenient to locate the minimum-energy crossing point (MECP) within this $(N_{\text{int}}-2)$-dimensional seam. Two separate minimum-energy pathway searches, one on the excited state starting from the ground-state geometry and terminating at the MECP, and the other on the ground state starting from the MECP and terminating at the ground-state geometry, then affords a photochemical mechanism. (See Ref. 112 for a simple example.) In some sense, then, the MECP is to photochemistry what the transition state is to reactions that occur on a single Born-Oppenheimer potential energy surface. One should be wary of pushing this analogy too far, because whereas a transition state reasonably be considered to be a bottleneck point on the reaction pathway, the path through a conical intersection may be downhill and perhaps therefore more likely to proceed from one surface to the other at a point "near" the intersection, and in addition there can be multiple conical intersections between the same pair of states so more than one photochemical mechanism may be at play. Such complexity could be explored, albeit at significantly increased cost, using nonadiabatic "surface hopping" *ab initio* molecular dynamics, as described in Section 9.9.7. Here we describe the computationally-simpler procedure of locating an MECP along a conical seam.

Recall that the branching space around a conical intersection between electronic states J and K is spanned by two vectors, $\mathbf{g}_{JK}$ [Eq. (9.48)] and $\mathbf{h}_{JK}$ [Eq. (9.49)]. While the former is readily available in analytic form for any electronic structure method that has analytic excited-state gradients, the nonadiabatic coupling vector $\mathbf{h}_{JK}$ is *not* available for most methods. For this reason, several algorithms have been developed to optimize MECPs without the need to evaluate $\mathbf{h}_{JK}$, and three such algorithms are available in Q-CHEM.

Martínez and coworkers[62] developed a penalty-constrained MECP optimization algorithm that consists of minimizing the objective function

$$F_\sigma(\mathbf{R}) = \tfrac{1}{2}\big[E_I(\mathbf{R}) + E_J(\mathbf{R})\big] + \sigma\left(\frac{\big[E_I(\mathbf{R}) - E_J(\mathbf{R})\big]^2}{E_I(\mathbf{R}) - E_J(\mathbf{R}) + \alpha}\right) , \tag{9.51}$$

where α is a fixed parameter to avoid singularities and σ is a Lagrange multiplier for a penalty function meant to drive the energy gap to zero. Minimization of F_σ is performed iteratively for increasingly large values σ.

A second MECP optimization algorithm is a simplification of the penalty-constrained approach that we call the "direct" method. Here, the gradient of the objective function is

$$\mathbf{G} = \mathbf{P}\mathbf{G}_{\rm mean} + 2(E_K - E_J)\mathbf{G}_{\rm diff} , \tag{9.52}$$

where

$$\mathbf{G}_{\rm mean} = \tfrac{1}{2}(\mathbf{G}_J + \mathbf{G}_K) \tag{9.53}$$

is the mean energy gradient, with $\mathbf{G}_i = \partial E_i/\partial\mathbf{R}$ being the nuclear gradient for state i, and

$$\mathbf{G}_{\rm diff} = \frac{\mathbf{G}_K - \mathbf{G}_J}{||\mathbf{G}_K - \mathbf{G}_J||} \tag{9.54}$$

is the normalized difference gradient. Finally,

$$\mathbf{P} = \mathbf{1} - \mathbf{G}_{\rm diff}\mathbf{G}_{\rm diff}^\top \tag{9.55}$$

projects the gradient difference direction out of the mean energy gradient in Eq. (9.52). The algorithm then consists in minimizing along the gradient $\mathbf{G}$, with for the iterative cycle over a Lagrange multiplier, which can sometimes be slow to converge.

The third and final MECP optimization algorithm that is available in Q-CHEM is the branching-plane updating method developed by Morokuma and coworkers[64] and implemented in Q-CHEM by Zhang and Herbert.[112] This algorithm uses a gradient that is similar to that in Eq. (9.52) but projects out not just $\mathbf{G}_{\rm diff}$ in Eq. (9.55) but also a second vector that is orthogonal to it, representing an iteratively-updated approximation to the branching space.

None of these three methods requires evaluation of nonadiabatic couplings, and all three can be used to optimize MECPs at the CIS, SF-CIS, TDDFT, SF-TDDFT, and SOS-CIS(D0) levels. The direct algorithm can also be used for EOM-XX-CCSD methods (XX = EE, IP, or EA). It should be noted that since EOM-XX-CCSD is a linear response method, it suffers from the same topology problem around conical intersections involving the ground state that was described in regards to TDDFT in Section 9.8.1. With spin-flip approaches, correct topology is obtained.[113]

Analytic derivative couplings are available for (SF-)CIS and (SF-)TDDFT, so for these methods one can alternatively employ an optimization algorithm that makes use of both $\mathbf{g}_{JK}$ and $\mathbf{h}_{JK}$. Such an algorithm, due to Schlegel and coworkers,[13] is available in Q-CHEM and consists of optimization along the gradient in Eq. (9.52) but with a projector

$$\mathbf{P} = \mathbf{1} - \mathbf{G}_{\rm diff}\mathbf{G}_{\rm diff}^\top - \mathbf{y}\mathbf{y}^\top \tag{9.56}$$

where

$$\mathbf{y} = \frac{(\mathbf{1} - \mathbf{x}\mathbf{x}^\top)\mathbf{h}_{JK}}{||(\mathbf{1} - \mathbf{x}\mathbf{x}^\top)\mathbf{h}_{JK}||} , \tag{9.57}$$

in place of the projector in Eq. (9.55). Equation (9.56) has the effect of projecting the span of $\mathbf{g}_{JK}$ and $\mathbf{h}_{JK}$ (*i.e.*, the branching space) out of state-averaged gradient in Eq. (9.52). The tends to reduce the number of iterations necessary to converge the MECP, and since calculation of the (optional) $\mathbf{h}_{JK}$ vector represents only a slight amount of overhead on top of the (required) $\mathbf{g}_{JK}$ vector, this last algorithm tends to yield significant speed-ups relative to the other three.[113] As such, it is the recommended choice for (SF-)CIS and (SF-)TDDFT.

It should be noted that while the spin-flip methods cure the topology problem around conical intersections that involve the ground state, this method tends to exacerbate spin contamination relative to the corresponding spin-conserving approaches.[49,115] While spin contamination is certainly present in traditional, spin-conserving CIS and TDDFT, it presents the following unique challenge in spin-flip methods. Suppose, for definiteness, that one is interested in singlet excited states. Then the reference state for the spin-flip methods should be the high-spin triplet. A spin-flipping excitation will then generate S_0, S_1, S_2, ... but will also generate the $M_S = 0$ component of the triplet reference state, which therefore appears in what is ostensibly the singlet manifold. Q-CHEM attempts to identify this automatically, based on a threshold for $\langle \hat{S}^2 \rangle$, but severe spin contamination can sometimes defeat this algorithm,[112] hampering Q-CHEM's ability to distinguish singlets from triplets (in this particular example). An alternative might be the state-tracking procedure that is described in Section 9.8.5. The spin-adapted spin-flip (SA-SF) method, either at the CIS or the TDDFT level, can be used to eliminate spin contamination by adding the minimal number of determinants necessary to obtain $\hat{S}^2$ eigenstates.[115] The SA-SF approach is described in Section 7.2.3.3. It presently lacks analytic gradients but can be used to spot-check trajectories or MECP optimization paths obtained using SF-TDDFT.

9.8.4 Job Control and Examples

For MECP optimization, set MECP_OPT = TRUE in the *$rem* section, and note that the *$derivative_coupling* input section discussed in Section 9.8.2 is not necessary in this case.

MECP_OPT

Determines whether we are doing MECP optimizations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do MECP optimization.

FALSE Do not do MECP optimization.

RECOMMENDATION:

None.

MECP_METHODS

Determines which method to be used.

TYPE:

STRING

DEFAULT:

BRANCHING_PLANE

OPTIONS:

BRANCHING_PLANE Use the branching-plane updating method.

MECP_DIRECT Use the direct method.

PENALTY_FUNCTION Use the penalty-constrained method.

RECOMMENDATION:

The direct method is stable for small molecules or molecules with high symmetry. The branching-plane updating method is more efficient for larger molecules but does not work if the two states have different symmetries. If using the branching-plane updating method, GEOM_OPT_COORDS must be set to 0 in the *$rem* section, as this algorithm is available in Cartesian coordinates only. The penalty-constrained method converges slowly and is suggested only if other methods fail.

MECP_STATE1

Sets the first Born-Oppenheimer state for MECP optimization.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[i,j] Find the jth excited state with the total spin i; $j = 0$ means the SCF ground state.

RECOMMENDATION:

i is ignored for restricted calculations; for unrestricted calculations, i can only be 0 or 1.

MECP_STATE2

Sets the second Born-Oppenheimer state for MECP optimization.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[i,j] Find the jth excited state with the total spin i; $j = 0$ means the SCF ground state.

RECOMMENDATION:

i is ignored for restricted calculations; for unrestricted calculations, i can only be 0 or 1.

CIS_S2_THRESH

Determines whether a state is a singlet or triplet in unrestricted calculations.

TYPE:

INTEGER

DEFAULT:

120

OPTIONS:

n Sets the $\langle \hat{S}^2 \rangle$ threshold to $n/100$

RECOMMENDATION:

For the default case, states with $\langle \hat{S}^2 \rangle > 1.2$ are treated as triplet states and other states are treated as singlets.

MECP_PROJ_HESS

Determines whether to project out the coupling vector from the Hessian when using branching plane updating method.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Use the default.

Example 9.28 MECP optimization of an intersection between the S_2 and S_3 states of NO_2^-, using the direct method at the SOS-CIS(D0) level.

```
$molecule
   -1 1
   N1
   O2  N1  RNO
   O3  N1  RNO  O2  AONO

   RNO  = 1.50
   AONO = 100
$end

$rem
   JOBTYPE         = opt
   METHOD          = soscis(d0)
   BASIS           = aug-cc-pVDZ
   AUX_BASIS       = rimp2-aug-cc-pVDZ
   PURECART        = 1111
   CIS_N_ROOTS     = 4
   CIS_TRIPLETS    = false
   CIS_SINGLETS    = true
   MEM_STATIC      = 900
   MEM_TOTAL       = 1950
   MECP_OPT        = true
   MECP_STATE1     = [0,2]
   MECP_STATE2     = [0,3]
   MECP_METHODS    = mecp_direct
$end
```

Example 9.29 Optimization of the ethylidene MECP between S_0 and S_1 at the SF-TDDFT level using the branching-plane updating method.

```
$molecule
   0 3
   C   0.044626  -0.2419240   0.357157
   C   0.008905   0.6727548   1.460500
   H   0.928425  -0.1459163  -0.272095
   H  -0.831032  -0.1926895  -0.288529
   H  -0.009238   0.9611331   2.479936
   H   0.068314  -1.2533580   0.778847
$end

$rem
   JOBTYPE           opt
   METHOD            bhhlyp
   BASIS             6-31G(d,p)
   MECP_OPT          true
   MECP_METHODS      branching_plane
   MECP_PROJ_HESS    true  ! project out y vector from the hessian
   GEOM_OPT_COORDS   0     ! currently only works for Cartesian coordinate
   GEOM_OPT_DRIVER   optimize
   SPIN_FLIP         true
   UNRESTRICTED      true
   CIS_N_ROOTS       4
   MECP_STATE1       [0,1]
   MECP_STATE2       [0,2]
   CIS_S2_THRESH     120
$end
```

Example 9.30 Optimization of the twisted-pyramidalized ethylene MECP between S_0 and S_1 at the SF-TDDFT level using the penalty-constrained updating method.

```
$molecule
   0 3
   C  -0.015889   0.073532  -0.059559
   C   0.012427  -0.002468   1.315694
   H   0.857876   0.147014  -0.710529
   H  -0.936470  -0.011696  -0.626761
   H   0.764557   0.663381   1.762573
   H   0.740773  -0.869764   1.328583
$end

$rem
   JOBTYPE         opt
   MECP_OPT        true
   MECP_METHODS    penalty_function
   METHOD          bhhlyp
   SPIN_FLIP       true
   UNRESTRICTED    true
   BASIS           6-31G(d,p)
   CIS_N_ROOTS     4
   MECP_STATE1     [0,1]
   MECP_STATE2     [0,2]
   CIS_S2_THRESH   120
   GEOM_OPT_DRIVER optimize
$end
```

Example 9.31 Optimization of the $\tilde{B}^1A_2$ and $\tilde{A}^1B_2$ states of N_3^+ at the EOM-EE-CCSD level using the direct method. See Section 7.10.27 for the specifics of performing MECP optimization with coupled cluster.

```
$molecule
   1 1
   N1
   N2 N1 rNN
   N3 N2 rNN N1 aNNN

   rNN  =  1.54
   aNNN = 50.0
$end

$rem
   JOBTYPE                opt
   MECP_OPT               true
   MECP_METHODS           mecp_direct
   METHOD                 eom-ccsd
   BASIS                  6-31g
   EE_SINGLETS            [0,2,0,2]
   XOPT_STATE_1           [0,2,2]
   XOPT_STATE_2           [0,4,1]
   CCMAN2                 false
   GEOM_OPT_TOL_GRADIENT  30
   GEOM_OPT_DRIVER        optimize
$end
```

Example 9.32 Optimization of the ethylidene MECP between S_0 and S_1 at the SF-TDDFT level with analytic derivative couplings.

```
$molecule
   0 3
   C    0.044626  -0.241924   0.357157
   C    0.008905   0.672754   1.460500
   H    0.928425  -0.145916  -0.272095
   H   -0.831032  -0.192689  -0.288529
   H   -0.009238   0.961133   2.479936
   H    0.068314  -1.253358   0.778847
$end

$rem
   JOBTYPE            opt
   MECP_OPT           true
   MECP_METHODS       branching_plane
   MECP_PROJ_HESS     true
   GEOM_OPT_COORDS    0
   MECP_STATE1        [0,1]
   MECP_STATE2        [0,2]
   UNRESTRICTED       true
   SPIN_FLIP          true
   CIS_N_ROOTS        4
   CALC_NAC           true
   CIS_DER_NUMSTATE   2
   MAX_CIS_CYCLES     50
   EXCHANGE           bhhlyp
   BASIS              6-31G(d,p)
   INTEGRAL_SYMMETRY  true
$end
```

9.8.5 State-Tracking Algorithm

For optimizing excited-state geometries and other applications, it can be important to find and follow electronically excited states of a particular character as the geometry changes. Various state-tracking procedures have been proposed for such cases.[22,44,115] One such algorithm, based on the overlap of the attachment/detachment densities at successive steps, in available in Q-CHEM (see Section 7.14.2).[22] Using the densities avoids any issues that may be introduced by sign changes in the orbitals or configuration-interaction coefficients.

Two parameters are used to influence the choice of the electronic surface. One (γ_E) controls the energy window for states included in the search, and the other (γ_S) controls how well the states must overlap in order to be considered of the same character. These can be set by the user or generated automatically based on the magnitude of the nuclear displacement. The energy window is defined relative to the estimated energy for the current step (*i.e.*, $E_{\rm est} \pm \gamma_E$), which in turn is based on the energy, gradient and nuclear displacement of previous steps. This estimated energy is specific to the type of calculation (*e.g.*, geometry optimization).

The similarity metric for the overlap is defined as

$$\mathcal{S} = 1 - \tfrac{1}{2}\big(||\Delta\mathbf{A}|| + ||\Delta\mathbf{D}||\big) \tag{9.58}$$

where $\Delta\mathbf{A} = \mathbf{A}_{t+1} - \mathbf{A}_t$ is the difference in attachment density matrices (Eq. (7.169)) and $\Delta\mathbf{D} = \mathbf{D}_{t+1} - \mathbf{D}_t$ is the difference in detachment density matrices (Eq. (7.167)), at successive steps. Equation (9.58) uses the matrix spectral norm,

$$||\mathbf{M}|| = \big(\lambda_{\rm max}\mathbf{M}^\dagger\mathbf{M}\big)^{1/2} \tag{9.59}$$

where $\lambda_{\rm max}$ is the largest eigenvalue of $\mathbf{M}$.

The selected state always satisfies one of the following

1. It is the only state in the window defined by γ_E.
2. It is the state with the largest overlap, provided at least one state has $\mathcal{S} \geq \gamma_S$.
3. It is the nearest state energetically if all states in the window have $\mathcal{S} < \gamma_S$, or if there are no states in the energy window.

State-following can currently be used with CIS or TDDFT excited states and is initiated with the *$rem* variable STATE_FOLLOW. It can be used with geometry optimization, *ab initio* molecular dynamics,[22] or with the freezing/growing-string method. The desired state is specified using CIS_STATE_DERIV for optimization or dynamics, or using SET_STATE_REACTANT and SET_STATE_PRODUCT for the freezing- or growing-string methods. The results for geometry optimizations can be affected by the step size (GEOM_OPT_DMAX), and using a step size smaller than the default value can provide better results. Also, it is often challenging to converge the strings in freezing/growing-string calculations.

STATE_FOLLOW

Turns on state following.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use state-following.

TRUE Use state-following.

RECOMMENDATION:

None.

FOLLOW_ENERGY

Adjusts the energy window for near states

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use dynamic thresholds, based on energy difference between steps.

n Search over selected state $E_{\mathrm{est}} \pm n \times 10^{-6}\ E_h$.

RECOMMENDATION:

Use a wider energy window to follow a state diabatically, smaller window to remain on the adiabatic state most of the time.

FOLLOW_OVERLAP

Adjusts the threshold for states of similar character.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use dynamic thresholds, based on energy difference between steps.

n Percentage overlap for previous step and current step.

RECOMMENDATION:

Use a higher value to require states have higher degree of similarity to be considered the same (more often selected based on energy).

9.9 *Ab Initio* Molecular Dynamics

9.9.1 Introduction

Q-CHEM can propagate classical molecular dynamics trajectories on the Born-Oppenheimer potential energy surface generated by a particular theoretical model chemistry (*e.g.*, B3LYP/6-31G* or MP2/aug-cc-pVTZ). This procedure, in which the forces on the nuclei are evaluated on-the-fly, is known variously as "direct dynamics", "*ab initio* molecular dynamics" (AIMD), or "Born-Oppenheimer molecular dynamics" (BOMD). In its most straightforward form, a BOMD calculation consists of an energy + gradient calculation at each molecular dynamics time step, and thus each time step is comparable in cost to one geometry optimization step. A BOMD calculation may be requested using any SCF energy + gradient method available in Q-CHEM, including excited-state dynamics in cases where excited-state analytic gradients are available. As usual, Q-CHEM will automatically evaluate derivatives by finite-difference if the analytic versions are not available for the requested method, but in AIMD applications this is very likely to be prohibitively expensive.

While the number of time steps required in most AIMD trajectories dictates that economical (typically SCF-based) underlying electronic structure methods are required, any method with available analytic gradients can reasonably be used for BOMD, including (within Q-CHEM) HF, DFT, MP2, RI-MP2, and CCSD. The RI-MP2 method, especially when combined with Fock matrix and Z-vector extrapolation (as described below) is particularly effective as an alternative to DFT-based dynamics.

9.9.2 Overview and Basic Job Control

Initial Cartesian coordinates and velocities must be specified for the nuclei. Coordinates are specified in the *$molecule* section as usual, while velocities can be specified using a *$velocity* section with the form:

```
$velocity
v_{x,1}  v_{y,1}  v_{z,1}
v_{x,2}  v_{y,2}  v_{z,2}
v_{x,N}  v_{y,N}  v_{z,N}
$end
```

Here $v_{x,i}$, $v_{y,i}$, and $v_{z,i}$ are the x, y, and z Cartesian velocities of the ith nucleus, specified in atomic units (bohrs per atomic unit of time, where 1 a.u. of time is approximately 0.0242 fs). The *$velocity* section thus has the same form as the *$molecule* section, but without atomic symbols and without the line specifying charge and multiplicity. The atoms must be ordered in the same manner in both the *$velocity* and *$molecule* sections.

As an alternative to a *$velocity* section, initial nuclear velocities can be sampled from certain distributions (*e.g.*, Maxwell-Boltzmann), using the AIMD_INIT_VELOC variable described below. AIMD_INIT_VELOC can also be set to QUASICLASSICAL, which triggers the use of quasi-classical trajectory molecular dynamics (see Section 9.9.6).

The nuclear mass can be initialized by a *$mass* section with the form:

```
$mass
m_1  m_2
m_3   ! mass of 3rd atom
m_4
$end
```

The total number in the *$mass* section must be equal to number of atoms. Unit is the atomic mass unit (amu), *e.g.*, mass of hydrogen-1 atom is 1.00783. If the *$mass* section is not initialized, the default mass will be used.

Although the Q-CHEM output file dutifully records the progress of any *ab initio* molecular dynamics job, the most useful information is printed not to the main output file but rather to a directory called "AIMD" that is a subdirectory of the job's scratch directory. (All *ab initio* molecular dynamics jobs should therefore use the *–save* option described in Section 2.2.) The AIMD directory consists of a set of files that record, in ASCII format, one line of information at each time step. Each file contains a few comment lines (indicated by "#") that describe its contents and which we summarize in the list below.

- `Cost`: Records the number of SCF cycles, the total CPU time, and the total memory use at each dynamics step.
- `EComponents`: Records various components of the total energy (all in hartree).
- `Energy`: Records the total energy and fluctuations therein.
- `MulMoments`: If multipole moments are requested, they are printed here.
- `NucCarts`: Records the nuclear Cartesian coordinates x_1, y_1, z_1, x_2, y_2, z_2, ..., x_N, y_N, z_N at each time step, in either bohrs or Ångstroms.
- `NucForces`: Records the Cartesian forces on the nuclei at each time step (same order as the coordinates, but given in atomic units).
- `NucVeloc`: Records the Cartesian velocities of the nuclei at each time step (same order as the coordinates, but given in atomic units).

- `TandV`: Records the kinetic and potential energy, as well as fluctuations in each.
- `View.xyz`: Cartesian-formatted version of `NucCarts` for viewing trajectories in an external visualization program.

For ELMD jobs, there are other output files as well:

- `ChangeInF`: Records the matrix norm and largest magnitude element of $\Delta\mathbf{F} = \mathbf{F}(t+\delta t) - \mathbf{F}(t)$ in the basis of Cholesky-orthogonalized AOs. The files ChangeInP, ChangeInL, and ChangeInZ provide analogous information for the density matrix $\mathbf{P}$ and the Cholesky orthogonalization matrices $\mathbf{L}$ and $\mathbf{Z}$ defined in Ref. 47.
- `DeltaNorm`: Records the norm and largest magnitude element of the curvy-steps rotation angle matrix Δ defined in Ref. 47. Matrix elements of Δ are the dynamical variables representing the electronic degrees of freedom. The output file DeltaDotNorm provides the same information for the electronic velocity matrix $d\Delta/dt$.
- `ElecGradNorm`: Records the norm and largest magnitude element of the electronic gradient matrix $\mathbf{FP} - \mathbf{PF}$ in the Cholesky basis.
- `dTfict`: Records the instantaneous time derivative of the fictitious kinetic energy at each time step, in atomic units.

Ab initio molecular dynamics jobs are requested by specifying JOBTYPE = AIMD. Initial velocities must be specified either using a *$velocity* section or via the AIMD_INIT_VELOC keyword described below. In addition, the following *$rem* variables must be specified for any *ab initio* molecular dynamics job:

AIMD_METHOD

Selects an *ab initio* molecular dynamics algorithm.

TYPE:

STRING

DEFAULT:

BOMD

OPTIONS:

BOMD Born-Oppenheimer molecular dynamics.

CURVY Curvy-steps Extended Lagrangian molecular dynamics.

QCMD Meyer-Miller nonadiabatic molecular dynamics.

RECOMMENDATION:

BOMD yields exact classical molecular dynamics, provided that the energy is tolerably conserved. ELMD is an approximation to exact classical dynamics whose validity should be tested for the properties of interest. QCMD initiates Meyer-Miller nonadiabatic dynamics methods including the symmetric quasiclassical approach or the traditional Ehrenfest method.

TIME_STEP

Specifies the molecular dynamics time step, in atomic units (1 a.u. = 0.0242 fs).

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

Smaller time steps lead to better energy conservation; too large a time step may cause the job to fail entirely. Make the time step as large as possible, consistent with tolerable energy conservation.

AIMD_TIME_STEP_CONVERSION

Modifies the molecular dynamics time step to increase granularity.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n The molecular dynamics time step is TIME_STEP/n a.u.

RECOMMENDATION:

None

AIMD_STEPS

Specifies the requested number of molecular dynamics steps.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

None.

Ab initio molecular dynamics calculations can be quite expensive, and thus Q-CHEM includes several algorithms designed to accelerate such calculations. At the self-consistent field (Hartree-Fock and DFT) level, BOMD calculations can be greatly accelerated by using information from previous time steps to construct a good initial guess for the new molecular orbitals or Fock matrix, thus hastening SCF convergence. A Fock matrix extrapolation procedure,[48] based on a suggestion by Pulay and Fogarasi,[83] is available for this purpose.

The Fock matrix elements $F_{\mu\nu}$ in the atomic orbital basis are oscillatory functions of the time t, and Q-CHEM's extrapolation procedure fits these oscillations to a power series in t:

$$F_{\mu\nu}(t) = \sum_{n=0}^{N} c_n \, t^n \tag{9.60}$$

The $N+1$ extrapolation coefficients c_n are determined by a fit to a set of M Fock matrices retained from previous time steps. Fock matrix extrapolation can significantly reduce the number of SCF iterations required at each time step, but for low-order extrapolations, or if SCF_CONVERGENCE is set too small, a systematic drift in the total energy may be observed. Benchmark calculations testing the limits of energy conservation can be found in Ref. 48, and demonstrate that numerically exact classical dynamics (without energy drift) can be obtained at significantly reduced cost.

Fock matrix extrapolation is requested by specifying values for N and M, as in the form of the following two *$rem* variables:

FOCK_EXTRAP_ORDER

Specifies the polynomial order N for Fock matrix extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

N Extrapolate using an Nth-order polynomial ($N > 0$).

RECOMMENDATION:

None

FOCK_EXTRAP_POINTS

Specifies the number M of old Fock matrices that are retained for use in extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

M Save M Fock matrices for use in extrapolation $(M > N)$

RECOMMENDATION:

Higher-order extrapolations with more saved Fock matrices are faster and conserve energy better than low-order extrapolations, up to a point. In many cases, the scheme (N = 6, M = 12), in conjunction with SCF_CONVERGENCE = 6, is found to provide about a 50% savings in computational cost while still conserving energy.

When nuclear forces are computed using underlying electronic structure methods with non-optimized orbitals (such as MP2), a set of response equations must be solved.[2] While these equations are linear, their dimensionality necessitates an iterative solution,[57,77] which, in practice, looks much like the SCF equations. Extrapolation is again useful here,[100] and the syntax for Z-vector (response) extrapolation is similar to Fock extrapolation.

Z_EXTRAP_ORDER

Specifies the polynomial order N for Z-vector extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Z-vector extrapolation.

OPTIONS:

N Extrapolate using an Nth-order polynomial ($N > 0$).

RECOMMENDATION:

None

Z_EXTRAP_POINTS

Specifies the number M of old Z-vectors that are retained for use in extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform response equation extrapolation.

OPTIONS:

M Save M previous Z-vectors for use in extrapolation $(M > N)$

RECOMMENDATION:

Using the default Z-vector convergence settings, a $(M, N) = (4, 2)$ extrapolation was shown to provide the greatest speedup. At this setting, a 2–3-fold reduction in iterations was demonstrated.

Assuming decent conservation, a BOMD calculation represents exact classical dynamics on the Born-Oppenheimer potential energy surface. In contrast, so-called extended Lagrangian molecular dynamics (ELMD) methods make an approximation to exact classical dynamics in order to expedite the calculations. ELMD methods—of which the most famous is Car–Parrinello molecular dynamics—introduce a fictitious dynamics for the electronic (orbital) degrees of freedom, which are then *propagated* alongside the nuclear degrees of freedom, rather than *optimized* at each time step as they are in a BOMD calculation. The fictitious electronic dynamics is controlled by a fictitious mass parameter μ, and the value of μ controls both the accuracy and the efficiency of the method. In the limit of small μ the nuclei and the orbitals propagate adiabatically, and ELMD mimics true classical dynamics. Larger values of μ slow down the electronic dynamics, allowing for larger time steps (and more computationally efficient dynamics), at the expense of an ever-greater approximation to true classical dynamics.

Q-CHEM's ELMD algorithm is based upon propagating the density matrix, expressed in a basis of atom-centered Gaussian orbitals, along shortest-distance paths (geodesics) of the manifold of allowed density matrices $\mathbf{P}$. Idempotency of $\mathbf{P}$ is maintained at every time step, by construction, and thus our algorithm requires neither density matrix purification, nor iterative solution for Lagrange multipliers (to enforce orthogonality of the molecular orbitals). We call this procedure "curvy steps" ELMD,[47] and in a sense it is a time-dependent implementation of the GDM algorithm (Section 4.5) for converging SCF single-point calculations.

The extent to which ELMD constitutes a significant approximation to BOMD continues to be debated. When assessing the accuracy of ELMD, the primary consideration is whether there exists a separation of time scales between nuclear oscillations, whose time scale τ_{nuc} is set by the period of the fastest vibrational frequency, and electronic oscillations, whose time scale τ_{elec} may be estimated according to[47]

$$\tau_{\mathrm{elec}} \geq \left(\frac{\mu}{\varepsilon_{\mathrm{LUMO}} - \varepsilon_{\mathrm{HOMO}}} \right)^{1/2} \tag{9.61}$$

A conservative estimate, suggested in Ref. 47, is that essentially exact classical dynamics is attained when $\tau_{\mathrm{nuc}} > 10\ \tau_{\mathrm{elec}}$. In practice, we recommend careful benchmarking to insure that ELMD faithfully reproduces the BOMD observables of interest.

Due to the existence of a fast time scale τ_{elec}, ELMD requires smaller time steps than BOMD. When BOMD is combined with Fock matrix extrapolation to accelerate convergence, it is no longer clear that ELMD methods are substantially more efficient, at least in Gaussian basis sets.[48,83]

The following *$rem* variables are required for ELMD jobs:

AIMD_FICT_MASS

Specifies the value of the fictitious electronic mass μ, in atomic units, where μ has dimensions of (energy)$\times$(time)^2.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified

RECOMMENDATION:

Values in the range of 50–200 a.u. have been employed in test calculations; consult Ref. 47 for examples and discussion.

AIMD_PRINT

Specifies the verbosity of the output in AIMD.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

User-specified

RECOMMENDATION:

Higher values result in more verbose output.

9.9.3 Additional Job Control and Examples

AIMD_INIT_VELOC

Specifies the method for selecting initial nuclear velocities.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

THERMAL	Random sampling of nuclear velocities from a Maxwell-Boltzmann distribution. The user must specify the temperature in Kelvin via the *$rem* variable AIMD_TEMP.
ZPE	Choose velocities in order to put zero-point vibrational energy into each normal mode, with random signs. This option requires that a frequency job to be run beforehand.
QUASICLASSICAL	Puts vibrational energy into each normal mode. In contrast to the ZPE option, here the vibrational energies are sampled from a Boltzmann distribution at the desired simulation temperature. This also triggers several other options, as described below.
OLD	Use the same initial velocities as the immediately preceding AIMD job.
RESTART	Use the final velocities from a previous AIMD job, reading them from disk.

RECOMMENDATION:

This variable need only be specified in the event that velocities are not specified explicitly in a *$velocity* section.

AIMD_INIT_VELOC_NANO_RANDOM

Uses a more precise random seed for generating random initial velocities.

TYPE:

LOGICAL

DEFAULT:

TRUE Use a more precise random seed.

OPTIONS:

FALSE Use a less precise random seed.

RECOMMENDATION:

Leave this set to TRUE unless necessary.

This option determines the source of the random seed used for sampling random initial velocities when AIMD_INIT_VELOC requires such. Setting the option to FALSE will have the seed based on the system time in seconds, meaning that two otherwise identical simulations starting in the same second will produce identical initial velocities. With the option set to TRUE, such collisions are virtually impossible.

The option is kept for legacy purposes. There should rarely ever be a need to set it to FALSE.

AIMD_MOMENTS

Requests that multipole moments be output at each time step.

TYPE:

INTEGER

DEFAULT:

0 Do not output multipole moments.

OPTIONS:

n Output the first n multipole moments.

RECOMMENDATION:

None

AIMD_TEMP

Specifies a temperature (in Kelvin) for Maxwell-Boltzmann velocity sampling.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified number of Kelvin.

RECOMMENDATION:

This variable is only useful in conjunction with AIMD_INIT_VELOC = THERMAL. Note that the simulations are run at constant energy, rather than constant temperature, so the mean nuclear kinetic energy will fluctuate in the course of the simulation.

DEUTERATE

Requests that all hydrogen atoms be replaces with deuterium.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not replace hydrogens.

OPTIONS:

TRUE Replace hydrogens with deuterium.

RECOMMENDATION:

Replacing hydrogen atoms reduces the fastest vibrational frequencies by a factor of 1.4, which allow for a larger fictitious mass and time step in ELMD calculations. There is no reason to replace hydrogens in BOMD calculations.

Example 9.33 Simulating thermal fluctuations of the water dimer at 298 K.

```
$molecule
   0  1
   O    1.386977   0.011218   0.109098
   H    1.748442   0.720970  -0.431026
   H    1.741280  -0.793653  -0.281811
   O   -1.511955  -0.009629  -0.120521
   H   -0.558095   0.008225   0.047352
   H   -1.910308   0.077777   0.749067
$end

$rem
   JOBTYPE              aimd
   AIMD_METHOD          bomd
   METHOD               b3lyp
   BASIS                6-31g*
   TIME_STEP            20        (20 a.u. = 0.48 fs)
   AIMD_STEPS           1000
   AIMD_INIT_VELOC      thermal
   AIMD_TEMP            298
   FOCK_EXTRAP_ORDER    6         request Fock matrix extrapolation
   FOCK_EXTRAP_POINTS   12
$end
```

Example 9.34 Propagating $F^-(H_2O)_4$ on its first excited-state potential energy surface, calculated at the CIS level.

```
$comment
  Note, only a few time steps are taken, a more appropriate
  number would be:
   AIMD_STEPS       827   500 fs
$end

$molecule
   -1   1
   O  -1.969902  -1.946636   0.714962
   H  -2.155172  -1.153127   1.216596
   H  -1.018352  -1.980061   0.682456
   O  -1.974264   0.720358   1.942703
   H  -2.153919   1.222737   1.148346
   H  -1.023012   0.684200   1.980531
   O  -1.962151   1.947857  -0.723321
   H  -2.143937   1.154349  -1.226245
   H  -1.010860   1.980414  -0.682958
   O  -1.957618  -0.718815  -1.950659
   H  -2.145835  -1.221322  -1.158379
   H  -1.005985  -0.682951  -1.978284
   F   1.431477   0.000499   0.010220
$end

$rem
   JOBTYPE           aimd
   AIMD_METHOD       bomd
   METHOD            hf
   BASIS             6-31+G*
   ECP               SRLC
   PURECART          1111
   CIS_N_ROOTS       3
   CIS_TRIPLETS      false
   CIS_STATE_DERIV   1         propagate on first excited state
   AIMD_INIT_VELOC   thermal
   AIMD_TEMP         150
   TIME_STEP         25
   AIMD_STEPS        10
$end
```

Example 9.35 Simulating vibrations of the NaCl molecule using ELMD.

```
$molecule
   0  1
   Na    0.000000    0.000000  -1.742298
   Cl    0.000000    0.000000   0.761479
$end

$rem
   JOBTYPE     freq
   METHOD      b3lyp
   ECP         fit-sbkjc
   BASIS       sbkjc
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE            aimd
   METHOD             b3lyp
   ECP                fit-sbkjc
   BASIS              sbkjc
   TIME_STEP          14
   AIMD_STEPS         500
   AIMD_METHOD        curvy
   AIMD_FICT_MASS     360
   AIMD_INIT_VELOC    zpe
$end
```

Q-CHEM has the ability to do AIMD with frozen bonds by using RATTLE algorithm.[3] It can be requested by setting the rem variable AIMD_INTEGRATION to RATTLE. Constraints are imposed via the *$rattle* input section, whose format is shown below.

```
$rattle
   bond  atom1  atom2  value
   ....  .....  .....  .....
$end
```

Note: The bond length values should be in Ångstrom units.

The convergence threshold and the number of maximum iterations for RATTLE steps are controlled by the following

$rem variables: RATTLE_THRESH (with a default value of 6) and RATTLE_MAXIT (with a default value of 100).

Example 9.36 Simulating water molecule using RATTLE algorithm.

```
$molecule
   0 1
   O
   H 1 0.95
   H 1 0.96 2 104.5
$end

$rem
   JOBTYPE             aimd
   METHOD              b3lyp
   BASIS               6-31G*
   TIME_STEP           15
   AIMD_STEPS          10
   AIMD_INIT_VELOC     thermal Boltzmann distribution
   AIMD_TEMP           300     (in Kelvin)
   AIMD_PRINT          1
   AIMD_INTEGRATION    RATTLE
   DEBUG_RANDOM_SEED   true
$end

$rattle
   bond 1 2 0.950
   bond 1 3 0.950
   bond 2 3 1.565
$end
```

Example 9.37 Excited-state AIMD with exciton analysis at each step.

```
$comment
AIMD trajectory with wave-function analysis performed at each step.
This is invoved by  STATE_ANALYSIS=true and AIMD_PRINT=2
$end

$molecule
0 1
  O   -2.1904701    0.5202006    0.0237374
  H   -1.2011437    0.4799213    0.0090967
  H   -2.4798356   -0.3698583   -0.2997841
$end

$rem
   JOBTYPE              aimd
   METHOD               hf
   BASIS                6-31+G*
   CIS_N_ROOTS          3
   CIS_TRIPLETS         false
   AIMD_STEPS           5
   TIME_STEP            40
   AIMD_INIT_VELOC      thermal
   AIMD_TEMP            500
   AIMD_INTEGRATION     vverlet
   CIS_STATE_DERIV      1
   STATE_ANALYSIS       true
   AIMD_PRINT           2
$end
```

9.9.4 Thermostats

Implicit in the discussion above was an assumption of conservation of energy, which implies dynamics run in the *microcanonical* (NVE) ensemble. Alternatively, the AIMD code in Q-CHEM can sample the *canonical* (NVT) ensemble with the aid of thermostats. These mimic the thermal effects of a surrounding temperature bath, and the time average of a trajectory (or trajectories) then affords thermodynamic averages at a chosen temperature. This option is appropriate in particular when multiple minima are thermally accessible. All sampled information is once again saved in the `AIMD/` subdirectory of the `$QCSCRATCH` directory for the job. Thermodynamic averages and error analysis may be performed externally, using these data. Two commonly used thermostat options, both of which yield proper canonical distributions of the classical molecular motion, are implemented in Q-CHEM and are described in more detail below. Constant-pressure barostats (for NPT simulations) are not yet implemented.

As with any canonical sampling, the trajectory evolves at the mercy of barrier heights. Short trajectories will sample only within the local minimum of the initial conditions, which may be desired for sampling the properties of a given isomer, for example. Due to the energy fluctuations induced by the thermostat, the trajectory is neither guaranteed to stay within this potential energy well nor guaranteed to overcome barriers to neighboring minima, except in the infinite-sampling limit for the latter case, which is likely never reached in practice. Importantly, the user should note that the introduction of a thermostat destroys the validity of any real-time trajectory information; thermostatted trajectories should *not* be used to assess real-time dynamical observables, but only to compute thermodynamic averages.

9.9.4.1 Langevin Thermostat

A stochastic, white-noise Langevin thermostat (AIMD_THERMOSTAT = LANGEVIN) combines random "kicks" to the nuclear momenta with a dissipative, friction term. The balance of these two contributions mimics the exchange of energy with a surrounding heat bath. The resulting trajectory, in the long-time sampling limit, generates the correct canonical distribution. The implementation in Q-CHEM follows the velocity Verlet formulation of Bussi and Parrinello,[19] which remains a valid propagator for all time steps and thermostat parameters. The thermostat is coupled to each degree of freedom in the simulated system. The MD integration time step (TIME_STEP) should be chosen in the same manner as in an NVE trajectory. The only user-controllable parameter for this thermostat, therefore, is the timescale over which the implied bath influences the trajectory. The AIMD_LANGEVIN_TIMESCALE keyword determines this parameter, in units of femtoseconds. For users who are more accustomed to thinking in terms of friction strength, this parameter is proportional to the inverse friction. A small value of the timescale parameter yields a "tight" thermostat, which strongly maintains the system at the chosen temperature but does not typically allow for rapid configurational flexibility. (Qualitatively, one may think of such simulations as sampling in molasses. This analogy, however, only applies to the thermodynamic sampling properties and does not suggest any electronic role of the solvent!) These small values are generally more appropriate for small systems, where the few degrees of freedom do not rapidly exchange energy and behave may behave in a non-ergodic fashion. Alternatively, large values of the time-scale parameter allow for more flexible configurational sampling, with the tradeoff of more (short-term) deviation from the desired average temperature. These larger values are more appropriate for larger systems since the inherent, microcanonical exchange of energy within the large number of degrees of freedom already tends toward canonical properties. (Think of this regime as sampling in a light, organic solvent.) Importantly, thermodynamic averages in the infinite-sampling limit are completely independent of this time-scale parameter. Instead, the time scale merely controls the efficiency with which the ensemble is explored. If maximum efficiency is desired, the user may externally compute lifetimes from the time correlation function of the desired observable and minimize the lifetime as a function of this timescale parameter. At the end of the trajectory, the average computed temperature is compared to the requested target temperature for

validation purposes.

Example 9.38 Canonical (NVT) sampling using AIMD with the Langevin thermostat

```
$comment
   Short example of using the Langevin thermostat
   for canonical (NVT) sampling
$end

$molecule
   0 1
   H
   O  1  1.0
   H  2  1.0  1  104.5
$end

$rem
   JOBTYPE                  aimd
   EXCHANGE                 hf
   BASIS                    sto-3g
   AIMD_TIME_STEP           20      !in au
   AIMD_STEPS               100
   AIMD_THERMOSTAT          langevin
   AIMD_INIT_VELOC          thermal
   AIMD_TEMP                298     !in K - initial conditions AND thermostat
   AIMD_LANGEVIN_TIMESCALE  100     !in fs
$end
```

9.9.4.2 Nosé-Hoover Thermostat

An alternative thermostat approach is also available, namely, the Nosé-Hoover thermostat[65] (also known as a Nosé-Hoover "chain"), which mimics the role of a surrounding thermal bath by performing a microcanonical (NVE) trajectory in an extended phase space. By allowing energy to be exchanged with a chain of fictitious particles that are coupled to the target system, NVT sampling is properly obtained for those degrees of freedom that represent the real system. (Only the target system properties are saved in `$QCSCRATCH/AIMD` for subsequent analysis and visualization, not the fictitious Nosé-Hoover degrees of freedom.) The implementation in Q-CHEM follows that of Martyna,[65] which augments the original extended-Lagrangian approach of Nosé[71,72] and Hoover,[51] using a chain of auxiliary degrees of freedom to restore ergodicity in stiff systems and thus afford the correct NVT ensemble. Unlike the Langevin thermostat, the collection of system and auxiliary chain particles can be propagated in a time-reversible fashion with no need for stochastic perturbations.

Rather than directly setting the masses and force constants of the auxiliary chain particles, the Q-CHEM implementation focuses instead, on the time scale of the thermostat, as was the case for the Langevin thermostat described above. The time-scale parameter is controlled by the keyword NOSE_HOOVER_TIMESCALE, given in units of femtoseconds. The only other user-controllable parameter for this function is the length of the Nosé-Hoover chain, which is typically chosen to be 3–6 fictitious particles. Importantly, the version in Q-CHEM is currently implemented as a *single* chain that is coupled to the system, as a whole. Comprehensive thermostatting in which every single degree of freedom is coupled to its own thermostat, which is sometimes used for particularly stiff systems, is not implemented and for such cases the Langevin thermostat is recommended instead. For large and/or fluxional systems, the single-chain

Nosé-Hoover approach is appropriate.

Example 9.39 Canonical (NVT) sampling using AIMD with the Nosé-Hoover chain thermostat

```
$molecule
   0 1
   H
   O 1 1.0
   H 2 1.0 1 104.5
$end

$rem
   JOBTYPE                 aimd
   EXCHANGE                hf
   BASIS                   sto-3g
   AIMD_TIME_STEP          20      !in au
   AIMD_STEPS              100
   AIMD_THERMOSTAT         nose_hoover
   AIMD_INIT_VELOC         thermal
   AIMD_TEMP               298     !in K - initial conditions AND thermostat
   NOSE_HOOVER_LENGTH      3       !chain length
   NOSE_HOOVER_TIMESCALE   100     !in fs
$end
```

AIMD_THERMOSTAT

Applies thermostatting to AIMD trajectories.

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

LANGEVIN	Stochastic, white-noise Langevin thermostat
NOSE_HOOVER	Time-reversible, Nosé-Hoovery chain thermostat

RECOMMENDATION:

Use either thermostat for sampling the canonical (NVT) ensemble.

AIMD_LANGEVIN_TIMESCALE

Sets the timescale (strength) of the Langevin thermostat

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

n	Thermostat timescale,asn n fs

RECOMMENDATION:

Smaller values (roughly 100) equate to tighter thermostats but may inhibit rapid sampling. Larger values (≥ 1000) allow for more rapid sampling but may take longer to reach thermal equilibrium.

NOSE_HOOVER_LENGTH

Sets the chain length for the Nosé-Hoover thermostat

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

n Chain length of n auxiliary variables

RECOMMENDATION:

Typically 3-6

NOSE_HOOVER_TIMESCALE

Sets the timescale (strength) of the Nosé-Hoover thermostat

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

n Thermostat timescale, as n fs

RECOMMENDATION:

Smaller values (roughly 100) equate to tighter thermostats but may inhibit rapid sampling. Larger values (≥ 1000) allow for more rapid sampling but may take longer to reach thermal equilibrium.

9.9.5 Vibrational Spectra

The inherent nuclear motion of molecules is experimentally observed by the molecules' response to impinging radiation. This response is typically calculated within the mechanical and electrical harmonic approximations (second derivative calculations) at critical-point structures. Spectra, including anharmonic effects, can also be obtained from dynamics simulations. These spectra are generated from dynamical response functions, which involve the Fourier transform of auto-correlation functions. Q-CHEM can provide both the vibrational spectral density from the velocity auto-correlation function

$$D(\omega) \propto \int_{-\infty}^{\infty} dt\, e^{-i\omega t} \langle \vec{v}(0) \cdot \vec{v}(t) \rangle \tag{9.62}$$

and infrared absorption intensity from the dipole auto-correlation function

$$I(\omega) \propto \frac{\omega}{2\pi} \int_{-\infty}^{\infty} dt\, e^{-i\omega t} \langle \vec{\mu}(0) \cdot \vec{\mu}(t) \rangle \tag{9.63}$$

These two features are activated by the AIMD_NUCL_VACF_POINTS and AIMD_NUCL_DACF_POINTS keywords, respectively, where values indicate the number of data points to include in the correlation function. Furthermore, the AIMD_NUCL_SAMPLE_RATE keyword controls the frequency at which these properties are sampled (entered as number of time steps). These spectra—generated at constant energy—should be averaged over a suitable distribution of initial conditions. The averaging indicated in the expressions above, for example, should be performed over a Boltzmann distribution of initial conditions.

Note that dipole auto-correlation functions can be contaminated with rotational or translational information, and while the initial conditions in Q-CHEM remove translation and rotation, numerical noise in the forces and propagation can lead to translation and rotation over time. A translation-rotation correction is activated by the PROJ_TRANSROT keyword.

AIMD_NUCL_VACF_POINTS

Number of time points to use in the velocity auto-correlation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute velocity auto-correlation function.
$1 \leq n \leq$ AIMD_STEPS	Compute velocity auto-correlation function for last n time steps of the trajectory.

RECOMMENDATION:

If the VACF is desired, set equal to the value of AIMD_STEPS.

AIMD_NUCL_DACF_POINTS

Number of time points to use in the dipole auto-correlation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute dipole auto-correlation function.
$1 \leq n \leq$ AIMD_STEPS	Compute dipole auto-correlation function for last n timesteps of the trajectory.

RECOMMENDATION:

If the DACF is desired, set equal to the value of AIMD_STEPS.

AIMD_NUCL_SAMPLE_RATE

The rate at which sampling is performed for the velocity and/or dipole auto-correlation function(s). Specified as a multiple of steps; *i.e.*, sampling every step is 1.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

$1 \leq n \leq$ AIMD_STEPS	Update the velocity/dipole auto-correlation function every n steps.

RECOMMENDATION:

Since the velocity and dipole moment are routinely calculated for *ab initio* methods, this variable should almost always be set to 1 when the VACF or DACF is desired.

PROJ_TRANSROT

Removes translational and rotational drift during AIMD trajectories.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not apply translation/rotation corrections.

TRUE Apply translation/rotation corrections.

RECOMMENDATION:

When computing spectra (see AIMD_NUCL_DACF_POINTS, for example), this option can be used to remove artificial, contaminating peaks stemming from translational and/or rotational motion. Recommend setting to TRUE for all dynamics-based spectral simulations.

9.9.6 Quasi-Classical Molecular Dynamics

So-called "quasi-classical" trajectories[56,79,80] (QCT) put vibrational energy into each mode in the initial velocity setup step, which can improve on the results of purely classical simulations, for example in the calculation of photoelectron[59] or infrared spectra.[85] Improvements include better agreement of spectral linewidths with experiment at lower temperatures and better agreement of vibrational frequencies with anharmonic calculations.

The improvements at low temperatures can be understood by recalling that even at low temperature there is nuclear motion due to zero-point motion. This is included in the quasi-classical initial velocities, thus leading to finite peak widths even at low temperatures. In contrast to that the classical simulations yield zero peak width in the low temperature limit, because the thermal kinetic energy goes to zero as temperature decreases. Likewise, even at room temperature the quantum vibrational energy for high-frequency modes is often significantly larger than the classical kinetic energy. QCT-MD therefore typically samples regions of the potential energy surface that are higher in energy and thus more anharmonic than the low-energy regions accessible to classical simulations. These two effects can lead to improved peak widths as well as a more realistic sampling of the anharmonic parts of the potential energy surface. However, the QCT-MD method also has important limitations which are described below and that the user has to monitor for carefully.

In our QCT-MD implementation the initial vibrational quantum numbers are generated as random numbers sampled from a vibrational Boltzmann distribution at the desired simulation temperature. In order to enable reproducibility of the results, each trajectory (and thus its set of vibrational quantum numbers) is denoted by a unique number using the AIMD_QCT_WHICH_TRAJECTORY variable. In order to loop over different initial conditions, run trajectories with different choices for AIMD_QCT_WHICH_TRAJECTORY. It is also possible to assign initial velocities corresponding to an average over a certain number of trajectories by choosing a negative value. Further technical details of our QCT-MD implementation are described in detail in Appendix A of Ref. 59.

AIMD_QCT_WHICH_TRAJECTORY

Picks a set of vibrational quantum numbers from a random distribution.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Picks the nth set of random initial velocities.

$-n$ Uses an average over n random initial velocities.

RECOMMENDATION:

Pick a positive number if you want the initial velocities to correspond to a particular set of vibrational occupation numbers and choose a different number for each of your trajectories. If initial velocities are desired that corresponds to an average over n trajectories, pick a negative number.

Below is a simple example input for running a QCT-MD simulation of the vibrational spectrum of water. Most input variables are the same as for classical MD as described above. The use of quasi-classical initial conditions is triggered

by setting the AIMD_INIT_VELOC variable to QUASICLASSICAL.

Example 9.40 Simulating the IR spectrum of water using QCT-MD.

```
$molecule
   0  1
   O     0.000000    0.000000    0.520401
   H    -1.475015    0.000000   -0.557186
   H     1.475015    0.000000   -0.557186
$end

$rem
   JOBTYPE                     freq
   INPUT_BOHR                  true
   METHOD                      hf
   BASIS                       3-21g
$end

@@@

$molecule
  read
$end

$rem
   JOBTYPE                     aimd
   INPUT_BOHR                  true
   METHOD                      hf
   BASIS                       3-21g
   SCF_CONVERGENCE             6
   TIME_STEP                   20    !  (in atomic units)
   AIMD_STEPS                  1250  !  600 fs total simulation time
   AIMD_TEMP                   12
   AIMD_PRINT                  2
   FOCK_EXTRAP_ORDER           6     ! Use a 6th-order extrapolation
   FOCK_EXTRAP_POINTS          12    ! of the previous 12 Fock matrices
   AIMD_MOMENTS                1
   AIMD_NUCL_SAMPLE_RATE       5
   AIMD_NUCL_VACF_POINTS       1000
   AIMD_INIT_VELOC             quasiclassical
   AIMD_QCT_WHICH_TRAJECTORY   1     ! Loop over several values to get
                                     ! the correct Boltzmann distribution.
$end
```

Other types of spectra can be calculated by calculating spectral properties along the trajectories. For example, we observed that photoelectron spectra can be approximated quite well by calculating vertical detachment energies (VDEs) along the trajectories and generating the spectrum as a histogram of the VDEs.[59] We have included several simple scripts in the `$QC/aimdman/tools` subdirectory that we hope the user will find helpful and that may serve as the basis for developing more sophisticated tools. For example, we include scripts that allow to perform calculations along a trajectory (`md_calculate_along_trajectory`) or to calculate vertical detachment energies along a trajectory (`calculate_rel_energies`).

Another application of the QCT code is to generate random geometries sampled from the vibrational wave function via a Monte Carlo algorithm. This is triggered by setting both the AIMD_QCT_INITPOS and AIMD_QCT_WHICH_TRAJECTORY variables to negative numbers, say $-m$ and $-n$, and setting AIMD_STEPS to zero. This will generate m random geometries sampled from the vibrational wave function corresponding to an average over n trajectories at the user-specified simulation temperature.

AIMD_QCT_INITPOS

Chooses the initial geometry in a QCT-MD simulation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Use the equilibrium geometry.
n	Picks a random geometry according to the harmonic vibrational wave function.
$-n$	Generates n random geometries sampled from the harmonic vibrational wave function.

RECOMMENDATION:

None.

For systems that are described well within the harmonic oscillator model and for properties that rely mainly on the ground-state dynamics, this simple MC approach may yield qualitatively correct spectra. In fact, one may argue that it is preferable over QCT-MD for describing vibrational effects at very low temperatures, since the geometries are sampled from a true quantum distribution (as opposed to classical and quasi-classical MD). We have included another script in

the *$QC/aimdman/tools* directory to help with the calculation of vibrationally averaged properties (`monte_geom`).

Example 9.41 MC sampling of the vibrational wave function for HCl. 1000 random geometries for HCl are generated based on the harmonic vibrational wave function at 1 Kelvin. The wave function is averaged over 1000 sets of random vibrational quantum numbers (*i.e.*, the ground state in this case due to the low temperature).

```
$molecule
   0 1
   H    0.000000    0.000000   -1.216166
   Cl   0.000000    0.000000    0.071539
$end

$rem
   JOBTYPE                    freq
   METHOD                     b3lyp
   BASIS                      6-311++G**
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE                    aimd
   METHOD                     B3LYP
   BASIS                      6-311++G**
   SCF_CONVERGENCE            1
   MAX_SCF_CYCLES             0
   TIME_STEP                  20     (in atomic units)
   AIMD_STEPS                 0
   AIMD_INIT_VELOC            quasiclassical
   AIMD_QCT_VIBSEED           1
   AIMD_QCT_VELSEED           2
   AIMD_TEMP                  1     (in Kelvin)
   AIMD_QCT_WHICH_TRAJECTORY  -1000 ! set to the desired trajectory number
   AIMD_QCT_INITPOS           -1000
$end
```

It is also possible make some modes inactive, *i.e.*, to put vibrational energy into a subset of modes (all other are set to zero). The list of active modes can be specified using the *$qct_active_modes* section. Furthermore, the vibrational quantum numbers for each mode can be specified explicitly using the *$qct_vib_distribution* input section. It is also

possible to set the phases using *$qct_vib_phase* (allowed values are 1 and −1). Below is a simple sample input:

Example 9.9.42 User control over the QCT variables. Makes the 1st vibrational mode QCT-active; all other ones receive zero kinetic energy. We choose the vibrational ground state and a positive phase for the velocity.

```
$qct_active_modes
  1
$end

$qct_vib_distribution
  0
$end

$qct_vib_phase
  1
$end
...
```

Finally we turn to a brief description of the limitations of QCT-MD. Perhaps the most severe limitation stems from the so-called "kinetic energy spilling" problem,[28] which means that there can be an artificial transfer of kinetic energy between modes. This can happen because the initial velocities are chosen according to quantum energy levels, which are usually much higher than those of the corresponding classical systems. Furthermore, the classical equations of motion also allow for the transfer of non-integer multiples of the zero-point energy between the modes, which leads to different selection rules for the transfer of kinetic energy. Typically, energy spills from high-energy into low-energy modes, leading to spurious "hot" dynamics. A second problem is that QCT-MD is actually based on classical Newtonian dynamics, which means that the probability distribution at low temperatures can be qualitatively wrong compared to the true quantum distribution.[59]

Q-CHEM implements a routine to monitor the kinetic energy within each normal mode along the trajectory and that is automatically switched on for quasi-classical simulations. It is thus possible to monitor for trajectories in which the kinetic energy in one or more modes becomes significantly larger than the initial energy. Such trajectories should be discarded. (Alternatively, see Ref. 28 for a different approach to the zero-point leakage problem.) Furthermore, this monitoring routine prints the squares of the (harmonic) vibrational wave function along the trajectory. This makes it possible to weight low-temperature results with the harmonic quantum distribution to alleviate the failure of classical dynamics for low temperatures.

9.9.7 Fewest-Switches Surface Hopping

As discussed in Section 9.8, optimization of minimum-energy crossing points (MECPs) along conical seams, followed by optimization of minimum-energy pathways that connect these MECPs to other points of interest on ground- and excited-state potential energy surfaces, affords an appealing one-dimensional picture of photochemical reactivity that is analogous to the "reactant → transition state → product" picture of ground-state chemistry. Just as the ground-state reaction is not obligated to proceed *exactly* through the transition-state geometry, however, an excited-state reaction need not proceed precisely through the MECP and the particulars of nuclear kinetic energy can lead to deviations. This is arguably more of an issue for excited-state reactions, where the existence of multiple conical intersections can easily lead to multiple potential reaction mechanisms. AIMD potentially offers a way to sample over the available mechanisms in order to deduce which ones are important in an automated way, but must be extended in the photochemical case to reactions that involve more than one Born-Oppenheimer potential energy surface.

The most widely-used trajectory-based method for nonadiabatic simulations is Tully's "fewest-switches" surface-hopping (FSSH) algorithm.[104] In this approach, classical trajectories are propagated on a single potential energy surface, but can undergo "hops" to a different potential surface in regions of near-degeneracy between surfaces. The

probability of these stochastic hops is governed by the magnitude of the nonadiabatic coupling [Eq. (9.49)]. Considering the ensemble average of a swarm of trajectories then provides information about, *e.g.*, branching ratios for photochemical reactions.

The FSSH algorithm, based on the AIMD code, is available in Q-CHEM for any electronic structure method where analytic derivative couplings are available, which at present means CIS, TDDFT, and their spin-flip analogues (see Section 9.8.1). The nuclear dynamics component of the simulation is specified just as in an AIMD calculation. Artificial decoherence can be added to the calculation at additional cost according to the augmented FSSH (AFSSH) method,[60,101,102] which enforces stochastic wave function collapse at a rate proportional to the difference in forces between the trajectory on the active surface and position moments propagated the other surfaces. At every time step, the component of the wave function on each active surface is printed to the output file. These amplitudes, as well as the position and momentum moments (if AFSSH is requested), is also printed to a text file called `SurfaceHopper` located in the `$QC/AIMD` sub-directory of the job's scratch directory.

In order to request a FSSH calculation, only a few additional *$rem* variables must be added to those necessary for an excited-state AIMD simulation. At present, FSSH calculations can only be performed using Born-Oppenheimer molecular dynamics (BOMD) method. Furthermore, the optimized velocity Verlet (OVV) integration method is not supported for FSSH calculations.

FSSH_LOWESTSURFACE

Specifies the lowest-energy state considered in a surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n Only states n and above are considered in a FSSH calculation.

RECOMMENDATION:

None

FSSH_NSURFACES

Specifies the number of states considered in a surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n n states are considered in the surface hopping calculation.

RECOMMENDATION:

Any states which may come close in energy to the active surface should be included in the surface hopping calculation.

FSSH_INITIALSURFACE

Specifies the initial state in a surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n An integer between FSSH_LOWESTSURFACE and FSSH_LOWESTSURFACE + FSSH_NSURFACES -1.

RECOMMENDATION:

None

AFSSH

Adds decoherence approximation to surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Traditional surface hopping, no decoherence.

1 Use augmented fewest-switches surface hopping (AFSSH).

RECOMMENDATION:

AFSSH will increase the cost of the calculation, but may improve accuracy for some systems. See Refs. 60,101,102 for more detail.

AIMD_SHORT_TIME_STEP

Specifies a shorter electronic time step for FSSH calculations.

TYPE:

INTEGER

DEFAULT:

TIME_STEP

OPTIONS:

n Specify an electronic time step duration of n/AIMD_TIME_STEP_CONVERSION a.u. If n is less than the nuclear time step variable TIME_STEP, the electronic wave function will be integrated multiple times per nuclear time step, using a linear interpolation of nuclear quantities such as the energy gradient and derivative coupling. Note that n must divide TIME_STEP evenly.

RECOMMENDATION:

Make AIMD_SHORT_TIME_STEP as large as possible while keeping the trace of the density matrix close to unity during long simulations. Note that while specifying an appropriate duration for the electronic time step is essential for maintaining accurate wave function time evolution, the electronic-only time steps employ linear interpolation to estimate important quantities. Consequently, a short electronic time step is not a substitute for a reasonable nuclear time step.

FSSH_CONTINUE

Restart a FSSH calculation from a previous run, using the file 396.0. When this is enabled, the initial conditions of the surface hopping calculation will be set, including the correct wave function amplitudes, initial surface, and position/momentum moments (if AFSSH) from the final step of some prior calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Start fresh calculation.

1 Restart from previous run.

RECOMMENDATION:

None

Example 9.43 FSSH simulation. Note that analytic derivative couplings must be requested via CALC_NAC, but it is unnecessary to include a *$derivative_coupling* section. The same is true for CIS_STATE_DERIV, which will be set to the initial active surface automatically. Finally, one must be careful to choose a small enough time step for systems that have energetic access to a region of large derivative coupling, hence the choice for AIMD_TIME_STEP_CONVERSION and TIME_STEP.

```
$molecule
   0 1
   C   -1.620294    0.348677   -0.008838
   C   -0.399206   -0.437493   -0.012535
   C   -0.105193   -1.296810   -1.081340
   H   -0.789110   -1.374693   -1.905080
   C    1.069016   -2.045054   -1.072304
   H    1.292495   -2.701157   -1.889686
   C    1.956240   -1.940324    0.002842
   H    2.859680   -2.517019    0.008420
   C    1.666259   -1.084065    1.071007
   H    2.348104   -1.005765    1.894140
   C    0.495542   -0.335701    1.065497
   H    0.253287    0.325843    1.871866
   O   -1.931045    1.124872    0.911738
   H   -2.269528    0.227813   -0.865645
$end

$rem
   JOBTYPE                   aimd
   EXCHANGE                  hf
   BASIS                     3-21g
   CIS_N_ROOTS               3
   CIS_SINGLETS              false
   CIS_TRIPLETS              true
   PROJ_TRANSROT             true
   FSSH_LOWESTSURFACE        1
   FSSH_NSURFACES            3   ! hop between T1 and T2
   FSSH_INITIALSURFACE       1   ! start on T1
   AFSSH                     0   ! no decoherence
   CALC_NAC                  true
   AIMD_STEPS                50  ! Typically more would be used
   TIME_STEP                 14
   AIMD_SHORT_TIME_STEP      2
   AIMD_TIME_STEP_CONVERSION 1   ! Do not alter time_step duration
   AIMD_PRINT                1
   AIMD_INIT_VELOC           thermal
   AIMD_TEMP                 300 # K
   AIMD_INTEGRATION          vverlet
   FOCK_EXTRAP_ORDER         6
   FOCK_EXTRAP_POINTS        12
   INTEGRAL_SYMMETRY         off
   POINT_GROUP_SYMMETRY      false
$end
```

9.9.8 Meyer-Miller Nonadiabatic Dynamics

As an alternative to Tully's FSSH algorithm (see Section 9.9.7), Q-CHEM can also perform trajectory-based electronically nonadiabatic simulations with the vibronic dynamics generated using the classical Meyer-Miller (MM) Hamiltonian.[67] Additionally, the dynamics can be subject to a symmetrical quasi-classical (SQC) quantization procedure,[23] effectively defining the electronic states.[24] Details of this approach as it pertains to the Q-CHEM implementation can be found in Ref. 103. In brief, the Meyer-Miller Hamiltonian maps the electronic degrees of freedom (DOF) in an electronically nonadiabatic process to a coupled set of classical harmonic oscillators, one for each electronic state. Movement of classical vibrational excitation amongst the oscillators then determines the active electronic state, or more precisely, the combination of electronic states which define an effective multi-state potential energy surface (PES) on which the nuclear degrees of freedom are also propagated classically.

In the adiabatic representation (relevant to the Born-Oppenheimer PESs generated by Q-CHEM), The Meyer-Miller Hamiltonian is

$$H(\mathbf{x},\mathbf{p},\mathbf{R},\mathbf{P}) = \frac{1}{2\boldsymbol{\mu}}\left(\mathbf{P} + \sum_{I,J}^{F}(x_I p_J - x_J p_I)\mathbf{d}_{IJ}(\mathbf{R})\right)^2 + \sum_{I}^{F}\left(\frac{1}{2}p_I^2 + \frac{1}{2}x_I^2 - \gamma_I\right)E_I(\mathbf{R}) \tag{9.64}$$

where $\{x_I, p_I\}$ are the coordinates and momenta of the "electronic oscillators" corresponding to a set of F electronic states, $\mathbf{R}$, $\mathbf{P}$ are the coordinates and momenta of the nuclear DOF having reduced masses $\boldsymbol{\mu}$, $E_I(\mathbf{R})$ is the Born-Oppenheimer PES corresponding to adiabatic state I, $\mathbf{d}_{IJ}(\mathbf{R}) = \left\langle \Psi_I \middle| \vec{\nabla}_{\mathbf{R}} \Psi_J \right\rangle$ is the standard first-order derivative coupling vector between electronic states I and J, and $\{\gamma_I\}$ are a set of zero point energy parameters. The evolution of the F classical oscillators in Eq. (9.64) thus describes the electronic configuration in the MM model and, in particular, the classical actions associated with each oscillator.

The symmetric quasi-classical Meyer-Miller (SQC/MM) approach is requested in Q-CHEM with the QCMD_METHOD rem variable. Quantization of the classical Hamiltonian dynamics produced by Eq. (9.64) is done symmetrically, *i.e.*, with respect to both the initial and final values of the dynamical electronic variables. This is performed initially by Monte Carlo sampling actions from a "windowing" function. The quantization at the prescribed final times is accomplished by "binning" the time-evolved actions according to the same windowing function. In Q-CHEM's implementation, only the "triangle" style of windowing function is employed,[25] which was found universally superior to the original histogram style windows of Ref. 24. Additionally, the option to use the γ-adjustment protocol of Ref. 26 is available and requested with the SQC_GAMADJUST keyword. This is generally recommended. The key point of the γ-adjustment procedure is to set the $\{\gamma_I\}$ in Eq. (9.64) per DOF (and per trajectory), so that the initial forces on the nuclei are that of the initial pure quantum state—*i.e.*, the single-PES forces. Ehrenfest simulations are also available where the dynamics of these are equivalent to the SQC calculations, but instead of using symmetric windowing functions for selecting initial conditions and estimating final populations, the Ehrenfest method uses integer initial electronic action variables with $\gamma = 0$ and uses the *values* of these action variables at each desired final time to estimate the electronic state populations.

Sampling distributions for the initial nuclear DOF are requested with the QCMD_INITNUC keyword. Currently, sampling both initial positions and velocity from either a Wigner or Boltzmann distribution is available. Sampling from either of these distributions requires a frequency calculation to be available. Alternatively, the user can input velocities using the *$velocity* section as described in Section 9.9.2.

QCMD_METHOD

Specifies the nonadiabatic Meyer-Miller molecular dynamics method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

Ehrenfest Traditional Ehrenfest molecular dynamics.

SQC Symmetric Quasi-Classical Meyer-Miller molecular dynamics

RECOMMENDATION:

None

QCMD_INITSTATE

Specifies the initially populated electronic state.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n An integer set less than CIS_N_ROOTS.

RECOMMENDATION:

None

QCMD_INITNUC

Specifies the distribution used when sampling initial nuclear positions and velocities.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

Wigner Wigner distribution.

Boltzmann Boltzmann distribution

RECOMMENDATION:

Used in conjunction with AIMD_TEMP.

QCMD_WARMUP

Specifies the number of linearly-interpolated steps between the initial and sampled configurations for accurate state following before the dynamics begin.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n

RECOMMENDATION:

None

SQC_GAMADJUST

Specifies the γ-adjustment protocol.

TYPE:

STRING

DEFAULT:

True

OPTIONS:

True use the γ-adjustment protocol.

False

RECOMMENDATION:

The γ-adjustment protocol is generally recommended.

Example 9.44 Ehrenfest simulation.

```
$molecule
0 1
 Na      0.00000000      0.00000000      0.93444743
  H      0.00000000      0.00000000     -0.93444743
$end

$rem
   JOBTYPE                  freq
   METHOD                   hflyp
   BASIS                    6-31g*
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE                aimd
   METHOD                 hflyp
   BASIS                  6-31g*
   CIS_N_ROOTS            6
   CIS_SINGLETS           true
   CIS_TRIPLETS           false
   AIMD_METHOD            qcmd      !initiates quasi-classical nonadiabatic dynamics
   QCMD_METHOD            ehrenfest
   TIME_STEP              10        !in atomic units
   AIMD_STEPS             340
   QCMD_INITSTATE         5         !start on S5
   QCMD_INITNUC           boltzmann !sample from a Boltzmann distribution
   AIMD_TEMP              300
   RPA                    0
$end
```

Example 9.45 SQC/MM simulation.

```
$molecule
0 1
  C     -1.10077021     -0.38619733      0.05617695
  C      0.09675762     -1.08610723     -0.30779113
  C      1.28662554     -0.41739388     -0.28021086
  O      1.41824439      0.84326764      0.06159343
  O     -1.12106364      0.80448471      0.39946439
  H      0.49060260      1.16255871      0.27980893
  H      0.06099112     -2.12846061     -0.60005353
  H      2.23038546     -0.89206633     -0.54524306
  H     -2.05177379     -0.94757617      0.02607669
$end

$rem
   JOBTYPE                  freq
   METHOD                   pbe0
   BASIS                    6-31g*
$end

@@@

$molecule
read
$end

$rem
   JOBTYPE                aimd
   METHOD                 pbe0
   BASIS                  6-31g*
   CIS_N_ROOTS            4
   CIS_SINGLETS           true
   CIS_TRIPLETS           false
   AIMD_METHOD            qcmd      !initiates quasi-classical nonadiabatic dynamics
   QCMD_METHOD            sqc
   TIME_STEP              10        !in atomic units
   AIMD_STEPS             340
   QCMD_INITSTATE         2         !start on S2
   QCMD_INITNUC           wigner    !sample from a Wigner distribution
   SQC_GAMADJUST          true
   AIMD_TEMP              0
   RPA                    0
$end
```

9.10 *Ab Initio* Path Integrals

9.10.1 Theory

Even in cases where the Born-Oppenheimer separation is valid, solving the electronic Schrödinger equation may only be half the battle. The remainder involves the solution of the *nuclear* Schrödinger equation for its resulting eigenvalues and eigenfunctions. This half is typically treated by the harmonic approximation at critical points, but anharmonicity, tunneling, and low-frequency ("floppy") motions can lead to extremely delocalized nuclear distributions, particularly for protons and for non-covalent interactions.

While the Born-Oppenheimer separation allows for a local solution of the electronic problem (in nuclear space), the nuclear half of the Schrödinger equation is entirely non-local and requires the computation of potential energy surfaces over large regions of configuration space. Grid-based methods, therefore, scale exponentially with the number of degrees of freedom, and are quickly rendered useless for all but very small molecules.

For equilibrium thermal distributions, the path integral (PI) formalism provides both an elegant and computationally feasible alternative. The equilibrium partition function can be written as a trace of the thermal, configuration-space density matrix,

$$Z = \mathrm{tr}\big(e^{-\beta\hat{H}}\big) = \int dx\, \big\langle x\big|e^{-\beta\hat{H}}\big|x\big\rangle = \int dx\, \rho(x,x;\beta)\,. \tag{9.65}$$

The density matrix at inverse temperature $\beta = (k_B T)^{-1}$ is defined by the last equality. Evaluating the integrals in Eq. (9.65) still requires computing eigenstates of $\hat{H}$, which is generally intractable. Inserting $N-1$ resolutions of the identity, however, one obtains

$$Z = \int dx_1 \int dx_2 \cdots \int dx_N\, \rho\left(x_1, x_2; \frac{\beta}{N}\right) \rho\left(x_2, x_3; \frac{\beta}{N}\right) \cdots \rho\left(x_N, x_1; \frac{\beta}{N}\right)\,. \tag{9.66}$$

Here, the density matrices appear at an inverse temperature β/N that corresponds to multiplying the actual temperature T by a factor of N.

The high-temperature form of the density matrix can be expressed as

$$\rho\left(x, x'; \frac{\beta}{N}\right) = \left(\frac{mN}{2\pi\beta\hbar^2}\right)^{1/2} \exp\left\{-\left(\frac{mN}{2\beta\hbar^2}\right)(x-x')^2 - \left(\frac{\beta}{2N}\right)\Big[V(x)+V(x')\Big]\right\} \tag{9.67}$$

which becomes exact as $T \to \infty$ (a limit in which quantum mechanics converges to classical mechanics), or in other words as $\beta \to 0$ or $N \to \infty$. Using N time slices, the partition function is therefore converted into the form

$$Z = \left(\frac{mN}{2\pi\beta\hbar^2}\right)^{N/2} \int dx_1 \int dx_2 \cdots \int dx_N\, \exp\left\{-\frac{\beta}{N}\left[\frac{mN^2}{2\beta^2\hbar^2}\sum_{i=1}^{N}(x_i - x_{i+1})^2 + \sum_{i=1}^{N} V(x_i)\right]\right\}\,, \tag{9.68}$$

with the implied cyclic condition $x_{N+1} \equiv x_1$. Here, $V(x)$ is the potential function on which the "beads" move, which is the electronic potential generated by Q-CHEM.

Equation 9.68 has the form

$$Z \propto \int e^{-\beta V_{\mathrm{eff}}}\,, \tag{9.69}$$

where the form of the effective potential V_{eff} is evident from the integrand in Eq. (9.68). Equation (9.69) reveals that the path-integral formulation of the quantum partition function affords a *classical* configurational integral for the partition function, albeit in an extended-dimensional space The effective potential describes a classical "ring polymer" with N beads, wherein neighboring beads are coupled by harmonic potentials that arise from the quantum nature of the kinetic energy. The exponentially-scaling, non-local nuclear quantum mechanics problem has therefore been mapped onto an entirely classical problem, which is amenable to standard treatments of configuration sampling. These methods typically involve molecular dynamics or Monte Carlo sampling. Importantly, the number of extended degrees of freedom, N, is reasonably small when the temperature is not too low: room-temperature systems involving hydrogen atoms typically are converged using roughly $N \approx 30$ beads. Therefore, fully quantum-mechanical nuclear distributions can be obtained at a cost only roughly 30 times that of a classical AIMD simulation. Path integral Monte Carlo (PIMC) is activated by setting JOBTYPE = PIMC.

The single-bead ($N = 1$) limit of the equations above is simply classical configuration sampling. When the temperature (controlled by the PIMC_TEMP keyword) is high, or where only heavy atoms are involved, the classical limit is often appropriate. The path integral machinery (with a single "bead") may be used to perform classical Boltzmann sampling. In this case, the partition function is simply

$$Z = \int dx\, e^{-\beta V(x)} \tag{9.70}$$

and this is what is ordinarily done in an AIMD simulation. Use of additional beads incorporates more quantum-mechanical delocalization, at a cost of roughly N times that of the classical AIMD simulation, and this is the primary input variable in a PI simulation. It is controlled by the keyword PIMC_NBEADSPERATOM. The ratio of the inverse temperature to beads (β/N) dictates convergence with respect to the number of beads, so as the temperature is lowered, a concomitant increase in the number of beads is required.

Integration over configuration space is performed by Metropolis Monte Carlo (MC). The number of MC steps is controlled by the PIMC_MCMAX keyword and should typically be $\gtrsim 10^5$, depending on the desired level of statistical convergence. A warm-up run, in which the PI ring polymer is allowed to equilibrate without accumulating statistics, can be performed using the PIMC_WARMUP_MCMAX keyword.

As in AIMD simulations, the main results of PIMC jobs in Q-CHEM are not in the job output file but are instead output to (`$QCSCRATCH/PIMC` in the user's scratch directory, thus PIMC jobs should always be run with the *-save* option. The output files do contain some useful information, however, including a basic data analysis of the simulation. Average energies (thermodynamic estimator), bond lengths (less than 5 Å), bond length standard deviations and errors are printed at the end of the output file. The `$QCSCRATCH/PIMC` directory additionally contains the following files:

- `BondAves`: running average of bond lengths for convergence testing.
- `BondBins`: normalized distribution of significant bond lengths, binned within 5 standard deviations of the average bond length.
- `ChainCarts`: human-readable file of configuration coordinates, likely to be used for further, external statistical analysis. This file can get quite large, so be sure to provide enough scratch space!
- `ChainView.xyz`: Cartesian-formatted file for viewing the ring-polymer sampling in an external visualization program. (The sampling is performed such that the center of mass of the ring polymer system remains centered.)
- `Vcorr`: potential correlation function for the assessment of statistical correlations in the sampling.

In each of the above files, the first few lines contain a description of how the data are arranged.

One of the unfortunate rites of passage in PIMC usage is the realization of the ramifications of the stiff bead-bead interactions as convergence (with respect to N) is approached. Nearing convergence—where quantum mechanical results are correct—the length of statistical correlations grows enormously, and special sampling techniques are required to avoid long (or non-convergent) simulations. Cartesian displacements or normal-mode displacements of the ring polymer lead to this severe stiffening. While both of these naïve sampling schemes are available in Q-CHEM, they are not recommended. Rather, the free-particle (harmonic bead-coupling) terms in the path integral action can be sampled directly. Several schemes are available for this purpose. Q-CHEM currently adopts the simplest of these options, Levy flights. An n-bead segment (with $n < N$) of the ring polymer is chosen at random, with the length n controlled by the PIMC_SNIP_LENGTH keyword. Between the endpoints of this segment, a free-particle path is generated by a Levy construction, which *exactly* samples the free-particle part of the action. Subsequent Metropolis testing of the resulting potential term—for which only the potential on the moved beads is required—then dictates acceptance.

Two measures of the sampling efficiency are provided in the job output file. The lifetime of the potential auto-correlation function $\langle V_0 V_\tau \rangle$ is provided in terms of the number of MC steps, τ. This number indicates the number of configurations that are statically correlated. Similarly, the mean-square displacement between MC configurations is also provided. Maximizing this number and/or minimizing the statistical lifetime leads to efficient sampling. Note that the optimally efficient acceptance rate may *not* be 50% in MC simulations. In Levy flights, the only variable controlling acceptance and sampling efficiency is the length of the snippet. The statistical efficiency can be obtained from relatively short runs, during which the length of the Levy snippet should be optimized by the user.

9.10.2 Job Control and Examples

PIMC_NBEADSPERATOM

Number of path integral time slices ("beads") used on each atom of a PIMC simulation.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

1 Perform classical Boltzmann sampling.

>1 Perform quantum-mechanical path integral sampling.

RECOMMENDATION:

This variable controls the inherent convergence of the path integral simulation. The one-bead limit represents classical sampling and the infinite-bead limit represents exact quantum-mechanical sampling. Using 32 beads is reasonably converged for room-temperature simulations of molecular systems.

PIMC_TEMP

Temperature, in Kelvin (K), of path integral simulations.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified number of Kelvin for PIMC or classical MC simulations.

RECOMMENDATION:

None.

PIMC_MCMAX

Number of Monte Carlo steps to sample.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified number of steps to sample.

RECOMMENDATION:

This variable dictates the statistical convergence of MC/PIMC simulations. For converged simulations at least 10^5 steps is recommended.

PIMC_WARMUP_MCMAX

Number of Monte Carlo steps to sample during an equilibration period of MC/PIMC simulations.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified number of steps to sample.

RECOMMENDATION:

Use this variable to equilibrate the molecule/ring polymer before collecting production statistics. Usually a short run of roughly 10% of PIMC_MCMAX is sufficient.

PIMC_MOVETYPE

Selects the type of displacements used in MC/PIMC simulations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Cartesian displacements of all beads, with occasional (1%) center-of-mass moves.

1 Normal-mode displacements of all modes, with occasional (1%) center-of-mass moves.

2 Levy flights without center-of-mass moves.

RECOMMENDATION:

Except for classical sampling (MC) or small bead-number quantum sampling (PIMC), Levy flights should be used. For Cartesian and normal-mode moves, the maximum displacement is adjusted during the warm-up run to the desired acceptance rate (controlled by PIMC_ACCEPT_RATE). For Levy flights, the acceptance is solely controlled by PIMC_SNIP_LENGTH.

PIMC_ACCEPT_RATE

Acceptance rate for MC/PIMC simulations when Cartesian or normal-mode displacements are used.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

$0 < n < 100$ User-specified rate, given as a whole-number percentage.

RECOMMENDATION:

Choose acceptance rate to maximize sampling efficiency, which is typically signified by the mean-square displacement (printed in the job output). Note that the maximum displacement is adjusted during the warm-up run to achieve roughly this acceptance rate.

PIMC_SNIP_LENGTH

Number of "beads" to use in the Levy flight movement of the ring polymer.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

$3 \leq n \leq$ PIMC_NBEADSPERATOM User-specified length of snippet.

RECOMMENDATION:

Choose the snip length to maximize sampling efficiency. The efficiency can be estimated by the mean-square displacement between configurations, printed at the end of the output file. This efficiency will typically, however, be a trade-off between the mean-square displacement (length of statistical correlations) and the number of beads moved. Only the moved beads require recomputing the potential, *i.e.*, a call to Q-CHEM for the electronic energy. (Note that the endpoints of the snippet remain fixed during a single move, so $n - 2$ beads are actually moved for a snip length of n. For 1 or 2 beads in the simulation, Cartesian moves should be used instead.)

Example 9.46 Path integral Monte Carlo simulation of H_2 at room temperature

```
$comment
The number of Monte Carlo steps is deliberately set low, more typical
values would be:
   PIMC_WARMUP_MCMAX    10000   !Equilibration run
   PIMC_MCMAX           100000  !Production run
$end

$molecule
   0  1
   H
   H 1 0.75
$end

$rem
   JOBTYPE              pimc
   METHOD               hf
   BASIS                sto-3g
   PIMC_TEMP            298
   PIMC_NBEADSPERATOM   32
   PIMC_WARMUP_MCMAX    100     !Equilibration run
   PIMC_MCMAX           1000    !Production run
   PIMC_MOVETYPE        2         !Levy flights
   PIMC_SNIP_LENGTH     10        !Moves 8 beads per MC step (10-endpts)
$end
```

Example 9.47 Classical Monte Carlo simulation of a water molecule at 500K

```
$comment
The number of Monte Carlo steps is deliberately set low, more typical
values would be:
   PIMC_WARMUP_MCMAX    10000   !Equilibration run
   PIMC_MCMAX           100000  !Production run
$end

$molecule
   0  1
   H
   O 1 1.0
   H 2 1.0 1 104.5
$end

$rem
   JOBTYPE              pimc
   METHOD               rimp2
   BASIS                cc-pvdz
   AUX_BASIS            rimp2-cc-pvdz
   PIMC_TEMP            500
   PIMC_NBEADSPERATOM   1       !1 bead is classical sampling
   PIMC_WARMUP_MCMAX    100     !Equilibration run
   PIMC_MCMAX           1000    !Production run
   PIMC_MOVETYPE        0       !Cartesian displacements (ok for 1 bead)
   PIMC_ACCEPT_RATE     40      !During warm-up, adjusts step size to 40% acceptance
$end
```

9.11 *Ab Initio* Molecular Dynamics with Complex Absorbing Potentials

9.11.1 Introduction

The study of metastable electronic states like temporary anions presents a major challenge for computational chemists. For example, finding a Hartree-Fock (HF) SCF solution that describes the electronic state of a given temporary anion is usually an arduous and tricky task.[46] That makes the prospect of performing a simple HF-based AIMD simulation of temporary anions even more daunting. On top of the inherent difficulties of the electronic structure problem, one also has to take into account the fact that upon its formation, the temporary anion is, in general, subject to two competing processes: electron autodetachment ($AB^- \rightarrow AB + e^-$) and dissociative electron attachment (DEA) ($AB^- \rightarrow A + B^-$). However, the need to be able to perform such an AIMD simulation cannot be overstated given that such an effort has the potential to offer very important insights into the mechanisms connecting the formation of temporary anions to their DEA products, for example.

Taking advantage of the analytic gradients for complex absorbing potential (CAPs) in Q-CHEM,[15], one may combine the general principles of AIMD simulations with the CAP method (Section 7.10.9), facilitating AIMD simulations for temporary anions. Based on the Born-Oppenheimer approximation, this "CAP-AIMD" method makes it possible to propagate the nuclei on a complex potential energy surface (CPES) computed on the fly.[43]

Starting a CAP-AIMD simulation on the right CPES is paramount to a successful simulation. Failure to do so will lead to wrong results. For this reason, before moving on to discuss how one can run CAP-AIMDs with Q-CHEM, it is useful to provide brief guidelines on how to find correct CAP-HF SCF solutions for temporary anions.

9.11.2 Finding Electronic Resonance States of Temporary Anions

We limit our discussion here to the HF method, but the general principles outlined also applies to higher level methods.

1. The correct description of the electronic state of temporary anions usually requires the use of extra diffuse functions in the basis set. You may thus consider having enough extra diffuse functions in your basis set.

2. It is best you use the core Hamiltonian as guess for the CAP-HF SCF procedure (*i.e.*, set SCF_GUESS = CORE).

3. For CAP calculations, Q-CHEM first solves the SCF problem using the old SCF drivers in Q-CHEM; the solution is then used as the starting point for the CAP part of the calculation. This preliminary SCF solution usually leads to wrong CAP-HF solutions. You may avoid this issue by setting SKIP_OLD_SCFMAN = TRUE in *$rem*. See Example 9.48.

4. Make sure the CAP contribution to the real and imaginary parts of the SCF solution found is negligible. You can get these CAP contributions by asking for an energy decomposition of the complex energy by setting CS_SCF_FINAL_PRINT = 1 in *$rem*. An alternative is to look at how close the imaginary (and real) parts of the CAP-HF energy and the corrected CAP-HF energy are; the ratio of the imaginary parts should be close to unity. See Example 9.48.

For CAP-HF calculations, set the *$rem* variable CS_STRICT = TRUE to print out the correct properties (*e.g.*, Mulliken charges and multipole moments) of the solution. See Example 9.48.

9.11.3 CAP-AIMD Job Control and Examples

The following three assignments are necessary in order to run a CAP-AIMD simulation:

- JOBTYPE = AIMD in *$rem*,
- COMPLEX_CCMAN = TRUE in *$rem* (see Section 7.10.9), and
- CS_HF = 1 in *$complex_ccman* (see Section 7.10.9).

For now, CAP-AIMD simulations are possible only with the cuboid CAP type, so setting CAP_TYPE = 1 is also necessary in the *$complex_ccman* section (see Section 7.10.9).

With CAP-AIMD simulations, one gets two additional files in the AIMD directory (§9.9.2):

- `CAP_EComponents`: Records for each step the total complex energy, the CAP-corrected total complex energy, and the real and imaginary parts of the CAP contribution to the total complex energy – all in atomic units (a.u.).
- `CAP_PositionAndWidth`: Records for each step the total complex energy (in a.u.) and the resonance width (in electron-Volt).

CAP_AIMD_SWITCH

Sets CAP_ETA to zero during a CAP-AIMD simulation when the real part of the last alpha occupied orbital's energy is negative

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Set CAP_ETA to zero when the real part of the last alpha occupied orbital's becomes negative.

FALSE Keep user's CAP_ETA constant throughout simulation.

RECOMMENDATION:

Use default.

CS_STRICT

Determines Mulliken charges, multipole moments and complex orbital energies for CAP-HF calculations by reading, when applicable, complex density matrix or complex molecular orbital coefficient file

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE determine Mulliken charges, multipole moments and complex orbital energies for CAP-HF calculations by reading – when applicable – the complex density matrix or complex molecular orbital coefficient file.

FALSE Don't read the complex density matrix or complex molecular orbital coefficient file when determining Mulliken charges, multipole moments and orbital energies for CAP-HF calculations.

RECOMMENDATION:

Set to 'TRUE' for CAP-HF calculations.

SKIP_OLD_SCFMAN

Skips *only* old SCF drivers

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Skip *only* old SCF drivers

FALSE Do not skip old SCF drivers

RECOMMENDATION:

When performing CAP calculations on temporary anions, it may help setting this variable to FALSE.

CS_SCF_FINAL_PRINT

Controls level of output from CAP-SCF procedure.

TYPE:

INTEGER

DEFAULT:

0 No extra print out.

OPTIONS:

1 Print direct breakdown of CAP-SCF energy.

2 Print breakdown of CAP-SCF energy based on the complex coefficient matrix. Also required if the options below are requested.

3 Level 2 plus diagonal elements of complex orbital energy matrix, **F**. Triggered by Level 2.

4 Level 2 plus diagonal elements of complex kinetic energy matrix, **T**. Triggered by Level 2

5 Level 2 plus diagonal elements of complex electron-nuclear Coulomb potential energy matrix, **V**. Triggered by Level 2.

6 Level 2 plus diagonal elements of CAP matrix, **W**. Triggered by Level 2.

7 Level 2 plus diagonal elements of total complex one-electron energy matrix, $\mathbf{T} + \mathbf{V} + \mathbf{W}$. Triggered by Level 2.

8 Level 2 plus diagonal elements of total complex electronic energy matrix, $\mathbf{T} + \mathbf{V} + \mathbf{W} + \mathbf{F}$. Triggered by Level 2.

9 Level 2 to 8. Triggered by Level 2.

RECOMMENDATION:

Level 1 is usually enough. Values for this *$rem* variable are transformed first into a set of distinct values; thus, for example, "1111" is equivalent to "1" and "28224" is equivalent to "248". To request Levels 3–9, please remember to request Level 2 as well.

Example 9.9.48 CAP-HF single point job for N_2^-, with energy decomposition of the complex energy. Basis set is cc-PVTZ+3p.

View input online

Example 9.9.49 CAP-AIMD simulation for N_2^-. Basis set is cc-PVTZ+3p.

View input online

Example 9.9.50 CAP-AIMD simulation for $C_2H_4^-$. Basis set is cc-PVTZ+3p(C).
A CAP single point calculation is first done. The CAP-SCF solution is then read as the initial guess for the CAP-AIMD part. This procedure is useful, for example, when one wants to use the MOM_START option (§4.5.14) to preserve orbital occupation in the course of the simulation.

View input online

9.12 Optimizing the Structure of Clusters

9.12.1 Introduction

The potential energy landscape of atomic and molecular clusters can be very complex with many minima which can have similar energies, and this complexity increases rapidly as the size of the clusters increases. Determining the global minimum of these clusters is challenging since it requires extensive searching over the potential energy surface. One approach to finding the low energy structures of these clusters is to perform many geometry optimizations starting at different initial coordinates. Q-CHEM is able to perform such random searches for molecular clusters containing up to two different molecule types. In these searches the molecules are subjected to translations and rotations of their structure to generate a new starting structure. These searches are initiated by the JOBTYPE = RAND and it is necessary to specify the number of molecules of the different types and the number of atoms in the different types of molecules. For the optimization of atomic clusters, SEARCH_ATOMIC = TRUE and the number of atom swops performed in the structure generation (N_SWOP) can be specified. Some care has to be taken with the specification of the input structure in the *$molecule* section. All the atoms of the molecules of molecule type 1 must come before those of molecule type

2. Furthermore, the atoms of the same molecule should be together. For examples of these studies see Refs. 31–33,63.

Example 9.51 A random search geometry optimization of the $NO^+.H_2O$ cluster.

```
$molecule
   1 1
   N    0.5682008336    0.1585044954   -0.9009280260
   O   -0.3450383302   -0.5598328271   -0.4634299478
   O    1.7303273568    0.3403569345    0.4364171165
   H    2.5236300547   -0.2494576134    0.1485689942
   H    2.1020812302    1.2823911654    0.2570156558
$end

$rem
   JOBTYPE                     RAND
   METHOD                      B3LYP
   BASIS                       STO-3G
   SCF_CONVERGENCE             6
   MAX_SCF_CYCLES              100
   NSEARCH                     10
   N_MOL_TYPE                  2
   NMOL1                       1
   N_ATOM_TYPE_1               2
   NMOL2                       1
   N_ATOM_TYPE_2               3
   N_MOVES                     20
   MAXBOX                      10000
   MIN_SEPARATION              25
   MAX_DISPLACE                25
   SCF_NOCRASH                 TRUE
   TIGHTEN_CONVERG             TRUE
   GEOM_OPT_MAX_CYCLES         200
   GEOM_OPT_COORDS             0
   GEOM_OPT_TOL_DISPLACEMENT   1000
   GEOM_OPT_TOL_GRADIENT       300
   GEOM_OPT_TOL_ENERGY         100
$end
```

Example 9.52 A random search geometry optimization of the He_3Ne_3 cluster.

```
$molecule
   0 1
   He   -1.3590894    3.0177788   -0.1662522
   He   -2.9853158    1.1444488    0.1036005
   He    0.5068109    1.3795209   -0.2168151
   Ne   -1.1002149   -0.5693061    0.0381894
   Ne    0.5981676    1.8697812    1.4685618
   Ne   -1.2376457    1.2597811   -0.0756066
$end

$rem
   JOBTYPE                     RAND
   METHOD                      B3LYP
   DFT_D                       EMPIRICAL_GRIMME
   BASIS                       STO-3G
   SCF_CONVERGENCE             6
   MAX_SCF_CYCLES              100

   NSEARCH                     10
   SEARCH_ATOMIC               TRUE
   N_SWOP                      4
   N_MOL_TYPE                  2
   NMOL1                       3
   N_ATOM_TYPE_1               1
   NMOL2                       3
   N_ATOM_TYPE_2               1
   N_MOVES                     20
   MAXBOX                      10000
   MIN_SEPARATION              25
   MAX_DISPLACE                25
   SCF_NOCRASH                 TRUE
   TIGHTEN_CONVERG             TRUE
   USE_INITIAL                 TRUE

   GEOM_OPT_MAX_CYCLES         200
   GEOM_OPT_COORDS             0
   GEOM_OPT_TOL_DISPLACEMENT   1000
   GEOM_OPT_TOL_GRADIENT       3000
   GEOM_OPT_TOL_ENERGY         1000
$end
```

Basin hopping (BH) is a more advanced technique for locating the global minimum on complex potential energy surfaces.[105] The BH algorithm is essentially a combination of the Metropolis sampling technique and a gradient-based local search method. This has the effect of sampling the energy basins, where an energy basin is a certain part of the configuration space around a minimum on the PES that contains all the configurations that will relax into this minimum using downhill relaxations, instead of sampling the configuration space. To enhance the efficiency of the method, BH with occasional jumping is used,[55] which incorporates a jumping move in addition to the standard Monte Carlo (MC) moves. Jumping is a MC move without local minimization at infinite temperature and, consequently, is always accepted. When the usual MC moves are rejected a number of times, the system is judged to be trapped at the local minimum. The temperature is raised to $T = \infty$, and the MC jumping moves are executed several times to allow the system to escape from the local minimum. This provides an efficient way to escape from a local minimum and to explore the next basin of the valley when it is separated by high barriers. Depending on the size and complexity of the

system being studied, a large number of MC_STEPS and/or MC_CYCLES to ensure the global minimum is found.

Example 9.53 A basin hopping search for the $NO^+(H_2O)$ cluster.

```
$molecule
   1 1
   N     0.5682008336    0.1585044954   -0.9009280260
   O    -0.3450383302   -0.5598328271   -0.4634299478
   O     1.7303273568    0.3403569345    0.4364171165
   H     2.5236300547   -0.2494576134    0.1485689942
   H     2.1020812302    1.2823911654    0.2570156558
$end

$rem
  JOBTYPE          BH
  METHOD           B3LYP
  BASIS            STO-3G
  SCF_CONVERGENCE  6
  MAX_SCF_CYCLES   100

  MC_CYCLES             4
  MC_STEPS              5
  MC_TEMP               300
  MAX_DISPLACE          25
  MIN_SEPARATION        25
  MAXBOX                5000
  N_MOVES               20

  N_MOL_TYPE       2
  NMOL1            1
  N_ATOM_TYPE_1    2
  NMOL2            1
  N_ATOM_TYPE_2    3
  N_MOVES          20
  MAXBOX           10000
  MIN_SEPARATION   25
  MAX_DISPLACE     25
  SCF_NOCRASH      TRUE

  GEOM_OPT_MAX_CYCLES       200
  GEOM_OPT_COORDS           0
  GEOM_OPT_TOL_DISPLACEMENT 2000
  GEOM_OPT_TOL_GRADIENT     4000
  GEOM_OPT_TOL_ENERGY       400
$end
```

9.12.2 Cluster Optimization Job Control

NSEARCH

INTEGER

TYPE:

Sets the number of structures that are generated and optimized.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

SEARCH_ATOMIC

Perform an optimization for atomic cluster.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Atomic cluster search will be performed.

False Molecular clusters search will be performed.

RECOMMENDATION:

Use N_SWOP to specify atomic number of atom swops in structure generation.

N_SWOP

INTEGER

TYPE:

Sets the number atom coordinate swops for atomic cluster search.

DEFAULT:

No default.

OPTIONS:

User defined

RECOMMENDATION:

None

N_MOL_TYPE

INTEGER

TYPE:

Sets the number of different atom or molecule types.

DEFAULT:

No default.

OPTIONS:

User defined : can be 1 or 2.

RECOMMENDATION:

None

NMOL1

INTEGER

TYPE:

Sets the number of molecules of type 1.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

N_ATOM_TYPE_1

INTEGER

TYPE:

Sets the number atoms in molecule type 1.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

NMOL2

INTEGER

TYPE:

Sets the number of molecules of type 2.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

N_ATOM_TYPE_2

INTEGER

TYPE:

Sets the number atoms in molecule type 2.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

MAXBOX

Sets the size of the box which the molecules are kept within.

TYPE:

INTEGER

DEFAULT:

20000

OPTIONS:

n Corresponding to MAXBOX = $n/1000$ bohr.

RECOMMENDATION:

Need to ensure that the cluster can fit within this box.

MIN_SEPARATION

Reject initial structures where the closest approach of molecules is less than this value.

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

n Corresponding to MIN_SEPARATION = $n/100$ bohr.

RECOMMENDATION:

MIN_SEPARATION of approximately 2.5 bohr.

MAX_DISPLACE

Sets the maximum distance a molecule will be moved during a translation.

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Corresponding to MAX_DISPLACE = $n/100$ bohr.

RECOMMENDATION:

None.

SCF_NOCRASH

Ensure the calculations continues if the SCF fails to converge for a given structure.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Ensure calculation will continue with next structure.

False Calculation will stop.

RECOMMENDATION:

Use SCF_NOCRASH = TRUE.

TIGHTEN_CONVERG

At the end of the search re-calculate the energies of the optimized structures with tighter SCF convergence criteria.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Additional calculations with tighter SCF convergence performed.

False No additional calculations performed.

RECOMMENDATION:

None.

USE_INITIAL

Include input structure as part of the search.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Input structure is included in the search.

False Input structure is not included in the search.

RECOMMENDATION:

None.

SEARCH_MOM

Allows the search to be performed in conjunction with MOM to explore excited states.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True A search with MOM is performed.

False Normal calculation without MOM.

RECOMMENDATION:

None.

MC_CYCLES

INTEGER

TYPE:

Sets the number of cycles in a basin hopping search.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

MC_STEPS

INTEGER

TYPE:

Sets the number of Monte Carlo steps in each MC_CYCLES. After MC_STEPS jumping is initiated.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

MC_TEMP

INTEGER

TYPE:

Sets the temperature (in Kelvin).

DEFAULT:

300

OPTIONS:

User defined.

RECOMMENDATION:

None

N_MOVES

INTEGER

TYPE:

Sets the number of structural changes/moves on each step.

DEFAULT:

2

OPTIONS:

User defined.

RECOMMENDATION:

None

MAX_JUMP

INTEGER

TYPE:

Sets the number of moves accepted on jumping.

DEFAULT:

10

OPTIONS:

User defined.

RECOMMENDATION:

None

References and Further Reading

[1] The VMD program may be downloaded from http://www.ks.uiuc.edu/Research/vmd.

[2] C. M. Aikens, S. P. Webb, R. L. Bell, G. D. Fletcher, M. W. Schmidt, and M. S. Gordon. *Theor. Chem. Acc.*, 110:233, 2004. DOI: 10.1007/s00214-003-0453-3.

[3] H. C. Andersen. *J. Comput. Phys.*, 52:24, 1983. DOI: 10.1016/0021-9991(83)90014-1.

[4] G. J. Atchity, S. S. Xantheas, and K. Ruedenberg. *J. Chem. Phys.*, 95:1862, 1991. DOI: 10.1063/1.461036.

[5] J. Baker. *J. Comput. Chem.*, 7:385, 1986. DOI: 10.1002/jcc.540070402.

[6] J. Baker. *J. Comput. Chem.*, 13:240, 1992. DOI: 10.1002/jcc.540130215.

[7] J. Baker. *J. Comput. Chem.*, 18:1079, 1997. DOI: 10.1002/(SICI)1096-987X(199706)18:8<1079::AID-JCC12>3.0.CO;2-8.

[8] J. Baker and D. Bergeron. *J. Comput. Chem.*, 14:1339, 1993. DOI: 10.1002/jcc.540141111.

[9] J. Baker and W. J. Hehre. *J. Comput. Chem.*, 12:606, 1991. DOI: 10.1002/jcc.540120510.

[10] J. Baker, A. Kessi, and B. Delley. *J. Chem. Phys.*, 105:192, 1996. DOI: 10.1063/1.471864.

[11] V. Bakken and T. Helgaker. *J. Chem. Phys.*, 117:9160, 2002. DOI: 10.1063/1.1515483.

[12] A. Banerjee, N. Adams, J. Simons, and R. Shepard. *J. Phys. Chem.*, 89:52, 1985. DOI: 10.1021/j100247a015.

[13] M. J. Bearpark, M. A. Robb, and H. B. Schlegel. *Chem. Phys. Lett.*, 223:269, 1994. DOI: 10.1016/0009-2614(94)00433-1.

[14] A. Behn, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon. *J. Chem. Phys.*, 135:224108, 2011. DOI: 10.1063/1.3664901.

[15] Z. Benda and T.-C. Jagau. *J. Chem. Phys.*, 146:031101, 2017. DOI: 10.1063/1.4974094.

[16] J. M. Bofill. *J. Comput. Chem.*, 15:1, 1994. DOI: 10.1002/jcc.540150102.

[17] P. E. Bowling, S. Dasgupta, and J. M. Herbert. *J. Chem. Inf. Model*, 64:3912, 2024. DOI: 10.1021/acs.jcim.4c00221.

[18] C. G. Broyden. *SIAM J. App. Math.*, 6:76, 1970. DOI: 10.1093/imamat/6.1.76.

[19] G. Bussi and M. Parrinello. *Phys. Rev. E*, 75:056707, 2007. DOI: 10.1103/PhysRevE.75.056707.

[20] S. Califano. *Vibrational States*. Wiley, London, 1976.

[21] C. J. Cerjan and W. H. Miller. *J. Chem. Phys.*, 75:2800, 1981. DOI: 10.1063/1.442352.

[22] K. D. Closser, O. Gessner, and M. Head-Gordon. *J. Chem. Phys.*, 140:134306, 2014. DOI: 10.1063/1.4869193.

[23] S. J. Cotton and W. H. Miller. *J. Phys. Chem. A*, 117:7190, 2013. DOI: 10.1021/jp401078u.

[24] S. J. Cotton and W. H. Miller. *J. Chem. Phys.*, 139:234112, 2013. DOI: 10.1063/1.4845235.

[25] S. J. Cotton and W. H. Miller. *J. Chem. Phys.*, 145:144108, 2016. DOI: 10.1063/1.4963914.

[26] S. J. Cotton and W.H. Miller. *J. Chem. Phys.*, 150:194110, 2019. DOI: 10.1063/1.5094458.

[27] P. Csaszar and P. Pulay. *J. Mol. Struct. (Theochem)*, 114:31, 1984. DOI: 10.1016/S0022-2860(84)87198-7.

[28] G. Czako, A. L. Kaledin, and J. M. Bowman. *J. Chem. Phys.*, 132:164103, 2010. DOI: 10.1063/1.3417999.

[29] S. Dasgupta and J. M. Herbert. *J. Phys. Chem. B*, 124:1137, 2020. DOI: 10.1021/acs.jpcb.9b11060.

[30] J. E. Dennis and R. B. Schnabel. *Numerical methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Englewood Cliffs, NJ, 1983.

[31] H. Do and N. A. Besley. *J. Chem. Phys.*, 137:134106, 2012. DOI: 10.1063/1.4755994.

[32] H. Do and N. A. Besley. *J. Phys. Chem. A*, 117:5385, 2013. DOI: 10.1021/jp405052g.

[33] H. Do and N. A. Besley. *Phys. Chem. Chem. Phys.*, 15:16214, 2013. DOI: 10.1021/jp405052g.

[34] W. E., W. Ren, and E. Vanden-Eijnden. *Phys. Rev. B*, 66:052301, 2002. DOI: 10.1103/PhysRevB.66.052301.

[35] Ö. Farkas and H. B. Schlegel. *J. Chem. Phys.*, 111:10806, 1999. DOI: 10.1063/1.480484.

[36] S. Fatehi, E. Alguire, Y. Shao, and J. E. Subotnik. *J. Chem. Phys.*, 135:234105, 2011. DOI: 10.1063/1.3665031.

[37] R. Fletcher. *Comput. J.*, 13:317, 1970. DOI: 10.1093/comjnl/13.3.317.

[38] R. Fletcher. *Practial Methods of Optimization*, volume 2. Wiley, New York, 1981.

[39] G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay. *J. Am. Chem. Soc.*, 114:8191, 1992. DOI: 10.1021/ja00047a032.

[40] K. Fukui. *J. Phys. Chem.*, 74:4161, 1970. DOI: 10.1021/j100717a029.

[41] G. Goldfarb. *Math. Comp.*, 24:23, 1970. DOI: 10.2307/2004873.

[42] M. Gray, A. Mandal, and J. M. Herbert. *J. Phys. Chem. A*, 129:3969, 2025. DOI: 10.1021/acs.jpca.5c01402.

[43] J. A. Gyamfi and J.-C. Jagau. *J. Phys. Chem. Lett.*, 13:8477, 2022. DOI: 10.1021/acs.jpclett.2c01969.

[44] Y. Harabuchi, K. Keipert, F. Zahariev, T. Taketsugu, and M. S. Gordon. *J. Phys. Chem. A*, 118:11987, 2014. DOI: 10.1021/jp5072428.

[45] G. Henkelman and H. Jónsson. *J. Chem. Phys.*, 113:9978, 2000. DOI: 10.1063/1.1323224.

[46] J. M. Herbert. The quantum chemistry of loosely-bound electrons. In A. L. Parill and K. Lipkowitz, editors, *Reviews in Computational Chemistry*, volume 28, chapter 8, pages 391–517. Wiley, 2015. DOI: 10.1002/9781118889886.ch8.

[47] J. M. Herbert and M. Head-Gordon. *J. Chem. Phys.*, 121:11542, 2004. DOI: 10.1063/1.1814934.

[48] J. M. Herbert and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 7:3269, 2005. DOI: 10.1039/b509494a.

[49] J. M. Herbert and A. Mandal. Spin-flip TDDFT for photochemistry. In C. Zhu, editor, *Time-Dependent Density Functional Theory: Nonadiabatic Dynamics*, chapter 10, pages 361–404. Jenny Sanford, 2023. DOI: 10.1201/9781003319214-10.

[50] J. M. Herbert, X. Zhang, A. F. Morrison, and J. Liu. *Acc. Chem. Res.*, 49:931, 2016. DOI: 10.1021/acs.accounts.6b00047.

[51] W. G. Hoover. *Phys. Rev. A*, 31:1695, 1985. DOI: 10.1103/PhysRevA.31.1695.

[52] M. Huix-Rotllant, B. Natarajan, A. Ipatov, C. M. Wawire, T. Deutsch, and M. E. Casida. *Phys. Chem. Chem. Phys.*, 12:12811, 2010. DOI: 10.1039/c0cp00273a.

[53] W. Humphrey, A. Dalke, and K. Schulten. *J. Molec. Graphics*, 14:33, 1996. DOI: 10.1016/0263-7855(96)00018-5.

[54] K. Ishida, K. Morokuma, and A. Komornicki. *J. Chem. Phys.*, 66:215, 1977. DOI: 10.1063/1.3077690.

[55] M. Iwamatsu and Y. Okabe. *Chem. Phys. Lett.*, 399:396, 2004. DOI: 10.1016/j.cplett.2004.10.032.

[56] M. Karplus, R. N. Porter, and R. D. Sharma. *J. Chem. Phys.*, 43:3259, 1965. DOI: 10.1063/1.1697301.

[57] P. P. Kombrath, J. Kong, T. R. Furlani, and M. Head-Gordon. *Mol. Phys.*, 100:1755, 2002. DOI: 10.1080/00268970110109466.

[58] Y. Kumeda, D. J. Wales, and L. J. Munro. *Chem. Phys. Lett.*, 341:185, 2001. DOI: 10.1016/S0009-2614(01)00334-7.

[59] D. S. Lambrecht, G. N. I. Clark, T. Head-Gordon, and M. Head-Gordon. *J. Phys. Chem. A*, 115:5928, 2011. DOI: 10.1021/jp110334w.

[60] B. R. Landry and J. E. Subotnik. *J. Chem. Phys.*, 137:22A513, 2012. DOI: 10.1063/1.4733675.

[61] B. G. Levine, C. Ko, J. Quenneville, and T. J. Martínez. *Mol. Phys.*, 104:1039, 2006. DOI: 10.1080/00268970500417762.

[62] B. G. Levine, J. D. Coe, and T. J. Martinez. *J. Phys. Chem. B*, 112:405, 2008. DOI: 10.1021/jp0761618.

[63] K. A. Linton, T. G. Wright, and N. A. Besley. *Proc. Roy. Soc. London A*, 376:20170152, 2018. DOI: 10.1098/rsta.2017.0152.

[64] S. Maeda, K. Ohno, and K. Morokuma. *J. Chem. Theory Comput.*, 6:1538, 2010. DOI: 10.1021/ct1000268.

[65] G. J. Martyna, M. L. Klein, and M. Tuckerman. *J. Chem. Phys.*, 97:2635, 1992. DOI: 10.1063/1.463940.

[66] S. Matsika and P. Krause. *Annu. Rev. Phys. Chem.*, 62:621, 2011. DOI: 10.1146/annurev-physchem-032210-103450.

[67] H.-D. Meyer and W. H. Miller. *J. Chem. Phys.*, 70:3214, 1979. DOI: 10.1063/1.437910.

[68] G. Mills and H. H. Jónsson. *Phys. Rev. Lett.*, 72:1124, 1994. DOI: 10.1103/PhysRevLett.72.1124.

[69] B. A. Murtagh and R. W. H. Sargent. *Comput. J.*, 13:185, 1970. DOI: 10.1093/comjnl/13.2.185.

[70] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 2006.

[71] S. Nosé. *J. Chem. Phys.*, 81:511, 1984. DOI: 10.1063/1.447334.

[72] S. Nosé. *Prog. Theor. Phys. Supp.*, 103:1, 1991. DOI: 10.1143/PTPS.103.1.

[73] M. T. Ong, J. Leiding, H. Tao, A. M. Virshup, and T. J. Martínez. *J. Am. Chem. Soc.*, 131:6377, 2009. DOI: 10.1021/ja8095834.

[74] Q. Ou, S. Fatehi, E. Alguire, Y. Shao, and J. E. Subotnik. *J. Chem. Phys.*, 141:024114, 2014. DOI: 10.1063/1.4887256.

[75] Q. Ou, G. D. Bellchambers, F. Furche, and J. E. Subotnik. *J. Chem. Phys.*, 142:064114, 2015. DOI: 10.1063/1.4906941.

[76] B. Peters, A. Heyden, A. T. Bell, and A. Chakraborty. *J. Chem. Phys.*, 120:7877, 2004. DOI: 10.1063/1.1691018.

[77] J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley. *Int. J. Quantum Chem. Symp.*, 13:225, 1979. DOI: 10.1002/qua.560160825.

[78] D. Poppinger. *Chem. Phys. Lett.*, 35:550, 1975. DOI: 10.1016/0009-2614(75)85665-X.

[79] R. Porter. *Annu. Rev. Phys. Chem.*, 25:317, 1974. DOI: 10.1146/annurev.pc.25.100174.001533.

[80] R. Porter, L. Raff, and W. H. Miller. *J. Chem. Phys.*, 63:2214, 1975. DOI: 10.1063/1.431603.

[81] P. Pulay. *J. Comput. Chem.*, 3:556, 1982. DOI: 10.1002/jcc.540030413.

[82] P. Pulay and G. Fogarasi. *J. Chem. Phys.*, 96:2856, 1992. DOI: 10.1063/1.462844.

[83] P. Pulay and G. Fogarasi. *Chem. Phys. Lett.*, 386:272, 2004. DOI: 10.1016/j.cplett.2004.01.069.

[84] P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs. *J. Am. Chem. Soc.*, 101:2550, 1979. DOI: 10.1021/ja00504a009.

[85] E. Ramos-Cordoba, D. S. Lambrecht, and M. Head-Gordon. *Faraday Discuss.*, 150:345, 2011. DOI: 10.1039/C1FD00004G.

[86] J. Ribas-Arino, M. Shiga, and D. Marx. *Angew. Chem. Int. Ed. Engl.*, 48:4190, 2009. DOI: 10.1002/anie.200900673.

[87] M. Scheurer, A. Dreuw, E. Epifanovsky, M. Head-Gordon, and T. Stauch. *J. Chem. Theory Comput.*, 17:583, 2021. DOI: 10.1021/acs.jctc.0c01212.

[88] H. B. Schlegel. *Theor. Chem. Acc.*, 66:333, 1984. DOI: 10.1007/BF00554788.

[89] M. W. Schmidt, M. S. Gordon, and M. Dupuis. *J. Am. Chem. Soc.*, 107:2585, 1985. DOI: 10.1021/ja00295a002.

[90] D. F. Shanno. *Math. Comp.*, 24:647, 1970. DOI: 10.2307/2004840.

[91] Y. Shao, M. Head-Gordon, and A. I. Krylov. *J. Chem. Phys.*, 118:4807, 2003. DOI: 10.1063/1.1545679.

[92] S. M. Sharada, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon. *J. Chem. Theory Comput.*, 8:5166, 2012. DOI: 10.1021/ct300659d.

[93] S. M. Sharada, A. T. Bell, and M. Head-Gordon. *J. Chem. Phys.*, 140:164115, 2014. DOI: 10.1063/1.4871660.

[94] P. E. M. Siegbahn and F. Himo. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 1:323, 2011. DOI: 10.1002/wcms.13.

[95] J. Simons, P. Jørgensen, H. Taylor, and J. Ozment. *J. Phys. Chem.*, 87:2745, 1983. DOI: 10.1021/j100238a013.

[96] T. Stauch. *J. Chem. Phys.*, 153:134503, 2020. DOI: 10.1063/5.0024671.

[97] T. Stauch. *Int. J. Quantum Chem.*, 121:e26208, 2021. DOI: 10.1002/qua.26208.

[98] T. Stauch and A. Dreuw. *Chem. Rev.*, 116:14137, 2016. DOI: 10.1021/acs.chemrev.6b00458.

[99] T. Stauch, R. Chakraborty, and M. Head-Gordon. *ChemPhysChem*, 20:2742, 2019. DOI: 10.1002/cphc.201900853.

[100] R. P. Steele and J. C. Tully. *Chem. Phys. Lett.*, 500:167, 2010. DOI: 10.1016/j.cplett.2010.10.003.

[101] J. E. Subotnik. *J. Phys. Chem. A*, 114:12083, 2011. DOI: 10.1021/jp206557h.

[102] J. E. Subotnik and N. Shenvi. *J. Chem. Phys.*, 134:024105, 2011. DOI: 10.1063/1.3506779.

[103] J. J. Talbot, M. Head-Gordon, and S. J. Cotton. *Mol. Phys.*, 121:e2153761, 2023. DOI: 10.1080/00268976.2022.2153761.

[104] J. C. Tully. *J. Chem. Phys.*, 93:1061, 1990. DOI: 10.1063/1.459170.

[105] D. J. Wales and J. P. K. Doye. *J. Phys. Chem. A*, 5111:101, 1997. DOI: 10.1021/jp970984n.

[106] R. Weiß, F. Zeller, and T. Neudecker. *J. Chem. Phys.*, 160:084101, 2024. DOI: 10.1063/5.0189887.

[107] E. B. Wilson, J. C. Decius, and P. C. Cross. *Molecular Vibrations*. McGraw-Hill, New York, 1955.

[108] J. M. Wittbrodt and H. B. Schlegel. *J. Mol. Struct. (Theochem)*, 398:55, 1997. DOI: 10.1016/S0166-1280(96)04928-7.

[109] K. Wolinski and J. Baker. *Mol. Phys.*, 107:2403, 2009. DOI: 10.1080/00268970903321348.

[110] D. R. Yarkony. *J. Chem. Phys.*, 114:2601, 2001. DOI: 10.1063/1.1329644.

[111] F. Zeller, P. Pracht, and T. Neudecker. *J. Phys. Chem. A*, 129:2108, 2025. DOI: 10.1021/acs.jpca.4c08065.

[112] X. Zhang and J. M. Herbert. *J. Phys. Chem. B*, 118:7806, 2014. DOI: 10.1021/jp412092f.

[113] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 141:064104, 2014. DOI: 10.1063/1.4891984.

[114] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 142:064109, 2015. DOI: 10.1063/1.4907376.

[115] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 143:234107, 2015. DOI: 10.1063/1.4937571.

[116] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 155:124111, 2021. DOI: 10.1063/5.0062757.

Chapter 10

Molecular Properties and Analysis

10.1 Introduction

Q-CHEM has incorporated a number of molecular properties and wave function analysis tools:

- Population analysis for ground and excited states
- Multipole moments for ground and excited states
- Extended excited-state analysis using reduced density matrices
- Calculation of molecular intracules
- Vibrational analysis (including isotopic substitution)
- Interface to the Natural Bond Orbital (NBO) package
- Molecular orbital symmetries
- Orbital localization
- Localized orbital bonding analysis
- Data generation for one- or two-dimensional plots
- Orbital visualization using the MOLDEN and MACMOLPLT programs
- Natural transition orbitals for excited states
- NMR shielding tensors and chemical shifts
- Molecular junctions

In addition, Chapter 12 describes energy decomposition analysis using the fragment-based absolutely-localized molecular orbital approach.

10.2 Wave Function Analysis

10.2.1 Introduction

Q-CHEM performs some standard wave function analyses by default. Setting WAVEFUNCTION_ANALYSIS = FALSE in the *$rem* section will turn off all wave function analysis features, or alternatively these can be controlled individually as described in this section. The Natural Bond Orbital (NBO) analysis program is interfaced with Q-CHEM and can perform more sophisticated analyses; see Section 10.3.1 for more details.

WAVEFUNCTION_ANALYSIS

Controls the running of the default wave function analysis tasks.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Perform default wave function analysis.

FALSE Do not perform default wave function analysis.

RECOMMENDATION:

None. This option has no effect on NBO analysis.

10.2.2 Atomic Partial Charges

10.2.2.1 Mulliken and Löwdin charges

The one-electron charge density,

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}\, \phi_\mu(\mathbf{r})\, \phi_\nu(\mathbf{r}) \ , \tag{10.1}$$

represents the probability of finding an electron at the point $\mathbf{r}$, but implies little regarding the number of electrons associated with a given nucleus in a molecule. However, since the number of electrons N is related to the occupied orbitals ψ_i by

$$N = 2 \sum_{a}^{N/2} \left|\psi_a(\mathbf{r})\right|^2 \tag{10.2}$$

We can substitute the atomic orbital (AO) basis expansion of ψ_a into Eq. (10.2) to obtain

$$N = \sum_{\mu\nu} P_{\mu\nu} S_{\mu\nu} = \sum_{\mu} (\mathbf{PS})_{\mu\mu} = \mathrm{tr}(\mathbf{PS}) \tag{10.3}$$

where we interpret $(\mathbf{PS})_{\mu\mu}$ as the number of electrons associated with ϕ_μ. If the basis functions are atom-centered, the number of electrons associated with a given atom can be obtained by summing over all the basis functions. This leads to the Mulliken formula for the net charge on atom A:

$$q_A = Z_A - \sum_{\mu \in A} (\mathbf{PS})_{\mu\mu} \tag{10.4}$$

where Z_A is the atom's nuclear charge. This is called *Mulliken population analysis*,[207] and it is performed by default.

Although conceptually simple, Mulliken population analyses suffer from a strong dependence on the basis set used, as well as the possibility of producing unphysical negative numbers of electrons. An alternative is that of *Löwdin*

population analysis,[128] which uses the Löwdin symmetrically orthogonalized basis set (which is still atom-tagged) to assign the electron density. This shows a reduced basis set dependence, but maintains the same essential features.

While Mulliken and Löwdin population analyses are commonly employed, and can be used to produce information about changes in electron density and also localized spin polarizations, they should not be interpreted as oxidation states of the atoms in the system. For such information we would recommend a bonding analysis technique (LOBA or NBO).

POP_MULLIKEN

Controls running of Mulliken population analysis.

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE (or 1)

OPTIONS:

FALSE (or 0)	Do not calculate Mulliken populations.
TRUE (or 1)	Calculate Mulliken populations.
2	Also calculate shell populations for each occupied orbital.
3	Same output as 2 and also orbital densities at the nuclear centers.
−1	Calculate Mulliken charges for both the ground state and any CIS, RPA, or TDDFT excited states.

RECOMMENDATION:

Leave as TRUE, unless excited-state charges are desired. Mulliken analysis is a trivial additional calculation, for ground or excited states.

LOWDIN_POPULATION

Run Löwdin population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Do not calculate Löwdin populations.
TRUE	Run Löwdin population analysis.

RECOMMENDATION:

None

10.2.2.2 Charges derived from the electrostatic potential

A more stable alternative to Mulliken or Löwdin charges are charges derived from the electrostatic potential (ESP),[60] of which there are several different types. So-called "ChElPG" charges,[31] whose name is an acronym for "Charges from the Electrostatic Potential on a Grid", are perhaps the most conceptually straightforward of the various ESP-derived charge schemes. By definition, the ChElPG atomic charges are the ones that provide the best fit to the molecular electrostatic potential, evaluated on a real-space grid outside of the van der Waals region and subject to the constraint that the sum of the ChElPG charges must equal the molecular charge. Q-CHEM's implementation of the ChElPG algorithm differs slightly from the one originally algorithm described by Breneman and Wiberg,[31] in that Q-CHEM weights the grid points with a smoothing function to ensure that the ChElPG charges vary continuously as the nuclei are displaced.[86] (For any particular geometry, however, numerical values of the charges are quite similar to those obtained using the original algorithm.) Note also that the Breneman-Wiberg approach uses a Cartesian grid and becomes

expensive for large systems, especially when ChElPG charges are used in QM/MM-Ewald calculations.[91] For that reason, an alternative procedure based on atom-centered Lebedev grids is also available,[91] which provides very similar charges using far fewer grid points. In order to use the Lebedev grid implementation the *$rem* variables CHELPG_H and CHELPG_HA must be set, which specify the number of Lebedev grid points for the hydrogen atoms and the heavy atoms, respectively.

CHELPG

Controls the calculation of CHELPG charges.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate ChElPG charges.

TRUE Compute ChElPG charges.

RECOMMENDATION:

Set to TRUE if desired. For large molecules, there is some overhead associated with computing ChElPG charges, especially if the number of grid points is large. Note that POP_MULLIKEN must also be set to TRUE (which is the default value) in order to get ChElPG charges.

CHELPG_HEAD

Sets the "head space"[31] (radial extent) of the ChElPG grid.

TYPE:

INTEGER

DEFAULT:

28

OPTIONS:

N Corresponding to a head space of $N/10$, in Å.

RECOMMENDATION:

Use the default, which is the value recommended by Breneman and Wiberg.[31]

CHELPG_DX

Sets the rectangular grid spacing for the traditional Cartesian ChElPG grid or the spacing between concentric Lebedev shells (when the variables CHELPG_HA and CHELPG_H are specified as well).

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

N Corresponding to a grid space of $N/20$, in Å.

RECOMMENDATION:

Use the default, which corresponds to the "dense grid" of Breneman and Wiberg,[31], unless the cost is prohibitive, in which case a larger value can be selected. Note that this default value is set with the Cartesian grid in mind and not the Lebedev grid. In the Lebedev case, a larger value can typically be used.

CHELPG_HA

Sets the Lebedev grid to use for heavy (non-hydrogen) atoms.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Use *N* Lebedev points per atom.

RECOMMENDATION:

N must be one of the defined values for a Lebedev grid; see Table 5.2.

CHELPG_H

Sets the Lebedev grid to use for hydrogen atoms.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Use *N* Lebedev points per atom.

RECOMMENDATION:

N must be one of the defined values for a Lebedev grid; see Table 5.2. Furthermore, CHELPG_H must always be less than or equal to CHELPG_HA. If it is greater, it will automatically be set to the value of CHELPG_HA.

A closely-related set of ESP-derived charges are the so-called "Merz-Kollman" charges,[24,193] in which the atom-centered charges are fit to reproduce the ESP on a small number of concentric atomic spheres (or van der Waals surfaces of the molecule), and in this respect the Merz-Kollman algorithm is similar to Q-CHEM's Lebedev-based implementation of the ChElPG charges. Q-CHEM's algorithm for computing Merz-Kollman charges uses surfaces constructed from atomic spheres whose radii are 1.4×, 1.6×, 1.8×, and 2.0× the atomic van der Waals radii. Lebedev or spherical-harmonics grid points are placed on each surface with a 0.5 Å default spacing between these grid points. These charges can be restricted to satisfy "chemical symmetry", where chemically equivalent atoms have the same atomic charge value, leading to the so-called "RESP" charges.[48]

Note:

1. When both ESP_CHARGES and RESP_CHARGES are turned on, only the RESP charges will be calcualted
2. Both ESP_CHARGES and RESP_CHARGES can be used to compute the atomic charges of any singlet excited state from a CIS or TDDFT calculation (RPA or TDA). For excited-state popular analysis, it is recommended to turn on CIS_RELAXED_DENSITY. Physically, the external electrostatic environment should feel the relaxed excited state density not the unrelaxed density.

ESP_CHARGES

Controls the calculations of Merz-Kollman ESP-derived charges.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 Use Lebedev grid points around each atom.

2 Use spherical harmonics grid points around each atom.

RECOMMENDATION:

NONE

RESP_CHARGES

Controls the calculations of RESP charges, where chemically equivalent atoms are restricted to have the same atomic charge value.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 Use Lebedev grid points around each atom.

2 Use spherical harmonics grid points around each atom.

RECOMMENDATION:

NONE

ESP_SURFACE_DENSITY

Controls the spacing between grid points on vdW surfaces.

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Spacing of $0.001 \times n$ (in Å)

RECOMMENDATION:

The default corresponds to 0.5 Å spacing.

10.2.2.3 Hirshfeld charges

Hirshfeld population analysis[89] provides yet another definition of atomic charges in a molecule:

$$q_A = Z_A - \int d\mathbf{r} \left(\frac{\rho_A^0(\mathbf{r})}{\sum_B \rho_B^0(\mathbf{r})} \right) \rho(\mathbf{r}), \tag{10.5}$$

where Z_A is the nuclear charge of A, ρ_B^0 is the isolated ground-state atomic density of atom B, and ρ is the molecular density. The sum goes over all atoms in the molecule. Thus computing Hirshfeld charges requires a self-consistent calculation of the isolated atomic densities (the promolecule) as well as the total molecule. Prior to the SCF calculation, the Hirshfeld atomic density matrix is constructed. After SCF convergence, numerical quadrature is used to evaluate the integral in Eq. (10.5). Neutral ground-state atoms are used, as the choice of appropriate reference for a charged molecule is ambiguous (such jobs will crash). As numerical integration (with default quadrature grid) is used, charges

may not sum precisely to zero. A larger XC_GRID may be used to improve the accuracy of the integration, but the magnitude of the Hirshfeld charges should be largely independent of grid choice.

The charges (and corresponding molecular dipole moments) obtained using Hirshfeld charges are typically underestimated as compared to other charge schemes or experimental data. To correct this, Marenich *et al.* introduced "Charge Model 5" (CM5),[135] which employs a single set of parameters to map the Hirshfeld charges onto a more reasonable representation of the electrostatic potential. CM5 charges generally lead to more accurate dipole moments as compared to the original Hirshfeld charges, at negligible additional cost. CM5 is available for molecules composed of elements H–Ca, Zn, Ge–Br, and I.

The use of neutral ground-state atoms to define the promolecular density in Hirshfeld scheme has no strict theoretical basis and there is no unique way to construct the promolecular densities. For example, Li^0F^0, Li^+F^-, or Li^-F^+ could each be used to construct the promolecular densities for LiF. Furthermore, the choice of appropriate reference for a charged molecule is ambiguous, and for this reason Hirshfeld analysis is disabled in Q-CHEM for any molecule with a net charge. A solution for charged molecules is to use the iterative "Hirshfeld-I" partitioning scheme proposed by Bultinck *et al.*,[34,215] in which the reference state is not predefined but rather determined self-consistently, thus eliminating the arbitrariness. The final self-consistent reference state for Hirshfeld-I partitioning usually consists of non-integer atomic populations.

In the first iteration, the Hirshfeld-I method uses neutral atomic densities (as in the original Hirshfeld scheme), $\rho_i^0(\mathbf{r})$ with electronic population $N_i^0 = \int d\mathbf{r}\, \rho_i^0(\mathbf{r}) = Z_i$. This affords charges

$$q_i^1 = Z_i - \int d\mathbf{r} \left(\frac{\rho_i^0(\mathbf{r})}{\sum_i^A \rho_i^0(\mathbf{r})} \right) \rho(\mathbf{r}) = Z_i - N_i^1 \tag{10.6}$$

on the first iteration. The new electronic population (number of electrons) for atom i is N_i^1, and is derived from the promolecular populations N_i^0. One then computes new isolated atomic densities with $N_i^1 = \int d\mathbf{r}\, \rho_i^1(\mathbf{r}_1)$ and uses them to construct the promolecular densities in the next iteration. In general, the new weighting function for atom i in the kth iteration is

$$w_{i,\mathrm{HI}}^k(\mathbf{r}) = \frac{\rho_i^{k-1}(\mathbf{r})}{\sum\limits_{i \in A} \rho_i^{k-1}(\mathbf{r})} \,. \tag{10.7}$$

The atomic densities $\rho_i^k(\mathbf{r})$ with corresponding fractional electron numbers N_i^k are obtained by linear interpolation between $\rho_i^{0,\lfloor N_i^k \rfloor}(\mathbf{r})$ and $\rho_i^{0,\lceil N_i^k \rceil}(\mathbf{r})$ of the same atom:[34,57]

$$\rho_i^k(\mathbf{r}) = \left(\lceil N_i^k \rceil - N_i^k \right) \rho_i^{0,\lfloor N_i^k \rfloor}(\mathbf{r}) + \left(N_i^k - \lfloor N_i^k \rfloor \right) \rho_i^{0,\lceil N_i^k \rceil}(\mathbf{r}) \,, \tag{10.8}$$

where $\lfloor N_i^k \rfloor$ and $\lceil N_i^k \rceil$ denote the integers that bracket N_i^k. The two atomic densities on the right side of Eq. (10.8) are obtained from densities $\rho_i^{0,Z_A-2}, \rho_i^{0,Z_A-1}, \ldots, \rho_i^{0,Z_A+2}$ that are computed in advance. (That is, the method uses the neutral atomic density along with the densities for the singly- and doubly-charged cations and anions of the element in equation.) The Hirshfeld-I iterations are converged once the atomic populations change insignificantly between iterations, say $|N_i^k - N_i^{k-1}| < 0.0005e$.[34,199]

The iterative Hirshfeld scheme generally affords more reasonable charges as compared to the original Hirshfeld scheme. In LiF, for example, the original Hirshfeld scheme predicts atomic charges of ± 0.57 while the iterative scheme increases these charges to ± 0.93. The integral in Eq. (10.6) is evaluated by numerical quadrature, and the cost of each iteration of Hirshfeld-I is equal to the cost of computing the original Hirshfeld charges. The *$rem* variable POINT_GROUP_SYMMETRY must be set to FALSE for Hirshfeld-I analysis.

HIRSHFELD

Controls running of Hirshfeld population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate Hirshfeld populations.

FALSE Do not calculate Hirshfeld populations.

RECOMMENDATION:

None

HIRSHFELD_READ

Switch to force reading in of isolated atomic densities.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Read in isolated atomic densities from previous Hirshfeld calculation from disk.

FALSE Generate new isolated atomic densities.

RECOMMENDATION:

Use the default unless system is large. Note, atoms should be in the same order with same basis set used as in the previous Hirshfeld calculation (although coordinates can change). The previous calculation should be run with the -save switch.

HIRSHFELD_SPHAVG

Controls whether atomic densities should be spherically averaged in pro-molecule.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Spherically average atomic densities.

FALSE Do not spherically average.

RECOMMENDATION:

Use the default.

CM5

Controls running of CM5 population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate CM5 populations.

FALSE Do not calculate CM5 populations.

RECOMMENDATION:

None

HIRSHITER

Controls running of iterative Hirshfeld population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate iterative Hirshfeld populations.

FALSE Do not calculate iterative Hirshfeld populations.

RECOMMENDATION:

None

HIRSHITER_THRESH

Controls the convergence criterion of iterative Hirshfeld population analysis.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

N Corresponding to the convergence criterion of $N/10000$, in e.

RECOMMENDATION:

Use the default, which is the value recommended in Ref. 34

Example 10.1 Iterative Hirshfeld population analysis for $F^-(H_2O)$

```
$molecule
 -1  1
  O     1.197566    -0.108087     0.000000
  H     1.415397     0.827014     0.000000
  H     0.134830    -0.084378     0.000000
  F    -1.236389     0.012239     0.000000
$end

$rem
point_group_symmetry False
   METHOD      B3LYP
   BASIS       6-31G*
   HIRSHITER   true
$end
```

10.2.3 Multipole Moments

This section discusses how to compute arbitrary electrostatic multipole moments for an entire molecule, including both ground- and excited-state electron densities. Occasionally, however, it is useful to decompose the electronic part of the multipole moments into contributions from individual MOs. This decomposition is especially useful for systems containing unpaired electrons,[226] where the first-order moments $\langle x \rangle$, $\langle y \rangle$, and $\langle z \rangle$ characterize the centroid (mean position) of the half-filled MO, and the second-order moments determine its radius of gyration, R_g, which characterizes the size of the MO. Upon setting PRINT_RADII_GYRE = TRUE, Q-CHEM will print out centroids and radii of gyration for each occupied MO and for the overall electron density of the Hartree-Fock or Kohn-Sham reference determinant. If CIS or TDDFT excited states are requested, then this keyword will also print out the centroids and radii of gyration for each excited-state electron density.

Note: These keywords only apply to SCF, DFT, CIS, and TDDFT calculations. To compute these quantities for correlated wavefunctions, use keywords that specify properties calculations for the corresponding method, as described in Chapters 6 and 7. For example, to compute CCSD or EOM-CCSD multipole moments and other properties, use CC_REF_PROP = TRUE and CC_EOM_PROP = TRUE.

PRINT_RADII_GYRE

Controls printing of MO centroids and radii of gyration.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

TRUE (or 1)	Print the centroid and radius of gyration for each occupied MO and each density.
2	Print centroids and radii of gyration for the virtual MOs as well.
FALSE (or 0)	Do not calculate these quantities.

RECOMMENDATION:

None

Q-CHEM can compute Cartesian multipole moments of the charge density to arbitrary order, both for the ground state and for excited states calculated using the CIS or TDDFT methods.

MULTIPOLE_ORDER

Determines highest order of multipole moments to print if wave function analysis requested.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n	Calculate moments to nth order.

RECOMMENDATION:

Use the default unless higher multipoles are required.

CIS_MOMENTS

Controls calculation of excited-state (CIS or TDDFT) multipole moments

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Do not calculate excited-state moments.
TRUE	Calculate moments for each excited state.

RECOMMENDATION:

Set to TRUE if excited-state moments are desired. (This is a trivial additional calculation.) The MULTIPOLE_ORDER controls how many multipole moments are printed.

10.2.4 Population of Effectively Unpaired Electrons

In a stretched hydrogen molecule the two electrons that are paired at equilibrium forming a bond become un-paired and localized on the individual H atoms. In singlet diradicals or doublet triradicals such a weak paring exists even at

equilibrium. At a single-determinant SCF level of the theory the valence electrons of a singlet system like H_2 remain perfectly paired, and one needs to include non-dynamical correlation to decouple the bond electron pair, giving rise to a population of effectively-unpaired ("odd", radicalized) electrons.[27,198,209] When the static correlation is strong, these electrons remain mostly unpaired and can be described as being localized on individual atoms.

These phenomena can be properly described within wave-function formalism. Within DFT, these effects can be described by broken-symmetry approach or by using SF-TDDFT (see Section 7.3.1). Below we describe how to derive this sort of information from pure DFT description of such low-spin open-shell systems without relying on spin-contaminated solutions.

The first-order reduced density matrix (1-RDM) corresponding to a single-determinant wave function (*e.g.*, SCF or Kohn-Sham DFT) is idempotent:

$$\begin{aligned} \rho_\sigma(\mathbf{r}_1) &= \int \gamma_\sigma^{\mathrm{SCF}}(1;2)\, \gamma_\sigma^{\mathrm{SCF}}(2;1)\, d\mathbf{r}_2 \\ \gamma_\sigma^{\mathrm{SCF}}(1;2) &= \sum_i^{\mathrm{occ}} \psi_{i\sigma}^{\mathrm{KS}}(1)\psi_{i\sigma}^{\mathrm{KS}}(2)\ , \end{aligned} \tag{10.9}$$

where $\rho_\sigma(1)$ is the electron density of spin σ at position $\mathbf{r}_1$, and $\gamma_\sigma^{\mathrm{SCF}}$ is the spin-resolved 1-RDM of a single Slater determinant. The cross product $\gamma_\sigma^{\mathrm{SCF}}(1;2)\, \gamma_\sigma^{\mathrm{SCF}}(2;1)$ reflects the Hartree-Fock exchange (or Kohn-Sham exact-exchange) governed by the HF exchange hole:

$$\begin{aligned} \gamma_\sigma^{\mathrm{SCF}}(1;2)\, \gamma_\sigma^{\mathrm{SCF}}(2;1) &= \rho_\alpha(1) h_{\mathrm{X}\sigma\sigma}(1,2) \\ \int h_{\mathrm{X}\sigma\sigma}(1,2)\, d\mathbf{r}_2 &= 1\ . \end{aligned} \tag{10.10}$$

When 1-RDM includes electron correlation, it becomes non-idempotent:

$$D_\sigma(1) \equiv \rho_\sigma(1) - \int \gamma_\sigma(1;2)\gamma_\sigma(2;1)\, d\mathbf{r}_2 \geq 0\,. \tag{10.11}$$

The function $D_\sigma(1)$ measures the deviation from idempotency of the correlated 1-RDM and yields the density of effectively-unpaired (odd) electrons of spin σ at point $\mathbf{r}_1$.[175,209] The formation of effectively-unpaired electrons in singlet systems is therefore exclusively a correlation based phenomenon. Summing $D_\sigma(1)$ over the spin components gives the total density of odd electrons, and integrating the latter over space gives the mean total number of odd electrons $\bar{N}_u$:

$$D_u(1) = 2\sum_\sigma D_\sigma(1) d\mathbf{r}_1,\ \ \bar{N}_u = \int D_u(1) d\mathbf{r}_1\,. \tag{10.12}$$

The appearance of a factor of 2 in Eq. (10.12) above is required for reasons discussed in Ref. 175. In Kohn-Sham DFT, the SCF 1-RDM is always idempotent which impedes the analysis of odd electron formation at that level of the theory. Ref. 178 has proposed a remedy to this situation. It was noted that the correlated 1-RDM cross product entering Eq. (10.11) reflects an effective exchange, also known as cumulant exchange.[27] The KS exact-exchange hole is itself artificially too delocalized. However, the total exchange-correlation interaction in a finite system with strong left-right (*i.e.*, static) correlation is normally fairly localized, largely confined within a region of roughly atomic size.[20] The effective exchange described with the correlated 1-RDM cross product should be fairly localized as well. With this in mind, the following form of the correlated 1-RDM cross product was proposed:[178]

$$\gamma_\sigma(1;2)\, \gamma_\sigma(2;1) = \rho_\sigma(1)\, \bar{h}_{\mathrm{X}\sigma\sigma}^{\mathrm{eff}}(1,2)\ . \tag{10.13}$$

The function $\bar{h}_{\mathrm{X}\sigma\sigma}^{\mathrm{eff}}(1;2)$ is a model DFT exchange hole of Becke-Roussel (BR) form used in Becke's B05 method.[21] The latter describes left-right static correlation effects in terms of certain effective exchange-correlation hole.[21] The extra delocalization of the HF exchange hole alone is compensated by certain physically motivated real-space corrections to it:[21]

$$\bar{h}_{\mathrm{XC}\alpha\alpha}(1,2) = \bar{h}_{\mathrm{X}\alpha\alpha}^{\mathrm{eff}}(1,2) + f_{\mathrm{c}}(1)\, \bar{h}_{\mathrm{X}\beta\beta}^{\mathrm{eff}}(1,2)\ . \tag{10.14}$$

The BR exchange hole $\bar{h}^{\text{eff}}_{X\sigma\sigma}$ is used in B05 as an auxiliary function, such that the potential from the relaxed BR hole equals that of the exact-exchange hole. This results in relaxed normalization of the auxiliary BR hole less than or equal to unity:

$$\int \bar{h}^{\text{eff}}_{X\sigma\sigma}(1;2)\, d\mathbf{r}_2 = N^{\text{eff}}_{X\sigma}(1) \leq 1\,. \tag{10.15}$$

The expression of the relaxed normalization $N^{\text{eff}}_{X\sigma}(\mathbf{r})$ is quite complicated, but it is possible to represent it in closed analytic form.[176,177] The smaller the relaxed normalization $N^{\text{eff}}_{X\alpha}(1)$, the more delocalized the corresponding exact-exchange hole.[21] The α–α exchange hole is further deepened by a fraction of the β–β exchange hole, $f_{\text{c}}(1)\,\bar{h}^{\text{eff}}_{X\beta\beta}(1,2)$, which gives rise to left-right static correlation. The local correlation factor f_{c} in Eq.(10.14) governs this deepening and hence the strength of the static correlation at each point:[21]

$$f_{\text{c}}(\mathbf{r}) = \min\big(f_\alpha(\mathbf{r}),\ f_\beta(\mathbf{r}),\ 1\big) \tag{10.16a}$$

$$0 \leq f_{\text{c}}(\mathbf{r}) \leq 1 \tag{10.16b}$$

$$f_\alpha(\mathbf{r}) = \frac{1 - N^{\text{eff}}_{X\alpha}(\mathbf{r})}{N^{\text{eff}}_{X\beta}(\mathbf{r})}\,. \tag{10.16c}$$

Using Eqs. (10.16), (10.12), and (10.13), the density of odd electrons becomes:

$$\begin{aligned} D_\alpha(1) &= \rho_\alpha(1)(1 - N^{\text{eff}}_{X\alpha}(1)) \\ &= \rho_\alpha(1) f_{\text{c}}(1)\, N^{\text{eff}}_{X\beta}(1)\,. \end{aligned} \tag{10.17}$$

The final formulas for the spin-summed odd electron density and the total mean number of odd electrons read:

$$\begin{aligned} D_u(1) &= 4a^{\text{op}}_{\text{nd}}\, f_{\text{c}}(1)\left[\rho_\alpha(1)\, N^{\text{eff}}_{X\beta}(1) + \rho_\beta(1)\, N^{\text{eff}}_{X\alpha}(1)\right] \\ \bar{N}_u &= \int D_u(\mathbf{r_1})\, d\mathbf{r_1}\,. \end{aligned} \tag{10.18}$$

Here $a^{\text{nd-opp}}_{\text{c}} = 0.526$ is the SCF-optimized linear coefficient of the opposite-spin static correlation energy term of the B05 functional.[21,177]

It is informative to decompose the total mean number of odd electrons into atomic contributions. Partitioning in real space the mean total number of odd electrons $\bar{N}_u$ as a sum of atomic contributions, we obtain the atomic population of odd electrons (F^{r}_A) as:

$$F^{\text{r}}_A = \int_{\Omega_A} D_u(\mathbf{r_1})\, d\mathbf{r}_1\,. \tag{10.19}$$

Here Ω_A is a subregion assigned to atom A in the system. To define these atomic regions in a simple way, we use the partitioning of the grid space into atomic subgroups within Becke's grid-integration scheme.[19] Since the present method does not require symmetry breaking, singlet states are calculated in restricted Kohn-Sham (RKS) manner even at strongly stretched bonds. This way one avoids the destructive effects that the spin contamination has on F^{r}_A and on the Kohn-Sham orbitals. The calculation of F^{r}_A can be done fully self-consistently only with the RI-B05 and RI-mB05 functionals. In these cases no special keywords are needed, just the corresponding choice of EXCHANGE in the *$rem* section. Atomic population of odd electron can be estimated also with any other functional in two steps: first obtaining a converged SCF calculation with the chosen functional, then performing one single post-SCF iteration with RI-B05 or RI-mB05 functionals reading the guess from a preceding calculation, as shown on the input example below:

Example 10.10.2 To calculate the odd-electron atomic population and the correlated bond order in stretched H_2, with B3LYP/RI-mB05, and with fully SCF RI-mB05

View input online

Once the atomic population of odd electrons is obtained, a calculation of the corresponding correlated bond order of Mayer's type follows in the code, using certain exact relationships between F^{r}_A, F^{r}_B, and the correlated bond order of Mayer type B_{AB}. Both new properties are printed at the end of the output, right after the multipoles section. It is useful to compare the correlated bond order with Mayer's SCF bond order. To print the latter, use SCF_FINAL_PRINT = 1.

10.2.5 Symmetry Decomposition

Q-CHEM's default is to write the SCF wave function molecular orbital symmetries and energies to the output file. If requested, a symmetry decomposition of the kinetic and nuclear attraction energies can also be calculated.

SYMMETRY_DECOMPOSITION

Determines symmetry decompositions to calculate.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 No symmetry decomposition.

1 Calculate MO eigenvalues and symmetry (if available).

2 Perform symmetry decomposition of kinetic energy and nuclear attraction matrices.

RECOMMENDATION:

None

10.2.6 Localized Orbital Bonding Analysis

Localized orbital bonding analysis (LOBA) is a technique developed by Dr. Alex Thom and Eric Sundstrom at Berkeley with Prof. Martin Head-Gordon.[212] Inspired by the work of Rhee and Head-Gordon,[183] it makes use of the fact that the post-SCF localized occupied orbitals of a system provide a large amount of information about the bonding in the system.

While the canonical molecular orbitals can provide information about local reactivity and ionization energies, their delocalized nature makes them rather uninformative when looking at the bonding in larger molecules. Localized orbitals in contrast provide a convenient way to visualize and account for electrons. Transformations of the orbitals within the occupied subspace do not alter the resultant density; if a density can be represented as orbitals localized on individual atoms, then those orbitals can be regarded as non-bonding. If a maximally localized set of orbitals still requires some to be delocalized between atoms, these can be regarded as bonding electrons. A simple example is that of He_2 versus H_2. In the former, the delocalized σ_g and σ_u canonical orbitals may be transformed into 1s orbitals on each He atom, and there is no bond between them. This is not possible for the H_2 molecule, and so we can regard there being a bond between the atoms. In cases of multiple bonding, multiple delocalized orbitals are required.

While there are no absolute definitions of bonding and oxidation state, it has been shown that the localized orbitals match the chemically intuitive notions of core, non-bonded, single- and double-bonded electrons, *etc.* By combining these localized orbitals with population analyses, LOBA allows the nature of the bonding within a molecule to be quickly determined.

In addition, it has been found that by counting localized electrons, the oxidation states of transition metals can be easily found. Owing to polarization caused by ligands, an upper threshold is applied, populations above which are regarded as "localized" on an atom, which has been calibrated to a range of transition metals, recovering standard oxidation states ranging from $-$II to VII.

LOBA

Specifies the methods to use for LOBA

TYPE:

INTEGER

DEFAULT:

00

OPTIONS:

ab

a specifies the localization method

0 Perform Boys localization.

1 Perform PM localization.

2 Perform ER localization.

b specifies the population analysis method

0 Do not perform LOBA. This is the default.

1 Use Mulliken population analysis.

2 Use Löwdin population analysis.

RECOMMENDATION:

Boys Localization is the fastest. ER will require an auxiliary basis set.

LOBA 12 provides a reasonable speed/accuracy compromise.

LOBA_THRESH

Specifies the thresholds to use for LOBA

TYPE:

INTEGER

DEFAULT:

6015

OPTIONS:

$aabb$ aa specifies the threshold to use for localization

bb specifies the threshold to use for occupation

Both are given as percentages.

RECOMMENDATION:

Decrease bb to see the smaller contributions to orbitals. Values of aa between 40 and 75 have been shown to given meaningful results.

On a technical note, LOBA can function of both restricted and unrestricted SCF calculations. The figures printed in the bonding analysis count the number of electrons on each atom from that orbital (*i.e.*, up to 1 for unrestricted or singly occupied restricted orbitals, and up to 2 for double occupied restricted.)

10.2.7 Oxidation State Localized Orbitals

Oxidation State Localized Orbitals (OSLO) is a new localization scheme focused on molecular fragments for the purposes of oxidation state assignment.[68] The method has been developed to avoid some pitfalls encountered in the LOBA method (10.2.6) where the Pipek-Mezey orbitals can spread across many fragments without reaching the 60% threshold[67]. OSLO starts by looking at fragments' centers of charge and localizing on the radial spread (in the real space) to those centers. Then, it admits orbitals to its list of orbitals if they are above a slowly increasing threshold of a criterion called the *fragment orbital localization index* (FOLI). To understand this criterion, one needs to know Pipek's delocalization measure. This is defined as

$$d_i = \left[\sum_A (N_A^i)^2 \right]^{-1} \tag{10.20}$$

where the N_A^i is the population of the ith orbital on center A and the summation runs over all centers. When an orbital is entirely localized on a given center, its delocalization measure is $d_i = 1$. If the orbital is perfectly delocalized among two centers A and B then $N_A = N_B = 1/2$ and the delocalization measure is $d_i = 2$, etc. The square helps make it less sensitive to the ratio of the population to each other (insensitive to bond polarization) compared to how many are related. In fact, minimizing the sum over all occupied orbitals is precisely what leads to the Pipek-Mezey localization procedure.[168] However, for our purposes, out of the localized orbitals generated from fragment A with low delocalization measure, we are interested in those that are also highly localized on fragment A. Defining the *fragment orbital localization index* (FOLI),

$$d_i^A = \left(\frac{d_i}{N_A} \right)^{1/2} , \tag{10.21}$$

it is easy to see d_i^A tends to unity for fragment A when orbital i is perfectly localized on A ($d_i = N_A = 1$), tends to 2 when the localized orbital is perfectly delocalized over two fragments ($d_i = 2, N_A = 0.5$), and gradually increases as the orbital becomes more delocalized as well as less centered on fragment A. Thus, among the redundant set of Mn_{occ} localized orbitals, one selects all orbitals above the smallest FOLI by a threshold (0.01 by default) and assigns each orbital to the originator fragment. The orbitals are then symmetrically orthogonalized and projected out from the remaining space of (unassigned) occupied space. In case a set of orbitals is redundant due to symmetry in the system or simple covalency, then the orbitals are split over all contributing fragments (originators). After the iterative process, each molecular fragment has an associated set of localized orbitals derived from the simplest orbital spread criterion, which in turn determines the fragment's formal charge or oxidation state in a natural manner.

The new method expects fragments. Otherwise, it localizes on atomic centers instead. Although this method was developed for oxidation state, it produces a set of localized orbitals on fragments or atoms that can be used like any other localization method.

OSLO

Triggers OSLO procedure after a converged SCF

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Don't perform OSLO

1 Perform the OSLO procedure

RECOMMENDATION:

None

More precise control of OSLO goes under the *$loco* input section, with keywords that are introduced below. See the example for reference.

OSLO_POPANALTYPE

Specifies which population analysis is used for OSLOs.

INPUT SECTION: *$loco*

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 Mulliken

1 Lowdin

2 IAOs

RECOMMENDATION:

Use IAOs since they are insenstive to the basis set. When using IAOs, use AUTOSAD = TRUE in the *$loco* input section to use the new AUTOSAD-IAO algorithm, which uses AUTOSAD atomic densities instead of some minimal basis.

OSLO_THRESH

Specifies the threshold for FOLI when adding new orbitals for each iteration.

INPUT SECTION: *$loco*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use the default, 0.01

c Use threshold of c/1000

RECOMMENDATION:

Use the default

OSLO_BRANCHING

Specifies which iteration with too much overlap between its orbitals and next sets orbitals to choose the next set. The input here can be $abcdf\ldots$, which is a binary string for which iteration (only consider those with warning prints) to branch in, *e.g.*, 010011, means to branch in iterations 2, 5, and 6.

INPUT SECTION: *$loco*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$abcd\ldots$, for each digit:

0 Follow the default behavior

1 Favor the next set of orbitals over the current set

RECOMMENDATION:

Use the default, and only worry about this if the system involves multiple Lewis structures

Example 10.3 An OSLO job to analyze the oxidation state (or formal charges) of the water molecule, H_2O

```
$molecule
0 1
--
0 2
H  0.7493679  0.000000   0.44243272
--
0 3
O  0.0000000  0.000000  -0.16535063
--
0 2
H -0.7493679  0.000000   0.44243272
$end

$rem
jobtype        sp
method         wB97X-V
basis          def2-TZVP
ecp            def2-ecp
mem_total      64000
mem_static     4000
scf_algorithm DIIS
scf_convergence 9
thresh          14
max_scf_cycles  1000
oslo            6
point_group_symmetry false
integral_symmetry    false
$end

$loco
autosad 1
ibo_mem 500
print_level 0
oslo_popanaltype 2
$end
```

10.2.8 Intrinsic Atomic Orbitals

Due to the form of commonly used basis sets, SCF wave functions $|\Phi\rangle$ are often not easily interpretable in terms of their atomic orbital (AO) basis functions. Especially for large basis sets it is not possible to achieve an unambiguous association of a basis function's contribution to the wave function with the atom it is centered on. An expansion of the molecular orbitals over a minimal basis (of free-atom AOs), on the other hand, does allow for a simple interpretation of the wave function. However, free-atom atomic orbitals cannot yield a qualitatively correct minimal basis expansion as these AOs do not account for the polarization due to the molecular environment. The approach by Knizia overcomes this drawback by determining a set of *polarized* AOs $\{|\rho\rangle\}$, termed intrinsic atomic orbitals (IAOs)[113], which exactly express the occupied space $\mathcal{O} = \sum_i |i\rangle\langle i|$ of the wave function $|\Phi\rangle$. As a consequence, atomic properties, such as partial charges, become accessible through IAOs. The IAO-derived atomic charges of a closed-shell system are then given by

$$q_A = Z_A - \sum_{\rho \in A} \langle \rho | \gamma | \rho \rangle, \tag{10.22}$$

where Z_A is the nuclear charge of atom A and γ represents the closed-shell SCF density matrix $\gamma = 2\sum_i |i\rangle\langle i|$.

The resulting IAOs in conjunction with an orbital localization scheme can be used to construct bonding orbitals, termed

intrinsic bond orbitals (IBOs). The applied localization scheme is similar to the Pipek-Mezey approach[168] and effectively minimizes the number of atoms on which an orbital is centered by maximizing the functional

$$L = \sum_{A}^{\text{atoms}} \sum_{i'}^{\text{occ}} [n_A(i')]^4, \tag{10.23}$$

with respect to unitary orbital rotations. In the above equation, $n_A(i')$ is the number of electrons from rotated orbital $|i'\rangle$ located on the IAOs of atom A:

$$n_A(i') = 2 \sum_{\rho \in A} \langle i'|\rho\rangle\langle\rho|i'\rangle. \tag{10.24}$$

Intrinsic bond orbitals exactly represent the occupied molecular orbitals and can thus be used to interpret the electronic molecular structure.

Calculation of intrinsic bond orbitals is controlled by the following *$rem* variable:

DO_IBO

Enables IBO procedure

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate IBOs

TRUE Run the IBO procedure

RECOMMENDATION:

None

Additional control parameters for the IBO procedure can be declared in the *$loco* input section and are listed below.

MIN_BASIS

Specifies the free atom basis used for the construction of IAOs.

INPUT SECTION: *$loco*

TYPE:

STRING

DEFAULT:

MINAO

OPTIONS:

MINAO Use truncated cc-pVTZ basis (same as in original publication)

STO-3G Use STO-3G basis

STO-6G Use STO-6G basis

autoSAD Construct minimal basis from autoSAD procedure.

RECOMMENDATION:

Use autoSAD procedure.

EXP_FOUR

Whether to use an exponent of $p = 4$ in the localization functional (Eq. 10.23).

INPUT SECTION: *$loco*

TYPE:

BOOL

DEFAULT:

TRUE

OPTIONS:

TRUE Use exponent $p = 4$ (same as in original publication)

FALSE Use exponent $p = 2$

RECOMMENDATION:

An exponent of $p = 4$ is preferred as it leads to discrete localizations in aromatic systems ($p = 2$ does not).

IBO_POP_THRESH

Threshold for printing IBO occupations.

INPUT SECTION: *$loco*

TYPE:

FLOAT

DEFAULT:

0.1

OPTIONS:

0.1 Prints IBO occupations $> 10\%$

RECOMMENDATION:

Use the default.

IBO_MEM

Memory (in MB) for IBO procedure

INPUT SECTION: *$loco*

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

RECOMMENDATION:

500MB should suffice for most purposes. Increase only when needed.

EXPORT_MOLDEN

Whether to write IBOs in Molden format.

INPUT SECTION: *$loco*

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

TRUE Export IBOs in Molden format (`ibo.molden`).

FALSE Do not create Molden file.

RECOMMENDATION:

None

Example 10.4 Input for calculating atomic partial charges and bond populations via the IAO/IBO procedure.

```
$molecule
0 1
O        0.0000000000     0.0000000000     0.1172309766
H       -0.7626482594     0.0000000000    -0.4685977726
H        0.7626482594     0.0000000000    -0.4685977726
$end

$rem
method          = wB97M-V
basis           = def2-TZVP
mem_total       = 4000
mem_static      = 500
scf_convergence = 9
thresh          = 14
gui             = 2
do_ibo          = true
$end


$loco
min_basis     = autosad
export_molden = true
$end
```

10.2.9 Broken Bond Orbitals

Broken bond orbitals (BBOs) are a set of complete active space (CAS) orbitals developed by Sterling *et al*[200] to investigate the importance of orbital contraction on the formation of chemical bonds. BBOs are an set of uncontracted orbitals that span the space of isolated atoms/fragments that are then used to evaluate properties of a bonded system, effectively prohibiting orbital contraction effects (and some other orbital relaxation effects). As a result, the total (and kinetic) energy of the system can be decomposed into a sum of bonding effects that result from the total energy lowering that is possible using unrelaxed orbitals (ΔE_{BBO}), and the further energy lowering that accompanies orbital relaxation (ΔE_{rlx}):

$$\Delta E(r) = \Delta E_{\mathrm{BBO}}(r) + \Delta E_{\mathrm{rlx}}(r) \tag{10.25}$$

BBOs are generated according to the following procedure: (1) Obtain a set of localized orbitals $\{\phi\}$ at the CASSCF (complete active space self-consistent field) level by choosing a bond length at which the bond is unambiguously broken

– chosen to be 10 Å for uncharged fragments, and 1000 Å for charged fragments; (2) These BBOs are then used as a basis for a subsequent CASCI (complete active space configuration interaction) calculation, by translating the BBOs to the desired bond length. Since these initial BBOs are no longer orthogonal upon translation, a judiciously-chosen orthogonalization procedure was chosen that first symmetrically orthogonalizes all inactive (core) orbitals:

$$\tilde{\mathbf{C}}_{\mathrm{c}} = \mathbf{C}_{\mathrm{c}}\mathbf{S}_{\mathrm{c}}^{-\frac{1}{2}} \tag{10.26}$$

followed by projection of the orthogonalized core out of the active (valence) orbitals, and subsequent renormalization and symmetric orthogonalization of these projected valence orbitals:

$$\mathbf{C}_{\mathrm{v,proj}} = (\mathbf{I} - \tilde{\mathbf{P}}_{\mathrm{c}}\mathbf{S}_{\mu\nu})\mathbf{C}_{\mathrm{v}} \tag{10.27}$$

where $\mathbf{C}$ is the coefficient matrix, $\mathbf{S}$ is the overlap matrix, $\mathbf{P} = \mathbf{C}\mathbf{C}^{\mathbf{T}}$, subscripted c, v denote core and valence spaces, respectively, and μ, ν, denote atomic orbitals.

This approach ensures qualitatively correct bond dissociation, and enables evaluation of total and kinetic energy contributions to bond formation in the absence of orbital relaxation.

To run a BBO calculation, link a pair of CAS jobs, where the first job uses a CASSCF procedure to generate the set of BBOs, followed by a CASCI job that reads in the generated BBOs and calculates the energy at the desired bond length. Ensure that the SCF procedure in the second job is skipped to avoid unwanted modification of the BBOs. These BBOs can be visualized by specifying GUI = 2 in the *$rem* block, and their natural occupation numbers and kinetic energies are printed in the output file by default.

Example 10.5 Input for a BBO calculation on Li_2.

```
$molecule
0 1
Li
Li 1 R

R = 10.0  ! choose a long bond length to generate BBOs
$end

$rem
  JOBTYPE                   sp
  gen_scfman                TRUE
  EXCHANGE                  HF
  BASIS                     cc-pVDZ
  SCF_GUESS                 SAD
  SCF_ALGORITHM             GDM
  MAX_SCF_CYCLES            250
  CAS_METHOD                2     !1 for CAS-CI, 2 for CASSCF
  CAS_M_S                   0     !M_s value*2
  CAS_N_ELEC                2     !N_elec
  CAS_N_ORB                 2     !N_orb
  CAS_N_ROOTS               1     !N_roots
  CAS_SOLVER                0     !2=ASCI, 1=Olsen, 0=naive
  INTEGRAL_SYMMETRY         false
  POINT_GROUP_SYMMETRY      false
$end
```

10.2.10 Atomic dipoles and quadrupoles

It can sometimes be desirable to go beyond atomic partial charges to calculate atomic dipoles and quadrupoles. This is useful in particular for certain types of force fields, which use these terms to more accurately model charge anisotropy. QChem provides the ability to calculate atomic dipoles and quadrupoles, using a method inspired by the formulations of Ref.[196]. We simply evaluate atomic contributions to the multipole moment by assigning contributions from each atom's basis functions. However, we avert any basis set dependence and delocalization errors by using intrinsic atomic orbitals,[113] which have been shown to have excellent convergence with basis set.[11] The atomic quadrupoles available from this method are the traceless quadrupoles, which are the more commonly used type in force field contexts.

We calculate atomic dipoles and (traceless) quadrupoles using the following equations:

$$\mu_A^u = \sum_{\alpha \text{ on } A} \sum_{\beta} 2P_{\alpha\beta}^{IAO} \langle \chi_\alpha | u_A - \hat{u} | \chi_\beta \rangle \tag{10.28}$$

$$Q_A^{uv} = \sum_{\alpha \text{ on } A} \sum_{\beta} 2P_{\alpha\beta}^{IAO} \langle \chi_\alpha | \frac{3}{2}(u_A - \hat{u})(v_A - \hat{v}) - \frac{1}{2}||r_A - \hat{r}||^2 \delta_{uv} | \chi_\beta \rangle \tag{10.29}$$

Here A is the atom index, r is the vector (x, y, z), and $u = x, y,$ or z. δ is the Kronecker delta function, χ represents an intrinsic atomic orbital, α and β are the IAO indices, P is the one-particle density matrix, and the IAO superscript on the density matrix indicates that this is the density matrix in the IAO basis. Additionally, the inclusion of u_A, v_A, and r_A (the coordinates of atom A) ensures that the multipoles are origin-invariant.

The calculation of atomic dipoles and quadrupoles is controlled with the following *$rem* variable:

DO_ATOMIC_MULTIPOLES

Enables atomic multipole calculation

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate IAO atomic multipoles

TRUE Calculate IAO atomic multipoles

RECOMMENDATION:

None

Example 10.6 Input for calculating atomic dipoles and quadrupoles via the IAO/IBO procedure.

```
$molecule
0 1
F          0.0000000000    0.0000000000    0.2314949765
H          0.0000000000    0.0000000000    1.1541050235
$end

$rem
jobtyp sp
method hf
basis aug-cc-pvtz
do_ibo true
do_atomic_multipoles true
$end

$loco
min_basis autosad
ibo_mem 1000
$end
```

10.2.11 Excited-State Analysis for CIS and TDDFT

For CIS, TDHF, and TDDFT excited-state calculations, we have already mentioned that Mulliken population analysis of the excited-state electron densities may be requested by setting POP_MULLIKEN = -1, and multipole moments of the excited-state densities will be generated if CIS_MOMENTS = TRUE. Another useful decomposition for excited states is to separate the excitation into "particle" and "hole" components, which can then be analyzed separately.[84,185] To do this, we define a density matrix for the excited electron,

$$\mathbf{D}^{\text{elec}}_{ab} = \sum_i (\mathbf{X}+\mathbf{Y})^{\dagger}_{ai}(\mathbf{X}+\mathbf{Y})_{ib} \tag{10.30}$$

and a density matrix for the hole that is left behind in the occupied space:

$$\mathbf{D}^{\text{hole}}_{ij} = \sum_a (\mathbf{X}+\mathbf{Y})_{ia}(\mathbf{X}+\mathbf{Y})^{\dagger}_{aj} \tag{10.31}$$

The quantities $\mathbf{X}$ and $\mathbf{Y}$ are the transition density matrices, *i.e.*, the components of the TDDFT eigenvector.[54] The indices i and j denote MOs that occupied in the ground state, whereas a and b index virtual MOs. Note also that $\mathbf{D}^{elec} + \mathbf{D}^{hole} = \Delta\mathbf{P}$, the difference between the ground- and excited-state density matrices.

Upon transforming $\mathbf{D}^{\text{elec}}$ and $\mathbf{D}^{\text{hole}}$ into the AO basis, one can write

$$\Delta q = \sum_\mu (\mathbf{D}^{\text{elec}}\,\mathbf{S})_{\mu\mu} = -\sum_\mu (\mathbf{D}^{\text{hole}}\,\mathbf{S})_{\mu\mu} \tag{10.32}$$

where Δq is the total charge that is transferred from the occupied space to the virtual space. For a CIS calculation, or for TDDFT within the Tamm-Dancoff approximation,[87] $\Delta q = -1$. For full TDDFT calculations, Δq may be slightly different than -1.

Comparison of Eq. (10.32) to Eq. (10.3) suggests that the quantities $(\mathbf{D}^{\text{elec}}\,\mathbf{S})$ and $(\mathbf{D}^{\text{hole}}\,\mathbf{S})$ are amenable to population analyses of precisely the same sort used to analyze the ground-state density matrix. In particular, $(\mathbf{D}^{\text{elec}}\,\mathbf{S})_{\mu\mu}$ represents the μth AO's contribution to the excited electron, while $(\mathbf{D}^{\text{hole}}\,\mathbf{S})_{\mu\mu}$ is a contribution to the hole. The sum of these quantities,

$$\Delta q_\mu = (\mathbf{D}^{\text{elec}}\,\mathbf{S})_{\mu\mu} + (\mathbf{D}^{\text{hole}}\,\mathbf{S})_{\mu\mu} \tag{10.33}$$

represents the contribution to Δq arising from the μth AO. For the particle/hole density matrices, both Mulliken and Löwdin population analyses available, and are requested by setting CIS_MULLIKEN = TRUE.

CIS_MULLIKEN

Controls Mulliken and Löwdin population analyses for excited-state particle and hole density matrices.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform particle/hole population analysis.

TRUE Perform both Mulliken and Löwdin analysis of the particle and hole density matrices for each excited state.

RECOMMENDATION:

Set to TRUE if desired. This represents a trivial additional calculation.

Although the excited-state analysis features described in this section require very little computational effort, they are turned off by default, because they can generate a large amount of output, especially if a large number of excited states are requested. They can be turned on individually, or collectively by setting CIS_AMPL_ANAL = TRUE. This collective option also requests the calculation of natural transition orbitals (NTOs),[85] which were introduced in Section 7.14.3. (NTOs can also be requested without excited-state population analysis. Some practical aspects of calculating and visualizing NTOs are discussed below, in Section 10.5.3.)

CIS_AMPL_ANAL

Perform additional analysis of CIS and TDDFT excitation amplitudes, including generation of natural transition orbitals, excited-state multipole moments, and Mulliken analysis of the excited state densities and particle/hole density matrices.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform additional amplitude analysis.

FALSE Do not perform additional analysis.

RECOMMENDATION:

None

CIS_AMPL_PRINT

Sets the threshold for printing CIS and TDDFT excitation amplitudes.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

n Print if $|x_{ia}|$ or $|y_{ia}|$ is larger than $0.01 \times n$.

RECOMMENDATION:

Use the default unless you want to see more amplitudes.

EXCIT_ENERGY_COMPONENTS

Compute individual compnents of the CIS/TDDFT excitation energy.[165] The output is divided into the one-electron components (H); Fock-matrix type components representing the Coulomb (J1), non-local exchange (K1), and xc potentials (XC1); and true two-electron components (J2, K2, XC2). Note that H+J1+K1+XC1 is equivalent to a weighted sum of MO energy differences whereas J2, K2, and XC2 represent the post-MO terms. For more information see Ref. 110.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute excitation energy components.

FALSE Do not compute excitation energy components.

RECOMMENDATION:

Use if more detailed insight into excitation energies is required.

10.2.12 General Excited-State Analysis

Q-CHEM features a module for extended excited-state analysis via the wavefunction analysis library LIBWFA.[174] LIBWFA is interfaced to the ADC, CC/EOM-CC, CIS, and TDDFT/SF-TDDFT methods. These analyses are based on the state, transition and difference density matrices of the excited states. The theoretical background for such analysis is given in Chapter 7.14. A more extended discussion of different analysis methods and classification schemes for excited states is given in Ref. 111.

One-electron transition-density matrix (1TDM) based analyses include the construction and export of natural transition orbitals[136] (NTOs) and electron and hole densities,[172] the evaluation of charge transfer numbers,[170] an analysis of exciton multipole moments,[15,139,173] and quantification of electron-hole entanglement.[169] NTOs are obtained by singular value decomposition (SVD) of the 1TDM:

$$\gamma_{pq}^{\mathrm{IF}} = \langle \Psi_I | p^\dagger q | \Psi_F \rangle \tag{10.34}$$

$$\boldsymbol{\gamma} = \boldsymbol{\alpha}\boldsymbol{\sigma}\boldsymbol{\beta}^\dagger \,, \tag{10.35}$$

where $\boldsymbol{\sigma}$ is diagonal matrix containing singular values and unitary matrices $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ contain the respective particle and hole NTOs. Note that:

$$\|\boldsymbol{\gamma}\|^2 = \sum_{pq} \gamma_{pq}^2 = \sum_K \sigma_K^2 \equiv \Omega \tag{10.36}$$

Furthermore, the formation and export of state-averaged NTOs, and the decomposition of the excited states into transitions of state-averaged NTOs are implemented.[172] The difference and/or state densities can be exported themselves, as well as employed to construct and export natural orbitals, natural difference orbitals, and attachment and detachment densities.[85] Furthermore, two measures of unpaired electrons are computed.[78] In addition, a Mulliken or Löwdin population analysis and an exciton analysis can be performed based on the difference/state densities. The main descriptors of the various analyses that are printed for each excited state are given in Tables 10.1 and 10.2. For a detailed description with illustrative examples, see Refs. 171, 172, and 174.

LIBWFA also computes descriptors for differentiating between diradical and ionic excited states, as defined within valence-bond theory. These descriptors, denoted `LOC`, `LOCa`, `QT`, and `QTa`, are zero for perfect diradical states and increase with the ionic character of the state.[53]

To activate any excited-state analysis STATE_ANALYSIS has to be set to TRUE. For individual analyses there is currently only a limited amount of fine grained control. The construction and export of any type of orbitals is controlled by

Descriptor	**Explanation**
`Leading SVs`	Largest NTO occupation numbers
`Sum of SVs (Omega)`	$\Omega = \|\boldsymbol{\gamma}^{\mathrm{IF}}\|^2$, sum of NTO occupation numbers
`E(h)`	Energy of hole NTO, $E_I(h) = \sum_{pq} \alpha_{pI} F_{pq} \alpha_{qI}$
`E(p)`	Energy of particle NTO, $E_I(p) = \sum_{pq} \beta_{pI} F_{pq} \beta_{qI}$
`PR_NTO`	NTO participation ratio $\mathrm{PR}_{\mathrm{NTO}}$
`Entanglement entropy (S_HE)`	$S_{H\|E} = -\sum_i \lambda_i \log_2 \lambda_i$
`Nr of entangled states (Z_HE)`	$Z_{HE} = 2^{S_{H\|E}}$
`Renormalized S_HE/Z_HE`	Replace $\lambda_i \to \lambda_i/\Omega$ in the above two formulas
`<Phe>`	Expec. value of the particle-hole permutation operator, measuring de-excitations[109]
`<r_h> [Ang]`	Mean position of hole $\langle \vec{x}_h \rangle_{\mathrm{exc}}$
`<r_e> [Ang]`	Mean position of electron $\langle \vec{x}_e \rangle_{\mathrm{exc}}$
`|<r_e - r_h>| [Ang]`	Linear e/h distance $\vec{d}_{h\to e} = \langle \vec{x}_e - \vec{x}_h \rangle_{\mathrm{exc}}$
`Hole size [Ang]`	RMS hole size: $\sigma_h = (\langle \vec{x}_h^2 \rangle_{\mathrm{exc}} - \langle \vec{x}_h \rangle_{\mathrm{exc}}^2)^{1/2}$
`Electron size [Ang]`	RMS elec. size: $\sigma_e = (\langle \vec{x}_e^2 \rangle_{\mathrm{exc}} - \langle \vec{x}_e \rangle_{\mathrm{exc}}^2)^{1/2}$
`RMS electron-hole separation [Ang]`	$d_{\mathrm{exc}} = (\langle \|\vec{x}_e - \vec{x}_h\|^2 \rangle_{\mathrm{exc}})^{1/2}$
`Covariance(r_h, r_e) [Ang^2]`	$\mathrm{COV}\,(\vec{x}_h, \vec{x}_e) = \langle \vec{x}_h \cdot \vec{x}_e \rangle_{\mathrm{exc}} - \langle \vec{x}_h \rangle_{\mathrm{exc}} \cdot \langle \vec{x}_e \rangle_{\mathrm{exc}}$
`Correlation coefficient`	$R_{eh} = \mathrm{COV}\,(\vec{x}_h, \vec{x}_e)\,/\sigma_h \cdot \sigma_e$
`Center-of-mass size`	$0.5 * (\langle \|\vec{x}_e + \vec{x}_h\|^2 \rangle_{\mathrm{exc}} - \langle \vec{x}_e + \vec{x}_h \rangle_{\mathrm{exc}}^2)^{1/2}$

Table 10.1: Descriptors output by Q-CHEM for transition density matrix analysis. Note that squares of the SVs, which correspond to the weights of the respective NTO pairs, are printed. Ω equals the square of the norm of the one-electron transition density matrix (1TDM).

Descriptor	**Explanation**
`n_u`	Number of unpaired electrons $n_u = \sum_i \min(n_i, 2 - n_i)$
`n_u,nl`	Number of unpaired electrons $n_{u,nl} = \sum_i n_i^2 (2 - n_i)^2$
`PR_NO`	NO participation ratio $\mathrm{PR}_{\mathrm{NO}}$
`p_D` and `p_A`	Promotion number p_D and p_A
`PR_D` and `PR_A`	D/A participation ratio PR_D and PR_A
`<r_h> [Ang]`	Mean position of detachment density $\vec{d}_D$
`<r_e> [Ang]`	Mean position of attachment density $\vec{d}_A$
`|<r_e - r_h>| [Ang]`	Linear D/A distance $\vec{d}_{D\to A} = \vec{d}_A - \vec{d}_D$
`Hole size [Ang]`	RMS size of detachment density σ_D
`Electron size [Ang]`	RMS size of attachment density σ_A

Table 10.2: Descriptors output by Q-CHEM for difference/state density matrix analysis.

MOLDEN_FORMAT to export the orbitals as MOLDEN files, and NTO_PAIRS which specifies the number of important orbitals to print (note that the same keyword controls the number of natural orbitals, the number of natural difference orbitals, and the number of NTOs to be printed). Setting MAKE_CUBE_FILES to TRUE triggers the construction and export of densities in "cube file" format[82], which requires the specification of the *$plots* section in either old or new format (see Sections 10.5.4 and 10.5.5 for details). Setting IQMOL_FCHK = TRUE (equivalently, GUI = 2) will export data to the "`.fchk`" formatted checkpoint file, and switches off the generation of cube files. The population analyses are controlled by POP_MULLIKEN and LOWDIN_POPULATION. Setting the latter to TRUE will enforce Löwdin population analysis to be employed for regular populations as well as CT numbers, while by default the Mulliken population analysis is used.

Any MOLDEN or cube files generated by the excited state analyses can be found in the directory `plots` in the job's scratch directory. Their names always start with a unique identifier of the excited state (the exact form of this human readable identifier varies with the excited state method). The names of MOLDEN files are then followed by either `_no.mo`, `_ndo.mo`, or `_nto.mo` depending on the type of orbitals they contain. In case of cube files the state identifier is followed by `_dens`, `_diff`, `_trans`, `_attach`, `_detach`, `_elec`, or `_hole` for state, difference, transition, attachment, detachment, electron, or hole densities, respectively. All cube files have the suffix `.cube`. In unrestricted calculations an additional part is added to the file name before `.cube` which indicates α (`_a`) or β (`_b`) spin. The only exception is the state density for which `_tot` or `_sd` are added indicating the total or spin-density parts of the state density. Analysis of relaxed CIS or TDDFT densities can be triggered by CIS_RELAXED_DENSITY = TRUE. The corresponding output files are marked by `_rlx`. Computation of ESPs for state, transition, and electron/hole densities (see Ref. 109) can be triggered by setting ESP_GRID = -3. These are indicated by `_esp` as part of the file name.

The `ctnum_*.om` file created in the main directory serves as input for a charge transfer number analysis, as explained, *e.g.*, in Refs. 140,170. Use the external `TheoDORE` program (`theodore-qc.sourceforge.net`) to create electron/hole correlation plots and to compute fragment based descriptors.

Note:

- When doing excited-state calculations from an open-shell reference, LIBWFA will perform the analysis for both $\alpha\alpha$ and $\beta\beta$ transition densities. Make sure you look at the correct one. The way to figure it out is to remember that in open-shell references $N_\alpha > N_\beta$, *e.g.*, in doublet references, the unpaired electron is α and the hole is β. Thus, for transitions of the unpaired electron into the unoccupied orbitals you need $\alpha\alpha$ block, whereas for the transitions from doubly occupied orbitals into the singly unoccupied orbital (the hole) you need the $\beta\beta$ block.

- To activate all types of analyses in CCMAN2 jobs, you have to activate CC_REF_PROP, CC_EOM_PROP, and CC_TRANS_PROP. This is shown in Example 7.165.

- In Hermitian formalisms, γ^{IF} is a Hermitian conjugate of γ^{FI}, but in non-Hermitian approaches, such as coupled-cluster theory, the two are slightly different. While for quantitative interstate properties both γ^{IF} and γ^{FI} are computed, the qualitative trends in exciton properties derived from $(\gamma^{\mathrm{IF}})^\dagger$ and γ^{FI} are very similar. Only one 1TDM is analyzed for EOM-CC.

- In spin-restricted calculations, the LIBWFA module computes NTOs for the $\alpha\alpha$ block of transition density. Thus, when computing NTOs for the transitions between open-shell EOM-IP/EA states make sure to specify correct spin states. For example, use EOM_EA_ALPHA to visualize transitions involving the extra electron.

STATE_ANALYSIS

Triggers the general state analysis via LIBWFA.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not run excited state analysis.

TRUE Activate excited state analysis.

RECOMMENDATION:

This analysis produces only minimal computational overhead (as long as no cube files are produced) and can be activated whenever some additional information about the excited state is required.

WFA_LEVEL

Master variable for controlling the amount of output produced by LIBWFA.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

1 Only perform some population analyses.

2 Also perform exciton analysis and compute natural (transition/difference) orbitals.

3 Also perform charge transfer number analysis.

4 Maximal output including cube files for NTOs (this is needed to reproduce Ref. 109)

RECOMMENDATION:

Reduce if you want less print-out.

NTO_PAIRS

Controls how many hole/particle NTO pairs and frontier natural orbital pairs and natural difference orbital pairs are printed in the standard output.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N Write N NTO/NO/NDO pairs per excited state.

RECOMMENDATION:

This controls the print-out to the standard output. Use WFA_ORB_THRESH if you want to modify the number of orbitals exported.

WFA_ORB_THRESH

Controls the number of hole/particle NTO pairs and frontier natural orbital pairs and natural difference orbital pairs exported to the Molden files.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

N Export all NTO/NO/NDO pairs with a weight above 10^{-N}.

RECOMMENDATION:

WFA_REF_STATE

Controls the reference state for the transition and difference density matrices used by LIBWFA. This keyword works for CIS/TDDFT/SF-DTDDFT computations. Use CC_STATE_TO_OPT for EOM-CC.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

-1 Use default: ground-state for standard CIS/TDDFT computations, first response state for SF-TDDFT.

0 Reference state

N Nth excited state/response state.

RECOMMENDATION:

NONE

Example 10.7 Basic excited-state analysis example for formaldehyde at the ADC(2)/def2-SV(P) level: computation of various descriptors and output of NOs, NTOs, and NDOs in MOLDEN format. LIBWFA used in this basic mode does usually not add notable computational cost.

```
$molecule
   0 1
   C     0.000000   0.000000    0.523383
   O    -0.000000   0.000000   -0.671856
   H     0.931138   0.000000    1.11728
   H    -0.931138   0.000000    1.11728
$end

$rem
   METHOD           adc(2)
   BASIS            def2-sv(p)
   N_FROZEN_CORE    fc
   EE_SINGLETS      [0,1,1,0]
   STATE_ANALYSIS   true
   MOLDEN_FORMAT    true
$end
```

Example 10.8 Analysis of two TDDFT excited states of formaldehyde using LIBWFA including output of cube files. This uses the new *$plots* format (see Section 10.5.4.1) with grid points specified. The cubes files of HOMO and LUMO will also be generated.

```
$molecule
0 1
  C    0.0000000   -0.0000000   -0.6133791
  O   -0.0000000    0.0000000    0.6060734
  H    0.0000000    0.9391300   -1.1555819
  H    0.0000000   -0.9391300   -1.1555819
$end

$rem
   METHOD           PBE0
   BASIS            6-31+G(d)
   CIS_N_ROOTS      2
   CIS_TRIPLETS     false
   MAKE_CUBE_FILES  true
   PLOTS            true
   STATE_ANALYSIS   true
$end

$plots
   grid_points                    50 50 50
   alpha_molecular_orbital        8-9
$end
```

Note: For the output of orbitals and densities it is possible to use either the MOLDEN_FORMAT = TRUE (see Example 10.7), MAKE_CUBE_FILES = TRUE (see Example 10.8), or GUI = 2 options. These options are mutually exclusive when STATE_ANALYSIS is activated.

Example 10.9 Uracil computed at the PBE0/def2-SV(P) level. Activation of extended LIBWFA functionality including export of NTOs and densities in cube format and computation of the ESPs of these densities. This extended LIBWFA mode can add significant computational cost, especially if a fine grid is chosen.

```
$molecule
   0 1
   C    1.194380    1.102510    0.000000
   C   -0.008366    1.692430    0.000000
   N   -1.169600    0.978035    0.000000
   C   -1.212060   -0.402293    0.000000
   N    0.034691   -0.979140    0.000000
   C    1.281590   -0.348737    0.000000
   O   -2.243420   -1.023750    0.000000
   O    2.299180   -0.995854    0.000000
   H   -0.123160    2.767140    0.000000
   H   -2.061440    1.444100    0.000000
   H    0.044818   -1.989990    0.000000
   H    2.104720    1.679840    0.000000
$end

$rem
   METHOD            pbe0
   BASIS             def2-sv(p)
   CIS_N_ROOTS       4
   CIS_SINGLETS      true
   CIS_TRIPLETS      true
   RPA               false
   STATE_ANALYSIS    true
   NTO_PAIRS         3
   MAKE_CUBE_FILES   true
   ESP_GRID          -3
   WFA_LEVEL         4
   WFA_ORB_THRESH    3
$end

$plots
Write cube files for all 4 states
35  -3.5  3.5
35  -3.5  3.5
15  -1.5  1.5
0  4  0  0
1 2 3 4
$end
```

Other examples of LIBWFA uses:

- Example 7.30 illustrates wave-function analysis of the SF-DFT states in para-benzyne;
- Example 7.63 illustrates wave-function analysis of XAS transitions within CVS-EOM-EE;
- Example 7.165 illustrates wave-function analysis for transitions between EOM-IP states;
- Example 7.102 illustrates wave-function analysis of complex-valued densities within CAP-EOM-CCSD.

10.3 Orbital Analysis

In addition to analyzing properties of the total wave function, Q-CHEM contains a variety of tools to transform the MOs into representations that may be more useful for chemical analysis.

10.3.1 Interface to the NBO Package

Q-CHEM incorporates v. 5 of the Natural Bond Orbital (NBO) package for molecular properties and wave function analysis,[69,223] and can interface to v. 6[69] and v. 7[70] of the NBO program as well. The NBO5 package is invoked simply by setting NBO = TRUE in the *$rem* section and is initiated after the SCF wave function is obtained. If switched on for a geometry optimization, the NBO5 program will only be invoked at the end of the last optimization step.

To use either NBO6 or NBO7, the desired version of the NBO program must be purchased, downloaded, and installed separately from nbo.chem.wisc.edu. Q-CHEM can then interface to an external version of the NBO program by: (a) setting the NBOEXE environment variable appropriately, and (b) setting both NBO = TRUE and NBO_EXTERNAL = TRUE in the *$rem* section of the Q-CHEM input file. Note that support for (and new downloads of) the NBO6 program were discontinued in 2020. Existing copies should still interface to Q-CHEM, and the newer NBO7 program should interface in the same way. The older NBO5 program continues to be a part of Q-CHEM and the latter is what runs if NBO = TRUE is set without specifying NBO_EXTERNAL.

Users should consult the NBO User's Manual for options and details relating to NBO calculations. NBO analysis is also available for excited states calculated using CIS or TDDFT. Excited-state NBO analysis is less well-developed, and users should be aware that the convergence of the NBO search procedure may be less well-behaved for excited states than it is for ground states. Excited state may require specification of additional NBO parameters in the *$nbo* section that is described below. Consult Ref. 222 for details and suggestions.

NBO

Controls the use of the NBO package.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not invoke the NBO package.

1 Do invoke the NBO package, for the ground state.

2 Invoke the NBO package for the ground state, and also each CIS, RPA, or TDDFT excited state.

RECOMMENDATION:

None

NBOMEMORY

Controls how much memory is allocated to Q-CHEM's internal NBO (v. 5) package.

TYPE:

INTEGER

DEFAULT:

500000

OPTIONS:

N Allocate N double-precision words (= $8N$ bytes).

RECOMMENDATION:

This memory comes out of MEM_TOTAL.

The general format for passing options from Q-CHEM to the NBO program is shown below:

```
$nbo
   {NBO program keywords, parameters and options}
$end
```

A few usage notes:

1. *$rem* variable NBO must be set to TRUE before the *$nbo* keyword is recognized.
2. Q-CHEM does not support facets of the NBO package which require multiple job runs
3. Output of the NBOs can be triggered by the PRINT_ORBITALS and MOLDEN_FORMAT keywords. In this case two MOLDEN sections are written to *outfile*. The first one corresponds to the regular MOs, the second one to the NBOs.
4. Print-out of the full set of NAOs, NHOs, NBOs, and NLMOs can be triggered via the PLOT keyword in the $nbo section. The files of interest are `FILE.31` – `FILE.39` in `$QCSCRATCH/savename`. These files can be opened by, *e.g.*, the CHEMCRAFT and JMOL programs.
5. Q-CHEM's internal implementation of NBO v. 5 has an upper limit of 300 atoms and 3,000 basis functions.

Example 10.10 Basic input for NBO computation on formaldehyde. The NBOs are printed to *outfile* in MOLDEN format and the full set of files in native NBO format are written to `$QCSCRATCH/savename`.

```
$molecule
   0 1
   C    0.000000    0.000000    0.523383
   O    0.000000    0.000000   -0.671856
   H    0.931138    0.000000    1.117280
   H   -0.931138    0.000000    1.117280
$end

$rem
   METHOD           pbe0
   BASIS            def2-sv(p)
   NBO              1
   PRINT_ORBITALS   true
   MOLDEN_FORMAT    true
$end

$nbo
   PLOT
$end
```

Example 10.11 NBO calculation that provides the canonical MOs in the natural atomic orbital (NAO) basis. This is an easy way to establish the percentage AO character in each canonical MO.

```
$molecule
   0 1
   C    0.000000    0.000000    0.523383
   O    0.000000    0.000000   -0.671856
   H    0.931138    0.000000    1.117280
   H   -0.931138    0.000000    1.117280
$end

$rem
   METHOD           pbe0
   BASIS            def2-sv(p)
   NBO              true
$end

$nbo
   naomo
$end
```

10.3.2 Orbital Localization

10.3.2.1 Overview

The concept of localized orbitals has already been visited in this manual in the context of perfect-pairing and methods. As the SCF energy is independent of the partitioning of the electron density into orbitals, there is considerable flexibility as to how this may be done. The canonical picture, where the orbitals are eigenfunctions of the Fock operator is useful in determining reactivity, for, through Koopmans' theorem, the orbital energy eigenvalues give information about the corresponding ionization energies and electron affinities. As a consequence, the HOMO and LUMO are very informative as to the reactive sites of a molecule. In addition, in small molecules, the canonical orbitals lead us to the chemical description of σ and π bonds.

In large molecules, however, the canonical orbitals are often very delocalized, and so information about chemical bonding is not readily available from them. Here, orbital localization techniques can be of great value in visualizing the bonding, as localized orbitals often correspond to the chemically intuitive orbitals which might be expected.

Q-CHEM has three post-SCF localization methods available. These can be performed separately over both occupied and virtual spaces. The localization scheme attributed to Boys[29,30] minimizes the radial extent of the localized orbitals, *i.e.*, the second moment $\sum_i \langle ii||\mathbf{r}_1 - \mathbf{r}_2|^2|ii\rangle$, and although is relatively fast, does not separate σ and π orbitals, leading to two "banana-orbitals" in the case of a double bond.[168] Pipek-Mezey localized orbitals[168] maximize the locality of Mulliken populations, and are of a similar cost to Boys localized orbitals, but maintain $\sigma - \pi$ separation. Edmiston-Ruedenberg localized orbitals[56] maximize the self-repulsion of the orbitals, $\sum_i \langle ii|\frac{1}{r}|ii\rangle$. This is more computationally expensive to calculate as it requires a two-electron property to be evaluated, but can be reduced to cubic-scaling cost (with respect to the number of occupied orbitals), via the resolution of identity approximation.[202]

BOYSCALC

Specifies how Boys localized orbitals are to be calculated

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform any Boys localization.

1 Localize core and valence together.

2 Do separate localizations on core and valence.

RECOMMENDATION:

None

ERCALC

Specifies how Edmiston-Ruedenberg localized orbitals are to be calculated

TYPE:

INTEGER

DEFAULT:

06000

OPTIONS:

$aabcd$

aa specifies the convergence threshold.
If $aa > 3$, the threshold is set to 10^{-aa}. The default is 6.
If $aa = 1$, the calculation is aborted after the guess, allowing Pipek-Mezey orbitals to be extracted.

b specifies the guess:
0 Boys localized orbitals. This is the default
1 Pipek-Mezey localized orbitals.

c specifies restart options (if restarting from an ER calculation):
0 No restart. This is the default
1 Read in MOs from last ER calculation.
2 Read in MOs and RI integrals from last ER calculation.

d specifies how to treat core orbitals
0 Do not perform ER localization. This is the default.
1 Localize core and valence together.
2 Do separate localizations on core and valence.
3 Localize only the valence electrons.
4 Use the *$localize* section.

RECOMMENDATION:

ERCALC 1 will usually suffice, which uses threshold 10^{-6}.

The *$localize* section may be used to specify orbitals subject to ER localization if require. It contains a list of the orbitals to include in the localization. These may span multiple lines. If the user wishes to specify separate beta orbitals to localize, include a zero before listing the beta orbitals, which acts as a separator, *e.g.*,

```
$localize
   2 3 4 0
   2 3 4 5 6
$end
```

10.3.2.2 Virtual Orbital Localization

Virtual orbitals can be advantageous to be localized in many scenarios. One scenario where this is useful is generalized valence bond (GVB) methods, where each bonding orbital is paired with its antibonding orbital through Sano procedure. Currently this is done in GVBMAN when PP or CCVB is run. An improved guess has been proposed that has been shown to converge faster.[11] The new subroutine is a stand-alone version that can generate these antibonding orbitals and exit without initiating a GVB calculation. It can do Boys, Pipek-Mezey, or Edmiston-Rudenberg localization for the occupied space depending on GVB_LOCAL = 1, 2, or 3, respectively, while 0 performs it on the canonical orbitals. The subroutine also prints out each occupied orbital's Mulliken charge, delocalization measure, and variance, in which it automatically detects the bonding orbitals and generates an antibonding guess for each. A population analysis based on this effective minimal basis can also be done using EDA_POP_ANAL = 1. The number of bonds can be enforced

by taking the highest GVB_N_PAIRS specified, with no guarantee of them being bonding, *i.e.* they can be core or lone pairs. This is currently implemented for restricted and restricted Open-shell spin symmetries; work on the unrestricted case is underway.

ANTIBOND

Triggers Antibond subroutine to generate antibonding orbitals after a converged SCF

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Does not localize the virtual space.

1 Localizes the virtual space, one antibonding for every bond.

2,3 Fill the virtual space with antibonding orbitals-like guesses.

4 Does Frozen Natural Orbitals and leaves them on scratch for future jobs or visualization.

RECOMMENDATION:

None

DOMODSANO

Specifies whether to do modified Sano or the original one

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Does original Sano procedure (similar to GVBMAN).

1 Does an improved Sano procedure that's more localized.

2 Does another variation of Sano.

RECOMMENDATION:

1 is always better

10.3.3 Donor–Acceptor Orbital Overlaps

In donor–acceptor (D–A) materials, the overlap between the donor orbital (HOMO of D) and the acceptor orbital (LUMO of A) has been shown to correlate with excited-state charge-transfer character.[208] This overlap can be computed, taking into account relaxation of the monomer geometries upon formation of the D–A complex, using the *$rem* variable MO_OVERLAPS_TWO_GEOMS, as shown in Example 10.13 below. This requires two separate SCF calculations (one for either D or A in isolation, and another for the dimer D–A), and both calculations must use the same basis set. Therefore, ghost atoms should be added in the monomer calculation so that both SCF calculations use the dimer basis set. Output is delivered in a directory `OutputName.MO-overlaps` (where `OutputName` is the name of the Q-CHEM output file). This output consists of the α-spin MO coefficients for both systems (`CA-Geom1.txt` and `CA-Geom2.txt`), the AO overlap matrix between the two calculations (`AO-overlap-TwoGeoms.txt`, which need not be an identity matrix because the geometries could differ), and the MO overlap matrix between the two calculations (`MO-overlap-TwoGeoms.txt`).

Note: Currently, only overlaps between α-spin MOs are considered, hence why only α-spin MO coefficients are returned. This means that this features is implemented only for spin-restricted wave functions.

Alternatively, one might be interested in the degree of spatial overlap between donor and acceptor orbitals (*e.g.*, HOMO

and LUMO) within the same molecule or on two different molecules within the same supramolecular calculation. These can be computed using the same tools that are used to compute the Λ metric in TDDFT [Eq. (7.16)], and the contributing overlaps O_{ia} [Eq. (7.17)] can be printed using SPATIAL_OVERLAP_ANALYSIS. See Section 7.3.5.1 for job control options and Example 10.12 for an example.

MO_OVERLAPS_TWO_GEOMS

Specifies whether to compute molecular orbital overlaps at two different geometries.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute these overlaps.

1 Used to indicate the first of two required SCF calculations.

2 Used to indicate the second of two required SCF calculations.

RECOMMENDATION:

The atoms must be ordered in the same way for both calculations. (This is not checked.)

Example 10.12 Compute "spatial overlaps" of the frontier MOs, for a dimer of naphthalene and tetracyanoquinone. These are overlaps O_{ia} of the absolute values of the orbitals [Eq. (7.17)]. This feature is based on the Λ metric of TDDFT, therefore at least one excited state must be computed in order to trigger this analysis.

```
$molecule
0 1
  C   1.262951   1.945976  -0.561360
  C   0.648393   2.896730   0.296837
  C   2.662678   1.710349  -0.545738
  H   3.283172   2.277769   0.142405
  C   3.225026   0.771942  -1.373626
  H   4.295659   0.593998  -1.345666
  C   2.415782   0.024414  -2.263235
  H   2.874995  -0.720497  -2.905219
  C   1.061881   0.234151  -2.304223
  H   0.435114  -0.339033  -2.982450
  C   0.449076   1.198213  -1.458953
  C  -0.950361   1.427044  -1.468096
  C  -1.241593  -0.113527   1.511623
  H  -1.776720   0.630886   2.090246
  C   0.087508  -0.255323   1.653805
  H   0.634524   0.372807   2.348013
  C   0.826509  -1.220978   0.861471
  C   2.186874  -1.336026   0.974224
  C   0.076113  -2.057065  -0.061488
  H   0.616720  -2.788316  -0.652909
  C  -1.255888  -1.926290  -0.191214
  H  -1.804801  -2.556542  -0.883200
  C  -1.987167  -0.926165   0.566999
  C  -3.333669  -0.739067   0.378968
  C  -1.516839   2.349538  -0.623777
  C  -0.708145   3.093806   0.267607
  H   1.274799   3.469589   0.975647
  H  -1.571143   0.856542  -2.154642
  H  -2.590617   2.512925  -0.634219
  H  -1.169539   3.821896   0.927541
  C   2.935702  -0.523467   1.883035
  C   2.933880  -2.256250   0.173594
  N   3.523004   0.145261   2.626355
  N   3.522832  -3.002457  -0.490547
  C  -4.043584   0.294041   1.066080
  C  -4.077033  -1.521605  -0.558360
  N  -4.657467  -2.163965  -1.330296
  N  -4.588038   1.154569   1.621873
$end

$rem
method                wb97x-d
basis                 6-31G*
cis_triplets          false
cis_n_roots           1    ! needed for spatial overlap analysis
spatial_overlap_anal  true
spatial_overlap_print 1
$end
```

Example 10.13 Compute MO overlaps for a $C_2H_4 \cdots C_2F_4$ dimer.

```
$comment
Job 1: this is the dimer
$end

$molecule
0 1
          C      -0.6656405432    -0.0000000000    -3.3601967383
          H      -1.2362909837    -0.9253767714    -3.3623091560
          H      -1.2362909837     0.9253767714    -3.3623091560
          C       0.6656405432     0.0000000000    -3.3601967383
          H       1.2362909837     0.9253767714    -3.3623091560
          H       1.2362909837    -0.9253767714    -3.3623091560
          C      -0.6610251661    -0.0000000000     1.1109504097
          F      -1.3861639022    -1.1014627457     1.1119323203
          F      -1.3861639022     1.1014627457     1.1119323203
          C       0.6610251661    -0.0000000000     1.1109504097
          F       1.3861639022     1.1014627457     1.1119323203
          F       1.3861639022    -1.1014627457     1.1119323203
$end

$rem
method cam-b3lyp
basis 6-31+G*
mo_overlaps_two_geoms 1 ! for job #1
$end

@@@

$comment
Job 2:  monomer1, with ghost functions for monomer2
$end

$molecule
0 1
          C      -0.6656405432    -0.0000000000    -3.3601967383
          H      -1.2362909837    -0.9253767714    -3.3623091560
          H      -1.2362909837     0.9253767714    -3.3623091560
          C       0.6656405432     0.0000000000    -3.3601967383
          H       1.2362909837     0.9253767714    -3.3623091560
          H       1.2362909837    -0.9253767714    -3.3623091560
         @C      -0.6610251661    -0.0000000000     1.1109504097
         @F      -1.3861639022    -1.1014627457     1.1119323203
         @F      -1.3861639022     1.1014627457     1.1119323203
         @C       0.6610251661    -0.0000000000     1.1109504097
         @F       1.3861639022     1.1014627457     1.1119323203
         @F       1.3861639022    -1.1014627457     1.1119323203
$end

$rem
method          cam-b3lyp
basis           6-31+G*
mo_overlaps_two_geoms 2 ! for job #2
$end
```

10.4 Density Analysis

10.4.1 Spin and Charge Densities at the Nuclei

Gaussian basis sets violate nuclear cusp conditions.[105,161,179] This may lead to large errors in wave function at nuclei, particularly for spin density calculations.[44] This problem can be alleviated by using an averaging operator that compute wave function density based on constraints that wave function must satisfy near Coulomb singularity.[180,181] The derivation of operators is based on hyper virial theorem[88] and presented in Ref. 180. Application to molecular spin densities for spin-polarized[181] and DFT[219] wave functions show considerable improvement over traditional delta function operator.

One of the simplest forms of such operators is based on the Gaussian weight function $\exp[-(Z/r_0)^2(\mathbf{r}-\mathbf{R})^2]$ that samples the vicinity of a nucleus of charge Z located at $\mathbf{R}$. The parameter r_0 has to be small enough to neglect two-electron contributions of the order $\mathcal{O}(r_0^4)$ but large enough for meaningful averaging. The range of values between 0.15–0.3 a.u. has been shown to be adequate, with final answer being relatively insensitive to the exact choice of r_0.[180,181] The value of r_0 is chosen by RC_R0 keyword in the units of 0.001 a.u. The averaging operators are implemented for single determinant Hartree-Fock and DFT, and correlated SSG wave functions. Spin and charge densities are printed for all nuclei in a molecule, including ghost atoms.

RC_R0

Determines the parameter in the Gaussian weight function used to smooth the density at the nuclei.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Corresponds the traditional delta function spin and charge densities

n corresponding to $n \times 10^{-3}$ a.u.

RECOMMENDATION:

We recommend value of 250 for a typical spit valence basis. For basis sets with increased flexibility in the nuclear vicinity the smaller values of r_0 also yield adequate spin density.

10.4.2 Atoms in Molecules

Q-CHEM can output a file suitable for analysis with the Atoms in Molecules package (AIMPAC). The source for AIMPAC can be freely downloaded from the web site

http://www.chemistry.mcmaster.ca/aimpac/imagemap/imagemap.htm

Users should check this site for further information about installing and running AIMPAC. The AIMPAC input file is created by specifying a filename for the WRITE_WFN *$rem*.

WRITE_WFN

Specifies whether or not a `.wfn` file is created, which is suitable for use with AIMPAC. Note that the output to this file is currently limited to f orbitals, which is the highest angular momentum implemented in AIMPAC.

TYPE:

STRING

DEFAULT:

(NULL) No output file is created.

OPTIONS:

filename Specifies the output file name. The suffix `.wfn` will be appended to this name.

RECOMMENDATION:

None

10.5 Visualizing and Plotting Volumetric Quantities

10.5.1 Introduction

The free, open-source visualization program IQMOL (http://www.iqmol.org) provides a graphical user interface for Q-CHEM that can be used as a molecular structure builder, as a tool for local or remote submission of Q-CHEM jobs, and as a visualization tool for densities and molecular orbitals. Alternatively, Q-CHEM can generate orbital and density data in formats suitable for plotting with various third-party visualization programs.

10.5.2 Visualizing Orbitals Using MOLDEN and MACMOLPLT

Upon request, Q-CHEM will generate an input file for MOLDEN, a freely-available molecular visualization program.[1,186] MOLDEN can be used to view ball-and-stick molecular models (including stepwise visualization of a geometry optimization), molecular orbitals, vibrational normal modes, and vibrational spectra. MOLDEN also contains a powerful Z-matrix editor. In conjunction with Q-CHEM, orbital visualization via MOLDEN is currently supported for s, p, and d functions (pure or Cartesian), as well as pure f functions. Upon setting MOLDEN_FORMAT to TRUE, Q-CHEM will append a MOLDEN-formatted input file to the end of the Q-CHEM log file. As some versions of MOLDEN have difficulty parsing the Q-CHEM log file itself, we recommend that the user cut and paste the MOLDEN-formatted part of the Q-CHEM log file into a separate file to be read by MOLDEN.

MOLDEN_FORMAT

Requests a MOLDEN-formatted input file containing information from a Q-CHEM job.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Append MOLDEN input file at the end of the Q-CHEM output file.

RECOMMENDATION:

None.

MOLDEN-formatted files can also be read by MACMOLPLT, another freely-available visualization program.[28] [2] MACMOLPLT generates orbital iso-contour surfaces much more rapidly than MOLDEN, however, within MACMOLPLT these surfaces are only available for Cartesian Gaussian basis functions, *i.e.*, PURECART = 2222, which may not be the default.

Example 10.14 Generating a MOLDEN file for molecular orbital visualization.

```
$molecule
   0 1
   O
   H  1  0.95
   H  1  0.95  2  104.5
$end

$rem
   METHOD           hf
   BASIS            cc-pvtz
   PRINT_ORBITALS   true (default is to print 5 virtual orbitals)
   MOLDEN_FORMAT    true
$end
```

For geometry optimizations and vibrational frequency calculations, one need only set MOLDEN_FORMAT to TRUE, and the relevant geometry or normal mode information will automatically appear in the MOLDEN section of the Q-CHEM log file.

Example 10.15 Generating a MOLDEN file to step through a geometry optimization.

```
$molecule
   0  1
   O
   H  1  0.95
   H  1  0.95  2  104.5
$end

$rem
   JOBTYPE          opt
   METHOD           hf
   BASIS            6-31G*
   MOLDEN_FORMAT    true
$end
```

10.5.3 Visualization of Natural Transition Orbitals

For excited states calculated using the CIS, RPA, TDDFT, EOM-CC, and ADC methods, construction of Natural Transition Orbitals (NTOs), as described in Sections 7.14.3 and 10.2.12, is requested using the *$rem* variable NTO_PAIRS. This variable also determines the number of hole/particle NTO pairs that are output for each excited state and the number of natural orbitals or natural difference orbitlas. Although the total number of hole/particle pairs is equal to the number of occupied MOs, typically only a very small number of these pairs (often just one pair) have significant amplitudes. (Additional large-amplitude NTOs may be encountered in cases of strong electronic coupling between multiple chromophores or where there is significant multireference character in the excited state.[85])

NTO_PAIRS

Controls the writing of hole/particle NTO pairs for excited state.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N Write N NTO pairs per excited state.

RECOMMENDATION:

If activated ($N > 0$), a minimum of two NTO pairs will be printed for each state. Increase the value of N if additional NTOs are desired.

When NTO_PAIRS > 0, Q-CHEM will generate the NTOs in MOLDEN format. The NTOs are state-specific, in the sense that each excited state has its own NTOs, and therefore a separate MOLDEN file is required for each excited state. These files are written to the job's scratch directory, in a sub-directory called *NTOs*, so to obtain the NTOs the scratch directory must be saved using the *–save* option that is described in Section 2.2. The output files in the *NTOs* directory have an obvious file-naming convention. The "hole" NTOs (which are linear combinations of the occupied MOs) are printed to the MOLDEN files in order of increasing importance, with the corresponding excitation amplitudes replacing the canonical MO eigenvalues. (This is a formatting convention only; the excitation amplitudes are unrelated to the MO eigenvalues.) Following the holes are the "particle" NTOs, which represent the excited electron and are linear combinations of the virtual MOs. These are written in order of decreasing amplitude. To aid in distinguishing the two sets within the MOLDEN files, the amplitudes of the holes are listed with negative signs, while the corresponding NTO for the excited electron has the same amplitude with a positive sign.

Due to the manner in which the NTOs are constructed (see Section 7.14.3), NTO analysis is available only when the number of virtual orbitals exceeds the number of occupied orbitals, which may not be the case for minimal basis sets.

Example 10.16 Generating MOLDEN-formatted natural transition orbitals for several excited states of uracil.

```
$molecule
   0 1
   N    -2.181263     0.068208     0.000000
   C    -2.927088    -1.059037     0.000000
   N    -4.320029    -0.911094     0.000000
   C    -4.926706     0.301204     0.000000
   C    -4.185901     1.435062     0.000000
   C    -2.754591     1.274555     0.000000
   N    -1.954845     2.338369     0.000000
   H    -0.923072     2.224557     0.000000
   H    -2.343008     3.268581     0.000000
   H    -4.649401     2.414197     0.000000
   H    -6.012020     0.301371     0.000000
   H    -4.855603    -1.768832     0.000000
   O    -2.458932    -2.200499     0.000000
$end

$rem
   METHOD          B3LYP
   BASIS           6-31+G*
   CIS_N_ROOTS     3
   NTO_PAIRS       2
$end
```

10.5.4 Generation of Volumetric Data Using *$plots*

The simplest way to visualize the charge densities and molecular orbitals that Q-CHEM evaluates is via a graphical user interface, such as those described in the preceding section. An alternative procedure, which is often useful for generating high-quality images for publication, is to evaluate certain densities and orbitals on a user-specified grid of points. This is accomplished by invoking the *$plots* option, which is itself enabled by requesting IANLTY = 200.

The format of the *$plots* input is documented below. It permits plotting of molecular orbitals, the SCF ground-state density, and excited-state densities obtained from CIS, RPA or TDDFT/TDA, or TDDFT calculations. Also in connection with excited states, either transition densities, attachment/detachment densities, or natural transition orbitals (at the same levels of theory given above) can be plotted as well. See Ref. 85 for descriptions of these various possibilities. Please note that plotting complex-valued orbitals and electron densities from Complex-Absorbing-Potential Hartree-Fock calculations is so far only possible by direct generation of cube files (see Section 10.5.5).

By default, the output from the *$plots* command is one (or several) plain text files in the working directory, named `plot.mo`, *etc.*. The results then must be visualized with a third-party program capable of making 3-D plots. (Some suggestions are given in Section 10.5.5.)

An example of the use of the *$plots* option is the following input deck:

Example 10.17 A job that evaluates the H_2 HOMO and LUMO on a $1 \times 1 \times 15$ grid, along the bond axis. The plotting output is in an ASCII file called `plot.mo`, which lists for each grid point, x, y, z, and the value of each requested MO.

```
$molecule
   0  1
   H    0.0    0.0    0.35
   H    0.0    0.0   -0.35
$end

$rem
   METHOD      hf
   BASIS       6-31g**
   IANLTY      200
$end

$plots
   Plot the HOMO and the LUMO on a line
   1    0.0    0.0
   1    0.0    0.0
  15   -3.0    3.0
   2    0    0    0
   1    2
$end
```

General format for the *$plots* section of the Q-CHEM input deck.

$plots

One comment line

Specification of the 3-D mesh of points on 3 lines:

$N_x \quad x_{\min} \quad x_{\max}$

$N_y \quad y_{\min} \quad y_{\max}$

$N_z \quad z_{\min} \quad z_{\max}$

A line with 4 integers indicating how many things to plot:

$N_{\mathrm{MO}} \quad N_{\mathrm{Rho}} \quad N_{\mathrm{Trans}} \quad N_{\mathrm{DA}}$

An optional line with the integer list of MO's to evaluate (only if $N_{\mathrm{MO}} > 0$)

MO(1) MO(2) ... MO(N_{MO})

An optional line with the integer list of densities to evaluate (only if $N_{\mathrm{Rho}} > 0$)

Rho(1) Rho(2) ... Rho(N_{Rho})

An optional line with the integer list of transition densities (only if $N_{\mathrm{Trans}} > 0$)

Trans(1) Trans(2) ... Trans(N_{Trans})

An optional line with states for detachment/attachment densities (if $N_{\mathrm{DA}} > 0$)

DA(1) DA(2) ... DA(N_{DA})

$end

Line 1 of the *$plots* keyword section is reserved for comments. Lines 2–4 list the number of one dimension points and the range of the grid (note that coordinate ranges are in Ångstroms if INPUT_BOHR is not set, while all output is in atomic units). Line 5 must contain 4 non-negative integers indicating the number of: molecular orbitals (N_{MO}), electron densities (N_{Rho}), transition densities and attach/detach densities (N_{DA}), to have mesh values calculated.

The final lines specify which MOs, electron densities, transition densities and CIS attach/detach states are to be plotted (the line can be left blank, or removed, if the number of items to plot is zero). Molecular orbitals are numbered $1 \ldots N_\alpha, N_\alpha + 1 \ldots N_\alpha + N_\beta$; electron densities numbered where 0= ground state, 1 = first excited state, 2 = second excited state, *etc.*; and attach/detach specified from state $1 \rightarrow N_{\mathrm{DA}}$.

By default, all output data are printed to files in the working directory, overwriting any existing file of the same name.

- Molecular orbital data is printed to a file called `plot.mo`.
- Densities are plotted to `plots.hf`.
- Restricted unrelaxed attachment/detachment analysis is sent to:
  - `plot.attach.alpha`
  - `plot.detach.alpha`
- Unrestricted unrelaxed attachment/detachment analysis is sent to:
  - `plot.attach.alpha`
  - `plot.detach.alpha`
  - `plot.attach.beta`
  - `plot.detach.beta`
- Restricted relaxed attachments/detachment analysis is plotted in:
  - `plot.attach.rlx.alpha`
  - `plot.detach.rlx.alpha`
- Unrestricted relaxed attachment/detachment analysis is sent to:
  - `plot.attach.rlx.alpha`
  - `plot.attach.rlx.alpha`
  - `plot.attach.rlx.beta`
  - `plot.detach.rlx.beta`

Output is printed in atomic units, with coordinates first followed by item value, as shown below:

```
x1   y1   z1      a1   a2  ...  aN
x2   y1   z1      b1   b2  ...  bN
...
```

Instead of a standard one-, two-, or three-dimensional Cartesian grid, a user may wish to plot orbitals or densities on a set of grid points of his or her choosing. Such points are specified using a *$grid* input section whose format is simply the Cartesian coordinates of all user-specified grid points:

```
x1   y1   z1
x2   y2   z2
...
```

The *$plots* section must still be specified as described above, but if the *$grid* input section is present, then these user-specified grid points will override the ones specified in the *$plots* section.

The Q-CHEM *$plots* utility allows the user to plot transition densities and detachment/attachment densities directly from amplitudes saved from a previous calculation, without having to solve the post-SCF (CIS, RPA, TDA, or TDDFT) equations again. To take advantage of this feature, the same Q-CHEM scratch directory must be used, and the SKIP_CIS_RPA *$rem* variable must be set to TRUE. To further reduce computational time, the SCF_GUESS *$rem* can be set to READ.

SKIP_CIS_RPA

Skips the solution of the CIS, RPA, TDA or TDDFT equations for wave function analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE / FALSE

RECOMMENDATION:

Set to true to speed up the generation of plot data if the same calculation has been run previously with the scratch files saved.

10.5.4.1 New *$plots* input

New format for the *$plots* section provides readable and friendly input for generation of volumetric data. The input section can be divided into three parts. The first part contains basic plot options which define the 3-D mesh of points. The second part contains the selection of densities or orbitals. The advanced options are included in the last part.

With new plot format, there are multiple ways to define 3-D mesh points. If no plot option is given, the boundaries of the mesh box are the maximum/minimum molecular coordinates $\pm$ 3.0 Å. The default box can be simply enlarged or reduced by setting `grid_range` to a value larger or smaller than 3.0 (negative number is accepted), respectively. To customize the mesh box, set `grid_range` to desired boundaries:

```
$plots
  grid_range (-1,1)  (-1,1)  (-1,1)
$end
```

This defines a 2×2×2 mesh box centered at the molecular coordinate origin. Note that there is no space in the parentheses.

The number of one dimension points is the value of the box length divided by `grid_spacing`. The default grid point spacing is 0.3 Å. To override the usage of `grid_spacing` and customize the number of 3-D points, set `grid_points` to desired values.

To generate cube file (Section 10.5.5) using new plot format, just set MAKE_CUBE_FILES = TRUE in *$rem* section. The new plot format is enabled by requesting PLOTS = 1.

Table 10.3: Options for new *$plots* input section

Option	Explanation
Basic plot options	
grid_range	boundaries† of the mesh box or increment/decrement in the default boundaries†† in Å
grid_spacing	grid point spacing††† in Å
grid_points	$N_x\ N_y\ N_z$
Density/orbital selection⋆	
alpha_molecular_orbital	an integer range of alpha MO's to evaluate
beta_molecular_orbital	an integer range of beta MO's to evaluate
total_density	an integer range of total densities to evaluate
spin_density	an integer range of spin densities to evaluate
transition_density	an integer range of transition densities to evaluate
natural_transition_orbital	an integer range of excited states whose NTOs are evaluated
attachment_detachment_density	an integer range of det.-att. densities to evaluate
natural_bond_orbital	an integer range of NBOs for each state to evaluate
nbo_state	an integer range of states whose NBOs are evaluated
Advanced options	
reduced_density_gradient	invoke non-covalent interaction (NCI) plot
orbital_laplacians	evaluate orbital Laplacians
average_local_ionization	evaluate average local ionization energies [194] with a given contour value of the electron density. The default is $0.0135e/\text{Å}^3$ ($\approx 0.002e/a_0^3$).
elf	invoke electron localization function (ELF) plots

†the format: $(x_{\min}, x_{\max})(y_{\min}, y_{\max})(z_{\min}, z_{\max})$

††the default is 3.0 Å increment in the boundaries derived from the molecular coordinates

†††the default is 0.3 Å; it can be overridden by option 'grid_points'

⋆input format: n-m or n, indicating n-th oribtal or state; use 0 for the ground-state

Example 10.18 Generating the cube files: the total densities of the ground and the first two excited states, the transition and detachment/attachment densities of the first two excited states, and the 28th to 31th alpha molecular orbitals, with customized 3-D mesh box and points.

```
$molecule
0 1
  C    0.0000000   -0.0000000   -0.6133791
  O   -0.0000000    0.0000000    0.6060734
  H    0.0000000    0.9391300   -1.1555819
  H    0.0000000   -0.9391300   -1.1555819
$end

$rem
   METHOD           cis
   BASIS            6-31+G*
   CIS_N_ROOTS      4
   CIS_TRIPLETS     false
   MAKE_CUBE_FILES  true   ! triggers writing of cube files
   PLOTS            true
$end

$plots
   grid_range                     (-8,8) (-8,8) (-8,8)
   grid_points                    40 40 40
   total_density                  0-2
   transition_density             1-2
   attachment_detachment_density  1-2
   alpha_molecular_orbital        28-31
$end
```

Example 10.19 Generating the cube files of the average local ionization energies and the total density for the ground state of aniline.

```
$molecule
0 1
  H        -2.952725   -0.026758    0.000000
  C        -1.871492   -0.010683    0.000000
  C        -1.172124   -0.001127   -1.197270
  H        -1.709244   -0.009471   -2.137819
  C         0.211522    0.017487   -1.202676
  H         0.754733    0.024328   -2.137945
  C         0.916518    0.025234    0.000000
  N         2.357874    0.119819    0.000000
  H         2.747183   -0.346427   -0.829920
  H         2.747183   -0.346427    0.829920
  C         0.211522    0.017487    1.202676
  H         0.754733    0.024328    2.137945
  C        -1.172124   -0.001127    1.197270
  H        -1.709244   -0.009471    2.137819
$end

$rem
   EXCHANGE          hf
   BASIS             6-31g*
   MAKE_CUBE_FILES   true
   PLOTS             true
$end

$plots
   grid_spacing   0.1
   total_density  0
   average_local_ionization
$end
```

10.5.5 Direct Generation of "Cube" Files

As an alternative to the output format discussed above, all of the *$plots* data may be output directly to a sub-directory named `plots` in the job's scratch directory, which must therefore be saved using the *–save* option described in Section 2.2. The plotting data in this sub-directory are not written in the `plot.*` format described above, but rather in the form of so-called "cube" file, one for each orbital or density that is requested. The cube file format is a standard one for volumetric data and consists of a small header followed by the orbital or density values at each grid point, in ASCII format. (Consult Ref. 82 for the complete format specification.) Because the grid coordinates themselves are not printed (their locations are implicit from information contained in the header), each individual cube file is much smaller than the corresponding `plot.*` file would be. Cube files can be read by many standard (and freely-available) graphics programs, including MacMolPlt[282] and VMD.[944] VMD, in particular, is recommended for generation of high-quality images for publication. Cube files for the MOs and densities requested in the *$plots* section are requested by setting MAKE_CUBE_FILES to TRUE, with the *$plots* section specified as described in Section 10.5.4.

MAKE_CUBE_FILES

Requests generation of cube files for MOs, NTOs, or NBOs.

TYPE:

LOGICAL/STRING

DEFAULT:

FALSE

OPTIONS:

FALSE Do not generate cube files.

TRUE Generate cube files for MOs and densities.

NTOS Generate cube files for NTOs.

NBOS Generate cube files for NBOs.

RECOMMENDATION:

None

PLOT_SPIN_DENSITY

Requests the generation of spin densities, ρ_α and ρ_β.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not generate spin density cube files.

TRUE Generate spin density cube files.

RECOMMENDATION:

Set to TRUE if spin densities are desired in addition to total densities. Requires that MAKE_CUBE_FILES be set to TRUE as well, and that one or more total densities is requested in the *$plots* input section. The corresponding spin densities will then be generated also.

The following example illustrates the generation of cube files for a ground and an excited-state density, including the corresponding spin densities. In this example, the `plots` sub-directory of the job's scratch directory should contain files named `dens.N.cube` (total density for state N, where $N = 0$ or 1 represents the ground and first excited state, respectively), `dens_alpha.N.cube` and `dens_beta.N.cube` (ρ_α and ρ_β for each state), and

`dens_spin.N.cube` ($\rho_\alpha - \rho_\beta$ for each state.)

Example 10.20 Generating density and spin-density cube files for the ground and first excited state of the HOO radical.

```
$molecule
   0 2
   H     1.004123   -0.180454    0.000000
   O    -0.246002    0.596152    0.000000
   O    -1.312366   -0.230256    0.000000
$end

$rem
   PLOT_SPIN_DENSITY  true
   MAKE_CUBE_FILES    true
   SCF_CONVERGENCE    8
   METHOD             b3lyp
   BASIS              6-31+G*
   CIS_N_ROOTS        1
$end

$plots
grid information and request to plot 2 densities
   20  -5.0  5.0
   20  -5.0  5.0
   20  -5.0  5.0
   0  2  0  0
   0  1
$end
```

Cube files are also available for natural transition orbitals (Sections 7.14.3 and 10.5.3) by setting MAKE_CUBE_FILES to NTOS, although in this case the procedure is somewhat more complicated, due to the state-specific nature of these quantities. Cube files for the NTOs are generated only for a single excited state, whose identity is specified using CUBEFILE_STATE. Cube files for additional states are readily obtained using a sequence of Q-CHEM jobs, in which the second (and subsequent) jobs read in the converged ground- and excited-state information using SCF_GUESS and SKIP_CIS_RPA.

CUBEFILE_STATE

Determines which excited state is used to generate cube files

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n Generate cube files for the nth excited state

RECOMMENDATION:

None

An additional complication is the manner in which to specify which NTOs will be output as cube files. When MAKE_CUBE_FILES is set to TRUE, this is specified in the *$plots* section, in the same way that MOs would be specified for plotting. However, one must understand the order in which the NTOs are stored. For a system with N_α α-spin MOs, the occupied NTOs $1, 2, \ldots, N_\alpha$ are stored in order of increasing amplitudes, so that the N_α'th occupied NTO is the most important. The virtual NTOs are stored next, in order of *decreasing* importance. According to this convention, the principle NTO pair consists of the final occupied orbital and the first virtual orbital, for any particular excited state.

Thus, orbitals N_α and $N_\alpha + 1$ represent the most important NTO pair, while orbitals $N_\alpha - 1$ and $N_\alpha + 2$ represent the second most important NTO pair, *etc.*.

Example 10.21 Generating cube files for the excitation between the principle occupied and virtual NTOs of the second singlet excited state of uracil. Note that $N_\alpha = 29$ for uracil.

```
$molecule
   0 1
   N    -2.181263     0.068208     0.000000
   C    -2.927088    -1.059037     0.000000
   N    -4.320029    -0.911094     0.000000
   C    -4.926706     0.301204     0.000000
   C    -4.185901     1.435062     0.000000
   C    -2.754591     1.274555     0.000000
   N    -1.954845     2.338369     0.000000
   H    -0.923072     2.224557     0.000000
   H    -2.343008     3.268581     0.000000
   H    -4.649401     2.414197     0.000000
   H    -6.012020     0.301371     0.000000
   H    -4.855603    -1.768832     0.000000
   O    -2.458932    -2.200499     0.000000
$end

$plots
Plot the dominant NTO pair, N --&gt; N+1
   25  -5.0  5.0
   25  -5.0  5.0
   25  -5.0  5.0
   2  0  0  0
   29  30
$end

$rem
   METHOD           B3LYP
   BASIS            6-31+G*
   CIS_N_ROOTS      2
   CIS_TRIPLETS     FALSE
   NTO_PAIRS        TRUE    ! calculate the NTOs
   MAKE_CUBE_FILES  NTOS    ! generate NTO cube files...
   CUBEFILE_STATE   2       ! ...for the 2nd excited state
$end
```

Cube files for Natural Bond Orbitals (for either the ground state or any CIS, RPA, of TDDFT excited states) can be generated in much the same way, by setting MAKE_CUBE_FILES equal to NBOS, and using CUBEFILE_STATE to select the desired electronic state. CUBEFILE_STATE = 0 selects ground-state NBOs. The particular NBOs to be plotted are selected using the *$plots* section, recognizing that Q-CHEM stores the NBOs in order of decreasing occupancies, with all α-spin NBOs preceding any β-spin NBOs, in the case of an unrestricted SCF calculation. (For ground states, there is typically one strongly-occupied NBO for each electron.) NBO cube files are saved to the `plots` sub-directory of the job's scratch directory. One final caveat: to get NBO cube files, the user must specify the AONBO option in the

$nbo section, as shown in the following example.

Example 10.22 Generating cube files for the NBOs of the first excited state of H_2O.

```
$rem
   METHOD           CIS
   BASIS            CC-PVTZ
   CIS_N_ROOTS      1
   CIS_TRIPLETS     FALSE
   NBO              2      ! ground- and excited-state NBO
   MAKE_CUBE_FILES  NBOS   ! generate NBO cube files...
   CUBEFILE_STATE   1      ! ...for the first excited state
$end

$nbo
  AONBO
$end

$molecule
   0 1
   O
   H  1  0.95
   H  1  0.95  2  104.5
$end

$plots
Plot the 5 high-occupancy NBOs, one for each alpha electron
   40  -8.0  8.0
   40  -8.0  8.0
   40  -8.0  8.0
   5  0  0  0
   1  2  3  4  5
$end
```

Example 10.23 NTO and NBO plots with the new format for the *$plots* section: NTOs for the first excited state (two pairs) and five NBOs for both the ground and first excited state are plotted. The aonbo keyword in the nbo section below is required.

```
$molecule
  0 1
   O
   H 1 0.95
   H 1 0.95 2 104.5
$end

$rem
   METHOD      cis
   BASIS      cc-pvtz
   CIS_N_ROOTS       1
   CIS_TRIPLETS      false
   nbo               2  ! ground- and excited-state NBO
   NTO_PAIRS         2  ! triggers NTO analysis
   MAKE_CUBE_FILES   true
   PLOTS             true
$end

$nbo
   aonbo
$end

$plots
   grid_points 50 50 50
   nbo_state   0-1
   natural_bond_orbital 1-5
   natural_transition_orbital 1
$end
```

It is also possible to obtain Molecular Orbital (MO) or ground-state electron density cube files from Complex-Absorbing-Potential Hartree-Fock (CAP-HF) calculations. No adjustment in the plot section is required, however the user should pay attention in order to identify which cube file contains which spatial function. For each requested MO with index N, three files will be printed:

- mo.N.cube contains the real-valued MO prior to addition of the CAP.
- mo.$(1000 + N)$.cube contains the real part of the MO after convergence of the CAP-HF calculation.
- mo.$(1000 + N + o)$.cube contains the imaginary part of the MO after convergence of the CAP-HF calculation. o is equal to the total number of spin orbitals calculated for the molecule.

For the ground-state density, the three files will have indices 0, 1000 and 1001. They contain the density prior to CAP addition, and the real and imaginary density after CAP calculation respectively.

10.5.6 Noncovalent Interactions (NCI) Plots

Weitao Yang and co-workers[47,101] have shown that the reduced density gradient,

$$s(\mathbf{r}) = \left(\frac{1}{2(3\pi^2)^{1/3}}\right) \frac{\|\hat{\boldsymbol{\nabla}}\rho(\mathbf{r})\|}{\rho(\mathbf{r})^{4/3}} \tag{10.37}$$

provides a convenient indicator of noncovalent interactions, which are characterized by large density gradients in regions of space where the density itself is small, leading to very large values of $s(\mathbf{r})$. Q-CHEM can generate noncovalent interactions (NCI) plots of the function $s(\mathbf{r})$ in three-dimensional space. To generate these, set the PLOT_REDUCED_DENSITY_GRAD *$rem* variable to TRUE. (See the `nci-c8h14.in` input example in `$QC/samples` directory.)

10.5.7 Electron Localization Function

Formulated by Becke and Edgecombe,[22] the electron localization function (ELF),

$$\mathrm{ELF} = \left[1 + \left(\frac{\tau_\sigma - \|\hat{\nabla}\rho_\sigma\|^2/(4\rho_\sigma)}{\frac{3}{5}(6\pi^2)^{2/3}(\rho_\sigma^{5/3})}\right)^2\right]^{-1} \tag{10.38}$$

is a measure of electron localization. It is derived from the Hartree-Fock conditional pair probability and can reveal information about bonding and shell structure.[61] The function $\mathrm{ELF}(\mathbf{r})$ has values that lie between 0 and 1, with $\mathrm{ELF} = 1$ representing perfect localization and $\mathrm{ELF} = 1/2$ representing electron-gas-like pair probability. To generate ELF plots with Q-CHEM, set the PLOT_ELF *$rem* variable to TRUE. For closed-shell systems, only the α-spin ELF is calculated whereas for open-shell systems (spin-unrestricted calculations), both α- and β-spin ELFs are computed.[114]

The following example illustrates the calculation of the ELF for a water molecule.

Example 10.24 A job that evaluates the ELF for H_2O on a $50 \times 50 \times 50$ grid. The output is in a cube file called `elf_alpha.0.cube`.

```
$molecule
   0 1
   O    -4.5320698567   0.2524215916     0.0130780103
   H    -3.5641829319   0.2173288989    -0.0173259969
   H    -4.8109190521  -0.4489616171    -0.5945943692
$end

$rem
   JOBTYPE           opt
   METHOD            b3lyp
   BASIS             6-31g*
   PLOT_ELF          true
   MAKE_CUBE_FILES   true
   geom_opt_driver optimize
$end

$plots
water
50 -7 7
50 -4 4
50 -4 4
0 1 0 0
0
$end
```

Please refer also to `elf_methane.in` the `$QC/samples` directory, which uses the newer *$plots* format.

10.5.8 Electrostatic Potentials

Q-CHEM can evaluate electrostatic potentials on a grid of points. Electrostatic potential (ESP) evaluation is controlled by the *$rem* variable ESP_GRID.

Note: For backwards compatibility with the Q-CHEM/CHARMM interface, the name IGDESP is equivalent to ESP_GRID.

ESP_GRID

Controls evaluation of the electrostatic potential on a grid of points. If enabled, the output is in an ASCII file, `plot.esp`, in the format $x, y, z, \phi(x, y, z)$ for each point, where ϕ is the ESP.

TYPE:

INTEGER

DEFAULT:

-4

OPTIONS:

-1 read grid input via the *$plots* section of the input deck

-2 same as the option -1, plus evaluate the ESP of the *$external_charges*

-3 same as the option -1 but in connection with STATE_ANALYSIS = TRUE. This computes the ESP for all excited-state densities, transition densities, and electron/hole densities.

-4 No ESP evaluation

0 Generate the ESP values at all nuclear positions

$+n$ read n grid points **in bohr** from the ASCII file `ESPGrid`

RECOMMENDATION:

For the $+n$ option, the user should also specify NO_REORIENT = TRUE. This forces Q-CHEM to use the user's coordinate frame (as established in the *$molecule* section), which should be the same coordinate frame used in the `ESPGrid` file.

The following example illustrates the evaluation of electrostatic potentials on a grid. Note that IANLTY must also be set to 200.

Example 10.25 A job that evaluates the electrostatic potential for H_2 on a 1 by 1 by 15 grid, along the bond axis. The output is in an ASCII file called `plot.esp`, which lists for each grid point, x, y, z, and the electrostatic potential.

```
$molecule
   0  1
   H    0.0    0.0    0.35
   H    0.0    0.0   -0.35
$end

$rem
   METHOD      hf
   BASIS       6-31g**
   IANLTY      200
   ESP_GRID      -1
$end

$plots
   plot the electrostatic potential on a line
   1    0.0    0.0
   1    0.0    0.0
  15   -3.0    3.0
   0   0   0   0
   0
$end
```

The example below evaluates ESP on a plotting grid and generates a cube file, which can be loaded by IQmol to

visualize ESP on a molecular surface (*e.g.* vdW surface or electron density isosurface).

Example 10.26 Calculating the ESP of acetyl chloride on a user-specified grid and generate a cube file as the result; the new format for the *$plots* section is used

```
$molecule
 0 1
 C       0.00000     1.18959     1.29360
 H       0.00000     1.01580     2.37206
 H       0.88291     1.76694     1.00084
 H      -0.88291     1.76694     1.00084
 C       0.00000    -0.13357     0.57945
 O       0.00000    -1.22212     1.05977
 Cl      0.00000     0.06441    -1.23718
$end

$rem
 jobtype  sp
 method   b3lyp
 basis    6-31g(d)
 point_group_symmetry False
 scf_convergence  8
 esp_grid  -1
 make_cube_files  true
 plots    true
 gui      2
$end

$plots
 grid_points 50 50 50
 grid_range (-4,4) (-5,5) (-5,5)
$end
```

We can also compute the electrostatic potential for the transition density, which can be used, for example, to compute the Coulomb coupling in excitation energy transfer.

ESP_TRANS

Controls the calculation of the electrostatic potential of the transition density

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE compute the electrostatic potential of the excited state transition density

FALSE compute the electrostatic potential of the excited state electronic density

RECOMMENDATION:

NONE

The electrostatic potential is a complicated object and it is not uncommon to model it using a simplified representation based on atomic charges. For this purpose it is well known that Mulliken charges perform very poorly. Several definitions of ESP-derived atomic charges have been given in the literature, however, most of them rely on a least-squares fitting of the ESP evaluated on a selection of grid points. Although these grid points are usually chosen so that the ESP is well modeled in the "chemically important" region, it still remains that the calculated charges will change if the molecule is rotated. Recently an efficient rotationally invariant algorithm was proposed that sought to model the ESP not by direct fitting, but by fitting to the multipole moments.[192] By doing so it was found that the

fit to the ESP was superior to methods that relied on direct fitting of the ESP. The calculation requires the traceless form of the multipole moments and these are also printed out during the course of the calculations. To request these multipole-derived charges, set MM_CHARGES = TRUE in the *$rem* section.

MM_CHARGES

Requests the calculation of multipole-derived charges (MDCs).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculates the MDCs and also the traceless form of the multipole moments

RECOMMENDATION:

Set to TRUE if MDCs or the traceless form of the multipole moments are desired. The calculation does not take long.

10.6 Electric Fields

The derivatives of electrostatic potential (ESP) with respect to positions give electric fields, which is a fundamental physical quantity that has been shown to play an important role in applications ranging from vibrational spectroscopy to molecular and enzyme catalysis. Similar to the case of ESP, Q-CHEM is able to compute the values of E-field on nuclear positions or a given grid, which is also controlled by the *$rem* variable ESP_GRID. The calculation of electric field is turned on when the value of ESP_EFIELD > 0:

ESP_EFIELD

Triggers the calculation of ESP and/or electric field at nuclear positions or on a given grid of points

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Compute ESP only

1 Compute both ESP and electric field

2 Compute electric field only

RECOMMENDATION:

None

Example 10.27 Calculate the electric field on the nuclear positions of a water molecule

```
$molecule
   0 1
   O    -0.9112629280    1.0922672019    1.0200719528
   H    -1.7568362275    1.5186695533    1.2826042030
   H    -0.5592940377    1.7449530375    0.3694007293
$end

$rem
   METHOD       b3lyp
   BASIS        cc-pvtz
   ESP_GRID     0
   ESP_EFIELD   2 ! compute E-field on atomic positions
$end
```

Q-CHEM 6 supports two DFT-based electronic structure methods for the evaluation of electric fields (and also ESPs) arising from a chemical environment (*e.g.* solvents) at the specific sites of a "probe" molecule (typically at the atomic positions). These methods, whose details are provided in Ref. 236, involve a partition of the electron density of the entire system into those belonging to the central system and to the environment. The first approach is based on the SPADE partitioning scheme (see Section 11.6.1),[46] which transforms the converged occupied MOs obtained from a standard SCF calculation and then assigns them to different parts of a system. The second approach is based on DFT calculations using absolutely localized molecular orbitals (ALMOs, see Section 12.5.1),[107] which invokes fragmentation of the supersystem from the beginning and the fragment MOs are then variationally optimized when they are polarized by other fragments. Both of these methods then construct the electron density for the "environment" part of the supersystem ($\rho_{\rm E}$) using the correspondingly assigned occupied MOs, and the ESP and electric field at a specific site of the central "probe" system can be calculated using the electron density and nuclear charges of the environment. Denoting the embedded central system and its environment as S and E, respectively, the ESP and electric field vector at site $A \in$ S (ϕ_A and $\mathbf{F}_A$) can be evaluated using

$$\phi_A = \sum_{B\in \rm E} \frac{Z_B}{|\mathbf{r}_A - \mathbf{r}_B|} - \int \frac{\rho_{\rm E}(\mathbf{r})}{|\mathbf{r}_A - \mathbf{r}|} {\rm d}\mathbf{r} \tag{10.39}$$

and

$$\mathbf{F}_A = -\hat{\boldsymbol{\nabla}}\phi_A = \sum_{B\in \rm E} \frac{Z_B(\mathbf{r}_A - \mathbf{r}_B)}{|\mathbf{r}_A - \mathbf{r}_B|^3} - \int \frac{\rho_{\rm E}(\mathbf{r})(\mathbf{r}_A - \mathbf{r})}{|\mathbf{r}_A - \mathbf{r}|^3} {\rm d}\mathbf{r}\ . \tag{10.40}$$

The current implementation of these methods requires two Q-CHEM jobs to be performed for a single environment ESP and electric field calculation. In the first job, one performs an SCF or SCF-MI calculation and generate the occupied MOs or electron density assigned to the environment; in the second job, the environment MOs or electron density is read in and the ESP and electric fields are calculated using Eqs. (10.39) and (10.40). Note that in the second job, the embedded central system ("probe") is represented using ghost atoms to probe the nuclear positions; and for SPADE and ALMO the SCF_GUESS for the 2nd job must be READ and READ_DEN, respectively. To evaluate the potential and field at the nuclear positions of the central system, one should set ESP_GRID = 0; for other options (*e.g.* evaluating the ESP and its gradient on grid points), one should refer to the documentation of "ESP_GRID" in Section 10.5.8.

ALMO_EFIELD

Calculate the environment ESP/E-field using ALMO-based partitioning

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE In job 1, it saves the electron density for the environment constructed from ALMOs; In job 2, it reads in the electron density (must be together with SCF_GUESS = READ_DEN)

FALSE Don't do ALMO-based ESP/field calculations

RECOMMENDATION:

Required for both jobs in ALMO-based electric field calculations

ALMO_EFIELD_PROBE_FRGM

Specify the index of the probe fragment in ALMO-based ESP and electric field calculations

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Specify the nth fragment as the probe

RECOMMENDATION:

None

Example 10.28 Using the SPADE partitioning method to calculate the ESP and electric field arising from the environment (the second H_2O molecule) at the atomic positions of the 1st H_2O

```
$molecule
0 1
--
0 1
O  -1.551007  -0.114520   0.000000
H  -1.934259   0.762503   0.000000
H  -0.599677   0.040712   0.000000
--
0 1
O   1.350625   0.111469   0.000000
H   1.680398  -0.373741  -0.758561
H   1.680398  -0.373741   0.758561
$end

$rem
jobtype  sp
method   b3lyp
basis    6-31G(d)
env_method b3lyp
gen_scfman_embed true
spade_partition  true
scf_convergence  8
embedding_early_stop true ! skip the embedded SCF
integral_symmetry    false
point_group_symmetry false
$end

@@@

$molecule
0 1
@O  -1.551007  -0.114520   0.000000
@H  -1.934259   0.762503   0.000000
@H  -0.599677   0.040712   0.000000
O    1.350625   0.111469   0.000000
H    1.680398  -0.373741  -0.758561
H    1.680398  -0.373741   0.758561
$end

$rem
method      b3lyp
basis       6-31G(d)
scf_guess   read
skip_scfman true ! generate results directly from the MOs read in
esp_grid    0
esp_efield  1    ! compute ESP and E-field on atomic positions
integral_symmetry    false
point_group_symmetry false
$end
```

Example 10.29 Using ALMO-based partitioning to calculate the ESP and electric field arising from the environment (the second H_2O molecule) at the atomic positions of the 1st H_2O

```
$molecule
0 1
--
0 1
O  -1.551007  -0.114520   0.000000
H  -1.934259   0.762503   0.000000
H  -0.599677   0.040712   0.000000
--
0 1
O   1.350625   0.111469   0.000000
H   1.680398  -0.373741  -0.758561
H   1.680398  -0.373741   0.758561
$end

$rem
jobtype  sp
method   b3lyp
basis    6-31G(d)
scf_convergence  8
frgm_method   stoll  ! doing SCF-MI (ALMO) calculation
scfmi_mode    1
almo_efield   true   ! save electron density belonging to the 2nd fragment
integral_symmetry    false
point_group_symmetry false
$end

@@@

$molecule
0 1
@O  -1.551007  -0.114520   0.000000
@H  -1.934259   0.762503   0.000000
@H  -0.599677   0.040712   0.000000
O    1.350625   0.111469   0.000000
H    1.680398  -0.373741  -0.758561
H    1.680398  -0.373741   0.758561
$end

$rem
jobtype sp
method  b3lyp
basis   6-31G(d)
scf_guess  read_den
almo_efield true   ! with read_den, this will read in the density saved in the 1st job
skip_scfman true
esp_grid    0
esp_efield  1  ! compute ESP and E-field on atomic positions
integral_symmetry    false
point_group_symmetry false
$end
```

10.7 Harmonic Vibrational Analysis

10.7.1 Overview

Vibrational analysis is an extremely important tool for the quantum chemist, supplying a molecular fingerprint which is invaluable for aiding identification of molecular species in many experimental studies. Q-CHEM includes a vibrational analysis package that can calculate vibrational frequencies and their infrared and Raman activities.[99] Vibrational frequencies are calculated by either using an analytic Hessian (if available) or else by numerical finite difference of the gradient or, as a last resort, via double finite-difference of single-point energies. (See Table 9.2 for the availability of analytic gradients and Hessians. The finite-difference step size is controlled by FDIFF_STEPSIZE.) The default setting in Q-CHEM is to use the highest analytical derivative order available for the requested theoretical method. The performance of various *ab initio* theories in determining vibrational frequencies has been well documented.[100,143,188]

When calculating analytic frequencies at the HF and DFT levels of theory, the coupled-perturbed SCF equations must be solved. This is the most time-consuming step in the calculation, and also consumes the most memory. The amount of memory required is $\mathcal{O}(N^2M)$ where N is the number of basis functions, and M the number of atoms. This is an order more memory than is required for the SCF calculation, and is often the limiting consideration when treating larger systems analytically. Q-CHEM incorporates a new approach to this problem that avoids this memory bottleneck by solving the CPSCF equations in segments.[116] Instead of solving for all the perturbations at once, they are divided into several segments, and the CPSCF is applied for one segment at a time, resulting in a memory scaling of $\mathcal{O}(N^2M/N_{\mathrm{seg}})$, where N_{seg} is the number of segments. This option is invoked automatically by the program.

After a vibrational analysis, Q-CHEM computes useful statistical thermodynamic properties at standard temperature and pressure following the rigid-rotor-harmonic-oscillator (RRHO) approach. These include the zero-point vibration energy (ZPVE) and, translational, rotational and vibrational, entropies and enthalpies. Note: in the Q-CHEM output the "total enthalpy" actually means the total enthalpy correction to the internal energy. One must add this "total enthalpy" to the internal energy to obtain the total enthalpy in common sense. In addition to these thermal corrections, Q-CHEM also prints the interpolated vibrational entropy and enthalpy according to the quasi-RRHO (qRRHO) approach by Head-Gordon and co-workers,[120] which extends Grimme's previous scheme[71] to address low-frequency vibrations; Section 10.7.3.

10.7.1.1 Job Control

In order to carry out a frequency analysis users must *at a minimum* provide a molecule within the *$molecule* keyword and define an appropriate level of theory within the *$rem* keyword using the *$rem* variables EXCHANGE, CORRELATION (if required) (Chapter 4) and BASIS (Chapter 8). Since the default type of job (JOBTYPE) is a single point energy (SP) calculation, the JOBTYPE *$rem* variable must be set to FREQ.

It is very important to note that a vibrational frequency analysis must be performed at a stationary point on the potential surface that has been optimized at the same level of theory. Therefore a vibrational frequency analysis most naturally follows a geometry optimization in the same input deck, where the molecular geometry is obtained (see examples).

Users should also be aware that the quality of the quadrature grid used in DFT calculations is more important when calculating second derivatives. The default grid for some atoms has changed in Q-CHEM 3.0 (see Section 5.5) and for this reason vibrational frequencies may vary slightly form previous versions. It is recommended that a grid larger than the default grid is used when performing frequency calculations.

The standard output from a frequency analysis includes the following.

- Vibrational frequencies.

- Raman and IR activities and intensities (requires *$rem* DORAMAN).
- Atomic masses.
- Zero-point vibrational energy.
- Translational, rotational, and vibrational, entropies and enthalpies.

Several other *$rem* variables are available that control the vibrational frequency analysis. In detail, they are:

DORAMAN

Controls calculation of Raman intensities. Requires JOBTYPE to be set to FREQ

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate Raman intensities.

TRUE Do calculate Raman intensities.

RECOMMENDATION:

None

VIBMAN_PRINT

Controls level of extra print out for vibrational analysis.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Standard full information print out.
If VCI is TRUE, overtones and combination bands are also printed.

3 Level 1 plus vibrational frequencies in atomic units.

4 Level 3 plus mass-weighted Hessian matrix, projected mass-weighted Hessian matrix.

6 Level 4 plus vectors for translations and rotations projection matrix.

RECOMMENDATION:

Use the default, unless the Hessian is desired.

CPSCF_NSEG

Controls the number of segments used to calculate the CPSCF equations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Determine the number of segments based on the memory request and MEM_TOTAL

n User-defined. Use n segments when solving the CPSCF equations.

RECOMMENDATION:

Use the default.

Example 10.30 EDF1/6-31+G* optimization followed by vibrational analysis. Doing the vibrational analysis at a stationary point is necessary for the results to be valid.

```
$molecule
   0  1
   O
   C  1  co
   F  2  fc  1  fco
   H  2  hc  1  hco  3  180.0

   co  =   1.2
   fc  =   1.4
   hc  =   1.0
   fco = 120.0
   hco = 120.0
$end

$rem
   JOBTYPE     opt
   METHOD      edf1
   BASIS       6-31+G*
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE     freq
   METHOD      edf1
   BASIS       6-31+G*
$end
```

10.7.1.2 Interpreting the Output

Numerical values in the following discussion correspond to Example 10.30 and the referenced partition functions come from the textbook by McQuarrie & Simon.[138] Note that Q-CHEM assumes $T = 298.15$ K and $P = 1.00$ atm by default; see Section 10.7.2 for instructions on how to modify these choices.

The quantity listed as zero-point vibrational energy (ZPVE),

```
Zero point vibrational energy:        12.695 kcal/mol
```

corresponds to ZPVE $= \frac{1}{2}\sum_K h c \nu_K$. Within Q-CHEM the vibrational entropy (S_{vib}) is computed from the bottom of the well rather than from the zero-point level, and thus includes the ZPE. This corresponds to a vibrational partition function

$$q_{\text{vib}}(T) = \prod_K \frac{e^{-\Theta_{\nu,K}/2T}}{1 - e^{-\Theta_{\nu,K}/T}} \tag{10.41}$$

where K indexes vibrational modes and $\Theta_K = h\nu_K/k_B$ is the vibrational temperature of the Kth mode. This expression can be altered by a factor of $\Theta_K/2$ to start at the first vibrational level rather than the bottom of the well. The

vibrational entropy is then

$$\begin{aligned} S_{\text{vib}} &= k_B T \left(\frac{\partial \ln q_{\text{vib}}}{\partial T} \right)_V + k_B \ln q_{\text{vib}} \\ &= k_B T \sum_K \left[\frac{\Theta_K/T}{e^{-\Theta_K/T} - 1} - \ln(1 - e^{-\Theta_K/T}) \right] , \end{aligned} \tag{10.42}$$

and for Example 10.30 it is reported as

```
Vibrational Entropy:              0.620   cal/mol.K
```

The vibrational enthalpy includes the ZPVE along with an additional temperature-dependent term:

$$H_{\text{vib}} = hc \sum_K \left[\frac{\nu_K}{2} + \frac{\nu_K}{e^{\Theta_K/T} - 1} \right] . \tag{10.43}$$

It is reported as

```
Vibrational Enthalpy:          12.839 kcal/mol
```

The rotational and translational enthalpy are multiples of RT,

$$H_{\text{total}} = H_{\text{vib}} + H_{\text{rot}} + H_{\text{trans}} + RT , \tag{10.44}$$

where the final contribution of RT comes from the definition $H = U + PV$. The quantity H_{total} reported as

```
Total Enthalpy:                15.209 kcal/mol
```

Translational and rotational entropies can be computed from the corresponding ideal-gas partition functions.[138]

For thermochemical calculations performed in implicit solvent (using models described in Section 11.2), there is a subtle point with some controversy attached.[37,83,90,184] This regards whether vibrational contributions to the free energy should be included or not. In nearly all cases, continuum solvation models are parameterized in order to reproduce experimental free energies of solvation ($\Delta_{\text{solv}}G^\circ$) using rigid gas-phase geometries for the solute molecules. There is a potential double-counting problem if $-TS_{\text{vib}}$ from a vibrational frequency calculation in implicit solvent is added, which might be avoided by instead using gas-phase harmonic frequencies for the solute.[90] However, the difference between these procedures is negligible (~ 0.2 kcal/mol). for the small-molecule data sets that are used to train implicit solvent models,[184] and only by using solution-phase vibrational frequencies can one obtain corrections to S_{vib} arising from solvation-induced changes in geometry, which might be significant for larger molecules. Note that Q-CHEM's harmonic frequency engine assumes a gas-phase molecule (even in the presence of continuum solvent), so that rotational and translational contributions to the free energy are printed out in every case. These should not be included in solution-phase free energy differences.

10.7.1.3 Finite-Difference Frequency Calculation with Molecular Segments

For electronic structure methods whose analytic 2nd nuclear derivatives are not implemented in Q-CHEM, finite-difference hessian calculations based on nuclear gradients will be performed. The Hessian matrix elements are calculated as

$$H_{ij} = \frac{\partial^2 E}{\partial x_i \partial x_j} = \frac{g_i(x_j + \Delta x_j) - g_i(x_j - \Delta x_j)}{2\Delta x_j} \tag{10.45}$$

where $g_i = \partial E/\partial x_i$. In total, $6N_{\text{atom}}$ gradient calculations at displaced structures are needed to construct the Hessian.

By default, the $6N_{\mathrm{atom}}$ gradient calculations are performed in serial, which is extremely inefficient for molecules of a large number of atoms. Taking advantage of the fact that these finite-difference calculations are independent of each other, Q-CHEM provides the option to perform these calculations in molecular segments (*i.e.*, atom blocks). The user will need to specify the number of atoms in each segment using *$rem* variable FD2ND_BLOCK_SIZE. The total number of segments will then be $\lfloor N_{\mathrm{atom}}/\mathrm{blocksize} \rfloor + 1$ (note: the number of atoms in the last segment is allowed to be smaller than the user-specified block size). The user will then need to set up and execute a series of input with FD2ND_BLOCK_INDEX set to differnt values (starting from 0). Note that all these jobs should share the same value of FD2ND_BLOCK_SIZE. On a large computer cluster, these jobs can easily be executed simultaneously. These jobs will produce text files `grdntDisp.0`, `grdntDisp.1`, ... and `dipoleDisp.0`, `dipoleDisp.1`... under the working directory. Note that IDERIV = 1 needs to be added if it is not set automatically.

After all the calculations for molecule segments finish, the user will need to perform an additional job with FD2ND_BLOCK_INDEX set to -1. This job will collect the results for all the segments (from the text files in the working directory), construct the Hessian, and complete the vibrational frequency analysis. This approach will greatly speed up the finite-difference frequency calculations for large molecules.

FD2ND_BLOCK_SIZE

Controls the number of atoms in each segment of finite-difference Hessian calculation.

TYPE:

INTEGER

DEFAULT:

Uninitialized

OPTIONS:

n Having n atoms in each segment

RECOMMENDATION:

The value can be estimated based on the desired number of segments. It should be significantly smaller than the total number of atoms in the molecule.

FD2ND_BLOCK_INDEX

The flag that species which atom block is being processed in a given finite-difference calculation.

TYPE:

INTEGER

DEFAULT:

Uninitialized

OPTIONS:

$0, 1, ..., N_{\mathrm{frag}} - 1$ The index of the first fragment is 0 and the last should be $N_{\mathrm{frag}} - 1$

-1 Indicator of the job that wraps up the result.

RECOMMENDATION:

None

Example 10.31 Finite-difference frequency calculation for a water dimer with two segments. In a production calculation, the first two jobs should be completed first as two separate calculations, and the 3rd job will be executed afterwards as a wrap-up.

```
$molecule
0 1
O    -1.48239   -0.12728    0.00000
H    -1.90743    0.74236    0.00000
H    -0.52048    0.04679    0.00000
O     1.37121    0.12361    0.00000
H     1.71353   -0.35106    0.77313
H     1.71353   -0.35106   -0.77313
$end

$rem
JOBTYPE  FREQ
METHOD  B3LYP
BASIS  6-31+G(D)
SYM_IGNORE  TRUE
THRESH      14
SCF_CONVERGENCE  8
IDERIV      1
FD2ND_BLOCK_SIZE  3
FD2ND_BLOCK_INDEX 0
$end

@@@

$molecule
read
$end

$rem
JOBTYPE  FREQ
METHOD  B3LYP
BASIS  6-31+G(D)
SYM_IGNORE  TRUE
THRESH      14
SCF_CONVERGENCE  8
IDERIV      1
FD2ND_BLOCK_SIZE  3
FD2ND_BLOCK_INDEX 1
$end

@@@

$molecule
read
$end

$rem
JOBTYPE  FREQ
METHOD  B3LYP
BASIS  6-31+G(D)
SYM_IGNORE  TRUE
THRESH      14
SCF_CONVERGENCE  8
IDERIV      1
FD2ND_BLOCK_SIZE  3
FD2ND_BLOCK_INDEX -1
$end
```

10.7.2 Isotopic Substitutions and Changes in T and P

By default Q-CHEM calculates vibrational frequencies using the atomic masses of the most abundant isotopes.[221] Masses of other isotopes can be specified using the *$isotopes* section and by setting the ISOTOPES *$rem* variable to TRUE. The format of the *$isotopes* section is as follows:

```
$isotopes
   number_of_isotope_loops  tp_flag
   number_of_atoms  [temp pressure] (loop 1)
   atom_number1   mass1
   atom_number2   mass2
   ...
   number_of_atoms  [temp pressure] (loop 2)
   atom_number1   mass1
   atom_number2   mass2
   ...
$end
```

Note: Only the atoms whose masses are to be changed from the default values need to be specified. After each loop all masses are reset to the default values. Atoms are numbered according to the order in the *$molecule* section.

An initial loop using the default masses is always performed first. Subsequent loops use the user-specified atomic masses. Only those atoms whose masses are to be changed need to be included in the list, all other atoms will adopt the default masses. The output gives a full frequency analysis for each loop. Note that the calculation of vibrational frequencies in the additional loops only involves a rescaling of the computed Hessian, and therefore takes little additional computational time.

The first line of the *$isotopes* section specifies the number of substitution loops and also whether the temperature and pressure should be modified. The *tp_flag* setting should be set to 0 if the default temperature and pressure are to be used (298.15 K and 1 atm respectively), or else to 1 if they are to be altered. Note that the temperatures should be specified in Kelvin and pressures in atmospheres. See Example 10.33.

ISOTOPES

Specifies if non-default masses are to be used in the frequency calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use default masses only.

TRUE Read isotope masses from *$isotopes* section.

RECOMMENDATION:

None

Example 10.32 An EDF1/6-31G* optimization, followed by a vibrational analysis including isotopic substitution. Doing the vibrational analysis at a stationary point is necessary for the results to be valid.

```
$molecule
   0  1
   C   1.08900   0.00000   0.00000
   C  -1.08900   0.00000   0.00000
   H   2.08900   0.00000   0.00000
   H  -2.08900   0.00000   0.00000
$end

$rem
   BASIS    6-31G*
   JOBTYPE  opt
   METHOD   edf1
$end

@@@

$molecule
   read
$end

$rem
   BASIS       6-31G*
   JOBTYPE     freq
   METHOD      edf1
   SCF_GUESS   read
   ISOTOPES    1
$end

$isotopes
   2   0           ! two loops, both at std temp and pressure
   4
     1   13.00336  ! All atoms are given non-default masses
     2   13.00336
     3   2.01410
     4   2.01410
   2
     3   2.01410   ! Hs replaced with Ds
     4   2.01410
$end
```

Example 10.33 EDF1/6-31G* optimization followed by vibrational analysis in which the vibrational enthalpy and entropy and computed at different temperatures and pressures.

```
$molecule
   0 1
   O    0.000000    0.000000    0.106731
   H   -0.758095    0.000000   -0.528927
   H    0.758095    0.000000   -0.528927
$end

$rem
   JOBTYPE    opt
   METHOD     edf1
   BASIS      6-31G*
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE    freq
   METHOD     edf1
   BASIS      6-31G*
   ISOTOPES   true
$end

$isotopes
   3   1              ! "1" here means we will set T & P
   0   500.0   1.0    ! 500 K, 1 atm, standard masses
   0   298.0   10.0   ! 298 K, 10 atm, standard masses
   2   500.0   1.0    ! 500 K, 1 atm, subst. D for H
     2   2.0141018
     3   2.0141018
$end
```

10.7.3 Treatment of Low-Frequency Vibrational Modes

Low-frequency vibrational modes usually emerge due to hindered or near-free rotations around a single bond within a molecule. The harmonic approximation is problematic for such internal rotations and yields an infinite vibrational entropy S_{vib} in the limit of vanishing frequencies.[49] To fix this issue, Grimme[71] proposed to enforce a finite vibrational entropy by interpolating between the entropy of the free rotor S_{FR} and the harmonic vibrational entropy S_{HO} as

$$S_{\mathrm{vib}}(\nu_i) = \big(1-\omega(\nu_i)\big)S_{\mathrm{FR}}(\nu_i) + \omega(\nu_i)S_{\mathrm{HO}}(\nu_i), \tag{10.46}$$

where

$$\omega(\nu_i) = \frac{1}{1+(\nu_0/\nu_i)^{\alpha}} \tag{10.47}$$

is the damping function of Chai and Head-Gordon,[41] with α and ν_0 as fixed parameters. It is used as a weighting function and allows for a smooth transition from the free rotor entropy at small frequencies to the harmonic vibrational entropy for frequencies ν_i above the cutoff ν_0. This was later extended to interpolate the vibrational enthalpy contributions between a free rotor H_{FR} and a harmonic oscillator H_{HO} with zero-point vibrational energy H_{ZPVE}, as[120]

$$H_{\mathrm{vib}}(\nu_i) = \big[1-\omega(\nu_i)\big]H_{\mathrm{FR}}(\nu_i) + \omega(\nu_i)\big[H_{\mathrm{HO}}(\nu_i) + H_{\mathrm{ZPVE}}(\nu_i)\big]\ . \tag{10.48}$$

This again reduces the error associated with treating translational and rotational degrees of freedom as low-frequency vibrations, which is especially important for the adsorption or association of larger molecules. This procedure is known as the quasi-rigid-rotor-harmonic-oscillator (qRRHO) approach.

The qRRHO scheme is the default in Q-CHEM, and all thermodynamic quantities are printed for the RRHO (without interpolations) and qRRHO schemes at standard temperature and pressure (298.15 K and 1.00 atm). To change the latter, the user can specify an *$isotopes* section (see examples and Section 10.7.2 for details). α and ω_0 for both interpolator functions can be modified through QRRHO_ALPHA and QRRHO_OMEGA_CUTOFF respectively.

QRRHO_ALPHA

Specifies the exponent in the damping function of Chai and Head-Gordon, used for interpolating the vibrational enthalpy and entropy in the qRRHO scheme. Specify MRRHO_ALPHA to change the exponent for the entropy interpolation separately.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

α Dimensionless interpolator exponent used in the qRRHO scheme.

RECOMMENDATION:

Use the default.

QRRHO_OMEGA_CUTOFF

Sets the frequency cutoff in the Chai-Head-Gordon damping function for interpolating the vibrational enthalpy and entropy in the qRRHO scheme. Specify MRRHO_OMEGA_CUTOFF to change the frequency cutoff for the entropy interpolation separately.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

ω_0 Interpolator cutoff frequency used in the qRRHO scheme in cm^{-1}.

RECOMMENDATION:

Use the default.

Example 10.34 Harmonic vibrational analysis at the HF/3-21G level of theory, where the thermodynamic properties for RRHO and qRRHO are printed at 298.15 K and 1.00 atm.

```
$molecule
0 1
C     1.682185104800      0.240320237000      0.000000000000
O     0.894276382600      1.089998584000      0.000000000000
O     2.480949717100     -0.590833069200      0.000000000000
C    -1.682185094800     -0.240320333400      0.000000000000
O    -0.894276273900     -1.089998812500      0.000000000000
O    -2.480949835800      0.590833394100      0.000000000000
$end

$rem
JOBTYPE         opt
METHOD          hf
BASIS           3-21G
INTEGRAL_SYMMETRY    false
POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
read
$end

$rem
JOBTYPE         freq
METHOD          hf
BASIS           3-21G
INTEGRAL_SYMMETRY    false
POINT_GROUP_SYMMETRY false
$end
```

Example 10.35 Harmonic vibrational analysis at the HF/3-21G level of theory, where the thermodynamic properties for RRHO and qRRHO are printed at standard temperature and pressure as well as two additional temperatures (273.15 K and 313.15 K). Please see Section 10.7.2 for further details on the *$isotopes* section.

```
$molecule
0 1
C     1.682185104800      0.240320237000      0.000000000000
O     0.894276382600      1.089998584000      0.000000000000
O     2.480949717100     -0.590833069200      0.000000000000
C    -1.682185094800     -0.240320333400      0.000000000000
O    -0.894276273900     -1.089998812500      0.000000000000
O    -2.480949835800      0.590833394100      0.000000000000
$end

$rem
JOBTYPE         opt
METHOD          hf
BASIS           3-21G
INTEGRAL_SYMMETRY    false
POINT_GROUP_SYMMETRY false
$end

@@@

$molecule
read
$end

$rem
JOBTYPE         freq
METHOD          hf
BASIS           3-21G
ISOTOPES        true
INTEGRAL_SYMMETRY    false
POINT_GROUP_SYMMETRY false
$end

$isotopes
2 1
0 273.15 1.0
0 313.15 1.0
$end
```

10.7.4 Partial Hessian Vibrational Analysis

The computation of harmonic frequencies for systems with a very large number of atoms can become computationally expensive. However, in many cases only a few specific vibrational modes or vibrational modes localized in a region of the system are of interest. A typical example is the calculation of the vibrational modes of a molecule adsorbed on a surface. In such a case, only the vibrational modes of the adsorbate are useful, and the vibrational modes associated with the surface atoms are of less interest. If the vibrational modes of interest are only weakly coupled to the vibrational modes associated with the rest of the system, it can be appropriate to adopt a partial Hessian approach. In this approach,[25,26] only the part of the Hessian matrix comprising the second derivatives of a subset of the atoms defined by the user is computed. These atoms are defined in the *$alist* block. This results in a significant decrease in the cost of the calculation. Physically, this approximation corresponds to assigning an infinite mass to all the atoms excluded from the Hessian and will only yield sensible results if these atoms are not involved in the vibrational modes of interest. VPT2

and TOSH anharmonic frequencies can be computed following a partial Hessian calculation.[75] It is also possible to include a subset of the harmonic vibrational modes with an anharmonic frequency calculation by setting ANHAR_SEL = TRUE in the *$rem* section. This can be useful to reduce the computational cost of an anharmonic frequency calculation or to explore the coupling between specific vibrational modes.

Alternatively, vibrationally averaged interactions with the rest of the system can be folded into a partial Hessian calculation using vibrational subsystem analysis.[229,237] Based on an adiabatic approximation, this procedure reduces the cost of diagonalizing the full Hessian, while providing a local probe of fragments vibrations, and providing better than partial Hessian accuracy for the low frequency modes of large molecules.[65] Mass-effects from the rest of the system can be vibrationally averaged or excluded within this scheme.

PHESS

Controls whether partial Hessian calculations are performed.

TYPE:

INTEGER

DEFAULT:

0 Full Hessian calculation

OPTIONS:

1 Partial Hessian calculation.

2 Vibrational subsystem analysis (massless).

3 Vibrational subsystem analysis (weighted).

RECOMMENDATION:

None

N_SOL

Specifies number of atoms included in the Hessian.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

PH_FAST

Lowers integral cutoff in partial Hessian calculation is performed.

TYPE:

LOGICAL

DEFAULT:

FALSE Use default cutoffs

OPTIONS:

TRUE Lower integral cutoffs

RECOMMENDATION:

None

Example 10.36 This example shows a partial Hessian frequency calculation of the vibrational frequencies of acetylene on a model of the C(100) surface

```
$comment
   acetylene - C(100)
   partial Hessian calculation
$end

$molecule
0 1
   C  0.000  0.659 -2.173
   C  0.000 -0.659 -2.173
   H  0.000  1.406 -2.956
   H  0.000 -1.406 -2.956
   C  0.000  0.786 -0.647
   C  0.000 -0.786 -0.647
   C  1.253  1.192  0.164
   C -1.253  1.192  0.164
   C  1.253 -1.192  0.164
   C  1.297  0.000  1.155
   C -1.253 -1.192  0.164
   C  0.000  0.000  2.023
   C -1.297  0.000  1.155
   H -2.179  0.000  1.795
   H -1.148 -2.156  0.654
   H  0.000 -0.876  2.669
   H  2.179  0.000  1.795
   H -1.148  2.156  0.654
   H -2.153 -1.211 -0.446
   H  2.153 -1.211 -0.446
   H  1.148 -2.156  0.654
   H  1.148  2.156  0.654
   H  2.153  1.211 -0.446
   H -2.153  1.211 -0.446
   H  0.000  0.876  2.669
$end

$rem
   JOBTYPE           freq
   METHOD            hf
   BASIS             sto-3g
   PHESS             TRUE
   N_SOL             4
$end

$alist
1
2
3
4
$end
```

10.7.5 Localized Mode Vibrational Analysis

The computation of harmonic frequencies leads to molecular vibrations described by coordinates which are often highly de-localized. For larger molecules many vibrational modes can potentially contribute to a single observed spectral band, and information about the interaction between localized chemical units can become less readily available. In certain cases, localizing vibrational modes using procedures similar to the localized orbital schemes discussed

previously in this manual can therefore provide a more chemically intuitive way of analyzing spectral data,[95–97] interpreting two-dimensional vibrational spectra,[76] or improving calculations that go beyond the harmonic approximation.[43,74,162] It is also possible to include only a subset of the normal modes in the localization calculation by invoking the LOCALFREQ_SELECT rem variable. This can be useful to improve convergence in larger molecules or to explore the coupling between specific vibrational modes. These modes are defined in the *$alist* block. Alternatively it is possible to localize high and low frequency modes separately in a single calculation using LOCALFREQ_GROUPS and related inputs.

LOCALFREQ

Controls whether a vibrational mode localization calculation is performed.

TYPE:

INTEGER

DEFAULT:

0 Normal mode calculation.

OPTIONS:

1 Localized mode calculation with a Pipek-Mezey like criterion.

2 Localized mode calculation with a Boys like criterion.

RECOMMENDATION:

None

LOCALFREQ_THRESH

Mode localization is considered converged when the change in the localization criterion is less than $10^{-\mathrm{LOCALFREQ\_THRESH}}$.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n User-specified integer.

RECOMMENDATION:

None

LOCALFREQ_MAX_ITER

Controls the maximum number of mode localization sweeps permitted.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

n User-specified integer.

RECOMMENDATION:

None

LOCALFREQ_SELECT

Select a subset of normal modes for subsequent anharmonic frequency analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE Use all normal modes.

OPTIONS:

TRUE Select a subset of normal modes.

RECOMMENDATION:

None

LOCALFREQ_GROUPS

Select the number of groups of frequencies to be localized separately within a localized mode calculation. The size of the groups are then controlled using the LOCALFREQ_GROUP1, LOCALFREQ_GROUP2, and LOCALFREQ_GROUP3 keywords.

TYPE:

INTEGER

DEFAULT:

0 Localize all normal modes together.

OPTIONS:

1 Define one subset of modes to localize independently.

2 Define two subsets of modes to localize independently.

3 Define three subsets of modes to localize independently.

RECOMMENDATION:

None

LOCALFREQ_GROUP1

Select the number of modes to include in the first subset of modes to localize independently when the keyword LOCALFREQ_GROUPS > 0.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n User-specified integer.

RECOMMENDATION:

Modes will be included starting with the lowest frequency mode and then in ascending energy order up to the defined value.

LOCALFREQ_GROUP2 and LOCALFREQ_GROUP3 are defined similarly.

10.7.6 Resonance Raman intensities

The theory of resonance Raman spectroscopy is fully described by the Kramers-Heisenberg-Dirac dispersion formalism based on the Raman polarizability tensor[106]

$$\alpha_{\sigma\tau}(\omega_{\mathrm{L}},\omega_{\mathrm{S}}) = \sum_{v}\left[\frac{\langle f|\hat{r}_{\sigma}|v\rangle\langle v|\hat{r}_{\tau}|i\rangle}{\hbar\omega_{\mathrm{L}} - \hbar\omega_{vi} + i\Gamma_{iv}} + \frac{\langle f|\hat{r}_{\tau}|v\rangle\langle v|\hat{r}_{\sigma}|i\rangle}{\hbar\omega_{vi} + \hbar\omega_{\mathrm{S}} + i\Gamma_{iv}}\right] \tag{10.49}$$

between initial state $|i\rangle$ and final state $|f\rangle$. Here, ω_{L} and ω_{S} are the frequencies of the laser (incident photon) and of the scattered photon, respectively. Eq. (10.49) is inconvenient due to the sum over intermediate states υ (vibrational levels on all accessible electronic states), and the usual procedure is to expand the static molecular polarizability as a Taylor series in the normal coordinates,[9,210] which allows the Raman intensity to be decomposed into Franck–Condon (or "A-term") contributions and coordinate-dependent Herzberg–Teller ("B"- and "C"-term) contributions.[9,106,210] Nevertheless, each term contains sums over intermediate vibrational states and becomes difficult to evaluate for large molecules with numerous vibrational modes.

As such, in most cases only the lowest-lying Raman-active excited electronic state is considered in computing the RR spectrum. In principle one should consider the effects of Duschinsky rotation,[191] *i.e.*, the fact that the normal modes are different in each electronic state. Neglecting this effect for simplicity and thus using ground-state normal modes only, one arrives at the "independent-mode, displaced harmonic oscillator" (IMDHO) model,[166] in which resonant enhancements to the vibrational intensities (for modes 1 and 2, say) are expressed as ratios[50,81,144]

$$\frac{I_1}{I_2} \approx \left(\frac{\omega_1^{\mathrm{g}}\Delta_1}{\omega_2^{\mathrm{g}}\Delta_2}\right)^2 . \tag{10.50}$$

In this equation, ω_1^{g} and ω_2^{g} represent the ground-state vibrational frequencies for normal modes Q_1 and Q_2 and ω is the electronic excitation energy. The first equality in Eq. (10.50), written as an approximation here, is exact within the IMDHO model. The quantity

$$\Delta_k = \left(\frac{\omega_k}{\hbar}\right)^{1/2} \Delta Q_k \tag{10.51}$$

evaluated at the ground-state geometry ($\mathbf{Q} = \mathbf{0}$), is the slope of the excited-state potential energy surface along mode k. This leads to the second equality in Eq. (10.50).

The time-dependent picture provides means to derive this expression.[81] In this approach, the requisite polarizability tensor elements involving different electronic states are expressed as the Fourier transformation of the time-evolving overlap between initial- and final-state electronic wave functions:

$$\alpha(\omega_{\mathrm{L}}) \propto \int_0^\infty e^{i\omega_{\mathrm{L}}t - \Gamma t}\langle\psi_f|\psi_i(t)\rangle dt + \mathrm{NRT} . \tag{10.52}$$

Here, "NRT" indicates the non-resonant terms that are neglected in RR spectroscopy. Large molecules likely spend no more than 10–20 fs in the Franck-Condon region and the overlap integral is likely only significant on that timescale.[50,81,144] Within a model that considers only two electronic states, the RR intensity that one obtains is

$$I_k \propto \omega_{\mathrm{L}}(\omega_{\mathrm{L}} - \omega_k)^3(\omega_k\Delta_k)^2 \tag{10.53}$$

where μ_k is the reduced mass of the kth normal mode.

Assuming identical force constants for Q_k in both the ground and excited electronic state, one obtains a linear transformation between the displacement Δ_k of the equilibrium position of this mode, expressed in normal coordinates, and the displacements $\widetilde{\Delta}_i$ expressed in Cartesian coordinates:

$$\widetilde{\Delta}_i = \sum_{k=1}^{3N-6} \left(\frac{L_{ik}}{m_i^{1/2}}\right) \Delta_k . \tag{10.54}$$

In matrix form this is

$$\mathbf{\Delta}_Q = \lambda^{-1}\mathbf{L}^\dagger\mathbf{M}^{1/2}\mathbf{V}_X \tag{10.55}$$

where λ is the eigenvalues of mass-weighted Hessian matrix, M defines the matrix of atomic masses and V_X is the energy gradient in Cartesian coordinate Raman intensities are related to the dimensionless displacements

$$\Delta_k = \left(\frac{\lambda_k}{m_e}\right)^{1/4} \Delta_{Qk} . \tag{10.56}$$

Setting JOBTYPE = RRAMAN invokes the calculation of resonance Raman intensities.

RR_NO_NORMALISE

Controls whether frequency job calculates resonance Raman intensities

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Normalize RR intensities

True Do not normalize RR intensities

RECOMMENDATION:

False

Example 10.37 Calculating resonance Raman intensities.

```
$molecule
   0 1
   C     1.8288506578   -0.1219336002    0.0000000000
   C     0.6155951063    0.3987918905    0.0000000000
   C    -0.6155955606   -0.3987931260    0.0000000000
   C    -1.8288502653    0.1219348794    0.0000000000
   H     2.7085214046    0.4909328271    0.0000000000
   H     1.9881851899   -1.1843222290    0.0000000000
   H     0.4885913610    1.4671254626    0.0000000000
   H    -0.4885933454   -1.4671268234    0.0000000000
   H    -1.9881816088    1.1843239478    0.0000000000
   H    -2.7085226822   -0.4909289672    0.0000000000
$end

$rem
   JOBTYPE                RRAMAN
   METHOD                 hf
   BASIS                  3-21G
   CIS_N_ROOTS            1
   CIS_STATE_DERIVATIVE   1
$end
```

10.7.7 Vibrationally-Resolved Electronic and Resonance Raman Spectra

Optical spectroscopy is the study of the interaction between the light and matter, and the study that encompasses a wide range of physical and chemical behavior, which can be directly recorded by the spectrometers. Contemporary spectroscopic techniques have been applied to widespread research fields and have served as a popular tool to obtain the information of structural and dynamical features of the matter. However, the experimentally-measured spectra can't straightforwardly give the microscopic information of the matter. The theoretical calculations of the spectra can serve as a supplementary tool to the experimental measurements and provide a deeper understanding on the underlying physical and chemical phenomena.[17,18,164] One can easily determine geometrical, electronic and dynamical features of matters through a comparison between the experimental results and the calculated values. Moreover, the role of different effects in spectroscopic properties can also be quantified by the calculations.

Vibrationally-resolved one-photon absorption (OPA) and emission (OPE) spectra and resonance Raman scattering (RRS) spectra, each of which involves simultaneous changes in the vibrational and electronic states of a molecule, can reveal a reliable molecular structure–property relationship. Theoretical prediction of these spectra needs to combine both the electronic structure theories and quantum dynamics methods to obtain the structure parameters and de-

scribe quantum dynamics, respectively.[42,144,145,197,218,220] For RRS simulation using the IMDHO model (which neglects Duschinsky rotation), see Section 10.7.6.

10.7.7.1 Time-dependent approach to simulating spectra

On the basis of perturbation theory, the transition rate of one- or two-photon transition processes from the initial state $|I\rangle$ to the final state $|F\rangle$ is proportional to $k_{IF} = |\langle F|\hat{M}|I\rangle|^2\delta(\Delta\omega)$, where $\delta(\Delta\omega)$ is the line shape function with $\Delta\omega = \omega_i - \omega_{FI}$ for a one-photon transition and $\Delta\omega = \omega_1 + \omega_2 - \omega_{FI}$ for a two-photon process. Here ω_i, ω_1, and ω_2 denote the incident photon frequencies and the operator $\hat{M}$ is given by

$$\hat{M} = \begin{cases} \hat{\mu} & \text{(one-photon transition)} \\ \sum_L \left[\frac{\mu\cdot\hat{e}_2|L\rangle\langle L|\mu\cdot\hat{e}_1}{(\omega_1-\omega_{LI})} + \frac{\mu\cdot\hat{e}_1|L\rangle\langle L|\mu\cdot\hat{e}_2}{(\omega_2-\omega_{LI})} \right] & \text{(two-photon transition)} \end{cases} . \tag{10.57}$$

In the two-photon case there are intermediate or "virtual" states $|L\rangle$.

Raman scattering is a two-photon process. In this process, one photon with the frequency ω_i is absorbed, another photon with the frequency ω_S is emitted, and the transition from the initial to the final vibrational states takes place. Based on perturbation theory, the transition rate of the Raman process is proportional to $S(\omega_i, \omega_S) = |\langle F|\hat{M}|I\rangle|^2\delta(\Delta\omega)$, where

$$\hat{M} = \sum_L \left[\frac{\mu\cdot\hat{e}_2|L\rangle\langle L|\mu\cdot\hat{e}_1}{(\omega_i - \omega_{LI})} - \frac{\mu\cdot\hat{e}_1|L\rangle\langle L|\mu\cdot\hat{e}_2}{(\omega_S + \omega_{LI})} \right] \tag{10.58}$$

and $\Delta\omega = \omega_S - \omega_i + \omega_{FI}$. The differential photon scattering cross section is given by[9,127]

$$\sigma(\omega_i, \omega_S) \propto \frac{4\omega_i\omega_S^3}{9c^4} S(\omega_i, \omega_S). \tag{10.59}$$

RRS spectroscopy is a type of vibrational Raman spectroscopy in which the incident laser frequency is close to an electronic transition of the molecule or crystal studied. As the adiabatic energy gap ω_{LI} between the L state and the initial I state is close to the laser frequency ω_i, the intermediate L state will make the dominant contribution to RRS. Under the "resonant" condition, the contributions from the non-resonant electronic states can be neglected.

One may evaluate $M_{IF} = \langle\Phi_F|\hat{M}|\Phi_I\rangle$ by making use of the Herzberg-Teller (HT) expansion, *i.e.*, one expands the integrals about the nuclear equilibrium configuration $Q = 0$. Writing the pure-spin Born–Oppenheimer (psBO) functions as products of an electronic wavefunction Ψ and a vibrational wavefunction Λ

$$\Phi_n(q, Q) = \Psi_n(q, Q)\Lambda_n(Q) , \tag{10.60}$$

we have

$$\begin{aligned} \Psi_n(q, Q) &= \Psi_n(q, 0) + [\partial\Psi_n(q, 0)/\partial Q]_{Q=0}Q + \cdots , \\ M_{IF} &= M_{IF}(Q = 0) + [\partial M_{IF}/\partial Q]_{Q=0}Q + \cdots . \end{aligned} \tag{10.61}$$

The second term in Eq. (10.61) origins from the HT expansion. If we truncate the expansions after the lowest-order non-vanishing term, M_{IF} can be written as

$$\begin{aligned} M_{IF} = &\langle\Psi_F(q, 0)|\hat{M}|\Psi_I(q, 0)\rangle\langle\Lambda_F(Q')|\Lambda_I(Q)\rangle \\ &+ [(\partial/\partial Q)\langle\Psi_F(q, 0)|\hat{M}|\Psi_I(q, 0)\rangle]_{Q=0}\langle\Lambda_F(Q')|Q|\Lambda_I(Q)\rangle . \end{aligned} \tag{10.62}$$

If the first term, the direct transition, vanishes, this process is orbitally forbidden.

To evaluate the vibrational terms in the remaining part of the vibronic matrix elements, we can use the harmonic oscillator approximation. Then the vibrational part of the wave function is written as $\Lambda_n = \prod_{k=1}^{N} \chi_k^{(n)}(\nu_n)$, where N is the total number of normal modes χ_k and ν_n the vibrational quantum number associated with mode k in state $|n\rangle$.

The delta function $\delta(\Delta\omega)$ can be expressed as the Fourier integral

$$\delta(\Delta\omega) = \frac{1}{2\pi\hbar}\int_{-\infty}^{+\infty} \mathrm{e}^{(i\Delta\omega)t/\hbar}\, dt \;, \tag{10.63}$$

and then the transition rate from the initial state to the final state becomes

$$k_{IF} = \int_{-\infty}^{+\infty} \exp[i(\omega_0 + E_i - E_f)t/\hbar - \gamma t] C_i(t) dt \tag{10.64}$$

where γ is a damping factor and

$$C_i(t) = \frac{\mathrm{tr}[e^{-\beta\hat{H}_i}\mathrm{e}^{i\hat{H}_i t/\hbar}\hat{M}e^{-iH_f t/\hbar}\hat{M})}{\mathrm{tr}[\mathrm{e}^{-\beta\hat{H}_i}]} \;. \tag{10.65}$$

Here $\beta = 1/k_B T$, $\omega_0 = \omega_i$ in one-photon absorption and emission processes, $\omega_0 = \omega_i - \omega_S$ in Raman scattering process. The notation $\mathrm{tr}(\cdots)$ represents a trace over nuclear and electronic degrees of freedom, and $\hat{M} = |\Lambda_i\rangle M_{IF}\langle\Lambda_f| + |\Lambda_f\rangle M_{FI}\langle\Lambda_i|$. The quantities $\hat{H}_i$ and $\hat{H}_f$ denote the nuclear Hamiltonians of electronic ground and excited states, respectively.

The Hamiltonian of vibrational motions on the ground and excited states can be written as

$$H_g = \frac{1}{2}\sum_j^N \left[(P_{g,i})^2 + (\omega_j^g Q_{g,j})^2\right] \;, \tag{10.66a}$$

$$H_e = \frac{1}{2}\sum_j^N \left[(P_{e,i})^2 + (\omega_j^e Q_{e,j})^2\right] \;, \tag{10.66b}$$

where P and Q are the momenta and coordinates of vibrational normal modes, respectively. The normal mode coordinates of ground and excited states are correlated by the Duschinsky rotation matrix $\bar{D}$,[55] with $Q_e = \bar{D}Q_g + \bar{\Delta}$. The quantity $\bar{\Delta}$ is the displacement of normal mode coordinates between ground and excited states, *i.e.*, the same quantity that appears in the IMDHO theory of Section 10.7.6, Eq. (10.51). The dimensionless forms are correspondingly $\Delta_j = (\omega_j^e)^{1/2}\bar{\Delta}_j$ and $D_{ij} = (\omega_i^e/\omega_j^g)^{1/2}\bar{D}_{ij}$. From the above, the transition rate can be calculated directly in the time domain using the correlation function approach. This time-dependent approach has been implemented to calculate vibronic spectra.[121,122,129,130]

It is obvious that ground and excited electronic states have different potential energy surfaces leading to different vibrational frequencies and normal modes. The relation between mass-weighted Cartesian displacement coordinates x and normal mode coordinates Q is given by

$$\begin{aligned} x^g - x_0^g &= L_g Q_g \;, \\ x^e - x_0^e &= L_e Q_e \;, \end{aligned} \tag{10.67}$$

where x_0^g and x_0^e are the equilibrium structures of ground and excited states. For an ideal N-dimensional harmonic oscillator, the normal mode coordinates of ground and excited states are related by

$$\begin{aligned} Q_e &= (L_e)^\top L_g Q_g + (L_e)\top(x_0^g - x_0^e) \\ &= \bar{D}Q_g + \bar{\Delta} \;. \end{aligned} \tag{10.68}$$

The minimum points on the potential surface and the Hessian matrix are required to calculate the Duschinsky rotation matrix and displacement vector. It can be time-consuming to calculate the excited-state potential surface, especially for large molecules, and often problematic due to state crossings. The linear coupling model (LCM), which is also known as the vertical gradient (VG) approximation, has been proposed to avoid this issue.[39] Assuming that the excited-state potential surface is approximated by a shift in the ground state PES, namely $\omega_j^e = \omega_j^g$ and $L_e = L_g$, the displacement of Q_j can be calculated by the excited-state energy gradient $(\partial E/\partial Q)_j$ and Δ^{VG} can be written as

$$\Delta_j^{\mathrm{VG}} = (\omega_j^g)^{-3/2}\left(\frac{\partial E}{\partial Q}\right)_j = \sum_i (\omega_j^g)^{-3/2}\left(\frac{\partial E}{\partial x_i}\right) L_g^{ij} . \tag{10.69}$$

The VG approximation is equivalent to the IMDHO approximation that is discussed in Section 10.7.6.[72,73,103]

Generally, the Franck-Condon (FC) approximation is accurate enough for strongly one- or two-photon allowed transitions, while it breaks down for forbidden or weakly allowed transitions, and the FC term becomes nearly zero. In this situation, a correction to this deviation should be introduced by including the Herzberg-Teller (HT) or non-Condon effect.[9,10,125,160,195] HT-type vibronic coupling comes from the normal mode-coordinate dependence of the transition moments. When these quantities are expanded in terms of the normal mode coordinates, the contribution of the linear-coordinate-dependent terms is commonly called HT effect.[9,10] Many works, whether or not they account for the mode-mixing or Duschinsky rotation (DR) effect or not, have shown the importance of the HT effect in OPA, OPE, and RRS spectra.[122]

To predict OPA, OPE and RRS spectra, electronic structure calculations on ground and excited states should be performed. The necessary jobs at different level of approximation are summarized in the following:

- FC. This is available for OPA, OPE, and RRS spectra.
  1. $x_0^g \neq x_0^e$, $L_g = L_e$, and $\omega^g = \omega^e$. Geometry optimization on excited state potential surface is performed, followed by ground state optimization and frequency analysis.
  2. $x_0^g \neq x_0^e$, $L_g \neq L_e$, and $\omega^g \neq \omega^e$. Geometry optimization and frequency calculation are needed on both ground and excited states.
- FCHT. This is available for OPA, OPE, and RRS spectra.
  - $x_0^g \neq x_0^e$, $L_g \neq L_e$, and $\omega^g \neq \omega^e$. It is similar to the second kind of FC calculation, in which transition dipole derivative is obtained via frequency calculation on excited state. Geometry optimization and frequency calculation are needed on both ground and excited states.
- VG. This is available for OPA and RRS spectra.
  - $x_0^g = x_0^e$, $L_g = L_e$, and $\omega^g = \omega^e$. Only the geometry optimization and frequency calculation of the ground state is involved. Frequencies and normal modes of excited state are assumed to be the same as ground state. The displacement vector is approximated by Eq. (10.69), in which the gradient of excited state potential surface is produced by excited state force job. Of course VG model has only contribution from FC term.

10.7.7.2 Job Control

Since both ground state and excited state parameters are required, the routines to predict vibronic spectra are designed to have two steps. Firstly excited state calculation is performed and information about excited state will be saved in `$QCSCRATCH/savename`. Then the vibronic spectra utility is called to simulate the requested spectra after frequency analysis on ground state. POINT_GROUP_SYMMETRY should be set to FALSE in order to prevent the molecular geometry being transformed to the standard orientation. Therefore the vibronic spectra job input can be set up in two ways. First, the multiple jobs can be separated by the string `@@@` as described in Section 3.5. Or, jobs can be separated into individual inputs using `$QCSCRATCH/savename` as described below and given as examples 10.39 and 10.40.

```
qchem infile_excited_state outfile_excited_state savename
qchem infile_ground_state outfile_ground_state savename
```

There are two *$rem* variables and one section *$vibronic* involved in vibronic spectra calculations.

SAVE_VIBRONIC_PARAMS

Save information about excited state which is requested in vibronic spectra simulation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE

RECOMMENDATION:

TRUE

VIBRONIC_SPECTRA

Specifies which type of vibronic spectra will be predicted. Should be used in a frequency job (jobtype = Freq).

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No vibronic spectra is predicted.

1 OPA spectra is calculated.

2 OPE spectra is calculated.

3 RRS spectra is calculated.

RECOMMENDATION:

Use the default.

Variables in the *$vibronic* section:

MODEL

Specifies which kind of model will be used to simulate the vibronic spectra.

INPUT SECTION: *$vibronic*

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

1 FC.

2 FCHT.

3 VG.

RECOMMENDATION:

User defined

TEMPERATURE

Specifies the temperature in the vibronic spectra simulation.

INPUT SECTION: *$vibronic*

TYPE:

FLOAT

DEFAULT:

298.15

OPTIONS:

t temperature, in K.

RECOMMENDATION:

User defined

FREQ_RANGE

Specifies the frequency range of vibronic spectra.

INPUT SECTION: *$vibronic*

TYPE:

FLOAT

DEFAULT:

1.0 40000.0 10.0

OPTIONS:

ν_{min} ν_{max} $\delta\nu$ minimum, maximum and step size, in cm^{-1}.

RECOMMENDATION:

User defined

TIME_RANGE

Specifies the step size and the number of steps in time domain propagation.

INPUT SECTION: *$vibronic*

TYPE:

FLOAT and INTEGER

DEFAULT:

1.0 40000

OPTIONS:

δt n_{step} time step size in a.u., and the number of steps.

RECOMMENDATION:

User defined

DAMPING

Specifies the damping factor.

INPUT SECTION: *$vibronic*

TYPE:

FLOAT

DEFAULT:

300.0

OPTIONS:

γ damping factor, in cm^{-1}.

RECOMMENDATION:

User defined

FREQ_SCALE_FACTOR

Specifies the frequency scale factors.

INPUT SECTION: *$vibronic*

TYPE:

FLOAT

DEFAULT:

1.0 1.0 1.0 1.0

OPTIONS:

λ_{H}^{g} λ_{H}^{e} $\lambda_{\mathrm{ZPE}}^{g}$ $\lambda_{\mathrm{ZPE}}^{e}$ scale factor for ground state harmonic frequency, for excited state harmonic frequency, for ground state zero-point energy, and for excited state zero-point energy

RECOMMENDATION:

User defined

EPSILON

Specifies the spectral broadening factor. It is available only for RRS spectra simulation.

INPUT SECTION: *$vibronic*

TYPE:

FLOAT

DEFAULT:

25.0

OPTIONS:

ε broadening factor, in cm^{-1}.

RECOMMENDATION:

User defined

10.7.7.3 Vibronic Job Examples

Example 10.38 Input files for OPA spectra in the FCHT approximation of formaldehyde corresponding to the $S_0 \rightarrow S_1$ transition. In the first step, frequency analysis at the S_1 equilibrium geometry. Then run a ground state frequency analysis on the S_0 ground state optimized structure.

```
$molecule
  0 1
  O    -0.0367447359    -0.0007590817     0.6963163574
  C     0.1461299638     0.0026846285    -0.5839700302
  H    -0.0732270514     0.9340547891    -1.1138640182
  H    -0.0391581765    -0.9359803358    -1.1140167891
$end

$rem
  JOBTYPE                freq
  METHOD                 b3lyp
  BASIS                  def2-TZVP
  CIS_STATE_DERIV        1
  CIS_SINGLETS           true
  CIS_TRIPLETS           false
  CIS_N_ROOTS            10
point_group_symmetry False
  SAVE_VIBRONIC_PARAMS   true ! enables saving information of S1 state
$end

@@@

$molecule
   0 1
   O    0.0000000000     0.0000000000     0.6637077571
   C    0.0000000000     0.0000000000    -0.5351027012
   H    0.0000000000     0.9394749352    -1.1220697679
   H    0.0000000000    -0.9394749352    -1.1220697679
$end

$rem
   JOBTYPE            freq
   METHOD             b3lyp
   BASIS              def2-TZVP
point_group_symmetry False
   VIBRONIC_SPECTRA   1 !enables vibronic_spectra and reads saved information
$end

$vibronic
   model        2
   freq_range   20000. 60000. 10.
   time_range   1. 40000
   damping      40.
$end
```

Example 10.39 Vibrationally resolved fluorescence, *i.e.*, OPE with the first kind of FC model, is calculated as following. The emission from D_1 to D_0 of *p*-fluorobenzyl radical is used as an example. This is the first job of the total vibronic spectra simulation, by running the excited state geometry optimization retaining information in `$QCSCRATCH/savename`. Information from this job will be needed to complete the simulation in Example 10.40.

```
$molecule
   0 2
   C          1.4840482200    0.0000338155    0.0000000000
   C          0.7160497031    0.0000524901   -1.2119311870
   C         -0.7159596058    0.0000542629   -1.2126930961
   C         -1.4043236629    0.0000543088    0.0000000000
   C         -0.7159596058    0.0000542629    1.2126930961
   C          0.7160497031    0.0000524901    1.2119311870
   C          2.8748450131    0.0000120580    0.0000000000
   H          1.2370230923    0.0000896598   -2.1693859212
   H         -1.2717579173    0.0000412967   -2.1497435961
   H         -1.2717579173    0.0000412967    2.1497435961
   H          1.2370230923    0.0000896598    2.1693859212
   H          3.4346492051    0.0000003003   -0.9330758768
   H          3.4346492051    0.0000003003    0.9330758768
   F         -2.7508602624    0.0000394216    0.0000000000
$end

$rem
   JOBTYPE                opt
   METHOD                 b3lyp
   BASIS                  def2-SVP
   CIS_STATE_DERIV        1
   CIS_N_ROOTS            10
point_group_symmetry False
   SAVE_VIBRONIC_PARAMS   true !saved into $QCSCRATCH/savename
$end
```

Example 10.40 The final job for obtaining vibrationally resolved fluorescence of the D_1 to D_0 transition of *p*-fluorobenzyl radical using OPE with the first kind of FC model from Example 10.39.

```
$molecule
   0 2
   C       1.4578807306     0.0130092784     0.0000000000
   C       0.7102753558     0.0082793447    -1.2194714816
   C      -0.6772053823    -0.0007923729    -1.2210832164
   C      -1.3603507249    -0.0052928605     0.0000000000
   C      -0.6772053834    -0.0007923730     1.2210832170
   C       0.7102753546     0.0082793447     1.2194714789
   C       2.8669152234     0.0219746232     0.0000000000
   H       1.2502372081     0.0119050313    -2.1697312391
   H      -1.2498963812    -0.0045285858    -2.1510495277
   H      -1.2498963822    -0.0045285858     2.1510495285
   H       1.2502372136     0.0119050313     2.1697312323
   H       3.4299577819     0.0255536038    -0.9358170827
   H       3.4299577814     0.0255536038     0.9358170860
   F      -2.7010161586    -0.0142661666     0.0000000000
$end

$rem
   JOBTYPE            freq
   METHOD             b3lyp
   BASIS              def2-SVP
point_group_symmetry False
   VIBRONIC_SPECTRA   2
$end

$vibronic
   model        1
   temperature  0.
   freq_range   1. 40000. 10.
   time_range   1. 40000
   damping      20.
$end
```

Example 10.41 RRS spectra of phenoxyl radical ($D_0 \rightarrow D_3$ transition) with the VG approximation. Therefore the first job calculates the D_3 state force at the ground state optimized geometry, followed by the ground state frequency analysis. The excited state forces and ground state frequencies are calculated in the ground state equilibrium geometry.

```
$molecule
   0 2
   C       0.0000000000     1.2271514002    -1.0879472096
   C       0.0000000000     0.0000408897    -1.7873074655
   C       0.0000000000    -1.2270324440    -1.0880160727
   C       0.0000000000    -1.2409681161     0.2924435676
   C       0.0000000000    -0.0000313560     1.0551142042
   C       0.0000000000     1.2409428316     0.2924458686
   H       0.0000000000     2.1656442172    -1.6487551860
   H       0.0000000000    -0.0001767539    -2.8803293768
   H       0.0000000000    -2.1655968771    -1.6487220344
   H       0.0000000000    -2.1715667156     0.8648121894
   H       0.0000000000     2.1714692701     0.8649813475
   O       0.0000000000     0.0001236541     2.3063351676
$end

$rem
   JOBTYPE                force
   METHOD                 b3lyp
   BASIS                  def2-SVP
   CIS_STATE_DERIV        3
   CIS_N_ROOTS            10
point_group_symmetry False
   SAVE_VIBRONIC_PARAMS   true
$end

@@@

$molecule
   read !VG approximation uses the same geometry for ground and excited state
$end

$rem
   JOBTYPE            freq
   METHOD             b3lyp
   BASIS              def2-SVP
point_group_symmetry False
   VIBRONIC_SPECTRA   3
$end

$vibronic
   MODEL         3
   TEMPERATURE   0.
   FREQ_RANGE    1. 4000. 1.
   TIME_RANGE    1. 40000
   DAMPING       100.
   EPSILON       25.
$end
```

10.8 Anharmonic Vibrational Frequencies

10.8.1 Introduction

Computing vibrational spectra beyond the harmonic approximation has become an active area of research owing to the improved efficiency of computer techniques.[16,35,141,231] To calculate the exact vibrational spectrum within Born-Oppenheimer approximation, one has to solve the nuclear Schrödinger equation completely using numerical integration techniques, and consider the full configuration interaction of quanta in the vibrational states. This has only been carried out on di- or triatomic system.[36,167] The difficulty of this numerical integration arises because solving exact the nuclear Schrödinger equation requires a complete electronic basis set, consideration of all the nuclear vibrational configuration states, and a complete potential energy surface (PES). Simplification of the Nuclear Vibration Theory (NVT) and PES are the doorways to accelerating the anharmonic correction calculations. There are five aspects to simplifying the problem:

- Expand the potential energy surface using a Taylor series and examine the contribution from higher derivatives. Small contributions can be eliminated, which allows for the efficient calculation of the Hamiltonian.
- Investigate the effect on the number of configurations employed in a variational calculation.
- Avoid using variational theory (due to its expensive computational cost) by using other approximations, for example, perturbation theory.
- Obtain the PES indirectly by applying a self-consistent field procedure.
- Apply an *anharmonic* wave function which is more appropriate for describing the distribution of nuclear probability on an anharmonic potential energy surface.

To incorporate these simplifications, new formulae combining information from the Hessian, gradient and energy are used as a default procedure to calculate the cubic and quartic force field of a given potential energy surface.

Here, we also briefly describe various NVT methods. In the early stage of solving the nuclear Schrödinger equation (in the 1930s), second-order Vibrational Perturbation Theory (VPT2) was developed.[8,16,147,149,227] However, problems occur when resonances exist in the spectrum. This becomes more problematic for larger molecules due to the greater chance of accidental degeneracies occurring. To avoid this problem, one can do a direct integration of the secular matrix using Vibrational Configuration Interaction (VCI) theory.[225] It is the most accurate method and also the least favored due to its computational expense. In Q-CHEM 3.0, we introduce a new approach to treating the wave function, transition-optimized shifted Hermite (TOSH) theory,[123] which uses first-order perturbation theory, which avoids the degeneracy problems of VPT2, but which incorporates anharmonic effects into the wave function, thus increasing the accuracy of the predicted anharmonic energies.

10.8.2 Vibration Configuration Interaction Theory

To solve the nuclear vibrational Schrödinger equation, one can only use direct integration procedures for diatomic molecules.[36,167] For larger systems, a truncated version of full configuration interaction is considered to be the most accurate approach. When one applies the variational principle to the vibrational problem, a basis function for the nuclear wave function of the nth excited state of mode i is

$$\psi_i^{(n)} = \phi_i^{(n)} \prod_{j \neq i}^{m} \phi_j^{(0)} \tag{10.70}$$

where the $\phi_i^{(n)}$ represents the harmonic oscillator eigenfunctions for normal mode q_i. This can be expressed in terms of Hermite polynomials:

$$\phi_i^{(n)} = \left(\frac{\omega_i^{\frac{1}{2}}}{\pi^{\frac{1}{2}} 2^n n!} \right)^{\frac{1}{2}} e^{-\frac{\omega_i q_i^2}{2}} H_n(q_i \sqrt{\omega_i}) \tag{10.71}$$

With the basis function defined in Eq. (10.70), the nth wave function can be described as a linear combination of the Hermite polynomials:

$$\Psi^{(n)} = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \sum_{k=0}^{n_3} \cdots \sum_{m=0}^{n_m} c_{ijk\cdots m}^{(n)} \psi_{ijk\cdots m}^{(n)} \tag{10.72}$$

where n_i is the number of quanta in the ith mode. We propose the notation VCI(n) where n is the total number of quanta, *i.e.*:

$$n = n_1 + n_2 + n_3 + \cdots + n_m \tag{10.73}$$

To determine this expansion coefficient $c^{(n)}$, we integrate the $\hat{H}$, as in Eq. (4.1), with $\Psi^{(n)}$ to get the eigenvalues

$$c^{(n)} = E_{\mathrm{VCI}(n)}^{(n)} = \langle \Psi^{(n)} | \hat{H} | \Psi^{(n)} \rangle \tag{10.74}$$

This gives us frequencies that are corrected for anharmonicity to n quanta accuracy for a m-mode molecule. The size of the secular matrix on the right hand of Eq. (10.74) is $((n+m)!/n!m!)^2$, and the storage of this matrix can easily surpass the memory limit of a computer. Although this method is highly accurate, we need to seek for other approximations for computing large molecules.

10.8.3 Vibrational Perturbation Theory

Vibrational perturbation theory has been historically popular for calculating molecular spectroscopy. Nevertheless, it is notorious for the inability of dealing with resonance cases. In addition, the non-standard formulas for various symmetries of molecules forces the users to modify inputs on a case-by-case basis,[12,45,142] which narrows the accessibility of this method. VPT applies perturbation treatments on the same Hamiltonian as in Eq. (4.1), but divides it into an unperturbed part, $\hat{U}$,

$$\hat{U} = \sum_i^m \left(-\frac{1}{2} \frac{\partial^2}{\partial q_i^2} + \frac{{\omega_i}^2}{2} {q_i}^2 \right) \tag{10.75}$$

and a perturbed part, $\hat{V}$:

$$\hat{V} = \frac{1}{6} \sum_{ijk=1}^{m} \eta_{ijk} q_i q_j q_k + \frac{1}{24} \sum_{ijkl=1}^{m} \eta_{ijkl} q_i q_j q_k q_l \tag{10.76}$$

One can then apply second-order perturbation theory to get the ith excited state energy:

$$E^{(i)} = \hat{U}^{(i)} + \langle \Psi^{(i)} | \hat{V} | \Psi^{(i)} \rangle + \sum_{j \neq i} \frac{|\langle \Psi^{(i)} | \hat{V} | \Psi^{(j)} \rangle|^2}{\hat{U}^{(i)} - \hat{U}^{(j)}} \tag{10.77}$$

The denominator in Eq. (10.77) can be zero either because of symmetry or accidental degeneracy. Various solutions, which depend on the type of degeneracy that occurs, have been developed which ignore the zero-denominator elements from the Hamiltonian.[12,45,142,150] An alternative solution has been proposed by Barone,[16] which can be applied to all molecules by changing the masses of one or more nuclei in degenerate cases. The disadvantage of this method is that it will break the degeneracy which results in fundamental frequencies no longer retaining their correct symmetry. He proposed

$$E_i^{\mathrm{VPT2}} = \sum_j \omega_j (n_j + 1/2) + \sum_{i \leq j} x_{ij} (n_i + 1/2)(n_j + 1/2) \tag{10.78}$$

where, if rotational coupling is ignored, the anharmonic constants x_{ij} are given by

$$x_{ij} = \frac{1}{4\omega_i \omega_j} \left(\eta_{iijj} - \sum_k^m \frac{\eta_{iik}\eta_{jjk}}{\omega_k^2} + \sum_k^m \frac{2(\omega_i^2 + \omega_j^2 - \omega_k^2)\eta_{ijk}^2}{[(\omega_i + \omega_j)^2 - \omega_k^2]\,[(\omega_i - \omega_j)^2 - \omega_k^2]} \right) \tag{10.79}$$

10.8.4 Transition-Optimized Shifted Hermite Theory

So far, every aspect of solving the nuclear wave equation has been considered, except the wave function. Since Schrödinger proposed his equation, the nuclear wave function has traditionally be expressed in terms of Hermite functions, which are designed for the harmonic oscillator case. Recently a modified representation has been presented.[123] To demonstrate how this approximation works, we start with a simple example. For a diatomic molecule, the Hamiltonian with up to quartic derivatives can be written as

$$\hat{H} = -\frac{1}{2}\frac{\partial^2}{\partial q^2} + \frac{1}{2}\omega^2 q^2 + \eta_{iii} q^3 + \eta_{iiii} q^4 \tag{10.80}$$

and the wave function is expressed as in Eq. (10.71). Now, if we shift the center of the wave function by σ, which is equivalent to a translation of the normal coordinate q, the shape will still remain the same, but the anharmonic correction can now be incorporated into the wave function. For a ground vibrational state, the wave function is written as

$$\phi^{(0)} = \left(\frac{\omega}{\pi}\right)^{\frac{1}{4}} e^{-\frac{\omega}{2}(q-\sigma)^2} \tag{10.81}$$

Similarly, for the first excited vibrational state, we have

$$\phi^{(1)} = \left(\frac{4\omega^3}{\pi}\right)^{\frac{1}{4}} (q-\sigma)\, e^{\frac{\omega}{2}(q-\sigma)^2} \tag{10.82}$$

Therefore, the energy difference between the first vibrational excited state and the ground state is

$$\Delta E_{\mathrm{TOSH}} = \omega + \frac{\eta_{iiii}}{8\omega^2} + \frac{\eta_{iii}\sigma}{2\omega} + \frac{\eta_{iiii}\sigma^2}{4\omega} \tag{10.83}$$

This is the fundamental vibrational frequency from first-order perturbation theory.

Meanwhile, We know from the first-order perturbation theory with an ordinary wave function within a QFF PES, the energy is

$$\Delta E_{\mathrm{VPT1}} = \omega + \frac{\eta_{iiii}}{8\omega^2} \tag{10.84}$$

The differences between these two wave functions are the two extra terms arising from the shift in Eq. (10.83). To determine the shift, we compare the energy with that from second-order perturbation theory:

$$\Delta E_{\mathrm{VPT2}} = \omega + \frac{\eta_{iiii}}{8\omega^2} - \frac{5{\eta_{iii}}^2}{24\omega^4} \tag{10.85}$$

Since σ is a very small quantity compared with the other variables, we ignore the contribution of σ^2 and compare ΔE_{TOSH} with ΔE_{VPT2}, which yields an initial guess for σ:

$$\sigma = -\frac{5}{12}\frac{\eta_{iii}}{\omega^3} \tag{10.86}$$

Because the only difference between this approach and the ordinary wave function is the shift in the normal coordinate, we call it "transition-optimized shifted Hermite" (TOSH) functions.[123] This approximation gives second-order accuracy at only first-order cost.

For polyatomic molecules, we consider Eq. (10.83), and propose that the energy of the ith mode be expressed as:

$$\Delta E_i^{\mathrm{TOSH}} = \omega_i + \frac{1}{8\omega_i}\sum_j \frac{\eta_{iijj}}{\omega_j} + \frac{1}{2\omega_i}\sum_j \eta_{iij}\sigma_{ij} + \frac{1}{4\omega_i}\sum_{j,k} \eta_{iijk}\sigma_{ij}\sigma_{ik} \tag{10.87}$$

Following the same approach as for the diatomic case, by comparing this with the energy from second-order perturbation theory, we obtain the shift as

$$\sigma_{ij} = \frac{(\delta_{ij}-2)(\omega_i+\omega_j)\eta_{iij}}{4\omega_i\omega_j^2(2\omega_i+\omega_j)} - \sum_k \frac{\eta_{kkj}}{4\omega_k\omega_j^2} \tag{10.88}$$

10.8.5 Job Control

The following *$rem* variables can be used to control the calculation of anharmonic frequencies.

ANHAR

Performing various nuclear vibrational theory (TOSH, VPT2, VCI) calculations to obtain vibrational anharmonic frequencies.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Carry out the anharmonic frequency calculation.

FALSE Do harmonic frequency calculation.

RECOMMENDATION:

Since this calculation involves the third and fourth derivatives at the minimum of the potential energy surface, it is recommended that the GEOM_OPT_TOL_DISPLACEMENT, GEOM_OPT_TOL_GRADIENT and GEOM_OPT_TOL_ENERGY tolerances are set tighter. Note that VPT2 calculations may fail if the system involves accidental degenerate resonances. See the VCI *$rem* variable for more details about increasing the accuracy of anharmonic calculations.

VCI

Specifies the number of quanta involved in the VCI calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined. Maximum value is 10.

RECOMMENDATION:

The availability depends on the memory of the machine. Memory allocation for VCI calculation is the square of $2(N_{\mathrm{Vib}} + N_{\mathrm{VCI}})/N_{\mathrm{Vib}}N_{\mathrm{VCI}}$ with double precision. For example, a machine with 1.5 GB memory and for molecules with fewer than 4 atoms, VCI(10) can be carried out, for molecule containing fewer than 5 atoms, VCI(6) can be carried out, for molecule containing fewer than 6 atoms, VCI(5) can be carried out. For molecules containing fewer than 50 atoms, VCI(2) is available. VCI(1) and VCI(3) usually overestimated the true energy while VCI(4) usually gives an answer close to the converged energy.

FDIFF_DER

Controls what types of information are used to compute higher derivatives. The default uses a combination of energy, gradient and Hessian information, which makes the force field calculation faster.

TYPE:

INTEGER

DEFAULT:

3 for jobs where analytical 2nd derivatives are available.

0 for jobs with ECP.

OPTIONS:

0 Use energy information only.

1 Use gradient information only.

2 Use Hessian information only.

3 Use energy, gradient, and Hessian information.

RECOMMENDATION:

When the molecule is larger than benzene with small basis set, FDIFF_DER = 2 may be faster. Note that FDIFF_DER will be set lower if analytic derivatives of the requested order are not available. Please refers to IDERIV.

MODE_COUPLING

Number of modes coupling in the third and fourth derivatives calculation.

TYPE:

INTEGER

DEFAULT:

2 for two modes coupling.

OPTIONS:

n for n modes coupling, Maximum value is 4.

RECOMMENDATION:

Use the default.

IGNORE_LOW_FREQ

Low frequencies that should be treated as rotation can be ignored during anharmonic correction calculation.

TYPE:

INTEGER

DEFAULT:

300 Corresponding to 300 cm^{-1}.

OPTIONS:

n Any mode with harmonic frequency less than n will be ignored.

RECOMMENDATION:

Use the default.

FDIFF_STEPSIZE_QFF

Displacement used for calculating third and fourth derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

5291 Corresponding to 0.1 bohr. For calculating third and fourth derivatives.

OPTIONS:

n Use a step size of $n \times 10^{-5}$.

RECOMMENDATION:

Use the default, unless the potential surface is very flat, in which case a larger value should be used.

Example 10.42 A four-quanta anharmonic frequency calculation on formaldehyde at the EDF2/6-31G* optimized ground state geometry, which is obtained in the first part of the job. It is necessary to carry out the harmonic frequency first and this will print out an approximate time for the subsequent anharmonic frequency calculation. If a FREQ job has already been performed, the anharmonic calculation can be restarted using the saved scratch files from the harmonic calculation.

```
$molecule
   0 1
   C
   O, 1, CO
   H, 1, CH, 2, A
   H, 1, CH, 2, A, 3, D

   CO = 1.2
   CH = 1.0
   A  = 120.0
   D  = 180.0
$end

$rem
    JOBTYPE                     OPT
    METHOD                      EDF2
    BASIS                       6-31G*
    GEOM_OPT_TOL_DISPLACEMENT   1
    GEOM_OPT_TOL_GRADIENT       1
    GEOM_OPT_TOL_ENERGY         1
$end

@@@

$molecule
    READ
$end

$rem
    JOBTYPE                     FREQ
    METHOD                      EDF2
    BASIS                       6-31G*
    ANHAR                       TRUE
    VCI                         4
$end
```

Anharmonic frequencies can also be computed using the partial Hessian approximation (see Section 10.7.4).

ANHAR_SEL

Select a subset of normal modes for subsequent anharmonic frequency analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE Use all normal modes

OPTIONS:

TRUE Select subset of normal modes

RECOMMENDATION:

None

Example 10.43 This example shows an anharmonic frequency calculation for ethene where only the C–H stretching modes are included in the anharmonic analysis.

```
$comment
  ethene
  restricted anharmonic frequency analysis
$end

$molecule
0 1
  C   0.6665   0.0000   0.0000
  C  -0.6665   0.0000   0.0000
  H   1.2480   0.9304   0.0000
  H  -1.2480  -0.9304   0.0000
  H  -1.2480   0.9304   0.0000
  H   1.2480  -0.9304   0.0000
$end

$rem
   JOBTYPE           freq
   METHOD            hf
   BASIS             sto-3g
   ANHAR_SEL         TRUE
   N_SOL             4
$end

$alist
9
10
11
12
$end
```

10.9 Linear-Scaling Computation of Electric Properties

10.9.1 Introduction

The search for new optical devices is a major field of materials sciences. Here, polarizabilities and hyperpolarizabilities provide particularly important information on molecular systems. The response of the molecular systems in the presence of an external, monochromatic, oscillatory electric field is determined by the solution of the time-dependent SCF (TDSCF) equations. Within the dipole approximation, the perturbation is represented as the interaction of the molecule

with a single Fourier component of the external field, $\boldsymbol{\mathcal{E}}$:

$$\hat{H}_{\text{field}} = \tfrac{1}{2}\hat{\boldsymbol{\mu}} \cdot \boldsymbol{\mathcal{E}}(e^{-i\omega t} + e^{+i\omega t}) \tag{10.89}$$

with

$$\hat{\boldsymbol{\mu}} = -e \sum_{i}^{N_{\text{elec}}} \hat{\mathbf{r}}_i \ . \tag{10.90}$$

Here, ω is the field frequency and $\hat{\boldsymbol{\mu}}$ is the dipole moment operator. The TDSCF equations can be solved via standard techniques of perturbation theory.[189] As a solution, one obtains the first-order perturbed density matrix $[\mathbf{P}^x(\pm\omega)]$ and the second-order perturbed density matrices $[\mathbf{P}^{xy}(\pm\omega, \pm\omega')]$. From these quantities, the following properties can be calculated:

- Static polarizability: $\alpha_{xy}(0;0) = \text{tr}\left[\mathbf{H}^{\mu_x}\mathbf{P}^y(\omega = 0)\right]$
- Dynamic polarizability: $\alpha_{xy}(\pm\omega;\mp\omega) = \text{tr}\left[\mathbf{H}^{\mu_x}\mathbf{P}^y(\pm\omega)\right]$
- Static hyperpolarizability: $\beta_{xyz}(0;0,0) = \text{tr}\left[\mathbf{H}^{\mu_x}\mathbf{P}^{yz}(\omega = 0, \omega = 0)\right]$
- Second harmonic generation: $\beta_{xyz}(\mp 2\omega;\pm\omega,\pm\omega) = \text{tr}\left[\mathbf{H}^{\mu_x}\mathbf{P}^{yz}(\pm\omega,\pm\omega)\right]$
- Electro-optical Pockels effect: $\beta_{\text{xyz}}(\mp\omega;0,\pm\omega) = \text{tr}\left[\mathbf{H}^{\mu_x}\mathbf{P}^{yz}(\omega = 0,\pm\omega)\right]$
- Optical rectification: $\beta_{xyz}(0;\pm\omega,\mp\omega) = \text{tr}\left[\mathbf{H}^{\mu_x}\mathbf{P}^{yz}(\pm\omega,\mp\omega)\right]$

Here, $\mathbf{H}^{\mu_x}$ is the matrix representation of the x component of the dipole moment.

For third-order properties (β_{xyz}), rather than computing them using a second-order TDSCF calculation and solving for $\mathbf{P}^{yz}$ explicitly, we calculate them from first-order properties using Wigner's $2n + 1$ rule.[104]

The TDSCF calculation is more time-consuming than the SCF calculation that precedes it (where the field-free, unperturbed ground state of the molecule is obtained). Q-CHEM's implementation of the TDSCF equations is MO based and the cost therefore formally scales asymptotically as $\mathcal{O}(N^3)$. The prefactor of the cubic-scaling step is rather small, however, and in practice (over a wide range of molecular sizes) the calculation is dominated by the cost of contractions with two-electron integrals, which is formally $\mathcal{O}(N^2)$ scaling but with a very large prefactor. The cost of these integral contractions can be reduced from quadratic to $\mathcal{O}(N)$ using LinK/CFMM methods (Section 4.6).[119] All derivatives are computed analytically.

The TDSCF module in Q-CHEM is known as "MOProp", since it corresponds (formally) to time propagation of the molecular orbitals. (For *actual* time propagation of the MOs, see Section 7.4.) The MOProp module has the following features:

- LinK and CFMM support to evaluate Coulomb- and exchange-like matrices
- Analytic derivatives
- DIIS acceleration
- Both restricted and unrestricted implementations of CPSCF and TDSCF equations are available, for both Hartree-Fock and Kohn-Sham DFT.
- Support for LDA, GGA, Meta-GGA[137], global hybrid and common range-separated functionals. VV10 is the only non-local correlation functional supported.

10.9.2 *$fdpfreq* Input Section

For dynamic response properties (*i.e.*, $\omega \neq 0$), various values of ω might be of interest, and it is considerably cheaper to compute properties for multiple values of ω in a single calculation than it is to run several calculations for one frequency each. The *$fdpfreq* input section is used to specify the frequencies of interest. The format is:

```
$fdpfreq
   property
   frequencies
   units
$end
```

The first line is only required for third-order properties, to specify the flavor of first hyperpolarizability. The options are

- `StaticHyper` (static hyperpolarizability)
- `SHG` (second harmonic generation)
- `EOPockels` (electro-optical Pockels effect)
- `OptRect` (optical rectification)

The second line in the *$fdpfreq* section contains floating-point values representing the frequencies of interest. Alternatively, for dynamic polarizabilities an equidistant sequence of frequencies can be specified by the keyword WALK (see example below). The last line specifies the units of the input frequencies. Options are:

- `au` (atomic units of frequency)
- `eV` (frequency units, expressed in electron volts)
- `Hz` (frequency units, expressed in Hertz)
- `nm` (wavelength units, in nanometers)
- `cmInv` (wavenumber units, cm^{-1})

Example 10.10.44 Static and dynamic polarizabilities, atomic units:

```
$fdpfreq
   0.0 0.03 0.05
   au
$end
```

Example 10.10.45 Series of dynamic polarizabilities, starting with 0.00 incremented by 0.01 up to 0.10:

```
$fdpfreq
   walk 0.00 0.10 0.01
   au
$end
```

Example 10.10.46 Static first hyperpolarizability, second harmonic generation and electro-optical Pockels effect, wavelength in nm:

```
$fdpfreq
   StaticHyper SHG EOPockels
   1064
   nm
$end
```

10.9.3 Job Control for the MOProp Module

The MOProp module is invoked by specifying a job number using the MOPROP *$rem* variable. In addition to electric properties, this module can also compute NMR chemical shifts (MOPROP = NMR); this functionality is described in Section 10.10.

MOPROP

Specifies the job number for MOProp module.

TYPE:

STRING

DEFAULT:

0 Do not run the MOProp module.

OPTIONS:

NMR	NMR chemical shielding tensors.
STATIC_POLAR	Static polarizability.
ISSC	Indirect nuclear spin–spin coupling tensors.
DYN_POLAR	Dynamic polarizability.
HYPERPOL	First hyperpolarizability using Wigner's $2n + 1$ rule.

RECOMMENDATION:

None

MOPROP_PERTNUM

Set the number of perturbed densities that will to be treated together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 All at once.

n Treat the perturbed densities batch-wise.

RECOMMENDATION:

Use the default. For large systems, limiting this number may be required to avoid memory exhaustion.

MOPROP_CONV_1ST

Sets the convergence criteria for CPSCF and 1st order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$ Convergence threshold set to 10^{-n}.

RECOMMENDATION:

None

MOPROP_CONV_2ND

Sets the convergence criterion for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$ Convergence threshold set to 10^{-n}.

RECOMMENDATION:

None

MOPROP_MAXITER_1ST

The maximum number of iterations for CPSCF and first-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n Set maximum number of iterations to n.

RECOMMENDATION:

Use the default.

MOPROP_MAXITER_2ND

The maximum number of iterations for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n Set maximum number of iterations to n.

RECOMMENDATION:

Use the default.

MOPROP_ISSC_PRINT_REDUCED

Specifies whether the isotope-independent reduced coupling tensor $\mathbf{K}$ should be printed in addition to the isotope-dependent $\mathbf{J}$-tensor when calculating indirect nuclear spin-spin couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print $\mathbf{K}$.

TRUE Print $\mathbf{K}$.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_FC

Specifies whether to skip the calculation of the Fermi contact contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate Fermi contact contribution.

TRUE Skip Fermi contact contribution.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_SD

Specifies whether to skip the calculation of the spin-dipole contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate spin-dipole contribution.

TRUE Skip spin-dipole contribution.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_PSO

Specifies whether to skip the calculation of the paramagnetic spin-orbit contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate paramagnetic spin-orbit contribution.

TRUE Skip paramagnetic spin-orbit contribution.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_DSO

Specifies whether to skip the calculation of the diamagnetic spin-orbit contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate diamagnetic spin-orbit contribution.

TRUE Skip diamagnetic spin-orbit contribution.

RECOMMENDATION:

None

MOPROP_DIIS

Controls the use of Pulay's DIIS in solving the CPSCF equations.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

0 Turn off DIIS.

5 Turn on DIIS.

RECOMMENDATION:

None

MOPROP_DIIS_DIM_SS

Specified the DIIS subspace dimension.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

0 No DIIS.

n Use a subspace of dimension n.

RECOMMENDATION:

None

SAVE_LAST_GPX

Save the last $\mathbf{G}[\mathbf{P}^x]$ when calculating dynamic polarizabilities in order to call the MOProp code in a second run, via MOPROP = 104 (which is otherwise the same as MOPROP = HYPERPOL).

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 False

1 True

RECOMMENDATION:

None

MOPROP_RESTART

Specifies the option for restarting MOProp calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Not a restart calculation.

1 Restart from a previous calculation using the same scratch directory.

RECOMMENDATION:

Need to also include "SCF_GUESS READ" and "SKIP_SCFMAN TRUE" to ensure the same set of MOs.

10.9.4 Examples

Example 10.47 HF/def2-SVPD static polarizability calculation for water cation, computed analytically using the MOProp module

```
$molecule
 1 2
   O  0.003  1.517  0.000
   H  0.913  1.819  0.000
   H  0.081  0.555  0.000
$end

$rem
   METHOD              hf
   BASIS               def2-svpd
   SCF_CONVERGENCE     11
   THRESH              14
   MOPROP              2
   MOPROP_CONV_1ST     8
   MOPROP_MAXITER_1ST  200
   INTEGRAL_SYMMETRY   false
   POINT_GROUP_SYMMETRY false
$end
```

10.10 NMR and Other Magnetic Properties

10.10.1 Introduction

The importance of nuclear magnetic resonance (NMR) spectroscopy for modern chemistry and biochemistry cannot be overestimated. Since there is no direct relationship between the measured NMR signals and structural properties, the necessity for a reliable method to predict NMR chemical shifts arises and despite tremendous progress in experimental techniques, the understanding and reliable assignment of observed experimental spectra remains often a highly difficult task. As such, quantum chemical methods can be extremely useful, both in solution and in the solid state.[33,153,155–157]

Features of Q-CHEM's NMR package include:

- Restricted Hartree-Fock and DFT calculations of NMR chemical shifts using gauge-including atomic orbitals.

- Support of linear-scaling CFMM and LinK procedures (Section 4.6) to evaluate Coulomb- and exchange-like matrices.
- Density matrix-based coupled-perturbed SCF approach for linear-scaling NMR calculations.
- DIIS acceleration.
- Support for basis sets up to h functions.
- Support for LDA, GGA, Meta-GGA[137], global hybrid and common range-separated functionals (RSH only support s, p and d basis functions). VV10 is the only non-local correlation functional supported.

Calculation of NMR chemical shifts and indirect spin-spin couplings is discussed in Section 10.10.2. Additional magnetic properties can be computed, as described in Section 10.10.4. These include hyperfine interaction tensors (electron spin–nuclear spin interaction) and nuclear quadrupole interactions with electric field gradients.

10.10.2 NMR Chemical Shifts and *J*-Couplings

NMR calculations are available at both the Hartree-Fock and DFT levels of theory.[80,206] Q-CHEM computes NMR chemical shielding tensors using gauge-including atomic orbitals[52,77,228] (GIAOs), an approach that has proven to reliable and accurate for many applications.[62,79] The shielding tensor $\boldsymbol{\sigma}$ is a second-order property that depends upon the external magnetic field, $\mathbf{B}$, and the spin angular momentum $\boldsymbol{m}$ for a given nucleus:

$$\Delta E = -\boldsymbol{m} \cdot (\mathbf{1} - \boldsymbol{\sigma}) \cdot \mathbf{B} \; . \tag{10.91}$$

Using analytical derivative techniques to evaluate $\boldsymbol{\sigma}$, the components of this 3×3 tensor are computed as

$$\sigma_{ij} = \sum_{\mu\nu} P_{\mu\nu} \left(\frac{\partial^2 h_{\mu\nu}}{\partial B_i \partial m_j} \right) + \sum_{\mu\nu} \frac{\partial P_{\mu\nu}}{\partial B_i} \frac{\partial h_{\mu\nu}}{\partial m_j} \tag{10.92}$$

where $i, j \in \{x, y, z\}$ indicate Cartesian components. Note that there is a separate chemical shielding tensor for each $\boldsymbol{m}$, that is, for each nucleus. To compute σ_{ij} it is necessary to solve coupled-perturbed SCF (CPSCF) equations to obtain the perturbed densities $\partial P / \partial B_i$, which can be accomplished using the MO-based "MOProp" module whose use is described below. (Use of the MOProp module to compute optical properties of molecules was discussed in Section 10.9.) Alternatively, a linear-scaling, density matrix-based CPSCF (D-CPSCF) formulation is available,[119,157] which is described in Section 10.10.3.

In addition to chemical shifts, indirect nuclear spin-spin coupling constants, also known as scalar couplings or J-couplings, can be computed at the SCF level. The coupling tensor $\mathbf{J}^{AB}$ between atoms A and B is evaluated as the second derivative of the electronic energy with respect to the nuclear magnetic moments $\boldsymbol{m}$:

$$\mathbf{J}^{AB} = \frac{\partial^2 E}{\partial \boldsymbol{m}_A \partial \boldsymbol{m}_B} \; . \tag{10.93}$$

The indirect coupling tensor has five distinct contributions. The diamagnetic spin-orbit (DSO) contribution is calculated as an expectation value with the ground state wave function. The other contributions are the paramagnetic spin-orbit (PSO), spin-dipole (SD), Fermi contact (FC), and mixed SD/FC contributions. These terms require the electronic response of the systems to the perturbation due to the magnetic nuclei. Ten distinct CPSCF equations must be solved for each perturbing nucleus, which makes the calculation of J-coupling constants more time-consuming than that of chemical shifts.

Some authors have recommended calculating only the Fermi contact contribution,[14] and skipping the other contributions, for ^{1}H-^{1}H coupling constants. For that purpose, Q-CHEM allows the user to skip calculation of any of the four contributions: (FC, SD, PSO, or DSO. (The mixed SD/FC contributions is automatically calculated at no additional

cost whenever both the SD and FC contributions are computed.) See Section 10.9.3 for details. Note that omitting any of the contributions cannot be rationalized from a theoretical point of view. Results from such calculations should be interpreted extremely cautiously.

Note:

1. Specialized basis sets are highly recommended in any J-coupling calculation. The pcJ-n basis set family[98] has been added to the basis set library.

2. The Hartree-Fock level of theory is *not* suitable to obtain J-coupling constants of *any* degree of reliability. Use GGA or hybrid density functionals instead.

10.10.2.1 NMR Job Control and Examples

This section describes the use of Q-CHEM's MO-based CPSCF code, which is contained in the "MOProp" module that is also responsible for computing electric properties. NMR chemical shifts are requested by setting MOPROP = 1, and J-couplings by setting JOBTYPE = ISSC. The reader is referred to to Section 10.9.3 for additional job control variables associated with the MOProp module, as well as explanations of the ones that are invoked in the samples below. An alternative, $\mathcal{O}(N)$ density matrix-based implementation of NMR chemical shifts is also available and is described in Section 10.10.3. Setting JOBTYPE = NMR invokes the density-based code, not the MO-based code.

Example 10.48 MO-based NMR calculation.

```
$molecule
0  1
  H        0.00000        0.00000        0.00000
  C        1.10000        0.00000        0.00000
  F        1.52324        1.22917        0.00000
  F        1.52324       -0.61459        1.06450
  F        1.52324       -0.61459       -1.06450
$end

$rem
  METHOD             B3LYP
  BASIS              6-31G*
  MOPROP               1
  MOPROP_PERTNUM       0  ! do all perturbations at once
  MOPROP_CONV_1ST      7  ! sets the CPSCF convergence threshold
  MOPROP_DIIS_DIM_SS   4  ! no. of DIIS subspace vectors
  MOPROP_MAXITER_1ST 100  ! max iterations
  MOPROP_DIIS          5  ! turns on DIIS (=0 to turn off)
  MOPROP_DIIS_THRESH   1
  MOPROP_DIIS_SAVE     0
$end
```

In the following compound job, we show how to restart an NMR calculation should it exceed the maximum number of CPSCF iterations (specified with MOPROP_MAXITER_1ST, or should the calculation run out of time on a shared computer resource. Note that the first job is *intentionally* set up to exceed the maximum number of iterations, so will

crash. However, the calculation is restarted and completed in the second job.

Example 10.49 Illustrates how to restart an NMR calculation. In this first job, we *intentionally* set the max number of iterations too small, to force premature end so that we can demonstrate restart capability in the 2nd job.

```
$molecule
0  1
  H        0.00000        0.00000        0.00000
  C        1.10000        0.00000        0.00000
  F        1.52324        1.22917        0.00000
  F        1.52324       -0.61459        1.06450
  F        1.52324       -0.61459       -1.06450
$end

$rem
   METHOD               B3LYP
   BASIS                6-31G*
   SCF_ALGORITHM        DIIS
   MOPROP               1
   MOPROP_MAXITER_1ST   10   ! too small, for demonstration only
   GUESS_PX             1
   MOPROP_DIIS_SAVE     0   ! don't hang onto the subspace vectors
$end

@@@

$molecule
0  1
  H        0.00000        0.00000        0.00000
  C        1.10000        0.00000        0.00000
  F        1.52324        1.22917        0.00000
  F        1.52324       -0.61459        1.06450
  F        1.52324       -0.61459       -1.06450
$end

$rem
   METHOD               B3LYP
   BASIS                6-31G*
   SCF_GUESS            READ
   SKIP_SCFMAN          TRUE   ! no need to redo the SCF
   MOPROP               1
   MOPROP_RESTART       1
   MOPROP_MAXITER_1ST   100  ! more reasonable choice
   GUESS_PX             1
   MOPROP_DIIS_SAVE     0
$end
```

Example 10.50 J-coupling calculation: water molecule with B3LYP/cc-pVDZ

```
$molecule
   0 1
   O
   H1 O OH
   H2 O OH H1 HOH

   OH  = 0.947
   HOH = 105.5
$end

$rem
   JOBTYPE           ISSC
   EXCHANGE          B3LYP
   BASIS             cc-pVDZ
   LIN_K             FALSE
   MOPROP_CONV_1ST   6
   INTEGRAL_SYMMETRY TRUE
$end
```

for which the couplings are computed. Selection is done via the *$spin-spin* input section, which is zero-indexed. For example, the section

```
$spin-spin
0 1 5
$end
```

would compute couplings between all possible pairings of the first, second, and sixth atoms in the respective *$molecule* section: (1, 2), (1, 6), and (2,6). If the *$spin-spin* section is not specified, couplings between all possible pairs of atoms in *$molecule* will be computed.

10.10.2.2 Nucleus-Independent Chemical Shifts: Probes of Aromaticity

Unambiguous theoretical estimates of degree of aromaticity are still on high demand. The NMR chemical shift methodology offers one unique probe of aromaticity based on one defining characteristics of an aromatic system: its ability to sustain a diatropic ring current. This leads to a response to an imposed external magnetic field with a strong (negative) shielding at the center of the ring. Von Schleyer *et al.* have employed this phenomenon to justify a new unique probe of aromaticity.[213] They proposed the computed absolute magnetic shielding at ring centers (unweighted mean of the heavy-atoms ring coordinates) as a new aromaticity criterion, called nucleus-independent chemical shift (NICS). Aromatic rings show strong negative shielding at the ring center (negative NICS), while anti-aromatic systems reveal positive NICS at the ring center. As an example, a typical NICS value for benzene is about -11.5 ppm as estimated with Q-CHEM at the Hartree-Fock/6-31G* level. The same NICS value for benzene was also reported in Ref. 213. The calculated NICS value for furan of -13.9 ppm with Q-CHEM is about the same as the value reported for furan in Ref. 213. Below is one input example of how to the NICS of furan with Q-CHEM, using the ghost atom option. The ghost atom is placed at the center of the furan ring, and the basis set assigned to it within the basis mix option must be the basis used for hydrogen atom.

Example 10.51 Calculation of the NMR NICS probe of furan, HF/6-31G* level. Note the ghost atom at the center of the ring.

```
$molecule
   0 1
   C    -0.69480       -0.62270       -0.00550
   C     0.72110       -0.63490        0.00300
   C     1.11490        0.68300        0.00750
   O     0.03140        1.50200        0.00230
   C    -1.06600        0.70180       -0.00560
   H     2.07530        1.17930        0.01410
   H     1.37470       -1.49560        0.00550
   H    -1.36310       -1.47200       -0.01090
   H    -2.01770        1.21450       -0.01040
   @H    0.02132        0.32584        0.00034
$end

$rem
   JOBTYPE               NMR
   METHOD                HF
   BASIS                 6-31G*
   PURCAR                111
   SCF_CONVERGENCE       7
   NO_REORIENT           1
   POINT_GROUP_SYMMETRY  FALSE
$end
```

10.10.3 Linear-Scaling NMR Chemical Shift Calculations

In conventional implementations, the cost for computation of NMR chemical shifts within even the simplest quantum chemical methods such as Hartree-Fock of DFT increases cubically with molecular size M, $\mathcal{O}(M^3)$. As such, NMR chemical shift calculations have largely been limited to molecular systems on the order of 100 atoms, assuming no symmetry. For larger systems it is crucial to reduce the increase of the computational effort to linear, which is possible for systems with a nonzero HOMO/LUMO gaps and was reported for the first time by Kussmann and Ochsenfeld.[118,157] This approach incurs no loss of accuracy with respect to traditional cubic-scaling implementations, and makes feasible NMR chemical shift calculations using Hartree-Fock or DFT approaches in molecular systems with 1,000+ atoms. For many molecular systems the Hartree-Fock (GIAO-HF) approach provides typically an accuracy of 0.2–0.4 ppm for the computation of ^{1}H NMR chemical shifts, for example.[33,153,155–157] GIAO-HF/6-31G* calculations with 1,003 atoms and 8,593 basis functions, without symmetry, have been reported.[157] GIAO-DFT calculations are even simpler and faster for density functionals that do not contain Hartree-Fock exchange.

The present implementation of NMR shieldings employs the LinK (linear exchange, "K") method[152,154] for the formation of exchange contributions.[157] Since the derivative of the density matrix with respect to the magnetic field is skew-symmetric, its Coulomb-type contractions vanish. For the remaining Coulomb-type matrices the CFMM method[224] is used.[157] In addition, a multitude of different approaches for the solution of the CPSCF equations can be selected within Q-CHEM.

To request a NMR chemical shift calculation using the density matrix approach, set JOBTYPE to NMR in the *$rem* section. Additional job-control variables can be found below.

D_CPSCF_PERTNUM

Specifies whether to do the perturbations one at a time, or all together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perturbed densities to be calculated all together.

1 Perturbed densities to be calculated one at a time.

RECOMMENDATION:

None

D_SCF_CONV_1

Sets the convergence criterion for the level-1 iterations. This preconditions the density for the level-2 calculation, and does not include any two-electron integrals.

TYPE:

INTEGER

DEFAULT:

4 corresponding to a threshold of 10^{-4}.

OPTIONS:

$n < 10$ Sets convergence threshold to 10^{-n}.

RECOMMENDATION:

The criterion for level-1 convergence must be less than or equal to the level-2 criterion, otherwise the D-CPSCF will not converge.

D_SCF_CONV_2

Sets the convergence criterion for the level-2 iterations.

TYPE:

INTEGER

DEFAULT:

4 Corresponding to a threshold of 10^{-4}.

OPTIONS:

$n < 10$ Sets convergence threshold to 10^{-n}.

RECOMMENDATION:

None

D_SCF_MAX_1

Sets the maximum number of level-1 iterations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n User defined.

RECOMMENDATION:

Use the default.

D_SCF_MAX_2

Sets the maximum number of level-2 iterations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User defined.

RECOMMENDATION:

Use the default.

D_SCF_DIIS

Specifies the number of matrices to use in the DIIS extrapolation in the D-CPSCF.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

n n = 0 specifies no DIIS extrapolation is to be used.

RECOMMENDATION:

Use the default.

Example 10.52 NMR chemical shifts via the D-CPSCF method, showing all input options.

```
$molecule
0  1
H        0.00000        0.00000        0.00000
C        1.10000        0.00000        0.00000
F        1.52324        1.22917        0.00000
F        1.52324       -0.61459        1.06450
F        1.52324       -0.61459       -1.06450
$end

$rem
   JOBTYPE            NMR
   EXCHANGE           B3LYP
   BASIS              6-31G*
   D_CPSCF_PERTNUM    0    D-CPSCF number of perturbations at once
   D_SCF_SOLVER       430  D-SCF   leqs_solver
   D_SCF_CONV_1       4    D-SCF   leqs_conv1
   D_SCF_CONV_2       4    D-SCF   leqs_conv2
   D_SCF_MAX_1        200  D-SCF   maxiter level 1
   D_SCF_MAX_2        50   D-SCF   maxiter level 2
   D_SCF_DIIS         11   D-SCF   DIIS
   D_SCF_ITOL         2    D-SCF   conv. criterion
$end
```

10.10.4 Additional Magnetic Field-Related Properties

In addition to NMR chemical shieldings and spin-spin couplings, other magnetic properties available in Q-CHEM are

- hyperfine interaction tensors,
- the nuclear quadrupole interaction from electric field gradient tensors, and
- the electronic g-tensor,

10.10.4.1 Hyperfine Interaction

The hyperfine interaction tensor describes the interaction the interaction of unpaired electron spin with an atom's nuclear spin levels:

$$\hat{H}_{\text{HFI}}/h = \hat{\mathbf{S}} \cdot \mathbf{A} \cdot \hat{\mathbf{I}}, \tag{10.94}$$

which is broken down into Fermi contact (FC), spin-dipole (SD), and orbital Zeeman/spin-orbit coupling (OZ/SOC) terms:

$$A_{ab}^{\text{tot}}(N) = A_{ab}^{\text{FC}}(N)\delta_{ab} + A_{ab}^{\text{SD}}(N) + A_{ab}^{\text{OZ/SOC}}, \tag{10.95}$$

where the Fermi contact (FC) contribution is

$$A^{\text{FC}}(N) = \frac{\alpha}{2}\frac{1}{S}\frac{8\pi}{3} g_e g_N \mu_N \sum_{\mu\nu} P_{\mu\nu}^{\alpha-\beta} \langle \chi_\mu | \delta(\mathbf{r}_N) | \chi_\nu \rangle \tag{10.96}$$

and the spin-dipole (SD) contribution is

$$A_{ab}^{\text{SD}}(N) = \frac{\alpha}{2}\frac{1}{S} g_e g_N \mu_N \sum_{\mu\nu} P_{\mu\nu}^{\alpha-\beta} \left\langle \chi_\mu \left| \frac{3r_{N,a}r_{N,b} - \delta_{ab}r_N^2}{r_N^5} \right| \chi_\nu \right\rangle \tag{10.97}$$

for a nucleus N. The orbital Zeeman/spin-orbit coupling cross-term (OZ/SOC) is currently not available.

Hyperfine interaction tensors are available for all SCF-based methods with an unrestricted (not restricted open-shell) reference. Post-HF methods are unavailable.

Calculation of excited state (CIS/TDA-TDDFT) singlet-triplet hyperfine couplings are also available, but formatted differently than the ground state unrestricted counterpart. Excited state couplings are printed as contributions from FC and SD for each pair of excited states and for each spin operator S_x, S_y, S_z. For this method, the nuclear spin states are averaged to get final coupling contributions[13].

10.10.4.2 Nuclear Quadrupole Interaction

Another sensitive probe of the individual nuclear environments in a molecule is the nuclear quadrupole interaction (NQI), which is a measure of how a nuclear quadrupole moment interacts with the local electric field gradient:

$$\hat{H}_{\text{NQI}}/h = \hat{\mathbf{I}} \cdot \mathbf{Q} \cdot \hat{\mathbf{I}}, \tag{10.98}$$

$$\begin{aligned} Q_{ab}(N) &= \frac{\partial^2 V_{eN}}{\partial X_{N,a} \partial X_{N,b}} + \frac{\partial^2 V_{NN}}{\partial X_{N,a} \partial X_{N,b}} \\ &= -\sum_{\mu\nu} P_{\mu\nu}^{\alpha+\beta} \left\langle \chi_\mu \left| \frac{3 r_{N,a} r_{N,b} - \delta_{ab} r_N^2}{r_N^5} \right| \chi_\nu \right\rangle \\ &\quad + \sum_{A \neq N} Z_A \frac{3 R_{AN,a} R_{AN,b} - \delta_{ab} R_{AN}^2}{R_{AN}^5} \end{aligned} \tag{10.99}$$

for a nucleus N. Diagonalizing the tensor gives three principal values, ordered $|Q_1| \leq |Q_2| \leq |Q_3|$, which are components of the asymmetry parameter eta:

$$\eta = \frac{Q_1 - Q_2}{Q_3} \tag{10.100}$$

10.10.4.3 Electronic *g*-tensor

The electronic g-tensor is a measure of the electron describes the coupling of unpaired electron spins with an external magnetic field, represented by the phenomenological Hamiltonian

$$\hat{H}^{g-tensor} = \mu_B \mathbf{S} \cdot \mathbf{g} \cdot \mathbf{B}, \tag{10.101}$$

where μ_B is the Bohr magneton, $\mathbf{S}$ is the intrinsic molecular spin vector, and $\mathbf{B}$ is the incident magnetic field vector.

The g-tensor is comprised of the Spin-Zeeman term and the g-tensor shift that includes the relativistic mass correction $\mathbf{g}^{rmc}$, diamagnetic spin-orbit coupling $\mathbf{g}^{dso}$ and paramagnetic spin-orbit coupling $\mathbf{g}^{pso}$ terms

$$\mathbf{g} = g_e \mathbf{I} + \mathbf{g}^{rmc} + \mathbf{g}^{dso} + \mathbf{g}^{pso}. \tag{10.102}$$

For the Spin-Zeeman term the contribution is isotropic and equals the free electron g-factor. The relativistic interaction terms are added as perturbations following the Breit-Pauli ansatz resulting the the following expressions. The relativistic mass correction shift term g^{rmc} is

$$g_{pq}^{rmc} = -\frac{\alpha^2 g_e}{2S} \delta_{pq} \sum_{\mu\nu} P_{\mu\nu}^{\alpha-\beta} T_{\mu\nu} \tag{10.103}$$

with α as the fine-structure constant, $P^{\alpha-\beta}$ as spin density and T as kinetic energy integrals. The diamagnetic spin-orbit term g^{dso} is currently not implemented in Q-CHEM and therefore excluded but typically also only of minor importance for lighter elements or first to second row transition metal systems.

The paramagnetic spin-orbit coupling term g^{pso} is a second-order term in the perturbation series but constitutes the main contribution to the g-tensor shift

$$g^{pso} = \frac{1}{\alpha S} \sum_{N} \frac{\langle \Psi_0 | h^{SO} | \Psi_N \rangle \langle \Psi_N | h^{OZ} | \Psi_0 \rangle}{E_N - E_0} \tag{10.104}$$

where h^{SO} is the spin-orbit coupling interaction where a spin-orbit mean-field approach[58] is used by default and h^{OZ} the orbital Zeeman interaction

$$h^{OZ} = \mu_B \mathbf{L} \cdot \mathbf{B} \tag{10.105}$$

with $\mathbf{L}$ as angular momentum.

In this implementation the paramagnetic spin-orbit coupling term is evaluated using a response theory approach, as first demonstrated by Gauss *et al.*[63], but with a computational approach following that used in the Q-CHEM polarization code[146]. At the moment the g-tensor is only implemented at the CCSD level.

10.10.4.4 Job Control and Examples

Only one keyword is necessary in the *$rem* section to activate the magnetic property module.

MAGNET
Activate the magnetic property module.
TYPE:
LOGICAL
DEFAULT:
FALSE
OPTIONS:
FALSE (or 0) Don't activate the magnetic property module.
TRUE (or 1) Activate the magnetic property module.
RECOMMENDATION:
None.

All other options are controlled through the *$magnet* input section, which has the same key-value format as the *$rem* section (see section 3.4). Current options are:

HYPERFINE
Activate the calculation of hyperfine interaction tensors.
INPUT SECTION: *$magnet*
TYPE:
LOGICAL
DEFAULT:
FALSE
OPTIONS:
FALSE (or 0) Don't calculate hyperfine interaction tensors.
TRUE (or 1) Calculate hyperfine interaction tensors.
RECOMMENDATION:
None. Due to the nature of the property, which requires the spin density $\rho^{\alpha-\beta}(\mathbf{r}) \equiv \rho^{\alpha}(\mathbf{r}) - \rho^{\beta}(\mathbf{r})$, this is not meaningful for restricted (RHF) references. Only UHF (not ROHF) is available.

ELECTRIC

Activate the calculation of electric field gradient tensors.

INPUT SECTION: *$magnet*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Don't calculate EFG tensors and nuclear quadrupole parameters.

TRUE (or 1) Calculate EFG tensors and nuclear quadrupole parameters.

RECOMMENDATION:

None.

For both hyperfine and EFG tensors, the results for all nuclei are automatically calculated.

HYPERFINE_FULL

Activate calculation of excited state hyperfine couplings.

INPUT SECTION: *$magnet*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

False (or 0) Don't calculate excited state hyperfine couplings.

True (or 1) Calculate excited state hyperfine couplings.

RECOMMENDATION:

None.

HYPERFINE_FULL_NROOTS

Specify number of roots for excited state hyperfine calculation.

INPUT SECTION: *$magnet*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Calculate hyperfine couplings between lowest n CIS/TDA-TDDFT states.

RECOMMENDATION:

None.

HYPERFINE_FULL_ROOT_OFFSET

Specify offset for roots for excited state hyperfine calculation.

INPUT SECTION: *$magnet*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

m Calculate hyperfine couplings between lowest n CIS/TDA-TDDFT states, starting at m^{th} root.

RECOMMENDATION:

None.

HYPERFINE_FULL_SINGLET_OTHER_SINGLET

Specify whether to calculate couplings between singlet excited states.

INPUT SECTION: *$magnet*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

False (or 0) Don't calculate excited state hyperfine couplings between singlets.

True (or 1) Calculate excited state hyperfine couplings between singlets.

RECOMMENDATION:

None.

HYPERFINE_FULL_TRIPLET_OTHER_TRIPLET

Specify whether to calculate couplings between triplet excited states.

INPUT SECTION: *$magnet*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

False (or 0) Don't calculate excited state hyperfine couplings between triplets.

True (or 1) Calculate excited state hyperfine couplings between triplets.

RECOMMENDATION:

None.

Calculation of g-tensor is activated by specifying the G_TENSOR keyword in the *$rem* section. Example 10.54 illustrates g-tensor calculation for water cation.

G_TENSOR

Activates g-tensor calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Don't calculate g-tensor

TRUE (or 1) Calculate g-tensor.

RECOMMENDATION:

None.

Example 10.53 Calculating hyperfine and EFG tensors for the glycine cation.

```
$molecule
   1 2
   N     0.0000000000     0.0000000000     0.0000000000
   C     1.4467530000     0.0000000000     0.0000000000
   C     1.9682482963     0.0000000000     1.4334965024
   O     1.2385450522     0.0000000000     2.4218667010
   H     1.7988742211    -0.8959881458    -0.5223754133
   H     1.7997303368     0.8930070757    -0.5235632630
   H    -0.4722340827    -0.0025218132     0.8996536532
   H    -0.5080000000     0.0766867527    -0.8765335943
   O     3.3107284257    -0.0000000000     1.5849828121
   H     3.9426948542    -0.0000000000     0.7289954096
$end

$rem
   METHOD                hf
   BASIS                 def2-sv(p)
   SCF_CONVERGENCE       11
   THRESH                14
   MAGNET                true
   INTEGRAL_SYMMETRY     false
   POINT_GROUP_SYMMETRY  false
$end

$magnet
   HYPERFINE  true
   ELECTRIC   true
$end
```

Example 10.54 Calculating g-tensor for the water cation.

```
$molecule
   1 2
   O    0.00000000   0.00000000   0.13475163
   H    0.00000000  -1.70748899  -1.06930309
   H    0.00000000   1.70748899  -1.06930309
$end

$rem
   INPUT_BOHR      = true
   METHOD          = ccsd
   BASIS           = 3-21g
   CC_REF_PROP     = true
   G_TENSOR        = true
   N_FROZEN_CORE   = 0
   NO_REORIENT     = true
   SCF_CONVERGENCE = 12
   CC_CONVERGENCE  = 12
   POINT_GROUP_SYMMETRY = false
$end

$gauge_origin
0.000000 0.000000  0.0172393
$end
```

Excited state HFC values are printed as

```
Nucleus:  IDX
...
STATE1 / STATE2
FC: S_x_contribution S_y_contribution S_z_contribution
SD: S_x_contribution S_y_contribution S_z_contribution
...
```

Example 10.55 Calculating excited state hyperfine coupling for Pyrene-Dimethylaniline radical pair.

```
$molecule
   READ    pydma_xyz.txt
$end

$rem
   EXCHANGE             wb97x-d3
   BASIS                def2-svpd
   CIS_SINGLETS         true
   CIS_TRIPLETS         true
   CIS_MULLIKEN         true
   CIS_N_ROOTS          15        ! 15 TDA-TDDFT roots for each spin state
   RPA                  false
   MAGNET               true      ! Turn on magnetman
   CC_PRINT_PREC        12        ! Print precision affects precision of printed hfc
   POINT_GROUP_SYMMETRY false
   CIS_MAX_CYCLES       500
   SCF_MAX_CYCLES       500
   SCF_CONVERGENCE      10
   CIS_CONVERGENCE      8
   XC_GRID              3
$end

$magnet
   HYPERFINE_FULL                       true  ! Activate excited state hfc calculation
   HYPERFINE_FULL_NROOTS                30    ! Calculate all roots
   HYPERFINE_FULL_SINGLET_OTHER_SINGLET false ! Neglect singlet-singlet couplings
   HYPERFINE_FULL_TRIPLET_OTHER_TRIPLET false ! Neglect triplet-triplet couplings
$end
```

10.11 Vibrational Circular Dichroism (VCD)

The VCD signals in dilute solution in terms of the differential molar extinction coefficient, $\Delta\epsilon = \epsilon_{\text{left}} - \epsilon_{\text{right}}$, is given by

$$\Delta\epsilon = 4\gamma\nu \sum_i \alpha_g R_{gi} f(\nu_{gi}, \nu) \,, \tag{10.106}$$

where

$$\gamma = \frac{N_A \pi^3}{375 \ln(10) hc} \,, \tag{10.107}$$

N_A is Avogadro constant, h is Planck's constant and c is the speed of light, ν is the frequency of the incident light, and α_g is the population of the ground state The quantity R_{gi} is the rotational strength, with indices g and i represent the ground state and the ith vibrational excited state, respectively; $f(\nu_{gi}$ is the lineshape function; and ν_{gi} is the transition frequency between vibrational states g and i. The rotational strength is given by

$$R_{gi} = \Im[\boldsymbol{P}_i \cdot \boldsymbol{M}_i] \,, \tag{10.108}$$

where $\Im[\cdots]$ means the the imaginary part of a complex number, $\boldsymbol{M}_i$ is the vibrational magnetic transition dipole vector, and $\boldsymbol{P}_i$ is the vibrational electric transition dipole vector. The latter can be expressed as

$$\boldsymbol{P}_i = \langle 0|\hat{\boldsymbol{\mu}}_{\rm e}|1\rangle_i = \sqrt{\frac{\hbar}{4\pi\nu_i}} \left(\frac{\partial \langle \Psi_g|\hat{\boldsymbol{\mu}}_{\rm e}|\Psi_g\rangle}{\partial Q_i} \right)\Bigg|_{R_0} . \tag{10.109}$$

Ψ_g denotes the ground state wave function, Q_i is the i-h normal mode, and this derivative is evaluated at the equilibrium geometry R_0. The quantity

$$\hat{\boldsymbol{\mu}}_{\rm e} = 1 + \sum_\lambda Z_\lambda \delta(r - R_\lambda) er \tag{10.110}$$

is the electric transition dipole operator, where Z_λ is the atomic number for atom λ, located at position R_λ.

Since the probability of the $1 \leftarrow 0$ transition of the ith normal mode is proportional to the modulus square of the corresponding electric transition dipole, $|\boldsymbol{P}_i|^2$, the quantity $\boldsymbol{P}_i$ is evaluated in the frequency subroutine. Suppose the linear transform relation between the normal modes Q_i and the atomic Cartesian displacement iis

$$X_{\lambda\alpha} = \sum_i S_{\lambda\alpha,i}\, Q_i \tag{10.111}$$

where λ is the index for nuclei and $\alpha \in \{x, y, z\}$. Matrix elements $S_{\lambda\alpha,i}$ define the normal modes and are obtained from a harmonic frequency calculation. We can define the *atomic polar tensor* (APT) having matrix elements

$$\boldsymbol{P}_{\lambda\alpha,\beta} = \left(\frac{\partial \langle \Psi_g|(\hat{\boldsymbol{\mu}}_{\rm e})_\beta|\Psi_g\rangle}{\partial \boldsymbol{X}_{\lambda\alpha}} \right)\Bigg|_{R_0} . \tag{10.112}$$

Thus,

$$\tilde{\boldsymbol{P}}_{i\beta} = \sqrt{\frac{\hbar}{4\pi\nu_i}} \sum_{\lambda\alpha} \boldsymbol{P}_{\lambda\alpha,\beta}\, \boldsymbol{S}_{\lambda\alpha,i} . \tag{10.113}$$

The quantity $\boldsymbol{P}_{\lambda\alpha,\beta}$ can be separated into the nuclear and electronic parts,

$$\boldsymbol{P}_{\lambda\alpha,\beta} = \boldsymbol{N}_{\lambda\alpha,\beta} + \boldsymbol{E}_{\lambda\alpha,\beta} , \tag{10.114}$$

where

$$\boldsymbol{N}_{\lambda\alpha,\beta} = Z_\lambda e \delta_{\alpha\beta} \tag{10.115}$$

and

$$\boldsymbol{P}_{\lambda\alpha,\beta} = \left(\frac{\partial \langle \Psi_g|e\boldsymbol{r}_\beta|\Psi_g\rangle}{\partial \boldsymbol{X}_{\lambda\alpha}} \right)\Bigg|_{R_0} = 2 \left\langle \psi_g \middle| e\boldsymbol{r}_\beta \middle| \frac{\partial \Psi_g}{\partial X_{\lambda\alpha}} \right\rangle . \tag{10.116}$$

Similarly, for the vibrational magnetic transition dipole $\tilde{\boldsymbol{M}}_i$, we have

$$\tilde{\boldsymbol{M}}_{i\beta} = \langle 0|\hat{\boldsymbol{\mu}}_{\rm m}|1\rangle_{i\beta} = -\sqrt{4\pi\hbar^3\nu_i} \sum_{\lambda\alpha} \boldsymbol{M}_{\lambda\alpha,\beta}\, S_{\lambda\alpha,i} \tag{10.117}$$

where

$$\hat{\boldsymbol{\mu}}_{\rm m} = \frac{e}{2m}(\hat{r} \times \hat{p}) \tag{10.118}$$

and $\boldsymbol{M}^\lambda_{\alpha\beta}$ is the *atomic axial tensor* (AAT). Moreover,

$$\boldsymbol{M}_{\lambda\alpha,\beta} = \boldsymbol{I}_{\lambda\alpha,\beta} + \boldsymbol{J}_{\lambda\alpha,\beta} \tag{10.119}$$

where

$$\boldsymbol{I}_{\lambda\alpha,\beta} = \left\langle \left(\frac{\partial \Psi_g}{\partial X_{\lambda\alpha}} \right)_{R_0} \middle| \left(\frac{\partial \Psi_g}{\partial H_\beta} \right)_{H_\beta = 0} \right\rangle \tag{10.120}$$

and

$$\boldsymbol{J}_{\lambda\alpha,\beta} = \frac{i}{4\hbar c} \sum_\gamma \epsilon_{\alpha\beta\gamma} Z_\lambda e R_{\lambda\gamma} . \tag{10.121}$$

The quantity $\epsilon_{\alpha\beta\gamma}$ is Levi-Civita symbol and $R_{\lambda\gamma}$ is some Cartesian component of R_γ ($\gamma \in \{x, y, z\}$). The derivative of the wavefunction with respect to the external magnetic field H_β is evaluated with a perturbation $H' = -\boldsymbol{\mu}_m \cdot H_\beta$ at the limit of $H_\beta \to 0$. Note that $\tilde{M}_i$ is pure imaginary if real basis functions are used.

The working equation for rotational strength can be expressed as

$$R_{gi} = \hbar^2 \Im \left[\sum_\beta \left(\sum_{\lambda\alpha} \boldsymbol{P}_{\lambda\alpha,\beta} S_{\lambda\alpha,i} \cdot \sum_{\lambda'\alpha'} \boldsymbol{M}_{\lambda'\alpha',\beta'} S_{\lambda'\alpha',i} \right) \right] \tag{10.122}$$

where λ', α', and β' are dummy indices for atoms and Cartesian coordinates, respectively. The $\boldsymbol{I}^\lambda_{\alpha\beta}$ tensor can be rewritten as

$$\boldsymbol{I}_{\lambda\alpha,\beta} = \sum_{\mu\nu} \left(D_{\mu\nu} \left\langle \chi_\mu^{X_{\lambda\alpha}} \middle| \chi_\nu^{H_\beta} \right\rangle + D_{\mu\nu}^{H_\beta} \left\langle \chi_\mu^{X_{\lambda\alpha}} \middle| \chi_\nu \right\rangle + D_{\mu\nu}^{X_{\lambda\alpha}} \left\langle \chi_\mu \middle| \chi_\nu^{H_\beta} \right\rangle + D_{\mu\nu}^{X_{\lambda\alpha},H_\beta} \left\langle \chi_\mu \middle| \chi_\nu \right\rangle \right) , \tag{10.123}$$

defining the intermediate quantities

$$D_{\mu\nu} = \sum_i^{\text{occ}} c^*_{\mu i} c_{\nu i} \tag{10.124a}$$

$$D_{\mu\nu}^{H_\beta} = \sum_i^{\text{occ}} c^*_{\mu i} c_{\nu i}^{H_\beta} \tag{10.124b}$$

$$D_{\mu\nu}^{X_{\lambda\alpha}} = \sum_i^{\text{occ}} c_{\mu i}^{X_{\lambda\alpha}*} c_{\nu i} \tag{10.124c}$$

$$D_{\mu\nu}^{X_{\lambda\alpha},H_\beta} = \sum_i c_{\mu i}^{X_{\lambda\alpha}*} c_{\nu i}^{H_\beta} \tag{10.124d}$$

and

$$c_{\nu i}^{H_\beta} = \left. \frac{\partial c_{\nu i}}{\partial H_\beta} \right|_{H_\beta = 0} \tag{10.125a}$$

$$c_{\mu i}^{X_{\lambda\alpha}} = \left. \frac{\partial c_{\mu i}}{\partial X_{\lambda\alpha}} \right|_{R_0 = 0} \tag{10.125b}$$

where $\chi_{\mu,\nu}$ are AO basis functions and $c_{\mu,\nu i}$ are MO coefficients. Superscripts of c indicate MO coefficient derivatives with respect to nuclear position and magnetic field. Derivatives of the MO coefficient are obtained by solving CPHF/ CPKS equations.

Electric transition dipoles are origin-independent while magnetic transition dipoles are origin-dependent with finite basis sets. In order to obtain origin-independent (gauge-invariant) VCD properties, one can employ the explicit field-dependent GIAO basis functions:

$$\chi_{\mu,\text{GIAO}}(\boldsymbol{H}) = \exp\left[-\frac{i}{2c} (\boldsymbol{H} \times \boldsymbol{R}_\mu) \cdot \boldsymbol{r} \right] \chi_\mu \tag{10.126}$$

To extract the VCD properties, one should (at a minimum) carry out a frequency analysis with the system. Several available *$rem* variables include:

VCD

Controls calculation of the VCD signals. Requires JOBTYPE to be set to FREQ

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate the VCD properties.

TRUE Do calculate the VCD properties.

RECOMMENDATION:

None

VCD_PRINT

Controls level of extra print out for the VCD calculations.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Standard full information print out.

2 Electronic part of AAT.

RECOMMENDATION:

Use the default.

Example 10.56 An PBE/STO-3G optimization, followed by a VCD calculation.

```
$comment
(-)-camphore
$end

$molecule
0 1
 O  -2.5217    0.3747   -0.2628
 C   0.9690    0.2807   -0.4137
 C  -0.2122    0.5268    0.5653
 C   0.5981   -1.2183   -0.5902
 C  -0.0177   -0.5488    1.6588
 C   0.5627   -1.7425    0.8680
 C  -0.8632   -1.1910   -1.1027
 C  -1.3692   -0.0289   -0.2712
 C   0.9102    1.0940   -1.7214
 C   2.3671    0.5229    0.1910
 C  -0.4232    1.9305    1.0788
 H   1.2655   -1.7945   -1.2356
 H  -0.9615   -0.8246    2.1436
 H   0.6774   -0.2257    2.4410
 H   1.5668   -1.9925    1.2263
 H  -0.0563   -2.6395    0.9757
 H  -1.4099   -2.1082   -0.8724
 H  -0.9272   -0.9564   -2.1673
 H   1.1849    2.1396   -1.5401
 H  -0.0766    1.1153   -2.1913
 H   1.6143    0.6876   -2.4563
 H   2.5320    0.0303    1.1519
 H   2.5399    1.5928    0.3537
 H   3.1441    0.1597   -0.4916
 H   0.4499    2.2884    1.6335
 H  -0.6167    2.6304    0.2592
 H  -1.2875    1.9748    1.7507
$end

$rem
   JOBTYPE     opt
   BASIS       sto-3g
   METHOD      pbe
   NO_REORIENT true
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

@@@

$molecule
read
$end

$rem
   JOBTYPE     freq
   BASIS       sto-3g
   METHOD      pbe
   VCD         1
   VCD_PRINT   2
   NO_REORIENT true
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end
```

10.12 Finite-Field Calculation of (Hyper)Polarizabilities

10.12.1 Introduction

The dipole moment vector ($\vec{\mu}$), polarizability tensor ($\overleftrightarrow{\alpha}$), first hyperpolarizability ($\vec{\overleftrightarrow{\beta}}$), and higher-order hyperpolarizabilities determine the response of the system to an applied electric field:

$$E(\vec{F}) = E(0) - \vec{\mu}(0) \cdot \vec{F} - \frac{1}{2!}\overleftrightarrow{\alpha} : \vec{F}\vec{F} - \frac{1}{3!}\vec{\overleftrightarrow{\beta}}\vdots\vec{F}\vec{F}\vec{F} - \cdots \ . \tag{10.127}$$

The various polarizability tensor elements are therefore derivatives of the energy with respect to one or more electric fields, which might be frequency-dependent (dynamic polarizabilities) or not (static polarizabilities). The most efficient way to compute these properties is by analytic gradient techniques, assuming that the required derivatives have been implemented at the desired level of theory. For DFT calculations using LDA, GGAs, or global hybrid functionals the requisite analytic gradients have been implemented and their use to compute static and dynamic (hyper)polarizabilities is described in Section 10.9.

10.12.2 Numerical Calculation of Static Polarizabilities

Where analytic gradients are not available, static polarizabilities (only) can be computed via finite-difference in the applied field, which is known as the *finite field* (FF) approach. Beginning with Q-CHEM 5.1, a sophisticated "Romberg" approach to FF differentiation is available, which includes procedures for assessing the stability of the results with respect to the finite-difference step size. The Romberg approach is described in Section 10.12.3. This section describes Q-CHEM's older approach to FF calculations based on straightforward application of small electric fields along the appropriate Cartesian directions.

Dipole moments can be calculated numerically as the first derivative of the energy with respect to $\vec{F}$ by setting JOBTYPE = DIPOLE and IDERIV = 0. If IDERIV is not specified explicitly, the dipole moment will be calculated analytically, which for post-Hartree–Fock levels of theory invokes a gradient calculation in order to utilize the relaxed wavefunction.

Similarly, set JOBTYPE = POLARIZABILITY for numerical evaluation of the static polarizability tensor $\overleftrightarrow{\alpha}$. This is performed by either first-order finite difference, taking first-order field derivatives of analytic dipole moments, or by second-order finite difference of the energy. The latter is useful (indeed, required) for methods where analytic gradients are not available, such as CCSD(T) for example. Note, however, that the electron cloud is formally unbound in the presence of static electric fields and therefore a bound solution is a consequence of using a finite basis set. (With analytic derivative techniques the perturbing field is infinitesimal so this is not an issue.) This fact, along with the overall sensitivity of numerical derivatives to the finite-difference step size, means that care must be taken in choosing the strength of the applied finite field.

To control the order for numerical differentiation with respect to the applied electric field, use IDERIV in the same manner as for geometric derivatives, *i.e.*, for polarizabilties use IDERIV = 0 for second-order finite-difference of the energy and IDERIV = 1 for first-order finite difference of gradients. In addition, for numerical polarizabilities at the Hartree-Fock or DFT level set RESPONSE_POLAR = -1 in order to disable the analytic polarizability code.

RESPONSE_POLAR

Control the use of analytic or numerical polarizabilities.

TYPE:

INTEGER

DEFAULT:

0 or -1 = 0 for HF or DFT, -1 for all other methods

OPTIONS:

0 Perform an analytic polarizability calculation.

-1 Perform a numeric polarizability calculation even when analytic 2nd derivatives are available.

RECOMMENDATION:

None

10.12.3 Romberg Finite-Field Procedure

Whereas the FF procedure described in Section 10.12.2 is a straightforward, finite-difference implementation of the derivatives suggested in Eq. (10.127), in the Romberg procedure[51] one combines energy values obtained for a succession of k external electric fields with amplitudes that form a geometric progression:

$$F(k) = a^k F_0 \ . \tag{10.128}$$

The FF expressions are obtained by combining truncated Taylor expansions of the energy with different amplitudes and/or external field directions. For example, in the case of the diagonal $\boldsymbol{\beta}$-tensor components the Romberg FF expression is

$$\beta_{iii}(k,0) = 3\left(\frac{\big[E(F_{-i}(k+1) - E(F_i(k+1))\big] - a\big[E(F_{-i}(k) - E(F_i(k))\big]}{a(a^2-1)(a^k F_0)^3}\right) \ . \tag{10.129}$$

The field index $\pm i$ refers to the possible field directions, *i.e.*, $\pm x$, $\pm y$ or $\pm z$. Truncation of the Taylor expansions means that the results are contaminated by higher-order hyperpolarizabilities, and to remove this contamination, successive "Romberg iterations" are performed using a recursive expression. For a component of a (hyper)polarizability tensor $\boldsymbol{\zeta}$, the recursive expression is

$$\zeta(k,n) = \frac{a^{2n}\zeta(k,n-1) - \zeta(k+1,n-1)}{a^{2n}-1} \ . \tag{10.130}$$

This expression leads to a triangular Romberg table enabling monitoring of the convergence of the numerical derivative.[51]

As with any finite-difference procedure, the FF method for computing (hyper)polarizabilities is sensitive to the details of numerical differentiation. The Romberg procedure allows one to find a field window, defined by its upper and lower bounds, where the finite-difference procedure is stable. Energy values for field amplitudes below that window suffer from too-large round-off errors, which are proportional to the energy convergence threshold. The upper bound is imposed by the critical field amplitude corresponding to the intersection between the ground and excited-state energies. In the Romberg procedure, this stability window defines a sub-triangle, determination of which is the primary goal in analyzing Romberg tables.

The automatic procedure based on the analysis of field amplitude errors is implemented in scripts provided with Q-CHEM's distribution. The field amplitude error is defined as the difference between ζ-values obtained for consecutive field amplitudes at the same Romberg iteration:

$$\epsilon_k(n) = \zeta(k+1,n) - \zeta(k,n) \ . \tag{10.131}$$

By virtue of Romberg's recursive expression, the field error is expected to decrease with each iteration. Convergence of the Romberg procedure can be probed using the iteration (order) error, defined as

$$\epsilon_n(k) = \zeta(k,n+1) - \zeta(k,n) \ . \tag{10.132}$$

Automatic analysis of these quantities is described in detail in Ref. 51.

Note:

- The automatic procedure can fail if the field window is not chosen wisely.
- The Romberg procedure can be performed either for one specific diagonal direction or for all Cartesian components. In the case of the second hyperpolarizability, only the $iiii$, $iijj$ components are available.

10.12.3.1 How to execute Romberg's differentiation procedure

To perform Romberg calculations of (hyper)polarizabilities, the following utilities are in the `$QC/bin/tools/Romberg` directory:

```
input-Q-Chem-t-rex.sh
parse-t-rex.sh
input-Q-Chem-t-rex-3.0
romberg_tddft_read
romberg_eom_read
T-REX-3.0.3
```

and this directory must be manually added to the `$PATH` environment variable depending on the shell being used:

```
# bash syntax for ~/.bashrc:
export PATH="$QC/bin/tools/Romberg:$PATH"
# csh/tcsh syntax for ~/.cshrc:
set path=($QC/bin/tools/Romberg $path)
```

To run the calculation, create an input called `input` that specifies your molecule, an electronic structure method, and several additional *$rem* variables that

- turn off symmetry (INTEGRAL_SYMMETRY and CC_SYMMETRY for CC/EOM calculations),
- request higher-precision printing (CC_PRINT_PREC), and
- set up very tight convergence (SCF_CONVERGENCE, CC_CONVERGENCE, EOM_DAVIDSON_CONVERGENCE, *etc.*).

An example of an input file is given below.

Example 10.57 Input for Romberg calculations of the ethylene molecule using B3LYP.

```
$molecule
0 1
 C        0.00000000   0.00000000 -0.66880000
 H        0.94859916   0.00000000 -1.19917145
 H       -0.94859916   0.00000000 -1.19917145
 C       -0.00000000   0.00000000  0.66880000
 H       -0.54409413   0.77704694  1.19917145
 H        0.54409413  -0.77704694  1.19917145
$end

$rem
   BASIS            =  sto-3g
   EXCHANGE         =  B3lyp
   SCF_CONVERGENCE  =  13  ! tight convergence for finite field calculations
   SCF_MAX_CYCLES   =  200
   CC_PRINT_PREC    =  16  ! decimal points of total energies
   INTEGRAL_SYMMETRY = false ! all symmetries need to be turned off
$end
```

Run the script `input-Q-Chem-t-rex.sh` and answer the questions regarding the parameters of the geometrical progression of field amplitudes (see example below). The script will create multiple input files for the FF calculations based on the input file template that you provided. The file name that is requested serves as a prefix to each automatically-generated FF input file. For example, if you specify the filename as `water`, input files named `water001.inp`, `water002.inp`, ... will be generated. These must be run independently of the Romberg procedure, such as on a computing cluster, or locally with GNU parallel or xargs. For example, with GNU parallel, to run 6 Q-CHEM jobs at a time, each with 4 threads,

```
ls hello*.inp | parallel -j6 'qchem -nt 4 {.}.inp {.}.out'
```

After running the calculations, parse the output files using the script `parse-t-rex.sh`. Then run the `T-REX` program. Answer the the questions to compute the dipole moment and (hyper)polarizabilities (see example below).

Romberg differentiation is only available for methods where printing the total energy to high precision has been enabled. In the current version of Q-CHEM, CC_PRINT_PREC is implemented for the following methods: HF, DFT, MP2, RI-MP2, MP3, CCD, CCSD, CCSD(T), QCISD, QCISD(T), TDDFT, and EOM-CCSD.

Note: When using excited-state methods such as TDDFT, CIS, and EOM-CC, state ordering may switch when the external field is large.

With the T-REX program, you can compute the static dipole moment, polarizability, and first and second hyperpolarizabilities for one specific diagonal direction or for all Cartesian components, except that second hyperpolarizabilities are limited to $iiii$ and $iijj$ components. When computing all the components, you can obtain the norm of the dipole moment and polarizability, the Hyper-Rayleigh scattering first hyperpolarizability, the mean of the second hyperpolarizability, and other information

10.12.3.2 Step-by-step Example of a Finite-Field Calculation

Put the sample `input` file (given above) for a DFT calculation in the new directory; the name of the input file should be `input`. Run the `input-Q-Chem-t-rex.sh` script. The following questions are asked:

```
Components: x=1 y=2 z=3 all_beta=4 all_alpha=5
```

```
4
File name
eth-
Number of field amplitudes
5
Smallest field amplitude F_0
0.0004
a    (F_k=a^k F_0)
2.0
Number of files created          101
```

In this example, the answers correspond to FF calculations for $F(k) = 2.0^k \times 0.0004$ and for a geometric progression ($k = 0, 1, 2, 3, 4, 5$) of external field amplitudes. The calculation is set up for all the components, with the input files `eth-001.inp`, `eth-002.inp`, `eth-003.inp`, ..., generated, where the exact number of inputs generated depends on the chosen geometric progression.

After the individual FF calculations (*i.e.*, after executing Q-CHEM jobs for all generated inputs), parse the outputs using the `parse-t-rex.sh` script. The energies are written in a file called `prelogfile`.

Run the `T-REX` program and answer the questions:

```
Components: x=1 y=2 z=3 all_beta=4
4
Number of methods
1
Number of field amplitudes k_max+1
5
Smallest field amplitude   F_0
0.0004
Step-size a
2.0
```

The energies are ordered in a file called `logfile` and the results are printed in the `results` file.

10.13 General Response Theory

10.13.1 Introduction

Many of the preceding sections of chapter 10 are concerned with properties that require the solution of underlying equations similar to those from TDDFT (see eq. (7.15)), but in the presence of a (time-dependent) perturbation:

$$\left[\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^* & \mathbf{A}^* \end{pmatrix} - \omega_f \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Delta} \\ -\boldsymbol{\Delta}^* & -\boldsymbol{\Sigma}^* \end{pmatrix}\right] \begin{pmatrix} \mathbf{X} \\ \mathbf{Y} \end{pmatrix} = \begin{pmatrix} \mathbf{V} \\ -\mathbf{V}^* \end{pmatrix}, \tag{10.133}$$

where $\boldsymbol{\Sigma} \to \mathbf{0}$ and $\boldsymbol{\Delta} \to \mathbf{1}$ for canonical HF/DFT MOs. The functionality for solving these equations with a general choice of operators representing a perturbation $\mathbf{V}$ is now available in Q-CHEM. Both singlet[102] and triplet[159] response are available for a variety of operators (see table 10.4).

An additional feature of the general response module is its ability to work with non-orthogonal MOs. In a formulation analogous to TDDFT(MI)[126], the linear response for molecular interactions[23], or LR(MI), method is available to solve the linear response equations on top of ALMOs.

The response solver can be used with any density functional available in Q-CHEM, including range-separated functionals (*e.g.* CAM-B3LYP, ωB97X) and meta-GGAs (*e.g.* M06-2X).

There are a few limitations:

- No post-HF/correlated methods are available yet.
- Currently, only linear response is implemented.
- Only calculations on top of restricted and unrestricted (not restricted open-shell) references are implemented.
- Density functionals including non-local dispersion (*e.g.* VV10, ωB97M-V) are not yet available.

10.13.2 Job Control

Only one keyword is necessary in the *$rem* section to activate the response module. All other options are controlled through the *$response* input section.

RESPONSE

Activate the general response property module.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Don't activate the general response property module.

TRUE (or 1) Activate the general response property module.

RECOMMENDATION:

None.

ORDER

Sets the maximum order of response theory to perform.

INPUT SECTION: *$response*

TYPE:

STRING

DEFAULT:

LINEAR

OPTIONS:

LINEAR Perform up through linear response.

RECOMMENDATION:

None. Currently, only linear response is implemented.

SOLVER

Sets the algorithm for solving the response equations.

INPUT SECTION: *$response*

TYPE:

STRING

DEFAULT:

DIIS

OPTIONS:

LINEAR Iteratively solve the response equations without convergence acceleration.

DIIS Iteratively solve the response equations using DIIS for convergence acceleration.

RECOMMENDATION:

DIIS

HAMILTONIAN

Sets the approximation used for the orbital Hessian.

INPUT SECTION: *$response*

TYPE:

STRING

DEFAULT:

RPA

OPTIONS:

RPA No approximations.

TDA Same as the CIS approximation.

CIS Synonym for TDA.

RECOMMENDATION:

None.

SPIN

Does the operator access same spin (singlet) or different spin (triplet) states?

INPUT SECTION: *$response*

TYPE:

STRING

DEFAULT:

SINGLET

OPTIONS:

SINGLET Operator is spin-conserving.

TRIPLET Operator is not spin-conserving.

RECOMMENDATION:

None. Care must be taken as *all* operators in a single calculation will be forced to follow this option.

MAXITER

Maximum number of iterations.

INPUT SECTION: *$response*

TYPE:

INTEGER

DEFAULT:

60

OPTIONS:

n Maximum number of iterations.

RECOMMENDATION:

Use the default value.

CONV

Convergence threshold. For the DIIS solver, this is the DIIS error norm. For the linear solver, this is the response vector RMSD between iterations.

INPUT SECTION: *$response*

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n Sets the convergence threshold to 10^{-n}.

RECOMMENDATION:

Use the default value.

DIIS_START

Iteration number to start DIIS. Before this, linear iterations are performed.

INPUT SECTION: *$response*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Iteration number to start DIIS.

RECOMMENDATION:

Use the default value.

DIIS_VECTORS

Maximum number of DIIS vectors to keep.

INPUT SECTION: *$response*

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

$n > 0$ Maximum number of DIIS vectors to keep.

RECOMMENDATION:

Use the default value.

RHF_AS_UHF

Should the response equations be solved as though an unrestricted reference is being used?

INPUT SECTION: *$response*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Treat an RHF wavefunction as though it were UHF.

FALSE Treat an RHF wavefunction as RHF.

RECOMMENDATION:

Use the default value. Only useful for debugging.

PRINT_LEVEL

Sets a general printing level across the response module.

INPUT SECTION: *$response*

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Print the initial guess and the final results.

2 1 + iterations and comments.

10 Kill trees.

RECOMMENDATION:

Use the default value.

RUN_TYPE

Should a single response calculation be performed, or should all permutations of the orbital Hessian and excitation type be performed?

INPUT SECTION: *$response*

TYPE:

STRING

DEFAULT:

SINGLE

OPTIONS:

SINGLE Use only the orbital Hessian and excitation type specified in their respective keywords.

ALL Use all permutations of RPA/TDA and singlet/triplet.

RECOMMENDATION:

Use the default value, unless a comparison between approximations and excitation types is desired.

SAVE

Save any quantities to disk?

INPUT SECTION: *$response*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Don't save any quantities to disk.

1 Save quantities in MO basis.

2 Save quantities in MO and AO bases.

RECOMMENDATION:

None.

READ

Read any quantities from disk?

INPUT SECTION: *$response*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Don't read any quantities from disk.

1 Read quantities in MO basis.

2 Read quantities in AO basis.

RECOMMENDATION:

None.

DUMP_AO_INTEGRALS

Should AO-basis property integrals be saved to disk?

INPUT SECTION: *$response*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Save AO-basis property integrals to disk.

FALSE Don't save AO-basis property integrals to disk.

RECOMMENDATION:

None.

FORCE_NOT_NONORTHOGONAL

Should the canonical response equations be solved, ignoring the identity of the underlying orbitals?

INPUT SECTION: *$response*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Leave as false. Using the standard (canonical) response equations with non-orthogonal MOs will give incorrect results.

FORCE_NONORTHOGONAL

Should the non-orthogonal response equations be solved, ignoring the identity of the underlying orbitals?

INPUT SECTION: *$response*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Leave as false. When used with canonical MOs, this should give the same answer as with the standard equations, but at greater computational cost.

FREQUENCY

Strength of one or more incident fields in atomic units. A separate response calculation will be performed for every field strength. 0.0 corresponds to the static limit.

INPUT SECTION: *$response*

TYPE:

DOUBLE

DEFAULT:

0.0

OPTIONS:

$l\,m\,n\,\ldots$ One or more field strengths separated by spaces.

RECOMMENDATION:

None.

10.13.3 *$response* Section and Operator Specification

The specification of operators used in solving for response vectors is designed to be very flexible. The general form of the *$response* input section is given by

```
$response
  keyword_1 setting_1
  keyword_2 setting_2
  ...
  [operator_1_label, operator_1_origin]
  [operator_2_label, operator_2_origin]
  [operator_3_label, operator_3_origin]
  ...
$end
```

where the keywords are those found in section 10.13.2 (with the exception of RESPONSE).

The specification of an operator is given within a line contained by `[]`, where the first element is a label from table 10.4, and the second element is a label from table 10.5. Operator specifications may appear in any order. Response values are calculated for all possible permutations of operators and their components.

For the Cartesian moment operator, a third field within `[]` may be specified for the order of the expansion, entered as (i, j, k). For example, the molecular response to the moment of order (2, 5, 4) with its origin at (0.2, 0.3, 0.4) a.u. can be found with the operator specification

```
[multipole, (0.2, 0.3, 0.4), (2, 5, 4)]
```

Table 10.4: Available operators

Operator Label	**Description**	**Integral**
`dipole` or `diplen`	dipole (length gauge)	$\langle\chi_\mu\|\mathbf{r}_O\|\chi_\nu\rangle$
`quadrupole`	second moment (length gauge)	$\langle\chi_\mu\|\mathbf{r}\mathbf{r}^T\|\chi_\nu\rangle$
`multipole`	arbitrary-order Cartesian moment (length gauge)	$\langle\chi_\mu\|x^i y^j z^k\|\chi_\nu\rangle$
`fermi` or `fc`	Fermi contact	$\frac{4\pi g_e}{3}\langle\chi_\mu\|\delta(\mathbf{r}_K)\|\chi_\nu\rangle$
`spindip` or `sd`	spin dipole	$\frac{g_e}{2}\langle\chi_\mu\|\frac{3\mathbf{r}_K\mathbf{r}_K^T - r_K^2}{r_K^5}\|\chi_\nu\rangle$
`angmom` or `dipmag`	angular momentum	$\langle\chi_\mu\|\mathbf{L}_O\|\chi_\nu\rangle$
`dipvel`	dipole (velocity gauge)	$\langle\chi_\mu\|\nabla\|\chi_\nu\rangle$

Table 10.5: Available operator origins

Origin Label	**Description**
`zero`	Cartesian origin, same as (0.0, 0.0, 0.0)
`(x, y, z)`	arbitrary point (double precision, units are bohrs)

10.13.4 Examples Including *$response* Section

Example 10.58 Input for calculating all components of the static (dipole) polarizability at the Cartesian origin for tryptophan. All of the options given are defaults.

```
$molecule
0 1
N     -0.0699826875    0.3321987191    0.2821283177
C      1.3728035449    0.0970713322   -0.0129587739
C      2.0969275417   -0.0523593054    1.3682652221
O      3.1382490088   -0.6563684788    1.5380162924
C      1.9529664597    1.3136139853   -0.7956021969
H      1.8442727348    2.2050605044   -0.1801631789
H      1.3455899915    1.4594935008   -1.6885689523
C      3.4053646872    1.1270611844   -1.1918075237
C      4.4845249667    1.6235038050   -0.5598918002
N      5.6509089647    1.2379326369   -1.2284610654
H      6.6009314349    1.4112351003   -0.9028629397
C      5.2921619642    0.4356274269   -2.3131617003
C      3.8942019475    0.3557998019   -2.3263315791
C      3.2659168792   -0.3832607567   -3.3431309548
H      2.1864306677   -0.4577058843   -3.3815918670
C      4.0381762333   -1.0087512639   -4.2870993776
H      3.5696890585   -1.5824763141   -5.0755609734
C      5.4445159165   -0.9194874753   -4.2519002882
H      6.0229926396   -1.4277973542   -5.0130007062
C      6.0869576238   -0.2024044961   -3.2767702726
H      7.1656650647   -0.1287762497   -3.2458650647
H      4.5457621618    2.2425310766    0.3253979653
H     -0.5159777859    0.7478905868   -0.5487661007
H      1.5420526570   -0.8143939718   -0.5935463196
H     -0.5302278747   -0.5823989653    0.4084507634
O      1.4575846656    0.5996887308    2.4093500287
H      0.5990015339    0.8842421241    2.0047830456
$end

$rem
   METHOD           = hf
   BASIS            = sto-3g
   SCF_CONVERGENCE  = 9
   THRESH           = 12
   RESPONSE         = true
$end

$response
   ORDER             linear
   SOLVER            diis
   HAMILTONIAN       rpa
   SPIN              singlet
   MAXITER           60
   CONV              8
   DIIS_START        1
   DIIS_VECTORS      7
   RHF_AS_UHF        false
   PRINT_LEVEL       2
   RUN_TYPE          single
   FREQUENCY         0.0
   [dipole, zero]
$end
```

Example 10.59 Functionally identical input for calculating all components of the static (dipole) polarizability at the Cartesian origin for tryptophan.

```
$molecule
   0 1
   N     -0.0699826875    0.3321987191    0.2821283177
   C      1.3728035449    0.0970713322   -0.0129587739
   C      2.0969275417   -0.0523593054    1.3682652221
   O      3.1382490088   -0.6563684788    1.5380162924
   C      1.9529664597    1.3136139853   -0.7956021969
   H      1.8442727348    2.2050605044   -0.1801631789
   H      1.3455899915    1.4594935008   -1.6885689523
   C      3.4053646872    1.1270611844   -1.1918075237
   C      4.4845249667    1.6235038050   -0.5598918002
   N      5.6509089647    1.2379326369   -1.2284610654
   H      6.6009314349    1.4112351003   -0.9028629397
   C      5.2921619642    0.4356274269   -2.3131617003
   C      3.8942019475    0.3557998019   -2.3263315791
   C      3.2659168792   -0.3832607567   -3.3431309548
   H      2.1864306677   -0.4577058843   -3.3815918670
   C      4.0381762333   -1.0087512639   -4.2870993776
   H      3.5696890585   -1.5824763141   -5.0755609734
   C      5.4445159165   -0.9194874753   -4.2519002882
   H      6.0229926396   -1.4277973542   -5.0130007062
   C      6.0869576238   -0.2024044961   -3.2767702726
   H      7.1656650647   -0.1287762497   -3.2458650647
   H      4.5457621618    2.2425310766    0.3253979653
   H     -0.5159777859    0.7478905868   -0.5487661007
   H      1.5420526570   -0.8143939718   -0.5935463196
   H     -0.5302278747   -0.5823989653    0.4084507634
   O      1.4575846656    0.5996887308    2.4093500287
   H      0.5990015339    0.8842421241    2.0047830456
$end

$rem
   JOBTYPE         =  polarizability
   METHOD          =  hf
   BASIS           =  sto-3g
   SCF_CONVERGENCE =  9
   THRESH          =  12
$end
```

10.14 Electronic Couplings for Electron- and Energy Transfer

10.14.1 Eigenstate-Based Methods

For electron transfer (ET) and excitation energy transfer (EET) processes, the electronic coupling is one of the important parameters that determine their reaction rates. For ET, Q-CHEM provides the coupling values calculated with the generalized Mulliken-Hush (GMH),[38] fragment charge difference (FCD),[216] Boys localization,[203] and Edmiston-Ruedenbeg[204] localization schemes. For EET, options include fragment excitation difference (FED),[93] fragment spin difference (FSD),[233] occupied-virtual separated Boys localization,[205] or Edmiston-Ruedenberg localization.[204] In all these schemes, a vertical excitation approach such as CIS or TDDFT is required, and the GMH, FCD, FED, FSD, Boys or ER coupling values are calculated based on the excited state results. More recently, the FED and FCD schemes have been extended to work with RAS-CI wavefunctions,[124,131] which are multi-configurational in nature.

10.14.1.1 Two-state approximation

Under the two-state approximation, the diabatic reactant and product states are assumed to be a linear combination of the eigenstates. For ET, the choice of such linear combination is determined by a zero transition dipoles (GMH) or maximum charge differences (FCD). In the latter, a 2×2 donor–acceptor charge difference matrix $\Delta\mathbf{q}$ is defined, with elements

$$\Delta q_{mn} = q_{mn}^{\mathrm{D}} - q_{mn}^{\mathrm{A}} = \int_{\mathbf{r}\in\mathrm{D}} \rho_{mn}(\mathbf{r})d\mathbf{r} - \int_{\mathbf{r}\in\mathrm{A}} \rho_{mn}(\mathbf{r})d\mathbf{r} \tag{10.134}$$

where $\rho_{mn}(\mathbf{r})$ is the matrix element of the density operator between states $|m\rangle$ and $|n\rangle$.

For EET, a maximum excitation difference is assumed in the FED, in which an excitation difference matrix is similarly defined with elements

$$\Delta x_{mn} = x_{mn}^{\mathrm{D}} - x_{mn}^{\mathrm{A}} = \int_{\mathbf{r}\in\mathrm{D}} \rho_{\mathrm{ex}}^{(mn)}(\mathbf{r})d\mathbf{r} - \int_{\mathbf{r}\in\mathrm{A}} \rho_{\mathrm{ex}}^{(mn)}(\mathbf{r})d\mathbf{r} \tag{10.135}$$

where $\rho_{\mathrm{ex}}^{(mn)}(\mathbf{r})$ is the sum of attachment and detachment densities for transition $|m\rangle \rightarrow |n\rangle$, as they correspond to the electron and hole densities in an excitation. In the FSD, a maximum spin difference is used and the corresponding spin difference matrix is defined with its elements as,

$$\Delta s_{mn} = s_{mn}^{\mathrm{D}} - s_{mn}^{\mathrm{A}} = \int_{\mathbf{r}\in\mathrm{D}} \sigma_{(mn)}(\mathbf{r})d\mathbf{r} - \int_{\mathbf{r}\in\mathrm{A}} \sigma_{(mn)}(\mathbf{r})d\mathbf{r} \tag{10.136}$$

where $\sigma_{mn}(\mathbf{r})$ is the spin density, difference between α-spin and β-spin densities, for transition from $|m\rangle \rightarrow |n\rangle$.

Since Q-CHEM uses a Mulliken population analysis for the integrations in Eqs. (10.134), (10.135), and (10.136), the matrices $\Delta\mathbf{q}$, $\Delta\mathbf{x}$ and $\Delta\mathbf{s}$ are not symmetric. To obtain a pair of orthogonal states as the diabatic reactant and product states, $\Delta\mathbf{q}$, $\Delta\mathbf{x}$ and $\Delta\mathbf{s}$ are symmetrized in Q-CHEM. Specifically,

$$\overline{\Delta q}_{mn} = (\Delta q_{mn} + \Delta q_{nm})/2 \tag{10.137a}$$
$$\overline{\Delta x}_{mn} = (\Delta x_{mn} + \Delta x_{nm})/2 \tag{10.137b}$$
$$\overline{\Delta s}_{mn} = (\Delta s_{mn} + \Delta s_{nm})/2 \tag{10.137c}$$

The final coupling values are obtained as listed below:

- For GMH,
$$V_{\mathrm{ET}} = \frac{(E_n - E_m)\|\vec{\mu}_{mn}\|}{\sqrt{(\vec{\mu}_m - \vec{\mu}_n)^2 + 4\|\vec{\mu}_{mn}\|^2}} \tag{10.138}$$
- For FCD,
$$V_{\mathrm{ET}} = \frac{(E_n - E_m)\overline{\Delta q}_{mn}}{\sqrt{(\Delta q_m - \Delta q_n)^2 + 4(\overline{\Delta q}_{mn}^2}} \tag{10.139}$$
- For FED,
$$V_{\mathrm{EET}} = \frac{(E_n - E_m)\overline{\Delta x}_{mn}}{\sqrt{(\Delta x_m - \Delta x_n)^2 + 4(\overline{\Delta x}_{mn})^2}} \tag{10.140}$$
- For FSD,
$$V_{\mathrm{EET}} = \frac{(E_n - E_m)\overline{\Delta s}_{mn}}{\sqrt{(\Delta s_m - \Delta s_n)^2 + 4(\overline{\Delta s}_{mn})^2}} \tag{10.141}$$

Q-CHEM provides the option to control FED, FSD, FCD and GMH calculations after a single-excitation calculation, such as CIS and TDDFT. To obtain ET coupling values, set STS_GMH = TRUE (for GMH) or STS_FCD = TRUE (for FCD). Similarly, set STS_FED = TRUE for FED and STS_FSD = TRUE for FSD. In FCD, FED and FSD calculations, the donor and acceptor fragments are defined via the *$rem* variables STS_DONOR and STS_ACCEPTOR. It is necessary

to arrange the atomic order in the *$molecule* section such that the atoms in the donor (acceptor) fragment is in one consecutive block. The ordering numbers of beginning and ending atoms for the donor and acceptor blocks are included in *$rem* variables STS_DONOR and STS_ACCEPTOR.

The couplings will be calculated between all choices of excited states with the same spin. In FSD, FCD and GMH calculations, the coupling value between the excited and reference (ground) states will be included, but in FED, the ground state is not included in the analysis. It is important to select excited states properly, according to the distribution of charge or excitation, among other characteristics, such that the coupling obtained can properly describe the electronic coupling of the corresponding process in the two-state approximation.

10.14.1.2 FED and FCD with RAS-CI

Within the ambit of the single excitation theory such as the CIS or TDDFT, one can easily obtain analytical expressions for the matrix elements of the excitation density and can therefore, use Eq. 10.140 to compute electronic couplings between adiabatic states. However, for multi-excitation wavefunctions such as those obtained from RAS-CI no simple expressions exist for the off-diagonal elements in the excitation difference (Δx_{mn} in Eq. 10.140). To circumvent this challenge, a new scheme was developed known as θ-FED.[117,124,131] In this approach, the diabatic states are assumed to be functions of a mixing angle θ. Consequently, the excitation difference density ($\mathbf{\Delta x}$ in Eqs 10.135 and 10.140) is dependent on θ. In order to obtain 'ideal' diabatic states, a scan of θ is performed from $-\pi/4$ to $\pi/4$ to maximize the difference of the excitation, *i.e.*,

$$\theta_{\text{max}} = \underset{-\pi/4<\theta<\pi/4}{\arg\max} \, |\mathbf{\Delta x}_{\text{i}}(\theta) - \mathbf{\Delta x}_{\text{f}}(\theta)|, \tag{10.142}$$

with 'i' and 'f' indicating the initial and final diabatic states, respectively. The corresponding θ-dependent coupling can then be written as

$$V_{\theta\text{-FED}} = \tfrac{1}{2}(E_n - E_m)\sin(2\theta_{\text{max}}) \, , \tag{10.143}$$

Fortunately, one can still use Eq. 10.139 to compute ET couplings between two adiabatic states for FCD with RAS-CI. This is because the charge difference matrix ($\mathbf{\Delta q}$ in Eqs 10.134 and 10.139) depends on the one-particle (for $\Delta q_{m/n}$) and transition density matrices (for $\Delta q_{mn/nm}$), which are also easily obtainable with the RAS-CI wavefunctions.

The *$rem* variables STS_FED, STS_FCD, STS_DONOR, and STS_ACCEPTOR also apply to FCD and FED calculations with RAS-CI.

STS_GMH

Control the calculation of GMH for ET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform a GMH calculation.

TRUE Include a GMH calculation.

RECOMMENDATION:

When set to true computes Mulliken-Hush electronic couplings. It yields the generalized Mulliken-Hush couplings as well as the transition dipole moments for each pair of excited states and for each excited state with the ground state.

STS_FCD

Control the calculation of FCD for ET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform an FCD calculation.

TRUE Include an FCD calculation.

RECOMMENDATION:

None

STS_FED

Control the calculation of FED for EET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform a FED calculation.

TRUE Include a FED calculation.

RECOMMENDATION:

None

STS_FSD

Control the calculation of FSD for EET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform a FSD calculation.

TRUE Include a FSD calculation.

RECOMMENDATION:

For RCIS triplets, FSD and FED are equivalent. FSD will be automatically switched off and perform a FED calculation.

STS_DONOR

Define the donor fragment.

TYPE:

STRING

DEFAULT:

0 No donor fragment is defined.

OPTIONS:

i-j Donor fragment is in the ith atom to the jth atom.

RECOMMENDATION:

Note no space between the hyphen and the numbers i and j.

STS_ACCEPTOR

Define the acceptor molecular fragment.

TYPE:

STRING

DEFAULT:

0 No acceptor fragment is defined.

OPTIONS:

i-j Acceptor fragment is in the ith atom to the jth atom.

RECOMMENDATION:

Note no space between the hyphen and the numbers i and j.

STS_MOM

Control calculation of the transition moments between excited states in CIS and TDDFT calculations (including spin-flip variants).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate state-to-state transition moments.

TRUE Do calculate state-to-state transition moments.

RECOMMENDATION:

When set to true requests the state-to-state dipole transition moments for all pairs of excited states and for each excited state with the ground state. This is not available for restricted open-shell wavefunctions.

Example 10.60 GMH and FCD calculation to analyze electron transfer couplings in a $(C_2H_2)\cdots(CH_4N^+)$ cation dimer.

```
$molecule
   1  1
   C     0.679952    0.000000    0.000000
   N    -0.600337    0.000000    0.000000
   H     1.210416    0.940723    0.000000
   H     1.210416   -0.940723    0.000000
   H    -1.131897   -0.866630    0.000000
   H    -1.131897    0.866630    0.000000
   C    -5.600337    0.000000    0.000000
   C    -6.937337    0.000000    0.000000
   H    -5.034682    0.927055    0.000000
   H    -5.034682   -0.927055    0.000000
   H    -7.502992   -0.927055    0.000000
   H    -7.502992    0.927055    0.000000
$end

$rem
   METHOD        CIS
   BASIS         6-31+G
   CIS_N_ROOTS   20
   CIS_SINGLETS  true
   CIS_TRIPLETS  false
   STS_GMH       true !turns on the GMH calculation
   STS_FCD       true !turns on the FCD calculation
   STS_DONOR     1-6  !define the donor fragment as atoms 1-6 for FCD calc.
   STS_ACCEPTOR  7-12 !define the acceptor fragment as atoms 7-12 for FCD calc.
   MEM_STATIC    200  !increase static memory for a CIS job with larger basis set
$end
```

Example 10.61 An FED calculation to analyze excitation energy transfer couplings in a pair of stacked ethylenes.

```
$molecule
   0  1
   C     0.670518    0.000000    0.000000
   H     1.241372    0.927754    0.000000
   H     1.241372   -0.927754    0.000000
   C    -0.670518    0.000000    0.000000
   H    -1.241372   -0.927754    0.000000
   H    -1.241372    0.927754    0.000000
   C     0.774635    0.000000    4.500000
   H     1.323105    0.936763    4.500000
   H     1.323105   -0.936763    4.500000
   C    -0.774635    0.000000    4.500000
   H    -1.323105   -0.936763    4.500000
   H    -1.323105    0.936763    4.500000
$end

$rem
   METHOD            CIS
   BASIS             3-21G
   CIS_N_ROOTS       20
   CIS_SINGLETS      true
   CIS_TRIPLETS      false
   STS_FED           true
   STS_DONOR         1-6
   STS_ACCEPTOR      7-12
$end
```

Example 10.62 A RAS-FCD calculation to analyze electron transfer couplings in an ethylene dimer.

```
$comment
RASCI for Hole Transfer
 Stacked-Ethylene / DZ*
$end

$molecule
  1  2
   C     0.670518    0.000000    0.000000
   H     1.241372    0.927754    0.000000
   H     1.241372   -0.927754    0.000000
   C    -0.670518    0.000000    0.000000
   H    -1.241372   -0.927754    0.000000
   H    -1.241372    0.927754    0.000000
   C     0.774635    0.000000    4.000000
   H     1.323105    0.936763    4.000000
   H     1.323105   -0.936763    4.000000
   C    -0.774635    0.000000    4.000000
   H    -1.323105   -0.936763    4.000000
   H    -1.323105    0.936763    4.000000
$end

$rem
   JOBTYPE                   SP
   BASIS                     DZ*
   CORRELATION               RASCI
   UNRESTRICTED              FALSE
   RAS_ROOTS                 5
   RAS_ACT                   4
   RAS_ELEC_ALPHA            2
   RAS_ELEC_BETA             1
   RAS_OCC                   14
   STS_FCD                   TRUE
   STS_ACCEPTOR              1-6
   STS_DONOR                 7-12
   RAS_SPIN_MULT             1
$end
```

Example 10.63 A RAS-FED calculation to analyze excitation energy transfer couplings in an ethylene dimer.

```
$comment
 RASCI for Excitation Energy Transfer
 Stacked-Ethylene / DZ*
$end

$molecule
  0  1
   C     0.670518    0.000000    0.000000
   H     1.241372    0.927754    0.000000
   H     1.241372   -0.927754    0.000000
   C    -0.670518    0.000000    0.000000
   H    -1.241372   -0.927754    0.000000
   H    -1.241372    0.927754    0.000000
   C     0.774635    0.000000    4.000000
   H     1.323105    0.936763    4.000000
   H     1.323105   -0.936763    4.000000
   C    -0.774635    0.000000    4.000000
   H    -1.323105   -0.936763    4.000000
   H    -1.323105    0.936763    4.000000
$end

$rem
   JOBTYPE                   SP
   BASIS                     DZ*
   CORRELATION               RASCI
   UNRESTRICTED              FALSE
   RAS_ROOTS                 5
   RAS_ACT                   4
   RAS_ELEC_ALPHA            2
   RAS_ELEC_BETA             2
   RAS_OCC                   14
   STS_FED                   TRUE
   STS_ACCEPTOR              1-6
   STS_DONOR                 7-12
   RAS_SPIN_MULT             1
$end
```

10.14.1.3 Multi-state treatments

When dealing with multiple charge or electronic excitation centers, diabatic states can be constructed with Boys[203] or Edmiston-Ruedenberg[204] localization. In this case, we construct diabatic states $\{|\Xi_I\rangle\}$ as linear combinations of adiabatic states $\{|\Phi_I\rangle\}$ with a general rotation matrix $\mathbf{U}$ that is $N_{\text{state}} \times N_{\text{state}}$ in size:

$$|\Xi_I\rangle = \sum_{J=1}^{N_{\text{states}}} |\Phi_J\rangle\, U_{ji} \qquad\qquad I = 1 \ldots N_{\text{states}} \tag{10.144}$$

The adiabatic states can be produced with any method, in principle, but the Boys/ER-localized diabatization methods have been implemented thus far only for CIS, TDDFT or RASCI (section 7.12.6) methods in Q-CHEM. In analogy to orbital localization, Boys-localized diabatization corresponds to maximizing the charge separation between diabatic state centers:

$$f_{\text{Boys}}(\mathbf{U}) = f_{\text{Boys}}(\{\Xi_I\}) = \sum_{I,J=1}^{N_{\text{states}}} \left|\langle\Xi_I|\vec{\mu}|\Xi_I\rangle - \langle\Xi_J|\vec{\mu}|\Xi_J\rangle\right|^2 \tag{10.145}$$

Here, $\vec{\mu}$ represents the dipole operator. ER-localized diabatization prescribes maximizing self-interaction energy:

$$f_{ER}(\mathbf{U}) = f_{\mathrm{ER}}(\{\Xi_I\}) = \sum_{I=1}^{N_{\mathrm{states}}} \int d\vec{\mathcal{R}}_1 \int d\vec{\mathcal{R}}_2 \frac{\langle \Xi_I|\hat{\rho}(\vec{\mathcal{R}}_2)|\Xi_I\rangle\langle \Xi_I|\hat{\rho}(\vec{\mathcal{R}}_1)|\Xi_I\rangle}{|\vec{\mathcal{R}}_1 - \vec{\mathcal{R}}_2|} \tag{10.146}$$

where the density operator at position $\vec{\mathcal{R}}$ is

$$\hat{\rho}(\vec{\mathcal{R}}) = \sum_j \delta(\vec{\mathcal{R}} - \vec{r}^{(j)}) \tag{10.147}$$

Here, $\vec{r}^{(j)}$ represents the position of the jth electron.

These models reflect different assumptions about the interaction of our quantum system with some fictitious external electric field/potential: (i) if we assume a fictitious field that is linear in space, we arrive at Boys localization; (ii) if we assume a fictitious potential energy that responds linearly to the charge density of our system, we arrive at ER localization. Note that in the two-state limit, Boys localized diabatization reduces nearly exactly to GMH.[203]

As written down in Eq. (10.145), Boys localized diabatization applies only to charge transfer, not to energy transfer. Within the context of CIS or TDDFT calculations, one can easily extend Boys localized diabatization[205] by separately localizing the occupied and virtual components of $\vec{\mu}$, $\vec{\mu}^{\mathrm{occ}}$, and $\vec{\mu}^{\mathrm{virt}}$:

$$\begin{aligned} f_{\mathrm{BoysOV}}(\mathbf{U}) &= f_{\mathrm{BoysOV}}(\{\Xi_I\}) \\ &= \sum_{I,J=1}^{N_{\mathrm{states}}} \left(\left|\langle \Xi_I|\vec{\mu}^{\mathrm{occ}}|\Xi_I\rangle - \langle \Xi_J|\vec{\mu}^{\mathrm{occ}}|\Xi_J\rangle\right|^2 + \left|\langle \Xi_I|\vec{\mu}^{\mathrm{virt}}|\Xi_I\rangle - \langle \Xi_J|\vec{\mu}^{\mathrm{virt}}|\Xi_J\rangle\right|^2 \right) \end{aligned} \tag{10.148}$$

where

$$|\Xi_I\rangle = \sum_{ia} t_i^{Ia}|\Phi_i^a\rangle \tag{10.149}$$

and the occupied/virtual components are defined by

$$\langle \Xi_I|\vec{\mu}|\Xi_J\rangle = \underbrace{\delta_{IJ}\sum_i \vec{\mu}_{ii} - \sum_{aij} t_i^{Ia}t_j^{Ja}\vec{\mu}_{ij}}_{\langle \Xi_I|\vec{\mu}^{\mathrm{occ}}|\Xi_J\rangle} + \underbrace{\sum_{iba} t_i^{Ia}t_i^{Jb}\vec{\mu}_{ab}}_{\langle \Xi_I|\vec{\mu}^{\mathrm{virt}}|\Xi_J\rangle} \ . \tag{10.150}$$

Note that when we maximize the Boys OV function, we are simply performing Boys-localized diabatization separately on the electron attachment and detachment densities.

Finally, for energy transfer, it can be helpful to understand the origin of the diabatic couplings. To that end, we now provide the ability to decompose the diabatic coupling between diabatic states into Coulomb (J), Exchange (K) and one-electron (O) components:[217]

$$\langle \Xi_P|H|\Xi_Q\rangle = \underbrace{\sum_{iab} t_i^{Pa}t_i^{Qb}F_{ab} - \sum_{ija} t_i^{Pa}t_j^{Qa}F_{ij}}_{O} + \underbrace{\sum_{ijab} t_i^{Pa}t_j^{Qb}(ia|jb)}_{J} - \underbrace{\sum_{ijab} t_i^{Pa}t_j^{Qb}(ij|ab)}_{K} \ . \tag{10.151}$$

BOYS_CIS_NUMSTATE

Define how many states to mix with Boys localized diabatization. These states must be specified in the *$localized_diabatization* section.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Boys localized diabatization.

OPTIONS:

2 to N where N is the number of CIS states requested (CIS_N_ROOTS)

RECOMMENDATION:

It is usually not wise to mix adiabatic states that are separated by more than a few eV or a typical reorganization energy in solvent.

ER_CIS_NUMSTATE

Define how many states to mix with ER localized diabatization. These states must be specified in the *$localized_diabatization* section.

TYPE:

INTEGER

DEFAULT:

0 Do not perform ER localized diabatization.

OPTIONS:

2 to N where N is the number of CIS states requested (CIS_N_ROOTS)

RECOMMENDATION:

It is usually not wise to mix adiabatic states that are separated by more than a few eV or a typical reorganization energy in solvent.

LOC_CIS_OV_SEPARATE

Decide whether or not to localized the "occupied" and "virtual" components of the localized diabatization function, *i.e.*, whether to localize the electron attachments and detachments separately.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not separately localize electron attachments and detachments.

OPTIONS:

TRUE

RECOMMENDATION:

If one wants to use Boys localized diabatization for energy transfer (as opposed to electron transfer) , this is a necessary option. ER is more rigorous technique, and does not require this OV feature, but will be somewhat slower.

CIS_DIABATH_DECOMPOSE

Decide whether or not to decompose the diabatic coupling into Coulomb, exchange, and one-electron terms.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not decompose the diabatic coupling.

OPTIONS:

TRUE

RECOMMENDATION:

These decompositions are most meaningful for electronic excitation transfer processes. Currently, available only for CIS, not for TDDFT diabatic states.

Example 10.64 A calculation using ER localized diabatization to construct the diabatic Hamiltonian and couplings between a square of singly-excited Helium atoms.

```
$molecule
   0 1
   he    0  -1.0    1.0
   he    0  -1.0   -1.0
   he    0   1.0   -1.0
   he    0   1.0    1.0
$end

$rem
   METHOD                 cis
   CIS_N_ROOTS            4
   CIS_SINGLETS           false
   CIS_TRIPLETS           true
   BASIS                  6-31g**
   SCF_CONVERGENCE        8
   RPA                    false
   LOC_CIS_OV_SEPARATE    false  ! NOT localizing attachments/detachments separately.
   ER_CIS_NUMSTATE        4      ! using ER to mix 4 adiabatic states.
   CIS_DIABATh_DECOMPOSE  true   ! decompose diabatic couplings into
                                 ! Coulomb, exchange, and one-electron components.
   INTEGRAL_SYMMETRY      false
   POINT_GROUP_SYMMETRY   false
$end

$localized_diabatization
   On the next line, list which excited adiabatic states we want to mix.
   1 2 3 4
$end
```

10.14.2 Diabatic-State-Based Methods

10.14.2.1 Electronic coupling in charge transfer

A charge transfer involves a change in the electron numbers in a pair of molecular fragments. As an example, we will use the following reaction when necessary, and a generalization to other cases is straightforward:

$$\mathrm{D^{-}A} \longrightarrow \mathrm{DA^{-}} \tag{10.152}$$

where an extra electron is localized to the donor (D) initially, and it becomes localized to the acceptor (A) in the final state. The two-state secular equation for the initial and final electronic states can be written as

$$\mathbf{H} - E\mathbf{S} = \begin{pmatrix} H_{\mathrm{ii}} - S_{\mathrm{ii}}E & H_{\mathrm{if}} - S_{\mathrm{if}}E \\ H_{\mathrm{if}} - S_{\mathrm{if}}E & H_{\mathrm{ff}} - S_{\mathrm{ff}}E \end{pmatrix} = 0 \tag{10.153}$$

This is very close to an eigenvalue problem except for the non-orthogonality between the initial and final states. A standard eigenvalue form for Eq. (10.153) can be obtained by using the Löwdin transformation:

$$\mathbf{H}_{\mathrm{eff}} = \mathbf{S}^{-1/2}\mathbf{H}\mathbf{S}^{-1/2}, \tag{10.154}$$

where the off-diagonal element of the effective Hamiltonian matrix represents the *electronic coupling* for the reaction, and it is defined by

$$V = H_{\mathrm{if}}^{\mathrm{eff}} = \frac{H_{\mathrm{if}} - S_{\mathrm{if}}(H_{\mathrm{ii}} + H_{\mathrm{ff}})/2}{1 - S_{\mathrm{if}}^2} \tag{10.155}$$

In a general case where the initial and final states are not normalized, the electronic coupling is written as

$$V = \sqrt{S_{\mathrm{ii}} S_{\mathrm{ff}}} \times \frac{H_{\mathrm{if}} - S_{\mathrm{if}}(H_{\mathrm{ii}}/S_{\mathrm{ii}} + H_{\mathrm{ff}}/S_{\mathrm{ff}})/2}{S_{\mathrm{ii}} S_{\mathrm{ff}} - S_{\mathrm{if}}^2} \tag{10.156}$$

Thus, in principle, V can be obtained when the matrix elements for the Hamiltonian H and the overlap matrix S are calculated.

The direct coupling (DC) scheme calculates the electronic coupling values via Eq. (10.156), and it is widely used to calculate charge transfer coupling.[32,59,158,235] In the DC scheme, the coupling matrix element is calculated directly using charge-localized determinants (the "diabatic states" in electron transfer literature). In electron transfer systems, it has been shown that such charge-localized states can be approximated by symmetry-broken unrestricted Hartree-Fock (UHF) solutions.[32,148,158] The adiabatic eigenstates are assumed to be the symmetric and anti-symmetric linear combinations of the two symmetry-broken UHF solutions in a DC calculation. Therefore, DC couplings can be viewed as a result of two-configuration solutions that may recover the non-dynamical correlation.

The core of the DC method is based on the corresponding orbital transformation[112] and a calculation for Slater's determinants in H_{if} and S_{if}.[59,235] Unfortunately, the calculation of H_{if} is not available for DFT method because a functional of the two densities ρ_{i} and $\rho \mathrm{f}$ is unknown and there are no existing approximate forms for H_{if}.[230] To calculate charge transfer coupling with DFT, we can use the CDFT-CI method (Section 5.11.5), the frontier molecular orbital (FMO) approach (Section 10.14.2.5),[214,234] or a hybrid scheme, namely, DC with CDFT wave functions (Section 10.14.2.4).

10.14.2.2 Corresponding orbital transformation

Let $|\Psi_a\rangle$ and $|\Psi_b\rangle$ be two single Slater-determinant wave functions for the initial and final states, and $\mathbf{a}$ and $\mathbf{b}$ be the spin-orbital sets, respectively:

$$\mathbf{a} = (a_1, a_2, \cdots, a_N) \tag{10.157}$$

$$\mathbf{b} = (b_1, b_2, \cdots, b_N) \tag{10.158}$$

Since the two sets of spin-orbitals are not orthogonal, the overlap matrix $\mathbf{S}$ can be defined as:

$$\mathbf{S} = \int \mathbf{b}^\dagger \mathbf{a} \, d\tau. \tag{10.159}$$

We note that $\mathbf{S}$ is not Hermitian in general since the molecular orbitals of the initial and final states are separately determined. To calculate the matrix elements H_{ab} and S_{ab}, two sets of new orthogonal spin-orbitals can be used by the corresponding orbital transformation.[112] In this approach, each set of spin-orbitals $\mathbf{a}$ and $\mathbf{b}$ are linearly transformed,

$$\hat{\mathbf{a}} = \mathbf{a}\mathbf{V} \tag{10.160}$$

$$\hat{\mathbf{b}} = \mathbf{b}\mathbf{U} \tag{10.161}$$

where $\mathbf{V}$ and $\mathbf{U}$ are the left-singular and right-singular matrices, respectively, in the singular value decomposition (SVD) of $\mathbf{S}$:

$$\mathbf{S} = \mathbf{U}\hat{\mathbf{s}}\mathbf{V}^\dagger \tag{10.162}$$

The overlap matrix in the new basis is now diagonal

$$\int \hat{\mathbf{b}}^\dagger \hat{\mathbf{a}} = \mathbf{U}^\dagger \left(\int \mathbf{b}^\dagger \mathbf{a} \right) \mathbf{V} = \hat{\mathbf{s}} \tag{10.163}$$

10.14.2.3 Generalized density matrix

The Hamiltonian for electrons in molecules are a sum of one-electron and two-electron operators. In the following, we derive the expressions for the one- and two-electron operators

$$\Omega^{(1)} = \sum_{i=1}^{N} \omega(i) \tag{10.164a}$$

$$\Omega^{(2)} = \frac{1}{2} \sum_{i,j=1}^{N} \omega(i,j) \tag{10.164b}$$

where

$$\omega(i) = -\frac{1}{2}\hat{\nabla}_i^2 + V(i) \tag{10.165a}$$

$$\omega(i,j) = \frac{1}{r_{ij}} \; . \tag{10.165b}$$

Matrix elements of the Hamiltonian can then be expressed as $H_{ab} = \Omega_{ab}^{(1)} + \Omega_{ab}^{(2)}$, where

$$\Omega_{ab}^{(1)} = \langle \Psi_b | \Omega^{(1)} | \Psi_a \rangle = \det(\mathbf{U}) \det(\mathbf{V}^\dagger) \sum_{i=1}^{N} \langle \hat{b}_i | \omega(1) | \hat{a}_i \rangle \prod_{j \neq i}^{N} \hat{s}_{jj} \tag{10.166}$$

and

$$\Omega_{ab}^{(2)} = \langle \Psi_b | \Omega^{(2)} | \Psi_a \rangle = \frac{1}{2} \det(\mathbf{U}) \det(\mathbf{V}^\dagger) \sum_{ij}^{N} \langle \hat{b}_i \hat{b}_j | \omega(1,2)(1 - P_{12}) | \hat{a}_i \hat{a}_j \rangle \prod_{k \neq i,j}^{N} \hat{s}_{kk} \; . \tag{10.167}$$

Overlap integrals are

$$S_{ab} = \langle \Psi_b | \Psi_a \rangle = \det(\mathbf{U}) \det(\mathbf{V}^\dagger) \prod_{i=1}^{N} \hat{s}_{ii} \; . \tag{10.168}$$

In an atomic orbital basis set, $\{\chi\}$, we can expand the molecular spin orbitals $\mathbf{a}$ and $\mathbf{b}$,

$$\mathbf{a} = \chi \mathbf{A}, \qquad \hat{\mathbf{a}} = \chi \mathbf{A} \mathbf{V} = \chi \hat{\mathbf{A}} \tag{10.169}$$

$$\mathbf{b} = \chi \mathbf{B}, \qquad \hat{\mathbf{b}} = \chi \mathbf{B} \mathbf{U} = \chi \hat{\mathbf{B}} \tag{10.170}$$

The one-electron terms, Eq. (10.168), can be expressed as

$$\begin{aligned} \Omega_{ab}^{(1)} &= \sum_{i}^{N} \sum_{\lambda\sigma} \hat{A}_{\lambda i} T_{ii} \hat{B}_{i\sigma}^\dagger \langle \chi_\sigma | \omega(1) | \chi_\lambda \rangle \\ &= \sum_{\lambda\sigma} G_{\lambda\sigma} \omega_{\sigma\lambda} \end{aligned} \tag{10.171}$$

where $T_{ii} = S_{ab}/\hat{s}_{ii}$ and define a generalized density matrix, $\mathbf{G}$:

$$\mathbf{G} = \hat{\mathbf{A}} \mathbf{T} \hat{\mathbf{B}}^\dagger \; . \tag{10.172}$$

Similarly, the two-electron terms, Eq. (10.167), are

$$\begin{aligned} \Omega_{ab}^{(2)} &= \frac{1}{2} \sum_{ij} \sum_{\lambda\sigma} \sum_{\mu\nu} \hat{A}_{\lambda i} \hat{A}_{\sigma j} \left(\frac{1}{\hat{s}_{ii}} \right) T_{jj} \hat{B}_{i\mu}^\dagger \hat{B}_{j\nu}^\dagger \langle \chi_\mu \chi_\nu | \omega(1,2) | \chi_\lambda \chi_\sigma \rangle \\ &= \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{\mathrm{L}} G_{\sigma\nu}^{\mathrm{R}} \langle \mu\nu || \lambda\sigma \rangle \end{aligned} \tag{10.173}$$

where $\mathbf{G}^{\mathrm{R}}$ and $\mathbf{G}^{\mathrm{L}}$ are generalized density matrices as defined in Eq. (10.172) except T_{ii} in $\mathbf{G}^{\mathrm{L}}$ is replaced by $1/(2s_{ii})$.

The α- and β-spin orbitals are treated explicitly. In terms of the spatial orbitals, the one- and two-electron contributions can be reduced to

$$\Omega_{ab}^{(1)} = \sum_{\lambda\sigma} G_{\lambda\sigma}^{\alpha}\omega_{\sigma\lambda} + \sum_{\lambda\sigma} G_{\lambda\sigma}^{\beta}\omega_{\sigma\lambda} \tag{10.174}$$

$$\begin{aligned}\Omega_{ab}^{(2)} = &\sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{\mathrm{L}\alpha} G_{\sigma\nu}^{\mathrm{R}\alpha}(\langle\mu\nu|\lambda\sigma\rangle - \langle\mu\nu|\sigma\lambda\rangle) + \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{\mathrm{L}\beta} G_{\sigma\nu}^{\mathrm{R}\alpha}\langle\mu\nu|\lambda\sigma\rangle \\ &+ \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{\mathrm{L}\alpha} G_{\sigma\nu}^{\mathrm{R}\beta}\langle\mu\nu|\lambda\sigma\rangle + \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{\mathrm{L}\beta} G_{\sigma\nu}^{\mathrm{R}\beta}(\langle\mu\nu|\lambda\sigma\rangle - \langle\mu\nu|\sigma\lambda\rangle)\end{aligned} \tag{10.175}$$

The resulting one- and two-electron contributions, Eqs. (10.174) and (10.175) can be easily computed in terms of generalized density matrices using standard one- and two-electron integral routines in Q-CHEM.

10.14.2.4 Direct coupling method for electronic coupling

It is important to obtain proper charge-localized initial and final states for the DC scheme, and this step determines the quality of the coupling values. Q-CHEM provides three approaches to construct charge-localized states:

- **The "1+1" approach**
  Since the system consists of donor and acceptor molecules or fragments, with a charge being localized either donor or acceptor, it is intuitive to combine wave functions of individual donor and acceptor fragments to form a charge-localized wave function. We call this approach "1+1" since the zeroth order wave functions are composed of the HF wave functions of the two fragments.

  For example, for the case shown in Example (10.152), we can use Q-CHEM to calculate two HF wave functions: those of anionic donor and of neutral acceptor and they jointly form the initial state. For the final state, wave

functions of neutral donor and anionic acceptor are used. Then the coupling value is calculated via Eq. (10.156).

Example 10.65 Calculate the electron-transfer coupling for ethylene dimer with "1+1" charge-localized states

```
$molecule
  -1  2
--
  -1  2,  0  1
   C     0.662489     0.000000     0.000000
   H     1.227637     0.917083     0.000000
   H     1.227637    -0.917083     0.000000
   C    -0.662489     0.000000     0.000000
   H    -1.227637    -0.917083     0.000000
   H    -1.227637     0.917083     0.000000
--
   0  1, -1  2
   C     0.720595     0.000000     4.5
   H     1.288664     0.921368     4.5
   H     1.288664    -0.921368     4.5
   C    -0.720595     0.000000     4.5
   H    -1.288664    -0.921368     4.5
   H    -1.288664     0.921368     4.5
$end

$rem
   METHOD                HF
   BASIS                 6-31G(d)
   SCF_PRINT_FRGM        FALSE
   SCF_GUESS             FRAGMO
   STS_DC                TRUE
   POINT_GROUP_SYMMETRY  FALSE
$end
```

In the *$molecule* subsection, the first line is for the charge and multiplicity of the whole system. The following blocks are two inputs for the two molecular fragments (donor and acceptor). In each block the first line consists of the charge and spin multiplicity in the initial state of the corresponding fragment, a comma, then the charge and multiplicity in the final state. Next lines are nuclear species and their positions of the fragment. For example, in the above example, the first block indicates that the electron donor is a doublet ethylene anion initially, and it becomes a singlet neutral species in the final state. The second block is for another ethylene going from a singlet neutral molecule to a doublet anion.

Note that the last three *$rem* variables in this example, POINT_GROUP_SYMMETRY, SCF_GUESS and STS_DC must be set to be the values as in the example in order to perform DC calculation with "1+1" charge-localized states. An additional *$rem* variable, SCF_PRINT_FRGM is included. When it is TRUE a detailed output for the fragment HF self-consistent field calculation is given.

- **The "relaxed" approach**

  In "1+1" approach, the intermolecular interaction is neglected in the initial and final states, and so the final electronic coupling can be underestimated. As a second approach, Q-CHEM can use "1+1" wave function as an initial guess to look for the charge-localized wave function by further HF self-consistent field calculation. This approach would 'relax' the wave function constructed by "1+1" method and include the intermolecular interaction effects in the initial and final wave functions. However, this method may sometimes fail, leading to either convergence problems or a resulting HF wave function that cannot represent the desired charge-localized states. This is more likely to be a problem when calculations are performed with diffusive basis functions, or when the donor and acceptor molecules are very close to each other.

  To perform relaxed DC calculation, set STS_DC = RELAX.

- **Hybrid scheme: Constrained DFT charge-localized states**
  Constrained (C)DFT (Section 5.11) can be used to obtain charge-localized states. It is recommended to set both charge and spin constraints in order to generate proper charge localization. To perform DC calculation with CDFT states, set SAVE_SUBSYSTEM = 10 and SAVE_SUBSYSTEM = 20 to save CDFT molecular orbitals in the first two jobs of a batch jobs, and then in the third job of the batch job, set SCF_GUESS = READ and STS_DC = TRUE to compute electronic coupling values.

  Example 10.10.66 Calculate the electron-transfer coupling for ethylene dimer using CDFT charge-localized states

  View input online

10.14.2.5 The frontier molecular orbital approach

The frontier molecular orbital (FMO) approach is often used with DFT to calculate ET coupling.[214,234] FMO coupling value is essentially an off-diagonal Kohn-Sham matrix element that accounts for overlap,

$$V^{\mathrm{FMO}} = \frac{f_{\mathrm{DA}} - S\left(f_{\mathrm{DD}} + f_{\mathrm{AA}}\right)/2}{1 - S^2} \tag{10.176}$$

where $f_{\mathrm{DA}} = \langle \phi^{\mathrm{D}}_{\mathrm{FMO}} | \hat{f} | \phi^{\mathrm{A}}_{\mathrm{FMO}} \rangle$, with $\hat{f}$ being the Kohn-Sham operator of the donor-acceptor system. The quantities $\phi^{\mathrm{D}}_{\mathrm{FMO}}$ and $\phi^{\mathrm{A}}_{\mathrm{FMO}}$ are the Kohn-Sham frontier MOs for the donor and acceptor fragments, respectively, which represents one-particle scheme of a charge transfer process.

In this approach, computations are often performed separately in the two fragments, and the off-diagonal Kohn-Sham operator (and the overlap matrix) in the FMOs is subsequently calculated. To compute FMO couplings, Q-CHEM has a setup that is similar to the "1+1" approach.

It may be of interest to compute couplings for frontier orbitals beyond the HOMO of the donor and the LUMO of the acceptor. They *$rem* options STS_TRANS_DONOR and STS_TRANS_ACCEPTOR can be used to select a range of occupied and virtual orbitals for FMO coupling calculations.

STS_TRANS_DONOR

Control the range of frontier MOs for the donor in the FMO approach to ET couplings.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

o-*v* Compute couplings between o and v occupied and virtual MOs, respectively.

RECOMMENDATION:

None

STS_TRANS_ACCEPTOR

Control the range of frontier MOs for the acceptor in the FMO approach to ET couplings.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

o-v Compute couplings between o and v occupied and virtual MOs, respectively.

RECOMMENDATION:

None

Example 10.67 Calculate the electron-transfer coupling for a stacked ethylene dimer with the FMO approach

```
$molecule
 0  1
--
 0  1
 C    0.662489    0.000000    0.000000
 H    1.227637    0.917083    0.000000
 H    1.227637   -0.917083    0.000000
 C   -0.662489    0.000000    0.000000
 H   -1.227637   -0.917083    0.000000
 H   -1.227637    0.917083    0.000000
--
 0  1
 C    0.720595    0.000000    4.500000
 H    1.288664    0.921368    4.500000
 H    1.288664   -0.921368    4.500000
 C   -0.720595    0.000000    4.500000
 H   -1.288664   -0.921368    4.500000
 H   -1.288664    0.921368    4.500000
$end

$rem
   METHOD              =  lrcwpbe
   OMEGA               =  370
   BASIS               =  dz*
   SCF_PRINT_FRGM      =  true
   SCF_GUESS           =  fragmo
   STS_DC              =  fock
   STS_TRANS_DONOR     =  2-3  ! use HOMO, HOMO-1 and LUMO, LUMO+1, LUMO+2 of donor
   STS_TRANS_ACCEPTOR  =  1-2  ! use HOMO and LUMO, LUMO+1 of acceptor
   POINT_GROUP_SYMMETRY= False
$end

$rem_frgm
 print_orbitals = 5
$end
```

10.14.3 Fragment-Based Methods for Electronic Coupling

10.14.3.1 Approach based on absolutely localized molecular orbitals

One can use absolutely localized molecular orbitals (ALMOs, see Chapter 12) to construct charge-localized diabatic states directly from DFT calculations. The ALMOs on each fragment are expanded by the AO basis functions belonging

to the same fragment alone, whose corresponding MO coefficient matrix is fragment block-diagonal.[107] In energy decomposition analysis methods,[92,108] ALMOs are utilized to separate the effects of polarization and charge transfer in intermolecular binding, because they have the useful property that they do not allow for charge transfer between fragments under the Mulliken definition of charge population. Making use of this property, one can construct charge-localized diabats for hole and electron transfer. For example, considering the initial and final states of a hole transfer process, $|D^+A\rangle$ and $|DA^+\rangle$, the two diabats can be represented in the following form:

$$|\psi_a\rangle = \frac{1}{\sqrt{(N-1)!}}\det\left\{\phi_{D1}^{(a)}, \phi_{D2}^{(a)}, \ldots, \phi_{Dn_D-1}^{(a)}\phi_{A1}^{(a)}, \phi_{A2}^{(a)}, \ldots, \phi_{An_A}^{(a)}\right\} \tag{10.177a}$$

$$|\psi_b\rangle = \frac{1}{\sqrt{(N-1)!}}\det\left\{\phi_{D1}^{(b)}, \phi_{D2}^{(b)}, \ldots, \phi_{Dn_D}^{(b)}\phi_{A1}^{(b)}, \phi_{A2}^{(b)}, \ldots, \phi_{An_A-1}^{(b)}\right\} \tag{10.177b}$$

For systems where the donor and acceptor moieties are well-separated, one can construct the ALMO-based diabats by simply concatenating orbitals obtained from isolated fragment calculations: D^+ and A for one diabat, and D and A^+ for the other. The energy of each ALMO diabat can then be variationally optimized with respect to orbital rotations on fragment, using the SCFMI technique (see Section 12.5.1).[66,107,201] These ALMO-based diabatic states are variationally optimized such that the associated nuclear forces can be easily computed.[132] The mutual polarization of donor and acceptor moieties in the presence of each other is also taken into account.

To calculate the electronic coupling between two ALMO diabats, one should first construct the diabatic Hamiltonian in the ALMO state basis

$$\mathbf{H}' = \begin{pmatrix} H'_{aa} & H'_{ab} \\ H'_{ba} & H'_{bb} \end{pmatrix} \tag{10.178}$$

and then transform that into the Löwdin-orthogonalized basis

$$\mathbf{H} = \mathbf{S}^{-1/2}\mathbf{H}'\mathbf{S}^{-1/2} \tag{10.179}$$

whose off-diagonal element, H_{ab}, corresponds to the diabatic coupling to be evaluated. In the 2-state case, we have

$$H_{ab} = \frac{1}{1-S_{ab}^2}\left|H'_{ab} - \frac{H'_{aa}+H'_{bb}}{2}S_{ab}\right| \tag{10.180}$$

which requires the overlap between two ALMO diabats and the diagonal and off-diagonal elements of $\mathbf{H}'$. The interstate overlap is given by

$$S_{ab} = \langle\psi_a|\psi_b\rangle = \det\left[(\mathbf{C}_{\mathrm{o}}^{(a)})^\dagger\mathbf{S}\mathbf{C}_{\mathrm{o}}^{(b)}\right] . \tag{10.181}$$

where $\mathbf{C}_{\mathrm{o}}^{(a)}$ and $\mathbf{C}_{\mathrm{o}}^{(b)}$ are MO coefficients for the occupied orbitals in diabats $|\psi_a\rangle$ and $|\psi_b\rangle$, respectively, and $\mathbf{S}$ is the AO overlap matrix.

The elements of the diabatic Hamiltonian matrix can be evaluated using the multi-state DFT (MSDFT) approach.[40,133,182] For the diagonal elements, it is straightforward to employ the KS energies of the two diabats:

$$H'_{aa} = E_a^{\mathrm{KS}}[\mathbf{P}^{(a)}], \quad H'_{bb} = E_b^{\mathrm{KS}}[\mathbf{P}^{(b)}] \tag{10.182}$$

where $\mathbf{P}^{(a)}$ and $\mathbf{P}^{(b)}$ are the one-electron density matrices associated with two ALMO states $|\psi_a\rangle$ and $|\psi_b\rangle$, respectively. The approximation for the off-diagonal element is theoretically more challenging. In the original MSDFT scheme,[40,182]

$$H'_{ab} = S_{ab}\left[V_{\mathrm{nn}} + \mathbf{P}_{ab}\cdot\mathbf{h} + \frac{1}{2}\mathbf{P}_{ab}\cdot\mathbf{II}\cdot\mathbf{P}_{ab} + \frac{1}{2}(\Delta E_a^{\mathrm{c}} + \Delta E_b^{\mathrm{c}})\right] \tag{10.183}$$

where $\mathbf{P}_{ab}$ is the one-particle transition density matrix between two ALMO states

$$\mathbf{P}_{ab} = \mathbf{C}_{\mathrm{o}}^{(a)}\left[(\mathbf{C}_{\mathrm{o}}^{(b)})^\dagger\mathbf{S}\mathbf{C}_{\mathrm{o}}^{(a)}\right]^{-1}(\mathbf{C}_{\mathrm{o}}^{(b)})^\dagger \tag{10.184}$$

The first three terms on the right-hand side of Eq. (10.183) correspond to the contributions from nuclear repulsion, one-electron Hamiltonian (kinetic energy and nuclei-electron attraction), and full two-electron integrals (Coulomb and

full HF exchange), which can be derived as in non-orthogonal CI.[211] The last term accounts for the contribution from exchange-correlation (XC) functional as a correction to the HF coupling, which is given by the average of the difference between the KS and HF energies calculated from the same one-electron density matrix for each diabat:

$$\Delta E_a^{\mathrm{c}} = E_a^{\mathrm{KS}}[\mathbf{P}^{(a)}] - E_a^{\mathrm{HF}}[\mathbf{P}^{(a)}] \tag{10.185a}$$

$$\Delta E_b^{\mathrm{c}} = E_b^{\mathrm{KS}}[\mathbf{P}^{(b)}] - E_b^{\mathrm{HF}}[\mathbf{P}^{(b)}] \, . \tag{10.185b}$$

This approach was denoted as ALMO(MSDFT) in Ref. 133 and it was found to overestimate the electronic couplings for the tested hole and electron transfer systems. A modified approach, denoted as ALMO(MSDFT2), was proposed in Ref. 133, which evaluates the XC contribution using the XC energy of the symmetrized transition density matrix

$$H'_{ab} = S_{ab}\left(V_{\mathrm{nn}} + \mathbf{P}_{ab}\cdot\mathbf{h} + \tfrac{1}{2}\mathbf{P}_{ab}\cdot\mathbf{II}\cdot\mathbf{P}_{ab} + E_{\mathrm{xc}}[\tilde{\mathbf{P}}_{ab}]\right) \tag{10.186}$$

where

$$\tilde{\mathbf{P}}_{ab} = \frac{1}{2}(\mathbf{P}_{ab} + \mathbf{P}_{ba}) \, . \tag{10.187}$$

Note that in Eq. (10.186), $\mathbf{II}$ includes only Coulomb integrals and a fraction of exact exchange if hybrid functionals are employed.

According to the benchmark results in Ref. 133, ALMO(MSDFT2) shows better accuracy than the original MSDFT method for hole and electron transfer, and thus it is implemented as the default approach to compute electronic couplings between ALMO diabats in Q-CHEM. We note that the results given by Eq. (10.186) may become inaccurate when the overlap between two states becomes near-singular, as

$$\boldsymbol{\sigma}_{ba} = (\mathbf{C}_{\mathrm{o}}^{(b)})^{\dagger}\mathbf{S}\mathbf{C}_{\mathrm{o}}^{(a)} \tag{10.188}$$

is inverted when constructing the transition density [Eq. (10.184)]. To circumvent this numerical issue, one can replace the inverse in Eq. (10.184) with the Penrose pseudo-inverse, which was suggested for a similar objective in Ref. 163.

The evaluation of electronic coupling between ALMO-based states using the original MSDFT or MSDFT2 scheme has been extended to non-aufbau electronic states prepared using ALMO-based ΔSCF calculations.[134] This method, which can be referred to as Δ-ALMO(MSDFT) or Δ-ALMO(MSDFT2), is capable of calculating the coupling between locally excited (LE) and charge-transfer (CT) states. More details regarding the job control can be found in Sec. 10.14.3.4.

10.14.3.2 Projection operator Diabatization (POD)

Besides ALMO-based diabatization method, other fragment-based diabatization methods are available in Q-CHEM. The projection operator diabatization (POD) method[115] starts from a standard KS-DFT calculation of the system and post-processes the converged Fock matrix. It first transforms the Fock matrix into the Löwdin-orthogonalized AO basis and then partitions that into the donor and acceptor blocks, assuming that these orthogonalized AO basis functions still retain their original fragment tags:

$$\tilde{\mathbf{F}} = \mathbf{S}^{-1/2}\mathbf{F}\mathbf{S}^{-1/2} = \begin{pmatrix} \tilde{\mathbf{F}}_{dd} & \tilde{\mathbf{F}}_{da} \\ \tilde{\mathbf{F}}_{ad} & \tilde{\mathbf{F}}_{aa} \end{pmatrix} \tag{10.189}$$

One then diagonalizes $\tilde{\mathbf{F}}_{dd}$ and $\tilde{\mathbf{F}}_{aa}$ separately

$$\boldsymbol{\epsilon}_d = \mathbf{D}_d^{\dagger}\tilde{\mathbf{F}}_{dd}\mathbf{D}_d, \;\; \boldsymbol{\epsilon}_a = \mathbf{D}_a^{\dagger}\tilde{\mathbf{F}}_{aa}\mathbf{D}_a, \tag{10.190}$$

where the eigenvectors $\mathbf{D}_d$ and $\mathbf{D}_a$ define the single-particle "diabatic states":

$$\begin{aligned} \left|\bar{\varphi}_p^{(d)}\right\rangle &= \sum_{\mu} \left|\tilde{\chi}_{\mu}^{(d)}\right\rangle (D_d)^{\mu}_{\ p} \\ \left|\bar{\varphi}_p^{(a)}\right\rangle &= \sum_{\mu} \left|\tilde{\chi}_{\mu}^{(a)}\right\rangle (D_a)^{\mu}_{\ p}, \end{aligned} \tag{10.191}$$

and transforms the off-diagonal block of the Fock matrix into this diabatic basis

$$\bar{\mathbf{F}}_{da} = \mathbf{D}_d^{\dagger}\tilde{\mathbf{F}}_{da}\mathbf{D}_a \tag{10.192}$$

yielding

$$\bar{\mathbf{F}} = \begin{pmatrix} \boldsymbol{\epsilon}_d & \bar{\mathbf{F}}_{da} \\ \bar{\mathbf{F}}_{ad} & \boldsymbol{\epsilon}_a \end{pmatrix} \tag{10.193}$$

The couplings between these single-particle diabatic orbitals can then be directly read off from the elements of $\bar{\mathbf{F}}_{da}$.

The Q-CHEM implementation of the POD method follows the description in Refs. 115 and 232, where a closed-shell reference system is used to generate the Fock matrix to be processed, *i.e.*, $\mathbf{F}$ in Eq. (10.189). By default, only the D(HOMO)–A(HOMO) coupling is calculated for the hole transfer cases, and the D(LUMO)–A(LUMO) coupling for the electron transfer cases. To calculate the couplings between multiple pairs of donor and acceptor orbitals, the user can set *$rem* variable POD_MULTI_PAIRS = TRUE and control the number of orbitals pairs through POD_WINDOW. See the instruction in Section 10.14.3.4.

Because of the use of globally Löwdin orthogonalized orbitals in Eq. (10.189), the diabatic orbitals created by POD cannot be strictly localized on fragments. This renders the POD results unstable with the change of employed AO basis sets: when larger basis sets are used, the mixing between AO basis functions on different fragments becomes stronger, and the resulting H_{ab} decreases. To alleviate this problem, a revised POD method, which was named as "POD2", was proposed by Ghan *et al.*.[64] It avoids the global Löwdin orthogonalization of the AO basis; instead, it separately diagonalizes the the donor and acceptor blocks of the Fock matrix (in the original AO basis):

$$\mathbf{F}_{dd}\mathbf{C}_d = \mathbf{S}_{dd}\mathbf{C}_d\boldsymbol{\epsilon}_d, \quad \mathbf{F}_{aa}\mathbf{C}_a = \mathbf{S}_{aa}\mathbf{C}_a\boldsymbol{\epsilon}_a \tag{10.194}$$

The obtained diabatic MO coefficient matrix is fragment-block-diagonal in the AO basis:

$$\mathbf{C}_{\text{diab}} = \begin{pmatrix} \mathbf{C}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{C}_a \end{pmatrix} \tag{10.195}$$

Transforming the AO Fock matrix into this diabatic MO basis, the D–D and A–A blocks of the resulting matrix are diagonal matrices:

$$\bar{\mathbf{F}}_{\text{diab}} = \mathbf{C}_{\text{diab}}^T\mathbf{F}\mathbf{C}_{\text{diab}} = \begin{pmatrix} \boldsymbol{\epsilon}_d & \bar{\mathbf{F}}_{da} \\ \bar{\mathbf{F}}_{ad} & \boldsymbol{\epsilon}_a \end{pmatrix} \tag{10.196}$$

Using the matrix elements in the off-diagonal block ($\bar{\mathbf{F}}_{da}$) directly would yield overestimated couplings since the diabatic MOs $\mathbf{C}_d$ and $\mathbf{C}_a$ are not orthogonal to each other. Therefore, a final orthogonalization step is required to obtain the diabatic coupling between a pair of orbitals that are located on the donor and acceptor, respectively. Denoting this pair of orbital as ϕ_d and ϕ_a, one can construct the 2×2 Hamiltonian and overlap matrices:

$$\mathbf{H} = \begin{pmatrix} \epsilon_d & \bar{F}_{da} \\ \bar{F}_{ad} & \epsilon_a \end{pmatrix}, \quad \mathbf{S} = \begin{pmatrix} 1 & S_{da} \\ S_{ad} & 1 \end{pmatrix} \tag{10.197}$$

Two orthogonalization schemes have been investigated by Ghan *et al.*.[64] The first approach performs a Löwdin orthgonalization on ϕ_d and ϕ_a, which is denoted as POD2L. The resulting coupling between the orthogonalized diabatic orbitals are

$$H_{da}^{\text{eff}} = \frac{1}{1 - S_{da}^2}\left|\bar{F}_{da} - \frac{1}{2}(\epsilon_d + \epsilon_a)S_{da}\right| \tag{10.198}$$

The second approach employs the Gram-Schmidt orthogonalization, which keeps one of the two orbitals (ϕ_d or ϕ_a) intact while ensures that the other is strictly orthogonal to it. This approach is denoted as POD2GS, and it might be better choice for asymmetric cases (*e.g.* surface and adsorbates) where one can choose to retain the orbital on the less sizable fragment. These two POD2 variants afford significantly improved accuracy over the original POD method, especially in terms of the robustness with regard to the use of extensive basis sets.

10.14.3.3 Fragment Orbital DFT (FODFT)

Fragment orbital DFT (FODFT)[151,187,190] is an approach to compute the diabatic couplings for hole and electron transfer between fragments. There have been several different flavors of FODFT approaches developed in literature, and here we introduce the most recent variant by Schober *et al.*.[187] Considering a hole-transfer process, $\mathrm{D}^+ + \mathrm{A} \rightarrow \mathrm{D} + \mathrm{A}^+$, or alternatively an electron transfer process $\mathrm{D}^- + \mathrm{A} \rightarrow \mathrm{D} + \mathrm{A}^-$, where the donor (D) and acceptor (A) fragments have n_D and n_A electrons, respectively, the procedure is the following.

- Perform KS-DFT calculations for isolated donor and acceptor fragments; collect the converged fragment orbitals: $\{\phi_{\mathrm{D}1}, \phi_{\mathrm{D}2}, \ldots, \phi_{\mathrm{D}n_\mathrm{D}\pm1}\}$ and $\{\phi_{\mathrm{A}1}, \phi_{\mathrm{A}2}, \ldots, \phi_{\mathrm{A}n_\mathrm{A}}\}$
- Löwdin orthogonalize the occupied orbitals on two fragments. The reactant diabat ($\mathrm{D}^+\mathrm{A}$ or $\mathrm{D}^-\mathrm{A}$) can be represented as
$$|\bar{\psi}_a\rangle = \frac{1}{\sqrt{(N-1)!}} \det\left\{\bar{\phi}_{\mathrm{D}1}, \bar{\phi}_{\mathrm{D}2}, \ldots, \bar{\phi}_{\mathrm{D}n_\mathrm{D}\pm1}\bar{\phi}_{\mathrm{A}1}, \bar{\phi}_{\mathrm{A}2}, \ldots, \bar{\phi}_{\mathrm{A}n_\mathrm{A}}\right\} \tag{10.199}$$
where "$\bar{\phi}$" denotes Löwdin-orthogonalized orbitals, and $N = n_\mathrm{D} + n_\mathrm{A}$. Note that the lowest unoccupied orbital where the electron is transferring to, $\phi_{\mathrm{D}n_\mathrm{D}}$ in the case of HT or $\phi_{\mathrm{A}n_\mathrm{A}+1}$ in the case of ET, also needs to be made orthogonal to the space spanned by all occupied orbitals.
- Construct the product diabat (DA^+ or DA^-), simply by moving the hole from $\bar{\phi}_{\mathrm{D}n_\mathrm{D}}$ to $\bar{\phi}_{\mathrm{A}n_\mathrm{A}}$ (HT), or the excess electron from $\bar{\phi}_{\mathrm{D}n_\mathrm{D}+1}$ to $\bar{\phi}_{\mathrm{A}n_\mathrm{A}+1}$ (ET)
$$|\bar{\psi}_b\rangle = \frac{1}{\sqrt{(N-1)!}} \det\left\{\bar{\phi}_{\mathrm{D}1}, \bar{\phi}_{\mathrm{D}2}, \ldots, \bar{\phi}_{\mathrm{D}n_\mathrm{D}}\bar{\phi}_{\mathrm{A}1}, \bar{\phi}_{\mathrm{A}2}, \ldots, \bar{\phi}_{\mathrm{A}n_\mathrm{A}\pm1}\right\} \tag{10.200}$$
- Compute the electronic coupling between $|\bar{\psi}_a\rangle$ and $|\bar{\psi}_b\rangle$, which is approximated by the coupling of the orthogonalized fragment orbitals through the Kohn-Sham Fock operator, built from the reactant diabat:
$$\langle\bar{\psi}_a|\hat{H}|\bar{\psi}_b\rangle \approx \begin{cases} \langle\phi_{\mathrm{D}n_\mathrm{D}}|\hat{f}_\mathrm{KS}|\phi_{\mathrm{A}n_\mathrm{A}}\rangle & \mathrm{HT} \\ \langle\phi_{\mathrm{D}n_\mathrm{D}+1}|\hat{f}_\mathrm{KS}|\phi_{\mathrm{A}n_\mathrm{A}+1}\rangle & \mathrm{ET} \end{cases} \tag{10.201}$$

The approach described above is denoted as FODFT($2n-1$)@D^+A (HT)/FODFT($2n+1$)@D^-A (ET),[187] as the charged fragment is explicitly taken into account when preparing the fragment orbitals and the KS Fock matrix is built from $2n \mp 1$ occupied orbitals. Besides this, there are two other variants of FODFT:

1. FODFT($2n$)@DA:[190] fragment orbitals prepared with D and A both closed-shell; KS Fock operator constructed from $2n$ occupied orbitals.
2. FODFT(2n $-$ 1)@DA (HT) / FODFT($2n+1$)@$\mathrm{D}^-\mathrm{A}^-$ (ET):[151] fragment orbitals prepared with the system having one excess electron (DA for HT and $\mathrm{D}^-\mathrm{A}^-$ for ET), while one occupied orbital is removed when building the KS Fock operator.

According to the benchmark results,[133,187] FODFT($2n-1$)@D^+A (HT)/FODFT($2n+1$)@D^-A (ET) is the best-performing method, possibly because of its explicit account for charged fragments and consistent electron count in the preparation of fragment orbitals and in the construction of Fock matrix.

One issue associated with the FODFT methods is that for asymmetric systems, the results would depend on how one chooses the initial and final states for an electron or hole transfer process (*e.g.*, $\mathrm{D}^+\mathrm{A}$ vs. DA^+), especially for the two variants that build the Fock matrix with $2n \pm 1$ occupied orbitals.[133] The Q-CHEM implementation of FODFT(2n $-$ 1)@DA/FODFT($2n+1$)@$\mathrm{D}^-\mathrm{A}^-$ automatically computes H_{ab} in both ways and then reports the average, as it only requires an extra Fock matrix build. This, however, is not automatically done for FODFT($2n-1$)@D^+A/FODFT($2n+1$)@D^-A.

10.14.3.4 Job control for fragment-based diabatization methods

POD, FODFT, and ALMO(MSDFT) calculations in Q-CHEM require specification of fragments in the *$molecule* section (see Section 12.2). For ALMO(MSDFT) calculations, one also needs to specify the charge and multiplicity of each fragment in each diabatic state in the *$almo_coupling* section, where two hyphens indicate the separation of different diabats:

```
$almo_coupling
   charge_frag_1     mult_frag_1        !diabat 1
   charge_frag_2     mult_frag_2
   --
   charge_frag_1     mult_frag_1        !diabat 2
   charge_frag_2     mult_frag_2
$end
```

The method employed to calculate the diabatic coupling is specified through FRAG_DIABAT_METHOD. Note that the current implementation of FODFT is limited to (i) hole transfer between the HOMOs of two fragments or (ii) electron transfer between the LUMOs of two fragments.

The ALMO(MSDFT)/ALMO(MSDFT2) method can be used to calculate the diabatic couplings involved in ground-state electron or hole transfer processes. With the incorporation of ALMO-based ΔSCF (supported by Q-CHEM 6.2.1 and after), this method can further be used to calculate the coupling between LE and CT states in photoinduced electron or hole transfer processes. One will need set EX_ALMO_MSDFT = TRUE for these calculations. Additionally, a simple approach based on the off-diagonal elements of Löwdin orthogonalized Hamiltonian (Eq. 10.179 generalized to multi-state cases) has also been implemented to calculate the diabatic couplings with multiple (> 2) states taken into account at the same time.

The ALMO-based ΔSCF states will be specified through the *$scfmi_mom* section, for which the syntax is documented in detail in Sec. 12.6.7. The specifications for different diabatic states are separated by "--". For diabatic states that do not involve on-fragment excitations, simply write "--" with no additional specification.

FRAG_DIABAT_METHOD

Specify fragment based diabatization method

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

ALMO_MSDFT	Perform ALMO(MSDFT) diabatization
POD	Perform projection operator diabatization (the original method)
POD2_L	Perform POD2 with Löwdin orthogonalization
POD2_GS	Perform POD2 with Grad-Schmidt orthogonalization
ESID	The energy-split-in-dimer method,[214] which is equivalent to the FMO approach introduced in Section 10.14.2.5
FODFT	Calculate electronic coupling using fragment orbital DFT

RECOMMENDATION:

NONE

FRAG_DIABAT_DOHT

Specify whether hole or electron transfer is considered

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Do hole transfer

FALSE Do electron transfer

RECOMMENDATION:

Need to be specified for POD and FODFT calculations

FRAG_DIABAT_PRINT

Specify the print level for fragment based diabatization calculations

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No additional prints

≥ 1 Print additional details

RECOMMENDATION:

Use the default unless debug information is needed

MSDFT_METHOD

Specify the scheme for ALMO(MSDFT)

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 The original MSDFT scheme [Eq. (10.183)]

2 The ALMO(MSDFT2) approach [Eq. (10.186)]

RECOMMENDATION:

Use the default method. Note that the method will be automatically reset to 1 if a meta-GGA functional is requested.

MSDFT_PINV_THRESH

Set the threshold for pseudo-inverse of the interstate overlap

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n Set the threshold to 10^{-n}

RECOMMENDATION:

Use the default value

EX_ALMO_MSDFT

Enable MSDFT/MSDFT2 calculations using ALMO-based ΔSCF states

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Will calculate ALMO-ΔSCF states as specified in *$scfmi_mom*

FALSE Performing ground-state ALMO(MSDFT) calculations.

RECOMMENDATION:

None

POD_MULTI_PAIRS

Calculate the couplings between multiple pairs of donor and acceptor orbitals in POD

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate the couplings between multiple pairs of orbitals

FALSE Only calculate the D(HOMO)–A(HOMO) coupling (for hole transfer) or the D(LUMO)–A(LUMO) coupling (for electron transfer)

RECOMMENDATION:

None

POD_WINDOW

Specify the number of donor and acceptor orbitals when couplings between multiple pairs are requested

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n Including n frontier occupied orbitals (from HOMO $- n + 1$ to HOMO) and n frontier virtual orbitals (from LUMO to LUMO $+ n - 1$) for both donor and acceptor

RECOMMENDATION:

None

FODFT_METHOD

Specify the flavor of FODFT method

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 FODFT$(2\mathrm{n}-1)@D^{+}A$ (HT) / FODFT$(2\mathrm{n}+1)@D^{-}A$ (ET)

2 FODFT$(2\mathrm{n})@DA$

3 FODFT$(2\mathrm{n}-1)@DA$ (HT) / FODFT$(2\mathrm{n}+1)@D^{-}A^{-}$ (ET)

RECOMMENDATION:

The default approach shows the best overall performance

FODFT_DONOR

Specify the donor fragment in FODFT calculation

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 First fragment as donor

2 Second fragment as donor

RECOMMENDATION:

With FODFT_METHOD = 1, the charged fragment needs to be the donor fragment

Example 10.68 ALMO(MSDFT2) calculation for hole transfer in ethylene dimer

```
$molecule
1 2
--
1 2
  C      0.000000    0.000000    0.000000
  C      1.332000    0.000000    0.000000
  H     -0.574301    0.000000   -0.928785
  H     -0.574301    0.000000    0.928785
  H      1.906301    0.000000    0.928785
  H      1.906301    0.000000   -0.928785
--
0 1
  C     -0.000000    4.000000    0.000000
  C      1.332000    4.000000   -0.000000
  H     -0.574301    4.000000    0.928785
  H     -0.574301    4.000000   -0.928785
  H      1.906301    4.000000   -0.928785
  H      1.906301    4.000000    0.928785
$end

$rem
   METHOD               PBE0
   BASIS                6-31+G(D)
   UNRESTRICTED         TRUE
   THRESH               14
   SCF_CONVERGENCE      8
   SCFMI_MODE           1
   FRGM_METHOD          STOLL
   FRAG_DIABAT_METHOD   ALMO_MSDFT
   INTEGRAL_SYMMETRY    FALSE
   POINT_GROUP_SYMMETRY FALSE
$end

$almo_coupling
  1  2
  0  1
  --
  0  1
  1  2
$end
```

Example 10.69 ALMO(MSDFT2) for the hole transfer between Indole^{+*} and guanine. The indole radical cation is in a non-aufbau state, with one β electron promoted from HOMO to HOMO-1.

```
$molecule
1 2
--
1 2
C        2.36540       -2.78410        3.38000
C        3.73980       -2.83830        3.38000
C        2.00000       -1.40150        3.38000
C        3.20860       -0.65020        3.38000
C        0.77390       -0.70880        3.38000
C        3.22680        0.75070        3.38000
C        0.78630        0.67870        3.38000
C        2.00000        1.40150        3.38000
N        4.24650       -1.55510        3.38000
H        4.40550       -3.69200        3.38000
H        5.22930       -1.31740        3.38300
H       -0.16770       -1.25470        3.38000
H        4.16090        1.30910        3.38000
H       -0.15420        1.22450        3.38000
H        1.97300        2.48860        3.38000
H        1.69890       -3.63700        3.38000
--
0 1
C        1.63430       -2.77360        0.00200
C        1.25300       -0.68190        0.01800
C        1.32680        0.75830        0.00900
C       -1.17560        0.58330        0.00600
C        0.00000       -1.29190        0.00000
O        2.29090        1.51390       -0.00300
N        0.26680       -2.63600       -0.00400
N        2.26910       -1.61180        0.01600
N        0.00000        1.29190        0.00000
N       -2.33770        1.33540       -0.07400
N       -1.24290       -0.72560        0.01500
H        2.11200       -3.74490       -0.00100
H       -0.02860        2.30410       -0.09800
H       -2.35450        2.16800        0.50600
H       -3.15580        0.75300        0.07300
H       -0.42990       -3.37190       -0.01800
$end

$rem
JOBTYPE  SP
METHOD   B3LYP
BASIS    6-31G
SCF_CONVERGENCE  8
POINT_GROUP_SYMMETRY  FALSE
SCFMI_MODE   1
FRGM_METHOD  STOLL
FRAG_DIABAT_METHOD ALMO_MSDFT
MSDFT_METHOD    2
EX_ALMO_MSDFT   TRUE
$end

$almo_coupling
1  2
0  1
--
0  1
1  2
$end

$scfmi_mom
1 0 0 b
--
```

Example 10.70 POD diabatization method for hole transfer in ethylene dimer. FRAG_DIABAT_METHOD can be set to POD2_L or POD2_GS for POD2 diabatization methods.

```
$molecule
0 1
--
0 1
  C      0.000000    0.000000    0.000000
  C      1.332000    0.000000    0.000000
  H     -0.574301    0.000000   -0.928785
  H     -0.574301    0.000000    0.928785
  H      1.906301    0.000000    0.928785
  H      1.906301    0.000000   -0.928785
--
0 1
  C     -0.000000    4.000000    0.000000
  C      1.332000    4.000000   -0.000000
  H     -0.574301    4.000000    0.928785
  H     -0.574301    4.000000   -0.928785
  H      1.906301    4.000000   -0.928785
  H      1.906301    4.000000    0.928785
$end

$rem
   METHOD               lrc-wpbeh
   BASIS                6-31+g(d)
   FRAG_DIABAT_METHOD   pod
   SCF_CONVERGENCE      8
   THRESH               14
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 10.71 FODFT(2n − 1)@D^+A calculation for hole transfer in ethylene dimer

```
$molecule
1 2
--
1 2
  C      0.000000    0.000000    0.000000
  C      1.332000    0.000000    0.000000
  H     -0.574301    0.000000   -0.928785
  H     -0.574301    0.000000    0.928785
  H      1.906301    0.000000    0.928785
  H      1.906301    0.000000   -0.928785
--
0 1
  C     -0.000000    4.000000    0.000000
  C      1.332000    4.000000   -0.000000
  H     -0.574301    4.000000    0.928785
  H     -0.574301    4.000000   -0.928785
  H      1.906301    4.000000   -0.928785
  H      1.906301    4.000000    0.928785
$end

$rem
   METHOD               wb97x-d
   BASIS                6-31+g(d)
   UNRESTRICTED         true
   SCF_CONVERGENCE      8
   THRESH               14
   FRAG_DIABAT_METHOD   fodft
   FODFT_METHOD         1
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

References and Further Reading

[1] The MOLDEN program may be freely downloaded from http://www.cmbi.ru.nl/molden/molden.html.

[2] MACMOLPLT may be downloaded from https://brettbode.github.io/wxmacmolplt.

[3] NBO 6.0 manual: http://nbo6.chem.wisc.edu/.

[4] The VMD program may be downloaded from http://www.ks.uiuc.edu/Research/vmd.

[5] Ground-State Methods (Chapters 4 and 6).

[6] Excited-State Calculations (Chapter 7).

[7] Basis Sets (Chapter 8).

[8] A. Adel and D. M. Dennison. *Phys. Rev.*, 43:716, 1933. DOI: 10.1103/PhysRev.43.716.

[9] A. C. Albrecht. *J. Chem. Phys.*, 34:1476, 1961. DOI: 10.1063/1.1701032.

[10] A. C. Albrecht and M. C. Hutley. *J. Chem. Phys.*, 55:4438, 1971. DOI: 10.1063/1.1676771.

[11] A. Aldossary and M. Head-Gordon. *J. Chem. Phys.*, 157:094102, 2022. DOI: 10.1063/5.0095443.

[12] W. D. Allen, Y. Yamaguchi, A. G. Császár, D. A. Clabo, Jr., R. B. Remington, and H. F. Schaefer III. *Chem. Phys.*, 145:427, 1990. DOI: 10.1016/0301-0104(90)87051-C.

[13] S. R. May andd C. Climent, Z. Tao, S. A. Vinogradov, and J. E. Subotnik. *J. Phys. Chem. A*, 127:3591, 2023. DOI: 10.1021/acs.jpca.3c00294.

[14] T. Bally and P. R. Rablen. *J. Org. Chem*, 76:4818, 2011. DOI: 10.1021/jo200513q.

[15] S. A. Bäppler, F. Plasser, M. Wormit, and A. Dreuw. *Phys. Rev. A*, 90:052521, 2014. DOI: 10.1103/PhysRevA.90.052521.

[16] V. Barone. *J. Chem. Phys.*, 122:014108, 2005. DOI: 10.1063/1.1824881.

[17] V. Barone, M. Biczysko, and G. Brancato. *Adv. Quantum Chem.*, 59:17, 2010. DOI: 10.1016/S0065-3276(10)59002-6.

[18] V. Barone, A. Baiardi, M. Biczysko, J. Bloino, and C. Cappelli ad F. Lipparini. *Phys. Chem. Chem. Phys.*, 14: 12404, 2012. DOI: 10.1039/C2CP41006K.

[19] A. D. Becke. *J. Chem. Phys.*, 88:2547, 1988. DOI: 10.1063/1.454033.

[20] A. D. Becke. *J. Chem. Phys.*, 119:2972, 2003. DOI: 10.1063/1.1589733.

[21] A. D. Becke. *J. Chem. Phys.*, 122:064101, 2005. DOI: 10.1063/1.1844493.

[22] A. D. Becke and K. E. Edgecombe. *J. Chem. Phys.*, 92:5397, 1990. DOI: 10.1063/1.458517.

[23] E. J. Berquist and D. S. Lambrecht. A first principles approach for partitioning linear response properties into additive and cooperative contributions. *ChemRxiv*, 2018. DOI: 10.26434/chemrxiv.5773968.v1.

[24] B. H. Besler, K. M. Merz Jr., and P. A. Kollman. *J. Comput. Chem.*, 11:431, 1990. DOI: 10.1002/jcc.540110404.

[25] N. A. Besley and J. A. Bryan. *J. Phys. Chem. C*, 112:4308, 2008. DOI: 10.1021/jp076167x.

[26] N. A. Besley and K. A. Metcalf. *J. Chem. Phys.*, 126:035101, 2007. DOI: 10.1063/1.2426344.

[27] R. C. Bochicchio. *J. Mol. Struct. (Theochem)*, 429:229, 1998. DOI: 10.1016/S0166-1280(97)00357-6.

[28] B. M. Bode and M. S. Gordon. *J. Mol. Graphics Mod.*, 16:133, 1998. DOI: 10.1016/S1093-3263(99)00002-9.

[29] S. F. Boys. *Rev. Mod. Phys.*, 32:296, 1960. DOI: 10.1103/RevModPhys.32.296.

[30] S. F. Boys. Localized orbitals and localized adjustment functions. In P.-O. Löwdin, editor, *Quantum Theory of Atoms, Molecules, and the Solid State: A Tribute to John C. SSlater*, pages 253–262. Academic Press, New York, 1966.

[31] C. M. Breneman and K. B. Wiberg. *J. Comput. Chem.*, 11:361, 1990. DOI: 10.1002/jcc.540110311.

[32] A. Broo and S. Larsson. *Chem. Phys.*, 148:103, 1990. DOI: 10.1016/0301-0104(90)89011-E.

[33] S. P. Brown, T. Schaller, U. P. Seelbach, F. Koziol, C. Ochsenfeld, F.-G. Klärner, and H. W. Spiess. *Angew. Chem. Int. Ed. Engl.*, 40:717, 2001. DOI: 10.1002/1521-3773(20010216)40:4<717::AID-ANIE7170>3.0.CO;2-X.

[34] P. Bultinck, C. Van Alsenoy, P. W. Ayers, and R. Carbó-Dorca. *J. Chem. Phys.*, 126:144111, 2007. DOI: 10.1063/1.2715563.

[35] R. Burcl, N. C. Handy, and S. Carter. *Spectrochim. Acta A*, 59:1881, 2003. DOI: 10.1016/S1386-1425(02)00421-3.

[36] T. Carrington, Jr. Vibrational energy level calculations. In P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, and P. R. Schreiner, editors, *Encyclopedia of Computational Chemistry*, volume 5, pages 3157–3166. Wiley, Chichester, 1998.

[37] R. Casasnovas, J. Ortega-Castro, J. Frau, J. Donoso, and F. Muñoz. *Int. J. Quantum Chem.*, 114:1350, 2014. DOI: 10.1002/qua.24699.

[38] R. J. Cave and M. D. Newton. *Chem. Phys. Lett.*, 249:15, 1996. DOI: 10.1016/0009-2614(95)01310-5.

[39] L. S. Cederbaum and W. Domcke. *J. Chem. Phys.*, 64:603, 1976. DOI: 10.1063/1.432250.

[40] A. Cembran, L. Song, Y. Mo, and J. Gao. *J. Chem. Theory Comput.*, 5:2702, 2009. DOI: 10.1021/ct9002898.

[41] J.-D. Chai and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 10:6615, 2008. DOI: 10.1039/b810189b.

[42] P. M. Champion and A. C. Albrecht. *Annu. Rev. Phys. Chem.*, 33:353, 1982. DOI: 10.1146/annurev.pc.33.100182.002033.

[43] X. Cheng and R. P. Steele. *J. Chem. Phys.*, 141:104105, 2014. DOI: 10.1063/1.4894507.

[44] D. M. Chipman. *Theor. Chem. Acc.*, 76:73, 1989. DOI: 10.1007/BF00532125.

[45] D. A. Clabo, W. D. Allen, R. B. Remington, Y. Yamaguchi, and H. F. Schaefer III. *Chem. Phys.*, 123:187, 1988. DOI: 10.1016/0301-0104(88)87271-9.

[46] D. Claudino and N. J. Mayhall. *J. Chem. Theory Comput.*, 15:1053, 2019. DOI: 10.1021/acs.jctc.8b01112.

[47] J. Contreras-García, E. R. Johnson, S. Keinan, B. Chaudret, J.-P. Piquemal, D. N. Beratan, and W. Yang. *J. Chem. Theory Comput.*, 7:625, 2011. DOI: 10.1021/ct100641a.

[48] W. D. Cornell, P. Cieplak, C. I. Bayly, and P. A. Kollman. *J. Am. Chem. Soc.*, 115:9620, 1993. DOI: 10.1021/ja00074a030.

[49] C. J. Cramer. *Essentials of Computational Chemistry: Theories and Models*. Wiley, Chichester, West Sussex, England ; Hoboken, NJ, 2004.

[50] S. Dasgupta, B. Rana, and J. M. Herbert. *J. Phys. Chem. B*, 123:8074, 2019. DOI: 10.1021/acs.jpcb.9b04895.

[51] M. de Wergifosse, V. Liégeois, and B. Champagne. *Int. J. Quantum Chem.*, 114:900, 2014. DOI: 10.1002/qua.24685.

[52] R. Ditchfield. *Mol. Phys.*, 27:789, 1974. DOI: 10.1080/00268977400100711.

[53] S. A. do Monte, R. F. K. Spada, R. L. R. Alves, L. Belcher, R. Shepard ad H. Lischka, and F. Plasser. *J. Phys. Chem. A*, 127:9842, 2023. DOI: 10.1021/acs.jpca.3c05559.

[54] A. Dreuw and M. Head-Gordon. *Chem. Rev.*, 105:4009, 2005. DOI: 10.1021/cr0505627.

[55] F. Duschinsky. *Acta Physicochim. URSS*, 7:551, 1937.

[56] C. Edmiston and K. Ruedenberg. *Rev. Mod. Phys.*, 35:457, 1963. DOI: 10.1103/RevModPhys.35.457.

[57] D. M. Elking, L. Perera, and L. G. Pedersen. *Comput. Phys. Commun.*, 183:390, 2012. DOI: 10.1016/j.cpc.2011.10.003.

[58] E. Epifanovsky, K. Klein, S. Stopkowicz, J. Gauss, and A. I. Krylov. *J. Chem. Phys.*, 143:064102, 2015. DOI: 10.1063/1.4927785.

[59] A. Farazdel, M. Dupuis, E. Clementi, and A. Aviram. *J. Am. Chem. Soc.*, 112:4206, 1990. DOI: 10.1021/ja00167a016.

[60] M. M. Francl and L. E. Chirlian. The pluses and minuses of mapping atomic charges to electrostatic potentials. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 14, chapter 1, pages 1–32. Wiley-VCH, New York, 2000. DOI: 10.1002/9780470125915.ch1.

[61] P. Fuentealba, E. Chamorro, and J. C. Santos. Understanding and using the electron localization function. In A. Toro-Labbé, editor, *Theoretical Aspects of Chemical Reactivity*, volume 19 of *Theoretical and Computational Chemistry*, chapter 5, pages 57–85. 2007. DOI: 10.1016/S1380-7323(07)80006-9.

[62] J. Gauss. *Ber. Bunsenges. Phys. Chem.*, 99:1001, 1995. DOI: 10.1002/bbpc.199500022.

[63] J. Gauss, M. Kállay, and F. Neese. *J. Phys. Chem. A*, 113:11541–11549, 2009. DOI: 10.1021/jp9028535.

[64] S. Ghan, C. Kunkel, K. Reuter, and H. Oberhofer. *J. Chem. Theory Comput.*, 16:7431, 2020. DOI: 10.1021/acs.jctc.0c00887.

[65] A. Ghysels, V. Van Speybroeck, E. Pauwels, S. Catak, B. R. Brooks, D. Van Neck, and M. Waroquier. *J. Comput. Chem.*, 31:994, 2010. DOI: 10.1002/jcc.21386.

[66] E. Gianinetti, M. Raimondi, and E. Tornaghi. *Int. J. Quantum Chem.*, 60:157, 1996. DOI: 10.1002/(SICI)1097-461X(1996)60:1<157::AID-QUA17>3.0.CO;2-C.

[67] M. Gimferrer, J. Van der Mynsbrugge, A. T. Bell, P. Salvador, and M. Head-Gordon. *Inorg. Chem.*, 59:15410, 2020. DOI: 10.1021/acs.inorgchem.0c02405.

[68] M. Gimferrer, A. Aldossary, P. Salvador, and M. Head-Gordon. *J. Chem. Theory Comput.*, 18:309, 2022. DOI: 10.1021/acs.jctc.1c01011.

[69] E. D. Glendening, C. R. Landis, and F. Weinhold. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2:1, 2012. DOI: 10.1002/wcms.51.

[70] E. D. Glendening, C. R. Landis, and F. Weinhold. *J. Comput. Chem.*, 40:2234, 2019. DOI: 10.1002/jcc.25873.

[71] S. Grimme. *Chem. Eur. J*, 18:9955, 2012. DOI: 10.1002/chem.201200497.

[72] J. Guthmuller. *J. Chem. Phys.*, 144:064106, 2016. DOI: 10.1063/1.4941449.

[73] J. Guthmuller. Calculation of vibrational resonance Raman spectra of molecules using quantum chemistry methods. In Y. Ozaki, M. J. Wójcik, and J. Popp, editors, *Molecular Spectroscopy: A Quantum Chemistry Approach*, volume 1, chapter 17, pages 497–536. Wiley-VCH, Weinheim, 2019. DOI: 10.1002/9783527814596.ch17.

[74] M. W. D. Hanson-Heine. *J. Chem. Phys.*, 143:164104, 2015. DOI: 10.1063/1.4934234.

[75] M. W. D. Hanson-Heine, M. W. George, and N. A. Besley. *J. Chem. Phys.*, 136:224102, 2012. DOI: 10.1063/1.4727853.

[76] M. W. D. Hanson-Heine, F. S. Husseini, J. D. Hirst, and N. A. Besley. *J. Chem. Theory Comput.*, 12:1905, 2016. DOI: 10.1021/acs.jctc.5b01198.

[77] M. Häser, R. Ahlrichs, H. P. Baron, P. Weiss, and H. Horn. *Theor. Chem. Acc.*, 83:455, 1992. DOI: 10.1007/BF01113068.

[78] M. Head-Gordon. *Chem. Phys. Lett.*, 372:508, 2003. DOI: 10.1016/S0009-2614(03)00422-6.

[79] T. Helgaker and M. Jaszuński K. Ruud. *Chem. Rev.*, 99:293, 1990. DOI: 10.1021/cr960017t.

[80] T. Helgaker, M. Watson, and N. C. Handy. *J. Chem. Phys.*, 113:9402, 2000. DOI: 10.1063/1.1321296.

[81] E. J. Heller, R. L. Sundberg, and D. Tannor. *J. Phys. Chem.*, 86:1822, 1982. DOI: 10.1021/j100207a018.

[82] J. M. Herbert. The quantum chemistry of loosely-bound electrons. In A. L. Parill and K. Lipkowitz, editors, *Reviews in Computational Chemistry*, volume 28, chapter 8, pages 391–517. Wiley, 2015. DOI: 10.1002/9781118889886.ch8.

[83] J. M. Herbert. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 11:e1519, 2021. DOI: 10.1002/wcms.1519.

[84] J. M. Herbert. Density-functional theory for electronic excited states. In C. García-Iriepa and M. Marazzi, editors, *Theoretical and Computational Photochemistry: Fundamentals, Methods, Applications and Synergy with Experimental Approaches*, chapter 3, pages 69–118. Elsevier, 2023. DOI: 10.1016/B978-0-323-91738-4.00005-1.

[85] J. M. Herbert. *Phys. Chem. Chem. Phys.*, 26:3755, 2024. DOI: 10.1039/D3CP04226J.

[86] J. M. Herbert, L. D. Jacobson, K. U. Lao, and M. A. Rohrdanz. *Phys. Chem. Chem. Phys.*, 14:7679, 2012. DOI: 10.1039/c2cp24060b.

[87] S. Hirata, M. Nooijen, and R. J. Bartlett. *Chem. Phys. Lett.*, 326:255, 2000. DOI: 10.1016/S0009-2614(00)00772-7.

[88] J. O. Hirschfelder. *J. Chem. Phys.*, 33:1462, 1960. DOI: 10.1063/1.1731427.

[89] F. L. Hirshfeld. *Theor. Chem. Acc.*, 44:129, 1977. DOI: 10.1007/BF00549096.

[90] J. Ho, A. Klamt, and M. L. Coote. *J. Phys. Chem. A*, 114:13442, 2010. DOI: 10.1021/jp107136j.

[91] Z. C. Holden, R. M. Richard, and J. M. Herbert. *J. Chem. Phys.*, 139:244108, 2013. DOI: 10.1063/1.4850655.

[92] P. R. Horn, Y. Mao, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 18:23067, 2016. DOI: 10.1039/C6CP03784D.

[93] C.-P. Hsu, Z.-Q. You, and H.-C. Chen. *J. Phys. Chem. C*, 112:1204, 2008. DOI: 10.1021/jp076512i.

[94] W. Humphrey, A. Dalke, and K. Schulten. *J. Molec. Graphics*, 14:33, 1996. DOI: 10.1016/0263-7855(96)00018-5.

[95] C. R. Jacob and M. Reiher. *J. Chem. Phys.*, 130:084106, 2009. DOI: 10.1063/1.3077690.

[96] C. R. Jacob, S. Luber, and M. Reiher. *Chem. Eur. J*, 15:13491, 2009. DOI: 10.1002/chem.200901840.

[97] C. R. Jacob, S. Luber, and M. Reiher. *J. Phys. Chem. B*, 113:6558, 2009. DOI: 10.1021/jp900354g.

[98] F. Jensen. *J. Chem. Theory Comput.*, 2:1360, 2006. DOI: 10.1021/ct600166u.

[99] B. G. Johnson and J. Florián. *Chem. Phys. Lett.*, 247:120, 1995. DOI: 10.1016/0009-2614(95)01186-9.

[100] B. G. Johnson, P. M. W. Gill, and J. A. Pople. *J. Chem. Phys.*, 98:5612, 1993. DOI: 10.1063/1.464906.

[101] E. R. Johnson, S. Keinan, P. Mori-Sánchez, J. Contreras-García, A. J. Cohen, and W. Yang. *J. Am. Chem. Soc.*, 132:6498, 2010. DOI: 10.1021/ja100936w.

[102] P. Jørgensen, H. J. A. Jensen, and J. Olsen. *J. Chem. Phys.*, 89:3654, 1988. DOI: 10.1063/1.454885.

[103] K. A. Kane and L. Jensen. *J. Phys. Chem. C*, 114:5540, 2010. DOI: 10.1021/jp906152q.

[104] S. P. Karna and M. Dupuis. *J. Comput. Chem.*, 12:487, 1991. DOI: 10.1002/jcc.540120409.

[105] T. Kato. *Commun. Pure Appl. Math.*, 10:151, 1957. DOI: 10.1002/cpa.3160100201.

[106] A. M. Kelley. *J. Phys. Chem. A*, 112:11975, 2008. DOI: 10.1021/jp805530y.

[107] R. Z. Khaliullin, M. Head-Gordon, and A. T. Bell. *J. Chem. Phys.*, 124:204105, 2006. DOI: 10.1063/1.2191500.

[108] R. Z. Khaliullin, E. A. Cobar, R. C. Lochan, A. T. Bell, and M. Head-Gordon. *J. Phys. Chem. A*, 111:8753, 2007. DOI: 10.1021/jp073685z.

[109] P. Kimber and F. Plasser. *Phys. Chem. Chem. Phys.*, 22:6058, 2020. DOI: 10.1039/D0CP00369G.

[110] P. Kimber and F. Plasser. *J. Chem. Theory Comput.*, 19:2340, 2023. DOI: 10.1021/acs.jctc.3c00125.

[111] P. Kimber and F. Plasser. Classification and analysis of molecular excited states. In M. Yáñeez and R. J. Boyd, editors, *Comprehensive Computational Chemistry*, volume 4, pages 55–83. Elsevier, Oxford, first edition, 2024. DOI: 10.1016/B978-0-12-821978-2.00053-2.

[112] H. F. King, R. E. Stanton, H. Kim, R. E. Wyatt, and R. G. Parr. *J. Chem. Phys.*, 47:1936, 1967. DOI: 10.1063/1.1712221.

[113] G. Knizia. *J. Chem. Theory Comput.*, 9:4834, 2013. DOI: 10.1021/ct400687b.

[114] M. Kohout and A. Savin. *Int. J. Quantum Chem.*, 60:875, 1996. DOI: 10.1002/(SICI)1097-461X(1996)60:4<875::AID-QUA10>3.0.CO;2-4.

[115] I. Kondov, M. Čížek, C. Benesch, H. Wang, and M. Thoss. *J. Phys. Chem. C*, 111:11970, 2007. DOI: 10.1021/jp072217m.

[116] P. P. Korambath, J. Kong, T. R. Furlani, and M. Head-Gordon. *Mol. Phys.*, 100:1755, 2002. DOI: 10.1080/00268970110109466.

[117] K. Y. Kue, G. C. Claudio, and C.-P. Hsu. *J. Chem. Theory Comput.*, 14:1304, 2018. DOI: 10.1021/acs.jctc.7b01103.

[118] J. Kussmann and C. Ochsenfeld. *J. Chem. Phys.*, 127:054103, 2007. DOI: 10.1063/1.2749509.

[119] J. Kussmann and C. Ochsenfeld. *J. Chem. Phys.*, 127:204103, 2007. DOI: 10.1063/1.2794033.

[120] Y.-P. Li, J. Gomes, S. M. Sharada, A. T. Bell, and M. Head-Gordon. *J. Phys. Chem. C*, 119:1840, 2015. DOI: 10.1021/jp509921r.

[121] W. Liang, Y. Zhao, J. Sun, J. Song, S. Hu, and J. Yang. *J. Phys. Chem. B*, 110:9908, 2006. DOI: 10.1021/jp0572481.

[122] W. Liang, H. Ma, H. Zang, and C. Ye. *Int. J. Quantum Chem.*, 115:550, 2015. DOI: 10.1002/qua.24824.

[123] C. Y. Lin, A. T. B. Gilbert, and P. M. W. Gill. *Theor. Chem. Acc.*, 120:23, 2008. DOI: 10.1007/s00214-007-0292-8.

[124] H.-H. Lin, K. Y. Kue, G. C. Claudio, and C.-P. Hsu. *J. Chem. Theory Comput.*, 15:2246, 2019. DOI: 10.1021/acs.jctc.8b01185.

[125] S. H. Lin and H. Eyring. *Proc. Natl. Acad. Sci. USA*, 71:3802, 1974. DOI: 10.1073/pnas.71.10.3802.

[126] J. Liu and J. M. Herbert. *J. Chem. Phys.*, 143:034106, 2015. DOI: 10.1063/1.4926837.

[127] D. A. Long. *The Raman Effect: A Unified Treatment of the Theory of Raman Scattering By Molecules*. Wiley, 2002.

[128] P.-O. Löwdin. *J. Chem. Phys.*, 18:365, 1950. DOI: 10.1063/1.1747632.

[129] H. Ma, J. Liu, and W. Liang. *J. Chem. Theory Comput.*, 8:4474, 2012. DOI: 10.1021/ct300640c.

[130] H. Ma, Y. Zhao, and W. Liang. *J. Chem. Phys.*, 140:094107, 2014. DOI: 10.1063/1.4867273.

[131] A. Manjanath, C.-H. Yang, K. Kue, C.-I. Wang, G. C. Claudio, and C.-P. Hsu. *J. Chem. Theory Comput.*, 18: 1017, 2022. DOI: 10.1021/acs.jctc.1c00868.

[132] Y. Mao, P. R. Horn, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 19:5944, 2017. DOI: 10.1039/C6CP08039A.

[133] Y. Mao, A. Montoya-Castillo, and T. E. Markland. *J. Chem. Phys.*, 151:164114, 2019. DOI: 10.1063/1.5125275.

[134] Y. Mao, A. Montoya-Castillo, and T. E. Markland. *J. Chem. Phys.*, 153:244111, 2020. DOI: 10.1063/5.0035593.

[135] A. V. Marenich, S. V. Jerome, C. J. Cramer, and D. G. Truhlar. *J. Chem. Theory Comput.*, 8:527, 2012. DOI: 10.1021/ct200866d.

[136] R. L. Martin. *J. Chem. Phys.*, 118:4775, 2003. DOI: 10.1063/1.1558471.

[137] S. N. Maximoff and G. E. Scuseria. *Chem. Phys. Lett.*, 390:408, 2004. DOI: 10.1016/j.cplett.2004.04.049.

[138] D. A. McQuarrie and J. D. Simon. *Physical Chemistry: A Molecular Approach*. University Science Books, Sausalito, 1997.

[139] S. A. Mewes, F. Plasser, and A. Dreuw. *J. Chem. Phys.*, 143:171101, 2015. DOI: 10.1063/1.4935178.

[140] S. A. Mewes, J.-M. Mewes, A. Dreuw, and F. Plasser. *Phys. Chem. Chem. Phys.*, 18:2548, 2016. DOI: 10.1039/C5CP07077E.

[141] A. Miani, E. Cancès, P. Palmieri, A. Trombetti, and N. C. Handy. *J. Chem. Phys.*, 112:248, 2000. DOI: 10.1063/1.480577.

[142] I. M. Mills. Vibration-rotation structure in asymmetric and symmetric-top molecules. In K. N. Rao and C. W. Mathews, editors, *Molecular Spectroscopy: Modern Research*, chapter 3.2, pages 115–140. Academic Press, New York, 1972. DOI: 10.1016/B978-0-12-580640-4.50013-3.

[143] C. W. Murray, G. J. Laming, N. C. Handy, and R. D. Amos. *Chem. Phys. Lett.*, 199:551, 1992. DOI: 10.1016/0009-2614(92)85008-X.

[144] A. B. Myers. *Chem. Rev.*, 96:911, 1996. DOI: 10.1021/cr950249c.

[145] A. B. Myers. *Acc. Chem. Res.*, 30:519, 1997. DOI: 10.1021/ar960240c.

[146] K. Nanda and A. I. Krylov. *J. Chem. Phys.*, 145:204116, 2016. DOI: 10.1063/1.4967860.

[147] J. Neugebauer and B. A. Hess. *J. Chem. Phys.*, 118:7215, 2003. DOI: 10.1063/1.1561045.

[148] M. D. Newton. *Chem. Rev.*, 91:767, 1991. DOI: 10.1021/cr00005a007.

[149] H. H. Nielsen. *Phys. Rev.*, 60:794, 1941. DOI: 10.1103/PhysRev.60.794.

[150] H. H. Nielsen. *Rev. Mod. Phys.*, 23:90, 1951. DOI: 10.1103/RevModPhys.23.90.

[151] H. Oberhofer and J. Blumberger. *Phys. Chem. Chem. Phys.*, 14:13846, 2012. DOI: 10.1039/C2CP41348E.

[152] C. Ochsenfeld. *Chem. Phys. Lett.*, 327:216, 2000. DOI: 10.1016/S0009-2614(00)00865-4.

[153] C. Ochsenfeld. *Phys. Chem. Chem. Phys.*, 2:2153, 2000. DOI: 10.1039/b000174k.

[154] C. Ochsenfeld, C. A. White, and M. Head-Gordon. *J. Chem. Phys.*, 109:1663, 1998. DOI: 10.1063/1.476741.

[155] C. Ochsenfeld, S. P. Brown, I. Schnell, J. Gauss, and H. W. Spiess. *J. Am. Chem. Soc.*, 123:2597, 2001. DOI: 10.1021/ja0021823.

[156] C. Ochsenfeld, F. Koziol, S. P. Brown, T. Schaller, U. P. Seelbach, and F.-G. Klärner. *Solid State Nucl. Mag.*, 22: 128, 2002. DOI: 10.1006/snmr.2002.0085.

[157] C. Ochsenfeld, J. Kussmann, and F. Koziol. *Angew. Chem.*, 116:4585, 2004. DOI: 10.1002/ange.200460336.

[158] K. Ohta, G. L. Closs, K. Morokuma, and N. J. Green. *J. Am. Chem. Soc.*, 108:1319, 1986. DOI: 10.1021/ja00266a045.

[159] J. Olsen, D. L. Yeager, and P. Jørgensen. *J. Chem. Phys.*, 91:381, 1989. DOI: 10.1063/1.457471.

[160] G. Orlandi and W. Siebrand. *J. Chem. Phys.*, 58:4513, 1973. DOI: 10.1063/1.1679014.

[161] R. T. Pack and W. B. Brown. *J. Chem. Phys.*, 45:556, 1966. DOI: 10.1063/1.1727605.

[162] P. T. Panek and C. R. Jacob. *ChemPhysChem*, 15:3365, 2014. DOI: 10.1002/cphc.201402251.

[163] M. Pavanello, T. Van Voorhis, L. Visscher, and J. Neugebauer. *J. Chem. Phys.*, 138:054101, 2013. DOI: 10.1063/1.4789418.

[164] A. Pedone, M. Biczysko, and V. Barone. *ChemPhysChem*, 11:1812, 2010. DOI: 10.1002/cphc.200900976.

[165] Z. Pei, Q. Ou, Y. Mao, J. Yang, A. de la Lande, F. Plasser W. Liang, Z. Shuai, and Y. Shao. *J. Phys. Chem. Lett.*, 12:2712, 2021. DOI: 10.1021/acs.jpclett.1c00094.

[166] T. Petrenko and F. Neese. *J. Chem. Phys.*, 127:164319, 2007. DOI: 10.1063/1.2770706.

[167] S. D. Peyerimhoff. Spectroscopy: Computational methods. In P. v. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, and P. R. Schreiner, editors, *Encyclopedia of Computational Chemistry*, volume 4, pages 2646–2664. Wiley, Chichester, 1998.

[168] J. Pipek and P. G. Mezey. *J. Chem. Phys.*, 90:4916, 1989. DOI: 10.1063/1.456588.

[169] F. Plasser. *J. Chem. Phys.*, 144:194107, 2016. DOI: 10.1063/1.4949535.

[170] F. Plasser and H. Lischka. *J. Chem. Theory Comput.*, 8:2777, 2012. DOI: 10.1021/ct300307c.

[171] F. Plasser, S. A. Bäppler, M. Wormit, and A. Dreuw. *J. Chem. Phys.*, 141:024107, 2014. DOI: 10.1063/1.4885820.

[172] F. Plasser, M. Wormit, and A. Dreuw. *J. Chem. Phys.*, 141:024106, 2014. DOI: 10.1063/1.4885819.

[173] F. Plasser, B. Thomitzni, S. A. Bäppler, J. Wenzel, D. R. Rehn, M. Wormit, and A. Dreuw. *J. Comput. Chem.*, 36:1609, 2015. DOI: 10.1002/jcc.23975.

[174] F. Plasser, A. I. Krylov, and A. Dreuw. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 12:e1595, 2022. DOI: 10.1002/wcms.1595.

[175] E. Proynov. *J. Mol. Struct. (Theochem)*, 762:159, 2006. DOI: 10.1016/j.theochem.2005.08.037.

[176] E. Proynov, Y. Shao, and J. Kong. *Chem. Phys. Lett.*, 493:381, 2010. DOI: 10.1016/j.cplett.2010.05.029.

[177] E. Proynov, F. Liu, Y. Shao, and J. Kong. *J. Chem. Phys.*, 136:034102, 2012. DOI: 10.1063/1.3676726.

[178] E. Proynov, F. Liu, and J. Kong. *Phys. Rev. A*, 88:032510, 2013. DOI: 10.1103/PhysRevA.88.032510.

[179] V. A. Rassolov and D. M. Chipman. *J. Chem. Phys.*, 104:9908, 1996. DOI: 10.1063/1.471719.

[180] V. A. Rassolov and D. M. Chipman. *J. Chem. Phys.*, 105:1470, 1996. DOI: 10.1063/1.472009.

[181] V. A. Rassolov and D. M. Chipman. *J. Chem. Phys.*, 105:1479, 1996. DOI: 10.1063/1.472010.

[182] H. Ren, M. R. Provorse, P. Bao, Z. Qu, and J. Gao. *J. Phys. Chem. Lett.*, 7:2286, 2016. DOI: 10.1021/acs.jpclett.6b00915.

[183] Y. M. Rhee and M. Head-Gordon. *J. Am. Chem. Soc.*, 130:3878, 2008. DOI: 10.1021/ja0764916.

[184] R. F. Ribeiro, A. V. Marenich, C. J. Cramer, and D. G. Truhlar. *J. Phys. Chem. B*, 115:14556, 2011. DOI: 10.1021/jp205508z.

[185] R. M. Richard and J. M. Herbert. *J. Chem. Theory Comput.*, 7:1296, 2011. DOI: 10.1021/ct100607w.

[186] G. Schaftenaar and J. H. Noordik. *J. Comput.-Aided Mol. Design*, 14:123, 2000. DOI: 10.1023/A:1008193805436.

[187] C. Schober, K. Reuter, and H. Oberhofer. *J. Chem. Phys.*, 144:054103, 2016. DOI: 10.1063/1.4940920.

[188] A. P. Scott and L. Radom. *J. Phys. Chem.*, 100:16502, 1996. DOI: 10.1021/jp960976r.

[189] H. Sekino and R. J. Bartlett. *J. Chem. Phys.*, 85:976, 1986. DOI: 10.1063/1.451255.

[190] K. Senthilkumar, F. C. Grozema, F. M. Bickelhaupt, and L. D. A. Siebbeles. *J. Chem. Phys.*, 119:9809, 2003. DOI: 10.1063/1.1615476.

[191] T. E. Sharp and H. M. Rosenstock. *J. Chem. Phys.*, 41:3453, 1964. DOI: 10.1063/1.1725748.

[192] A. C. Simmonett, A. T. B. Gilbert, and P. M. W. Gill. *Mol. Phys.*, 103:2789, 2005. DOI: 10.1080/00268970500187910.

[193] U. C. Singh and P. A. Kollman. *J. Comput. Chem.*, 5:129, 1984. DOI: 10.1002/jcc.540050204.

[194] P. Sjoberg, J. S. Murray, T. Brinck, and P. Politzer. *Can. J. Chem.*, 68:1440, 1990. DOI: 10.1139/v90-220.

[195] G. J. Small. *J. Chem. Phys.*, 54:3300, 1971. DOI: 10.1063/1.1675343.

[196] W. A. Sokalski and R. A. Poirier. *Chem. Phys. Lett.*, 98:86, 1983. DOI: 10.1016/0009-2614(83)80208-5.

[197] T. G. Spiro and P. Stein. *Annu. Rev. Phys. Chem.*, 28:501, 1977. DOI: 10.1146/annurev.pc.28.100177.002441.

[198] V. N. Staroverov and E. R. Davidson. *Chem. Phys. Lett.*, 330:161, 2000. DOI: 10.1016/S0009-2614(00)01088-5.

[199] S. N. Steinmann and C. Corminbeoeuf. *J. Chem. Theory Comput.*, 6:1990, 2010. DOI: 10.1021/ct1001494.

[200] A. J. Sterling, D. S. Levine, A. Aldossary, and M. Head-Gordon. *J. Am. Chem. Soc.*, 146:9532, 2024. DOI: 10.1021/jacs.3c10633.

[201] H. Stoll, G. Wagenblast, and H. Preuss. *Theor. Chem. Acc.*, 57:169, 1980. DOI: 10.1007/BF00574903.

[202] J. E. Subotnik, Y. Shao, W. Liang, and M. Head-Gordon. *J. Chem. Phys.*, 121:9220, 2004. DOI: 10.1063/1.1790971.

[203] J. E. Subotnik, S. Yeganeh, R. J. Cave, and M. A. Ratner. *J. Chem. Phys.*, 129:244101, 2008. DOI: 10.1063/1.3042233.

[204] J. E. Subotnik, R. J. Cave, R. P. Steele, and N. Shenvi. *J. Chem. Phys.*, 130:234102, 2009. DOI: 10.1063/1.3148777.

[205] J. E. Subotnik, J. Vura-Weis, A. Sodt, and M. A. Ratner. *J. Phys. Chem. A*, 114:8665, 2010. DOI: 10.1021/jp101235a.

[206] V. Sychrovský, J. Gräfenstein, and D. Cremer. *J. Chem. Phys.*, 113:3530, 2000. DOI: 10.1063/1.1286806.

[207] A. Szabo and N. S. Ostlund. *Modern Quantum Chemistry*. Dover, 1996.

[208] A. A. Taka, J. M. Herbert, and L. M. McCaslin. *J. Phys. Chem. Lett.*, 14:11063, 2023. DOI: 10.1021/acs.jpclett.3c02787.

[209] K. Takatsuka, T. Fueno, and K. Yamaguchi. *Theor. Chem. Acc.*, 48:175, 1978. DOI: 10.1007/BF00549017.

[210] J. Tang and A. C. Albrecht. Developments in the theories of vibrational Raman intensities. In H. A. Szymanski, editor, *Raman Spectroscopy: Theory and Practice*, volume 2, pages 33–68. Plenum Press, New York, 1970. DOI: 10.1007/978-1-4684-3027-1_2.

[211] A. J. W. Thom and M. Head-Gordon. *J. Chem. Phys.*, 131:124113, 2009. DOI: 10.1063/1.3236841.

[212] A. J. W. Thom, E. J. Sundstrom, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 11:11297, 2009. DOI: 10.1039/b915364k.

[213] P. v. R. Schleyer, C. Maerker, A. Dransfield, H. Jiao, and N. J. R. van Eikema Hommes. *J. Am. Chem. Soc.*, 118: 6317, 1996. DOI: 10.1021/ja960582d.

[214] E. F. Valeev, V. Coropceanu, D. A. da Silva Filho, S. Salman, and J.-L. Brédas. *J. Am. Chem. Soc.*, 128:9882, 2006. DOI: 10.1021/ja061827h.

[215] D. E. P. Vanpoucke, P. Bultinck, and I. Van Driessche. *J. Comput. Chem.*, 34:405, 2013. DOI: 10.1002/jcc.23088.

[216] A. A. Voityuk and N. Rösch. *J. Chem. Phys.*, 117:5607, 2002. DOI: 10.1063/1.1502255.

[217] J. Vura-Weis, M. Wasielewski, M. D. Newton, and J. E. Subotnik. *J. Phys. Chem. C*, 114:20449, 2010. DOI: 10.1021/jp104783r.

[218] M. Wächtler, J. Guthmuller, L. González, and B. Dietzek. *Coord. Chem. Rev.*, 256:1479, 2012. DOI: 10.1016/j.ccr.2012.02.004.

[219] B. Wang, J. Baker, and P. Pulay. *Phys. Chem. Chem. Phys.*, 2:2131, 2000. DOI: 10.1039/b000026o.

[220] A. Warshel. *Annu. Rev. Biophys. Bioeng.*, 6:273, 1977. DOI: 10.1146/annurev.bb.06.060177.001421.

[221] R. C. Weast, editor. *CRC Handbook of Chemistry and Physics*. Chemical Rubber Company, Boca Rotan, 63rd edition, 1982.

[222] F. Weinhold. Natural bond orbital analysis of photochemical excitation, with illustrative applications to vinoxy radical. In A. G. Kutateladze, editor, *Computational Methods in Photochemistry*, volume 13 of *Molecular and Supramolecular Photochemistry*, chapter 7, pages 393–476. Taylor & Francis, New York, 2005.

[223] F. Weinhold, C. R. Landis, and E. D. Glendening. *Int. Rev. Phys. Chem.*, 35:399, 2016. DOI: 10.1080/0144235X.2016.1192262.

[224] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon. *Chem. Phys. Lett.*, 230:8, 1994. DOI: 10.1016/0009-2614(94)01128-1.

[225] R. J. Whitehead and N. C. Handy. *J. Mol. Spect.*, 55:356, 1975. DOI: 10.1016/0022-2852(75)90274-X.

[226] C. F. Williams and J. M. Herbert. *J. Phys. Chem. A*, 112:6171, 2008. DOI: 10.1021/jp802272r.

[227] E. B. Wilson and J. J. B. Howard. *J. Chem. Phys.*, 4:260, 1936. DOI: 10.1063/1.1749833.

[228] K. Wolinski, J. F. Hinton, and P. Pulay. *J. Am. Chem. Soc.*, 112:8251, 1990. DOI: 10.1021/ja00179a005.

[229] H. L. Woodcock, W. Zheng, A. Ghysels, Y. Shao, J. Kong, and B. R. Brooks. *J. Chem. Phys.*, 129:214109, 2008. DOI: 10.1063/1.3013558.

[230] Q. Wu and T. Van Voorhis. *J. Chem. Phys.*, 125:164105, 2006. DOI: 10.1063/1.2360263.

[231] K. Yagi, K. Hirao, T. Taketsuga, M. W. Schmidt, and M. S. Gordon. *J. Chem. Phys.*, 121:1383, 2004. DOI: 10.1063/1.1764501.

[232] C.-H. Yang, C. Y. Yam, and H. Wang. *Phys. Chem. Chem. Phys.*, 20:2571, 2018. DOI: 10.1039/c7cp06660k.

[233] Z.-Q. You and C.-P. Hsu. *J. Chem. Phys.*, 133:074105, 2010. DOI: 10.1063/1.3467882.

[234] Z.-Q. You, Y.-C. Hung, and C.-P. Hsu. *J. Phys. Chem. B*, 119:7480, 2015. DOI: 10.1021/jp511216c.

[235] L. Y. Zhang, R. A. Friesner, and R. B. Murphy. *J. Chem. Phys.*, 107:450, 1997. DOI: 10.1063/1.474406.

[236] C. Zheng, Y. Mao, J. Kozuch, A. O. Atsango, Z. Ji, T. E. Markland, and S. G. Boxer. *Nat. Chem.*, 14:891, 2022. DOI: 10.1038/s41557-022-00937-w.

[237] W. J. Zheng and B. R. Brooks. *Biophys. J.*, 88:3109, 2005. DOI: 10.1529/biophysj.104.058453.

Chapter 11

Molecules in Complex Environments: Solvent Models, QM/MM, QM/EFP, and Embedding Methods

11.1 Introduction

Q-CHEM has incorporated a number of methods for complex systems such as molecules in solutions, proteins, polymers, molecular clusters, *etc.*, summarized as follows:

- Implicit solvation models;
- QM/MM tools;
- EFP and QM/EFP approach (polarizable electrostatic embedding); and
- Density embedding methods.

11.2 Chemical Solvent Models

11.2.1 Introduction

Ab initio quantum chemistry makes possible the study of gas-phase molecular properties from first principles. In liquid solution, however, these properties may change significantly, especially in polar solvents. Although it is possible to model solvation effects by including explicit solvent molecules in the quantum-chemical calculation (*e.g.*, a supramolecular cluster calculation, averaged over different configurations of the molecules in the first solvation shell), such calculations are very computationally demanding. Furthermore, cluster calculations typically do not afford accurate solvation energies, owing to the importance of long-range electrostatic interactions. (Hybrid discrete/continuum models, which contain *some* explicit solvent, can be quite effective, however.[124,127]) Accurate prediction of solvation free energies is crucial for modeling of chemical reactions but also for relative conformational energies in solution.

Q-CHEM contains several different implicit solvent models, which differ greatly in their level of sophistication. These are generally known as self-consistent reaction field (SCRF) models, because the continuum solvent establishes a "reaction field" (additional terms in the solute Hamiltonian) that depends upon the solute electron density, and must

therefore be updated self-consistently during the iterative convergence of the wave function. The simplest and oldest of these models that is available in Q-CHEM is the multipole expansion method, also known as a Kirkwood-Onsager model,[62] in which the solute molecule is placed inside of a spherical cavity and its electrostatic potential is represented in terms of a single-center multipole expansion. (This should not be confused with the *Onsager model*, in which a dipole approximation is used, and to avoid confusion the *multipolar expansion method* is better terminology.) More sophisticated models, which use a molecule-shaped cavity and the full molecular electrostatic potential, include the conductor-like PCM (C-PCM),[6,41] the conductor-like screening model (COSMO),[81] the "surface and simulation of volume polarization for electrostatics" [SS(V)PE] model,[29] and the closely-related "integral equation formalism" (IEF-PCM).[21,22] For an overview of all of these methods and their interconnections, see the review by Herbert.[62]

The C-PCM, IEF-PCM, and SS(V)PE are examples of what are called "apparent surface charge" SCRF models, and the term *polarizable continuum models* (PCMs), as popularized by Tomasi and coworkers,[161] is now used almost universally to refer to this class of solvation models. Q-CHEM employs a Switching/Gaussian or "SwiG" implementation of these PCMs,[63,85,86,88,89] which resolves a long-standing (though little-publicized) problem with standard PCMs, namely, that the boundary-element methods used to discretize the solute/continuum interface may lead to discontinuities in the potential energy surface for the solute molecule. These discontinuities inhibit convergence of geometry optimizations, introduce serious artifacts in vibrational frequency calculations, and make *ab initio* molecular dynamics calculations virtually impossible.[85,86] In contrast, Q-CHEM's SwiG-PCMs afford potential energy surfaces that are rigorously continuous and smooth. Unlike earlier attempts to obtain smooth PCMs, the SwiG approach largely preserves the properties of the underlying integral-equation solvent models, so that solvation energies and molecular surface areas are hardly affected by the smoothing procedure.

Other solvent models available in Q-CHEM include the "Langevin dipoles" model;[49,50] as well as the SM8,[108] SM12,[112] and SMD models.[109] SM8 and SM12 are based upon the generalized Born method for electrostatics, augmented with atomic surface tensions intended to capture nonelectrostatic effects (cavitation, dispersion, exchange repulsion, and changes in solvent structure). These models have been carefully parameterized to reproduce experimental free energies of solvation.[43] The SMD model, in which the "D" is for "density", combines IEF-PCM with similar nonelectrostatic corrections. Statistically speaking, SMD is not any more or less accurate than other SMx models,[62] but has the advantage of being based on rigorous electrostatics derived from the solute's exact SCF density. The SM12 and SMD models can each be used in arbitrary basis sets. The SM8 model uses generalized Born electrostatics based on "CM4" charges,[77,122] which are themselves based on Löwdin atomic charges, and as such this model should only be used in the basis sets for which it was parameterized: 6-31G*, 6-31+G*, or 6-31+G**. (Other basis sets, if requested will use the 6-31G* parameters but this is not recommended.) The SM12 model also uses generalized Born electrostatics but substitutes CM4 charges for Hirshfeld charges; the latter are more stable with respect to changes in basis set and therefore SM12 is available in arbitrary basis sets. The trade-off is that an analytic gradient is available for SM8 but not SM12; see Table 11.2. For SMD, both analytic first and second derivatives (frequencies) are available.[14]

Table 11.1 summarizes the implicit solvent models that are available in Q-CHEM. Solvent models are invoked via the SOLVENT_METHOD keyword, as shown below. Additional details about each particular solvent model can be found in the sections that follow. In general, these methods are available for any SCF level of electronic structure theory, with the aforementioned caveat about basis sets for SM8. Post-Hartree–Fock calculations (such as MP2 or EOM-CC) can be performed by first running an SCF + PCM job, in which case the correlated wave function will employ MOs and Hartree-Fock energy levels that are polarized by the solvent. This represents a "zeroth-order" inclusion of the solvent effects at the correlated level of theory, but is perfectly adequate for many applications as higher-order corrections are usually small.[62,115] Table 11.2 also summarizes the availability of analytical energy gradients for implicit solvent models. (Finite-difference gradients and Hessians are requested automatically for calculations where the requisite analytic derivatives are not available.)

Note: The format for specifying implicit solvent models changed significantly starting in Q-CHEM v. 4.2.1. This change was made in an attempt to simply and unify the input notation for a large number of different models.

Model	**Cavity**		**Non-Electrostatic Terms?**	**Basis Sets Supported**
	Construction	Discretization		
Kirkwood-Onsager	spherical	point charges	no	all
Langevin Dipoles	atomic spheres (user-definable)	dipoles in 3-d space	no	all
Poisson Equation Solver	atomic spheres (user-definable)	grid in 3-d space	no	all
C-PCM	atomic spheres (user-definable)	point charges or smooth Gaussians	user-specified	all
SS(V)PE/ IEF-PCM	atomic spheres (user-definable)	point charges or smooth Gaussians	user-specified	all
COSMO	predefined atomic spheres	point charges	none	all
Isodensity SS(V)PE	isodensity contour	point charges	none	all
CMIRS	isodensity contour	point charges	automatic	all
SM8	predefined atomic spheres	N/A[a]	automatic	6-31G* 6-31+G* 6-31+G**
SM12	predefined atomic spheres	N/A[a]	automatic	all
SMD	predefined atomic spheres	point charges	automatic	all

[a] Generalized Born electrostatic model; does not require cavity construction.

Table 11.1: Summary of implicit solvation models available in Q-CHEM, indicating how the solute cavity is constructed and discretized, whether nonelectrostatic terms are (or can be) included, and which basis sets are available for use with each model.

Energy Derivatives	**Solvent Model**								
	C-PCM	SS(V)PE/ IEF-PCM	CMIRS	COSMO	SMx			PEqS	Langevin Dipoles
					SM8	SM12	SMD		
SCF energy gradient	yes[a]	yes[a]	no	yes	yes	no	yes	no	yes
SCF energy Hessian	yes[a]	no	no	yes[b]	no	no	yes[c]	no	no
CIS/TDDFT energy gradient	yes[a]	no	—	—	— unsupported —			—	—
CIS/TDDFT energy Hessian	yes[a]	no	—	—	— unsupported —			—	—
MP2 & double-hybrid[d]	—	—	—	—	— unsupported —			—	—
Coupled cluster methods[d]	—	—	—	—	— unsupported —			—	—

[a] Gradients available for vdW cavities and SAS cavity surfaces only.

[b] Hessians for COSMO with the outlying charge correction are not supported.

[c] Seminumerical Hessian, with a step size controllable via FDIFF_STEPSIZE.

[d] Gradients are not supported but single-point calculations can be performed using solvent-polarized MOs.

Table 11.2: Summary of analytic energy gradient and Hessian capabilities with implicit solvent models.

SOLVENT_METHOD

Sets the preferred solvent method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0	Do not use a solvation model.
KIRKWOOD	Use the Kirkwood-Onsager model (Section 11.2.2).
PCM	Use an apparent surface charge, polarizable continuum model (Section 11.2.3).
ISOSVP	Use the isodensity implementation of the SS(V)PE model (Section 11.2.6).
COSMO	Use COSMO (Section 11.2.8).
SM8	Use version 8 of the Cramer-Truhlar SMx model (Section 11.2.9.1).
SM12	Use version 12 of the SMx model (Section 11.2.9.2).
SMD	Use SMD (Section 11.2.9.3).
CHEM_SOL	Use the Langevin Dipoles model (Section 11.2.10).
PEQS	Use the Poisson Equation Solver (Section 11.2.11).

RECOMMENDATION:

Consult the literature (*e.g.*, Ref. 62). PCM is a collective name for a family of models and additional input options may be required in this case, in order to fully specify the model; see Section 11.2.3. Several versions of SM12 are available as well, as discussed in Section 11.2.9.2.

Before going into detail about each of these models, a few potential points of confusion warrant mention, with regards to nomenclature. First, "PCM" is used frequently (and sometimes ambiguously) to a family of models that includes COSMO, C-PCM, SS(V)PE, and IEF-PCM. The latter two models are formally equivalent at the level of integral equations,[21,62] but exhibit some some differences in their numerical implementation.[62,88,178] One or the other of these models can be selected by additional job control variables in a *$pcm* input section, as described in Section 11.2.3. COSMO is very similar to C-PCM but includes a correction for that part of the solute's electron density that penetrates beyond the cavity (the so-called "outlying charge"),[80] although later work cast doubt on the theoretical justification for this and other *ad hoc* charge renormalization procedures.[28] [The IEF-PCM and SS(V)PE methods already contain an *implicit* correction for outlying charge, as does the C-PCM method that is derived as an approximation to these models,[28,62] although this was not recognized at the time that COSMO was formulated. See Ref. 62 for a historical discussion of these developments.] In any case, COSMO is described in Section 11.2.8.

Two implementations of the SS(V)PE model are also available. The PCM implementation (which is requested by setting SOLVENT_METHOD = PCM in conjunction with appropriate job-control variables in the *$pcm* input section) uses a solute cavity constructed from atom-centered spheres, in keeping with other PCMs. On the other hand, setting SOLVENT_METHOD = ISOSVP requests an SS(V)PE calculation in which the solute cavity is defined by an isocontour of the solute's own electron density.[27,31,32] This is an appealing, one-parameter cavity construction that avoid many of the problems with cusps in "van der Waals" cavity surfaces that are constructed from atom-centered spheres, and the isodensity implementation of SS(V)PE forms the basis of a physics-based continuum solvation model called CMIRS that is described in Section 11.2.7.[128–131,177] CMIRS is competitive in accuracy (for solvation free energies) with the best-available SMx solvation models,[62] despite using far fewer fitting parameters. However, analytic energy gradients are not available for the isodensity cavity construction and therefore not available for CMIRS, whereas these gradients are available when the cavity surface is constructed from atom-centered spheres.

Regarding the accuracy of these models for solvation free energies (ΔG_{298}), the SMx models generally achieve sub-kcal/mol accuracy for neutral molecules, based on comparison to a large database of experimental values, although

average errors for ions are more like 4 kcal/mol.[43,62] (Note that the SM12 and SMD models generally do not improve too much upon SM8 in any statistical sense,[109,112] but do extend these models to arbitrary basis sets whereas SM8 is limited to a few small basis sets. Furthermore, SMD includes both gradients and frequencies.[14]) To achieve accuracy comparable to the SMx models within the PCM class of solvent models, nonelectrostatic terms must be added.[62] Among the various PCMs described above, the only one that constitutes a "black box" model for solvation energies is CMIRS, which slightly outperforms the SMx models in a statistical sense,[62,177] although it is only available for a few solvents.

The following sections provide more details regarding theory and job control for the various implicit solvent models that are available in Q-CHEM. Ref. 62 contains both formal comparisons amongst these models as well as a side-by-side comparison of the accuracy of solvation free energies.

11.2.2 Kirkwood-Onsager Multipole Expansion Method

The simplest implicit solvation model available in Q-CHEM is the multipolar expansion model,[62,79,117] which has also been called the Kirkwood-Onsager model or sometimes the generalized Kirkwood model.[62] In this approach, the solute is placed inside of a spherical cavity of radius a that is surrounded by a homogeneous dielectric medium whose dielectric constant is ε, and these constitute the only parameters of the model. The term *Onsager model* is sometimes synonymous with a point-dipole approximation for the solute's charge density, but Q-CHEM's version uses a single-center multipole expansion of the density (which can be extended to arbitrarily high order), in order to obtain an essentially exact description of the solute's electrostatic potential. The model then consists of using Kirkwood's analytic expressions for the solvation energy of each spherical harmonic function, in a spherical cavity inside of a dielectric continuum.[62]

Regarding the cavity radius a, Onsager's original suggestion is based on the molar volume V_m of the pure solute,

$$a = (3V_m/4\pi N_A)^{1/3} \tag{11.1}$$

where N_A is Avogadro's constant. This was later shown to be a poor choice in the context of modern quantum chemistry calculations.[62] It is also common to add 0.5 Å to the value of a in Eq. (11.1) in order to account for the first solvation shell,[173] or to set a equal to the maximum distance between the solute center of mass and the solute atoms, plus the relevant van der Waals radii. A third option is to set $2a$ (the cavity diameter) equal to the largest solute–solvent internuclear distance, plus the van der Waals radii of the relevant atoms. Clearly, there is quite a bit of arbitrariness in this choice and solvation energies are quite sensitive to the value of a,[62] and the PCMs that are described in Section 11.2.3 have largely made the multipolar expansion method obsolete, since the PCMs employ the exact electron density and can be used (if desired) with a spherical cavity, although the more typical choice is a molecule-shaped van der Waals cavity.

The Kirkwood-Onsager SCRF is requested by setting SOLVENT_METHOD = KIRKWOOD in the *$rem* section. Some additional options can be specified in the *$solvent* section, as described below, of which only **CavityRadius** is required. Energies and analytic gradients for the Kirkwood-Onsager solvent model are available for Hartree-Fock, DFT, and CCSD calculations. It is often advisable to perform a gas-phase calculation of the solute molecule first, which can serve as the initial guess for a subsequent Kirkwood-Onsager implicit solvent calculation. For coupled-cluster calculations using this model, one may set CC_SAVEAMPL = TRUE to retain the CC amplitudes from the gas-phase calculation, which will save some time in the subsequent solution-phase calculation.

Note: For CCSD calculations the Kirkwood-Onsager model works only with CCMAN2 = FALSE.

The following job-control options belong in the *$solvent* section, not the *$rem* section. As with other parts of the Q-CHEM input file, this input section is not case-sensitive.

CavityRadius

Sets the radius of the spherical solute cavity.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

No default.

OPTIONS:

a Cavity radius in Å.

RECOMMENDATION:

None

Dielectric

Sets the dielectric constant of the solvent continuum.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

ε Use a (dimensionless) value of ε.

RECOMMENDATION:

The default value corresponds to water at 25°C.

MultipoleOrder

Determines the order to which the multipole expansion of the solute charge density is carried out.

INPUT SECTION: *$solvent*

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

ℓ Include up to ℓth order multipoles.

RECOMMENDATION:

Use the default. The multipole expansion is usually converged by order $\ell = 15$.

Example 11.1 Multipole expansion model at the Hartree-Fock level; H_2O in acetonitrile

```
$molecule
   0 1
   O       0.00000000     0.00000000     0.11722303
   H      -0.75908339     0.00000000    -0.46889211
   H       0.75908339     0.00000000    -0.46889211
$end

$rem
   METHOD            HF
   BASIS             6-31g**
   SOLVENT_METHOD    Kirkwood
$end

$solvent
   CavityRadius      1.8    !   1.8 Angstrom Solute Radius
   Dielectric        35.9   !   Acetonitrile
   MultipoleOrder    15     !   this is the default value
$end
```

Example 11.2 Kirkwood-Onsager SCRF applied to hydrogen fluoride in water, performing a gas-phase calculation first.

```
$molecule
   0 1
   H       0.000000     0.000000    -0.862674
   F       0.000000     0.000000     0.043813
 $end

$rem
   METHOD          HF
   BASIS           6-31G*
$end

@@@

$molecule
  read
$end

$rem
   JOBTYPE           FORCE
   METHOD            HF
   BASIS             6-31G*
   SOLVENT_METHOD    KIRKWOOD
   SCF_GUESS         READ  ! read vacuum solution as a guess
$end

$solvent
   CavityRadius    2.5
$end
```

11.2.3 Polarizable Continuum Models

The multipolar expansion model is based on exact formulas for the solvation energy of a point multipole in a spherical cavity,[62,117] which is a crude approximation except (or perhaps even) for small molecules, and the Kirkwood-Onsager

Model	Literature Refs.	Matrix K	Matrix R	Scalar f_ε
COSMO[a]	81	$\mathbf{S}$	$-f_\varepsilon\mathbf{1}$	$(\varepsilon-1)/(\varepsilon+1/2)$
C-PCM	6,162	$\mathbf{S}$	$-f_\varepsilon\mathbf{1}$	$(\varepsilon-1)/\varepsilon$
IEF-PCM	21,29	$\mathbf{S}-(f_\varepsilon/2\pi)\mathbf{DAS}$	$-f_\varepsilon\left(\mathbf{1}-\frac{1}{2\pi}\mathbf{DA}\right)$	$(\varepsilon-1)/(\varepsilon+1)$
SS(V)PE	29	$\mathbf{S}-(f_\varepsilon/4\pi)\left(\mathbf{DAS}+\mathbf{SAD}^\dagger\right)$	$-f_\varepsilon\left(\mathbf{1}-\frac{1}{2\pi}\mathbf{DA}\right)$	$(\varepsilon-1)/(\varepsilon+1)$

[a] Also includes a charge renormalization correction; see Section 11.2.8.

Table 11.3: Definition of the matrices in Eq. (11.2) for the various PCMs that are available in Q-CHEM. The matrix $\mathbf{S}$ consists of Coulomb interactions between the cavity charges and $\mathbf{D}$ is the discretized version of the matrix that generates the outward-pointing normal electric field vector. (See Refs. 30,32,63 for detailed definitions.) The matrix $\mathbf{A}$ is diagonal and contains the surface areas of the cavity discretization elements, and $\mathbf{1}$ is a unit matrix.

model has been largely superseded by the more general class of "apparent surface charge" SCRF solvation models, typically known as PCMs.[62,161] These models improve upon the multipolar expansion method in two ways. Most importantly, they provide a much more realistic description of molecular shape, typically by constructing the "solute cavity" (*i.e.*, the interface between the atomistic region and the dielectric continuum) from a union of atom-centered spheres, an aspect of the model that is discussed in Section 11.2.3.2. In addition, the exact electron density of the solute (rather than a multipole expansion) is used to polarize the continuum. Electrostatic interactions between the solute and the continuum manifest as an induced charge density on the cavity surface, which is discretized into point charges for practical calculations. The surface charges are determined based upon the solute's electrostatic potential at the cavity surface, hence the surface charges and the solute wave function must be determined self-consistently.

11.2.3.1 Formal Theory and Discussion of Different Models

The PCM literature has a long history[161] and there are several different models in widespread use; connections between these models have not always been appreciated.[21,29,30,62,88] Chipman[29,30] has shown how various PCMs can be formulated within a common theoretical framework; see Ref. 62 for a review. The PCM takes the form of a set of linear equations,

$$\mathbf{Kq} = \mathbf{Rv}\ , \tag{11.2}$$

in which the induced charges q_i at the cavity surface discretization points [organized into a vector $\mathbf{q}$ in Eq. (11.2)] are computed from the values v_i of the solute's electrostatic potential at those same discretization points. The form of the matrices $\mathbf{K}$ and $\mathbf{R}$ depends upon the particular PCM in question. These matrices are given in Table 11.3 for the PCMs that are available in Q-CHEM.

The oldest PCM is the so-called D-PCM model of Tomasi and coworkers,[116] but unlike the models listed in Table 11.3, D-PCM requires explicit evaluation of the electric field normal to the cavity surface. This is undesirable, as evaluation of the electric field is both more expensive and more prone to numerical problems as compared to evaluation of the electrostatic potential. Moreover, the dependence on the electric field can be formally eliminated at the level of the integral equation whose discretized form is given in Eq. (11.2).[29] As such, D-PCM is essentially obsolete, and the PCMs available in Q-CHEM require only the evaluation of the electrostatic potential, not the electric field.

The simplest PCM that continues to enjoy widespread use is the conductor-like model, C-PCM.[6,41] Originally derived by Klamt and Schüürmann[81] based on arguments invoking the conductor limit ($\varepsilon \to \infty$), this model can also be derived as an approximation to more formally correct models.[28,62,87] Over the years, the dielectric-dependent factor

$$f_\varepsilon = \frac{\varepsilon - 1}{\varepsilon + x} \tag{11.3}$$

that appears in this model (see Table 11.3) has been used with different values of x. The value $x = 0$ is typically used in C-PCM calculations and $x = 1/2$ in COSMO calculations, although Klamt and co-workers later suggested using $x = 1/2$ for neutral solutes and $x = 0$ for ions.[83] The specific choice of f_ε is controllable via the *$pcm* input section that is described in Section 11.2.4.

Whereas from Table 11.3 the C-PCM and COSMO methods would appear to be the same up to a minor rescaling of the surface charge (*i.e.*, up to the precise choice of f_ε), historically the term "COSMO" has been used by Klamt to mean a particular "dual-cavity" implementation of this model that makes it different from other PCMs.[62] This construction is equivalent to the "outlying charge correction" that is discussed in Section 11.2.8, and was intended to account for the effects of the tail of the solute's charge density that penetrates beyond the cavity surface.[80] Subsequent work cast considerable doubt on the theoretical justification for this correction, since the C-PCM/COSMO *ansatz* was shown to include already an *implicit* correction for outlying charge.[28,62] Further discussion of this construction, and of COSMO, is deferred to Section 11.2.8,

As compared to C-PCM, a more sophisticated treatment of continuum electrostatic interactions is afforded by the "surface and simulation of volume polarization for electrostatics" [SS(V)PE] approach.[29] Formally speaking, this model provides an exact treatment of the *surface polarization* (*i.e.*, the surface charge induced by the solute charge that is contained within the solute cavity, which induces a surface polarization owing to the discontinuous change in dielectric constant across the cavity boundary) but also an approximate treatment of the *volume polarization* (arising from the aforementioned outlying charge). The "SS(V)PE" terminology is Chipman's notation,[29] but this model is formally equivalent, at the level of integral equations, to the "integral equation formalism" (IEF-PCM) that was developed originally by Cancès *et al.*.[22,160] Some difference do arise when the integral equations are discretized to form finite-dimensional matrix equations,[88] and it should be noted from Table 11.3 that SS(V)PE uses a symmetrized form of the $\mathbf{K}$ matrix as compared to IEF-PCM. The *asymmetric* IEF-PCM is the recommended approach,[88] although only the symmetrized version is available in the isodensity implementation of SS(V)PE that is discussed in Section 11.2.6. That said, differences between symmetry and asymmetric versions are only important in the case of van der Waals cavity surfaces; they are insignificant for the isodensity cavity construction.[62]

As with the obsolete D-PCM approach, the original version of IEF-PCM explicitly required evaluation of the normal electric field at the cavity surface, but it was later shown that this dependence could be eliminated to afford the version described in Table 11.3.[21,29] This version requires only the electrostatic potential, and is thus preferred, and it is this version that we designate as IEF-PCM. The C-PCM model becomes equivalent to SS(V)PE in the limit $\varepsilon \to \infty$,[29,88] which means that C-PCM must somehow include an *implicit* correction for volume polarization, even if this was not by design.[80] For $\varepsilon \gtrsim 50$, numerical calculations reveal that there is essentially no difference between SS(V)PE and C-PCM results.[88] Since C-PCM is less computationally involved as compared to SS(V)PE, it is the PCM of choice in high-dielectric solvents. The computational savings relative to SS(V)PE may be particularly significant for large QM/MM/PCM jobs.

For a more detailed discussion of the history of these models, see the lengthy and comprehensive review by Tomasi *et al.*.[161] For a briefer discussion of the connections between these models, see Refs. 30,63,88.

11.2.3.2 Cavity Construction and Discretization

Construction of the cavity surface is a crucial aspect of PCMs, as computed properties are quite sensitive to the details of the cavity construction. Most cavity constructions are based on a union of atom-centered spheres (see Fig. 11.1), but there are yet several different constructions whose nomenclature is occasionally confused in the literature. Simplest and most common is the van der Waals (vdW) surface consisting of a union of atom-centered spheres. The radius for the sphere centered on atom A can be written in the form

$$R_A = \alpha_{\text{vdW}} R_{\text{vdW},A} + R_{\text{probe}} \tag{11.4}$$

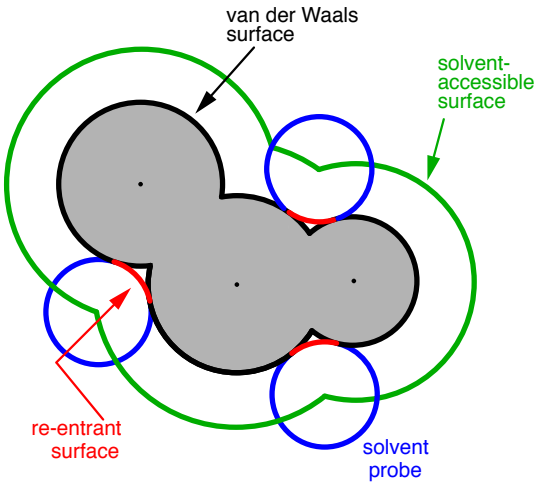


Figure 11.1: Illustration of various solute cavity surface definitions for PCMs.[89] The union of atomic van der Waals spheres (shown in gray) defines the van der Waals (vdW) surface, in black. Note that actual vdW radii from the literature are sometimes scaled in constructing the vdW surface. If a probe sphere (representing the assumed size of a solvent molecule) is rolled over the van der Waals surface, then its center point traces out the solvent accessible surface (SAS), shown in green; the SAS is equivalent to a vdW surface where the atomic radii are increases by the radius of the probe sphere. Finally, one can use the probe sphere to smooth out the sharp crevasses in the vdW surface using the re-entrant surface elements shown in red, resulting in the solvent-excluded surface (SES).

where $R_{\text{vdW},A}$ is the vdW radius for atom A, taken for example from the set of vdW radii published by Bondi.[12] Traditionally, the vdW radii $R_{\text{vdW},A}$ that are extracted from crystallographic data are scaled by a factor $\alpha_{\text{vdW}} = 1.1$–$1.2$.[11,62,159] This 20% augmentation is intended to mimic the fact that solvent molecules cannot approach all the way to the vdW radius of the solute atoms, though it's not altogether clear that the same value ought to be optimal in all cases. (The scaling factor defaults to $\alpha_{\text{vdW}} = 1.2$ but can be modified by the user.)

An alternative to scaling the atomic radii is to add a certain fixed increment R_{probe} to each, representing the approximate size of a solvent molecule, and leading to what is known as the *solvent accessible surface* (SAS). The choice $R_{\text{probe}} = 1.4$ Å is common for water and represents the approximate physical size of a water molecule, although values in the range $R_{\text{probe}} = 0.2$–0.5 Å often afford better solvation energies.[62] In any case, if a nonzero value of R_{probe} is used, then the scaling factor in Eq. (11.4) should be set to $\alpha_{\text{vdW}} = 1.0$, since these two parameters are intended to model the same effect, namely, that a solvent molecule's finite size prevents it from approaching all the way to the vdW radii of the solute.

Note from Fig. 11.1 that both the vdW surface and the SAS possess cusps where the atomic spheres intersect, although these become less pronounced as the atomic radii are scaled or augmented. These cusps are eliminated in what is known as the *solvent-accessible surface* (SES), sometimes called the Connolly surface or the "molecular surface". The SES uses the surface of the probe sphere at points where it is simultaneously tangent to two or more atomic spheres to define elements of a "re-entrant surface" that smoothly connects the atomic (or "contact") surface.[89]

Having chosen a model for the cavity surface, this surface is discretized using atom-centered Lebedev grids[90–92] of the same sort that are used to perform the numerical integrations in DFT. (Discretization of the re-entrant facets of the SES is somewhat more complicated but similar in spirit.[89]) Surface charges q_i are located at these grid points and the Lebedev quadrature weights can be used to define the surface area associated with each discretization point.[85]

A long-standing (though not well-publicized) problem with the aforementioned discretization procedure is that it fails to afford continuous potential energy surfaces as the solute atoms are displaced, because certain surface grid points

may emerge from, or disappear within, the solute cavity, as the atomic spheres that define the cavity are moved. This undesirable behavior can inhibit convergence of geometry optimizations and, in certain cases, lead to very large errors in vibrational frequency calculations.[85] It is also a fundamental hindrance to molecular dynamics calculations.[86] Building upon earlier work by York and Karplus,[176] Lange and Herbert[85,86,89] developed a general scheme for implementing apparent surface charge PCMs in a manner that affords smooth potential energy surfaces, even for *ab initio* molecular dynamics simulations involving bond breaking.[62,63,86] Notably, this approach is faithful to the properties of the underlying integral equation theory on which the PCMs are based, in the sense that the smoothing procedure does not significantly perturb solvation energies or cavity surface areas.[85,86] The smooth discretization procedure combines a switching function with Gaussian blurring of the cavity surface charge density, and is thus known as the "Switching/Gaussian" (SwiG) implementation of the PCM.

Both single-point energies and analytic energy gradients are available for SwiG-PCMs, when the solute is described using molecular mechanics or an SCF (Hartree-Fock or DFT) electronic structure model, except that for the SES cavity model only single-point energies are available. Analytic Hessians are available for the C-PCM model only. (As usual, vibrational frequencies for other models will be computed, if requested, by finite difference of analytic energy gradients.) Single-point energy calculations using correlated wave functions can be performed in conjunction with these solvent models, in which case the correlated wave function calculation will use Hartree-Fock molecular orbitals that are polarized in the presence of the continuum dielectric solvent (*i.e.*, there is no post-Hartree–Fock PCM correction). This represents a "zeroth-order" inclusion of solvent effects that captures the leading-order effect of continuum solvation on molecular properties.[62] Given the crudeness of the model itself, more consistent inclusion of post-Hartree–Fock solvation effects is not expected to be important.[62]

Researchers who use these PCMs are asked to cite Refs. 86 and 88, which provide the details of Q-CHEM's implementation; Ref. 89 if the SES is used, and Ref. 63 if the conjugate-gradient solver or other linear-scaling options are used. We point the reader in particular to Refs. 86 and 178, which provides an assessment of the discretization errors that can be anticipated using various PCMs and Lebedev grids; default grid values in Q-CHEM were established based on these tests. When publishing results based on PCM calculations, it is essential to specify both the precise model that is used (see Table 11.3) as well as how the cavity was constructed, and this should be done without resorting to software-specific keywords, the use of which has significantly muddled the literature on continuum electrostatics.[44,62] As an example of good practice, the default cavity construction in Q-CHEM is a vdW cavity using Bondi atomic radii,[12] except that for hydrogen we use the modified radius of 1.1 Å, following a reassessment that judged Bondi's original value of 1.2 Å for hydrogen to be too large.[139] Each of these radii $R_{\mathrm{vdW},A}$ in Eq. (11.4) is then scaled by a factor $\alpha_{\mathrm{vdW}} = 1.2$ for use in cavity construction. Radii for main-group elements that were not provided by Bondi are taken from Ref. 105. Absent details such as these, PCM calculations will be difficult to reproduce in other electronic structure programs.

11.2.3.3 Nonequilibrium Solvation for Vertical Excitation, Ionization and Emission

In vertical excitation or ionization, the solute undergoes a sudden change in its charge distribution. Various microscopic motions of the solvent have characteristic times to reach certain polarization response, and fast part of the solvent response (electrons) can follow such a dynamic process while the remaining degrees of freedom (nuclei) remain unchanged as in the initial state. Such splitting of the solvent response gives rise to nonequilibrium solvation. In the literature, two different approaches have been developed for describing nonequilibrium solvent effects: the linear response (LR) approach[18,39] and the state-specific (SS) approach.[19,37,69,159] Both are implemented in Q-CHEM,[178], at the SCF level for vertical ionization and at the corresponding level (CIS and TDDFT, see Section 7.3.6, or ADC, see Section 7.11.10) for vertical excitation. A brief introduction to these methods is given below, and users of the nonequilibrium PCM features are asked to cite Refs. 178 and 114. State-specific solvent-field equilibration for long-lived excited states to compute *e.g.* emission energies is implemented for the ADC-suite of methods as described in Section 7.11.10 and for TDDFT as described in Section 7.3.6.2. Users of this equilibrium-solvation PCM please cite

and be referred to Ref. 115.

The LR approach considers the solvation effects as a coupling between a pair of transitions, one for solute and the other for solvent. The transition frequencies when the interaction between the solute and solvent is turned on may be determined by considering such an interaction as a perturbation. In the framework of TDDFT, the interaction between solute and solvent is given by[66]

$$\begin{aligned}\omega' = \int d\mathbf{r} \int d\mathbf{r}' \int d\mathbf{r}'' \int d\mathbf{r}''' \rho^*_{\text{tr}}(\mathbf{r}) \left(\frac{1}{|\mathbf{r}-\mathbf{r}'|} + g_{\text{XC}}(\mathbf{r},\mathbf{r}') \right) \\ \times \chi^*(\mathbf{r}',\mathbf{r}'',\omega) \left(\frac{1}{|\mathbf{r}''-\mathbf{r}'''|} + g_{\text{XC}}(\mathbf{r}'',\mathbf{r}''') \right) \rho_{\text{tr}}(\mathbf{r}''') \, ,\end{aligned} \tag{11.5}$$

where χ is the charge density response function of the solvent and $\rho_{\text{tr}}(\mathbf{r})$ is the solute's transition density. This term accounts for a dynamical correction to the transition energy so that it is related to the response of the solvent to the charge density of the solute oscillating at the solute transition frequency (ω). Within a PCM, only classical Coulomb interactions are taken into account, and Eq. (11.5) becomes

$$\omega'_{\text{PCM}} = \int d\mathbf{r} \int d\mathbf{s} \, \frac{\rho^*_{\text{tr}}(\mathbf{r})}{|\mathbf{r}-\mathbf{s}|} \int d\mathbf{s}' \int d\mathbf{r}' \, \mathcal{Q}(\mathbf{s},\mathbf{s}',\varepsilon) \frac{\rho_{\text{tr}}(\mathbf{r}')}{|\mathbf{s}'-\mathbf{r}'|} \, , \tag{11.6}$$

where $\mathcal{Q}$ is PCM solvent response operator for a generic dielectric constant, ε. The integral of $\mathcal{Q}$ and the potential of the density $\rho_{\text{tr}}(\mathbf{r})$ gives the surface charge density for the solvent polarization.

The state-specific approach takes into account the capability of a part of the solvent degrees of freedom to respond instantaneously to changes in the solute wave function upon excitation. Such an effect is not accounted for in the LR approach. In SS, a generic solvated-solute excited state Ψ_i is obtained as a solution of a nonlinear Schrödinger equation

$$\left(\hat{H}^{\text{vac}} + \hat{V}_0^{\text{slow}} + \hat{V}_i^{\text{fast}} \right) |\Psi_i\rangle = E_i^{\text{SS}} |\Psi_i\rangle \tag{11.7}$$

that depends upon the solute's charge distribution. Here $\hat{H}^{\text{vac}}$ is the usual Hamiltonian for the solute in vacuum and the reaction field operator $\hat{V}_i$ generates the electrostatic potential of the apparent surface charge density (Section 11.2.3.1), corresponding to slow and fast polarization response. The solute is polarized self-consistently with respect to the solvent's reaction field. In case of vertical ionization rather than excitation, both the ionized and non-ionized states can be treated within a ground-state formalism. For vertical excitations, self-consistent SS models have been developed for various excited-state methods,[69,110] and are implemented for both CIS and TDDFT.

In a linear dielectric medium, the solvent polarization is governed by the electric susceptibility, $\chi = [\varepsilon(\omega) - 1]/4\pi$, where $\varepsilon(\omega)$ is the frequency-dependent permittivity. In case of very fast vertical transitions, the dielectric response is ruled by the optical dielectric constant, $\varepsilon_{\text{opt}} = n^2$, where n is the solvent's index of refraction. In both LR and SS, the fast part of the solvent's degrees of freedom is in equilibrium with the solute density change. Within PCM, the fast solvent polarization charges for the SS excited state i can be obtained by solving the following equation:[37]

$$\mathbf{K}_{\varepsilon_{\text{opt}}} \mathbf{q}_i^{\text{fast,SS}} = \mathbf{R}_{\varepsilon_{\text{opt}}} \left[\mathbf{v}_i + \mathbf{v}(\mathbf{q}_0^{\text{slow}}) \right] \, . \tag{11.8}$$

Here $\mathbf{q}^{\text{fast,SS}}$ is the discretized fast surface charge. The dielectric constants in the matrices $\mathbf{K}$ and $\mathbf{R}$ (Section 11.2.3.1) are replaced with the optical dielectric constant, and $\mathbf{v}_i$ is the potential of the solute's excited state density, ρ_i. The quantity $\mathbf{v}(\mathbf{q}_0^{\text{slow}})$ is the potential of the slow part of the apparent surface charges in the ground state, which are given by

$$\mathbf{q}_0^{\text{slow}} = \left(\frac{\varepsilon - \varepsilon_{\text{opt}}}{\varepsilon - 1} \right) \mathbf{q}_0 \, . \tag{11.9}$$

For LR-PCM, the solvent polarization is subjected to the first-order changes to the electron density (TDDFT linear density response), and thus Eq. (11.8) becomes

$$\mathbf{K}_{\varepsilon_{\text{opt}}} \mathbf{q}_i^{\text{fast,LR}} = \mathbf{R}_{\varepsilon_{\text{opt}}} \mathbf{v}(\rho_i^{\text{tr}}) \, . \tag{11.10}$$

The LR approach for CIS/TDDFT excitations and the self-consistent SS method (using the ground-state SCF) for vertical ionizations are available in Q-CHEM. The self-consistent SS method is also available for vertical excitations, but can be problematic for (near-) degeneracies between excited states, such as in the vicinity of a conical intersection. The fundamental problem in the SS approach is that each wave function Ψ_i is an eigenfunction of a *different* Hamiltonian, since Eq. (11.7) depend upon the specific state of interest. To avoid the ordering and the non-orthogonality problems, we compute the vertical excitation energy using a first-order, perturbative approximation to the SS approach,[20,23] in what we have termed the "ptSS" method.[114] The zeroth-order excited-state wave function can be calculated using various excited-state methods (currently available for CIS and TDDFT in Q-CHEM) with solvent-relaxed molecular orbitals obtained from a ground-state PCM calculation. As mentioned previously, LR and SS describe different solvent relaxation features in nonequilibrium solvation. In the perturbation scheme, we can calculate the LR contribution using the zeroth-order transition density, in what we have called the "ptLR" approach. The combination of ptSS and ptLR yields quantitatively good solvatochromatic shifts in combination with TDDFT but not with the correlated variants of ADC, for which the pure ptSS approach was shown to be superior.[114,178]

The LR and SS approaches can also be used in the study of photon emission processes.[70] An emission process can be treated as a vertical excitation at a stationary point on the excited-state potential surface. The basic requirement therefore is to prepare the solvent-relaxed geometry for the excited-state of interest. TDDFT/LR-C-PCM analytic gradients and Hessian are available.

Section 7.3.7 for computational details regarding excited-state geometry optimization with PCM. An emission process is slightly more complicated than the absorption case. Two scenarios are discussed in literature, depending on the lifetime of an excited state in question. In the limiting case of ultra-fast excited state decay, when only fast solvent degrees of freedom are expected to be equilibrated with the excited-state density. In this limit, the emission energy can be computed exactly in the same way as the vertical excitation energy. In this case, excited state geometry optimization should be performed in the nonequilibrium limit. The other limit is that of long-lived excited state, *e.g.*, strongly fluorescent species and phosphorescence. In the long-lived case, excited state geometry optimization should be performed with the solvent equilibrium limit. Thus, the excited state should be computed using an equilibrium LR or SS approach, and the ground state is calculated using nonequilibrium self-consistent SS approach. The latter approach is implemented for the ADC-based methods as described in Section 7.11.10 and TDDFT as described in Section 7.3.6.2.

11.2.3.4 Absorption and Emission Spectra Based on the Constrained Equilibrium Principle

For ultrafast processes in solution, such as electron transfer, photo-absorption/emission and photo-ionization, a continuum model should combine a proper nonequilibrium solvation theory to account for nonequilibrium solute–solvent interactions. In the traditional treatments, the nonequilibrium electrostatic solvation free energy was derived from the so-called reversible electric work integration along the path linking the initial equilibrium state (eq) and the intermediate nonequilibrium state (neq), *i.e.*,[106,107]

$$[\rho = 0, \Phi = 0] \xrightarrow{\varepsilon} [\rho_1, \Phi_1^{\mathrm{eq}}] \xrightarrow{\varepsilon_{\mathrm{opt}}} [\rho_2, \Phi_2^{\mathrm{neq}}] \tag{11.11}$$

where Φ denotes the total electric potential including both the potential ψ due to the solute charge ρ in vacuum and polarization potential φ due to the medium. In order to deal with electron absorption and emission spectra in solution, the numerical expression of nonequilibrium solvation free energy which was established by intuitively collecting a series of energy terms from the interactions of solute charges and polarized charges, has been implemented using TDDFT with PCM model.[37,38] It is easy to verify this numerical form can be achieved through the discretization of analytical expression of nonequilibrium solvation energy by traditional treatments.[98] However, there exist a number of doubts on the overestimation of the solvent reorganization energy in ultrafast processes by this reversible electric work method.[10,54] It becomes clear now that there is no possibility to find a reversible pathway between the initial equilibrium state and the intermediate nonequilibrium state. Thus, the integrated electric work can not equal to the change of electrostatic free energy.[98]

Xiangyuan Li *et al.*[98,99,134,175] established the new theory for nonequilibrium solvation by employing the constrained equilibrium principle,[94] using the following pathway:

$$[\rho_1, \varphi_1^{\mathrm{eq}}] \xrightarrow{\text{fast}} [\rho_2, \varphi_2^{\mathrm{neq}}] \xrightarrow{-\lambda_{\mathrm{s}}} [\rho_2, \varphi_2^{\mathrm{eq}}] \xrightarrow{+\rho_{\mathrm{ex}},\text{quasistatic}} C[\rho_2 + \rho_{\mathrm{ex}}, \varphi_2^{\mathrm{neq}}] \tag{11.12}$$

where C stands for the constrained equilibrium state which is constructed and mapped to the true nonequilibrium state by introducing the proper external charge ρ_{ex} which is used to equilibrate the "residual" polarization potential, $\varphi^{'} = \varphi_2^{\mathrm{neq}} - \varphi_2^{\mathrm{eq}}$. In this way the solvent reorganization energy can be derived as

$$\lambda_{\mathrm{s}} = -\frac{1}{2}\int_V \rho_{\mathrm{ex}}\ \varphi'\ dV \tag{11.13}$$

Then the nonequilibrium solvation free energy is simply given by

$$F_2^{\mathrm{neq}} = \frac{1}{2}\int_V \rho_2\ \varphi_2^{\mathrm{eq}}\ dV + \lambda_{\mathrm{s}} \tag{11.14}$$

For more detailed descriptions of the gain of the external (constraining) charge ρ_{ex}, or the equivalent constraining external electric field $\mathbf{E}_{\mathrm{ex}}$, please refer to the review.[98] Within the framework of continuum model, the discretization and numerical solution of Eq. (11.14) is expressed as[8]

$$F_i^{\mathrm{neq}} = \sum_m V_{i,m} Q_{i,m}^{\mathrm{neq}} - \frac{1}{2}\sum_m Q_{i,m}^{\mathrm{neq}} D(\varepsilon) Q_{i,m}^{\mathrm{neq}} \tag{11.15}$$

where the subscript i denotes the ground (i = 1) or excited (i = 2) electronic state. The value $V_{i,m}$ refers to the solute electrostatic potential at the mth tesserae. $Q_{i,m}^{\mathrm{neq}}$ is the apparent charge for the nonequilibrium state. The quantity $D(\varepsilon)$ is the square matrix arising in PCMs such as C-PCM and IEF-PCM (see Section 11.2.3.1).

The vertical excitation energy for absorption is defined as

$$h\nu_{\mathrm{ab}} = G_2^{\mathrm{neq}} - G_1^{\mathrm{eq}} \tag{11.16}$$

where G stands for total free energy of the solute in solution,

$$G_1^{\mathrm{eq}} = E_1 + F_1^{\mathrm{eq}} = E_1 + \frac{1}{2}\sum_m V_{1,m} Q_{1,m}^{\mathrm{eq}} \tag{11.17}$$

The quantity G_1^{eq} is calculated by the SCRF method based on equilibrium ground-state reaction field and E_1 means the ground-state electronic energy of the solute. Based on the equilibrium ground-state reaction field, with a self-consistent state-specific method in the framework of TDDFT, G_2^{neq} is given by[8]

$$G_2^{\mathrm{neq}} = E_1 + \omega_{\mathrm{ab}} + \sum_m V_{1,m} Q_{1,m}^{\mathrm{eq}} + \sum_m V_{2,m}(Q_{2,m}^{\mathrm{neq}} - Q_{1,m}^{\mathrm{eq}}) - \frac{1}{2}\sum_m Q_{2,m}^{\mathrm{neq}} D(\varepsilon) Q_{2,m}^{\mathrm{neq}} \tag{11.18}$$

where ω_{ab} is the TDDFT excitation energy. Alternatively, based on the nonequilibrium excited-state reaction field, G_2^{neq} is given by[9]

$$G_{2,k}^{\mathrm{neq}} = E_1^k + \omega_{\mathrm{ab}}^k - \sum_m V_{1,m}^k Q_{2,m}^{\mathrm{neq},k} - \frac{1}{2}\sum_m Q_{2,m}^{\mathrm{neq},k} D(\varepsilon) Q_{2,m}^{\mathrm{neq},k}\ , \tag{11.19}$$

where k represents the kth iteration of the nonequilibrium excited-state reaction field at the ground-state geometry of solute. ω_{ab}^k is the excitation energy from TDDFT calculation in the presence of the nonequilibrium excited-state reaction field. The quantity E_1^k is the ground-state electronic energy of solute at the kth iteration. Clearly, Eq. (11.19) is more physically-meaningful than Eq. (11.18).

Similarly, the vertical excitation energy for emission is given by[9]

$$h\nu_{\mathrm{em}} = G_2^{\mathrm{eq}} - G_1^{\mathrm{neq}} \tag{11.20}$$

where G_2^{eq} and G_1^{neq} represent the free energies of the equilibrium excited state and the nonequilibrium ground state at the excited-state equilibrium geometry, respectively. These can be expressed as[9]

$$G_1^{\text{neq}} = E_1'' + \sum_m V_{1,m} Q_{1,m}^{\text{neq}} - \frac{1}{2} \sum_m Q_{1,m}^{\text{neq}} D(\varepsilon) Q_{1,m}^{\text{neq}} \ . \tag{11.21a}$$

$$G_2^{\text{eq},k} = \omega_{\text{em}}^k + E_1^{k\prime} - \frac{1}{2} \sum_m V_{2,m}^k Q_{2,m}^{\text{eq},k} + \sum_m V_{1,m}^k Q_{2,m}^{\text{eq},k} \ . \tag{11.21b}$$

The quantity ω_{em}^k can be directly obtained by TDDFT calculation in the equilibrium excited-state reaction field at the kth iteration. The energies $E_1^{k\prime}$ and E_1'' are, respectively, the corresponding ground-state electronic energy of the solute and the ground-state electronic energy of the solute at the excited-state equilibrium geometry in the presence of the nonequilibrium ground-state reaction field.

To engage the constrained equilibrium principle for vertical absorption and emission, use the **Td_CEq_NonEq** keyword in the *$pcm* section. Users are asked to cite Ref. 98.

Note: This method is incompatible with spin-flip calculations.

11.2.3.5 Heterogeneous PCM

For modeling proteins and other anisotropic environments, a "heterogeneous PCM" (HetPCM) approach has been suggested, which amounts to using a different dielectric constant for each atomic sphere.[13,71,146,167] Unlike the isotropic PCM, which is derivable from Maxwell's equations based on well-defined assumptions,[62] HetPCM is entirely *ad hoc*. However, it has recently been tested (for the first time) against exact Poisson boundary conditions, implemented using the PEqS approach that is discussed in Section 11.2.11.[13] Considering HetPCM methods that use a pair of distinct dielectric constants (namely, $\varepsilon = 78$ to represent an aqueous environment and a smaller value, $\varepsilon_{\text{nonp}}$, to represent a nonpolar or hydrophobic region), it was determined that the heterogenous model is in reasonable agreement with solution of Poisson's equation for an analogous heterogeneous dielectric environment, provided that $\varepsilon_{\text{nonp}} \gtrsim 4$.[13] Fortunately, this is consistent with values that are often used to model the hydrophobic interiors of proteins.[62] See Example 11.6 for instructions on setting up HetPCM calculations. The analytic gradient is available for HetPCM.

11.2.4 PCM Job Control

A PCM calculation is requested by setting SOLVENT_METHOD = PCM in the *$rem* section. As mentioned above, there are a variety of different theoretical models that fall within the PCM family, so additional fine-tuning may be required, as described below.

11.2.4.1 *$pcm* section

Most PCM job control is accomplished via options specified in the *$pcm* input section, which allows the user to specify which flavor of PCM will be used, which algorithm will be used to solve the PCM equations, and other options. The format of the *$pcm* section is analogous to that of the *$rem* section:

```
$pcm
   <Keyword>  <parameter/option>
$end
```

Note: The following job control variables belong *only* in the *$pcm* section. Do not place them in the *$rem* section.

Theory

Specifies the which polarizable continuum model will be used.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

CPCM

OPTIONS:

CPCM	Conductor-like PCM of Ref. 41, with $f_\varepsilon = (\varepsilon - 1)/\varepsilon$.
CPCM1	Equivalent to CPCM.
CPCM2	Original conductor-like screening model of Ref. 81, with $f_\varepsilon = (\varepsilon - 1)/(\varepsilon + 1/2)$.
IEFPCM	IEF-PCM with an asymmetric $\mathbf{K}$ matrix.
SSVPE	SS(V)PE model, equivalent to IEF-PCM with a symmetric $\mathbf{K}$ matrix.

RECOMMENDATION:

The IEF-PCM/SS(V)PE model is more sophisticated model than C-PCM, and formally more appropriate for low-dielectric solvents, but it is also more complicated to work with. In practice there is often little difference between these models; see Ref. 62. Regarding the two versions of C-PCM and the choice of f_ε in Eq. (11.3), the choice $x = 0$ has been suggested for neutral solutes and $x = 1/2$ for ions; again, see Ref. 62 for a discussion of these distinctions.

Method

Specifies which surface discretization method will be used.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

SwiG

OPTIONS:

SwiG	Switching/Gaussian method
ISwiG	"Improved" Switching/Gaussian method with an alternative switching function
Spherical	Use a single, fixed sphere for the cavity surface.
Fixed	Use discretization point charges instead of smooth Gaussians.

RECOMMENDATION:

Use of **SwiG** is recommended only because it is slightly more efficient than the switching function of **ISwiG**. On the other hand, **ISwiG** offers some conceptually more appealing features and may be superior in certain cases. Consult Refs. 86,88 for a discussion of these differences. The **Fixed** option uses the Variable Tesserae Number (VTN) algorithm of Li and Jensen,[95] with Lebedev grid points. VTN uses point charges with no switching function or Gaussian blurring, and is therefore subject to discontinuities that may cause errors in analytic gradients and lead to slow convergence in geometry optimizations.[85] Its use is not recommended except to make contact with other calculations in the literature.

Solver

Specifies the algorithm used to solve the PCM equations.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

INVERSION

OPTIONS:

INVERSION	Direct matrix inversion
CG	Iterative conjugate gradient

RECOMMENDATION:

Matrix inversion is faster for small solutes because it needs to be performed only once in a single-point calculation. However, the CG solver (which must be applied at each SCF iteration) is recommended if the number of discretization points approaches or exceeds ∼10,000. Linear-scaling options for large-scale MM/PCM and QM/MM/PCM calculations are discussed in Section 11.2.5.

SwitchThresh

Threshold for discarding grid points on the cavity surface.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n	Discard grid points when the switching function is less than 10^{-n}.

RECOMMENDATION:

Use the default, which is found to avoid discontinuities within machine precision. Increasing n reduces the cost of PCM calculations but can introduce discontinuities in the potential energy surface. If the PCM part of the calculation becomes too expensive it is better to reduce the number of discretization points (**HeavyPoints** and **HPoints**, described below), or to use **Solver = CG**, than it is to adjust **SwitchThresh**.

Construction of the solute cavity is an important part of the model and users should consult the literature in this capacity, especially with regard to the radii used for the atomic spheres. The default values provided in Q-CHEM correspond to the consensus choice that has emerged over several decades, namely, to use vdW radii $\{R_{\text{vdW},A}\}$ scaled by a factor $\alpha_{\text{vdW}} = 1.2$; see Eq. (11.4). The most widely-used set of vdW radii are those determined from crystallographic data by Bondi,[12] although the radius for hydrogen was later adjusted to 1.1 Å,[139] and radii for those main-group elements not addressed by Bondi were provided later.[105] This extended set of vdW is used by default in Q-CHEM, and for simplicity we call these "Bondi radii" regardless of whether they come from Bondi's original paper or the later work. Alternatively, atomic radii from the Universal Force Field (UFF) are available.[133] The main appeal of UFF radii is that they are defined for all atoms of the periodic table, though the quality of these radii for PCM applications is unclear.

Radii

Specifies which set of atomic van der Waals radii will be used to define the solute cavity.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

Bondi

OPTIONS:

Bondi Use the (extended) set of Bondi radii (Ref. 139).

FF Use Lennard-Jones radii from a molecular mechanics force field.

UFF Use radii from the Universal Force Field (Ref. 133).

Read Read the atomic radii from a *$van_der_waals* input section.

RECOMMENDATION:

Bondi radii are widely used. The **FF** option requires the user to specify an MM force field using the FORCE_FIELD variable in the *$rem* section, and also to define the atom types in the *$molecule* section (see Section 11.3). This is not required for UFF radii. For user-defined radii, see Section 11.2.10 for the format of the *$van_der_waals* input section.

vdwScale

Scaling factor for the atomic van der Waals radii used to define the solute cavity.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

1.2

OPTIONS:

α Use a scaling factor of $\alpha > 0$.

RECOMMENDATION:

The default value is widely used in PCM calculations, although a value of 1.0 might be appropriate if using a solvent-accessible surface.

SASradius

Form a "solvent accessible" surface with the given solvent probe radius.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

0.0

OPTIONS:

r Use a solvent probe radius of r, in Å.

RECOMMENDATION:

The solvent probe radius is added to the scaled van der Waals radii of the solute atoms. A common solvent probe radius for water is 1.4 Å, but smaller values (0.2–0.5 Å) have also been used historically.[62]

SurfaceType

Selects the solute cavity surface construction.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

VDW_SAS

OPTIONS:

VDW_SAS	van der Waals or solvent-accessible surface
SES	solvent-excluded surface

RECOMMENDATION:

The vdW surface and the SAS are each comprised simply of atomic spheres and thus share a common option; the only difference is the specification of a solvent probe radius, **SASradius**. For a true vdW surface, the probe radius should be zero (which is the default), whereas for the SAS the atomic radii are traditionally not scaled, hence **vdwScale** should be set to 1.0 (which is *not* the default). For the SES, only SwiG discretization is available, but this can be used with any set of (scaled or unscaled) atomic radii, or with radii that are augmented by **SASradius**.

Note from the description of the **Radii** keyword that the user may specify his or her own radii for cavity construction, using a *$van_der_waals* input section. To specify all of the radii for a given molecule, use the format of the *$van_der_waals* section that is described in Section 11.2.10. Alternatively, if one simply needs to specify a radius that is missing from a given set of radii (such as Bondi's set, which is not defined for the entire periodic table), this can be accomplished without setting **Radii = Read**. Instead, leave **Radii = Bondi** (or some other choice) but include a *$van_der_waals* section having the format

```
$van_der_waals
atomic_number    vdW_radius
$end
```

No scaling factor is applied to user-defined radii. Note that $R = 0$ is allowed for a particular atomic radius, in which case the atom in question is not used to construct the cavity surface. This feature facilitates the construction of "united atom" cavities,[7] in which the hydrogen atoms do not get their own spheres and the heavy-atom radii are increased to compensate. As an alternative to scaling the vdW radii $\{R_{\mathrm{vdW},A}\}$, the user can choose to augment each of them with a probe radius R_{probe} [see Eq. (11.4)] to obtain the SAS cavity.

Historically, discretization of the cavity surface has involved "tessellation" methods that divide the cavity surface area into finite polygonal "tesserae". (The GEPOL algorithm[123] is perhaps the most widely-used tessellation scheme.) Tessellation methods, however, suffer not only from discontinuities in the cavity surface area and solvation energy as a function of the nuclear coordinates, but in addition they lead to analytic energy gradients that are complicated to derive and implement. To avoid these problems, Q-CHEM's SwiG-PCM implementation[85,86,88] uses Lebedev grids to discretize the atomic spheres. These are atom-centered grids with icosahedral symmetry, and may consist of anywhere from 26 to 5294 grid points per atomic sphere. The default values used by Q-CHEM were selected based on extensive numerical tests,[86,88] and they are looser for MM atoms (in MM/PCM or QM/MM/PCM jobs) than they are for QM atoms, reflecting the more complicated electrostatic potential that is generated by a QM density as compared to an MM point charge. For QM atoms, the default is to use $N = 110$ points for hydrogen atoms and $N = 194$ points for all other atoms, whereas for MM atoms the default is $N = 50$ for hydrogen and $N = 110$ for non-hydrogen. These default values exhibit good rotational invariance (< 0.1 kcal/mol differences in ΔG when the molecule is rotated[86]) and absolute solvation energies that typically lie within < 1 kcal/mol of the $N \to \infty$ limit,[86,88] at least for charge-neutral solutes.

Note that earlier versions of Q-CHEM used denser grids by default. (In versions up to and including Q-CHEM v. 4.2, the default was $N = 590$ for all QM atoms, but was switched to $N = 302$ beginning with v. 4.2.1. The defaults mentioned above are the current ones starting with v. 5.2.) However, grid errors of $\sim$0.5 kcal/mol are well within the intrinsic accuracy of $\Delta G_{\text{solvation}}$ for these models. Grid errors in solvatochromatic shifts for excitation energies tend to be $\sim$0.01–0.02 eV, which is well within the intrinsic accuracy of nearly any excited-state methodology. If questions about grid accuracy arise, we suggest using $N = 302$ as a high-quality option and $N = 590$ as an essentially converged option.[86,88] For large molecules it may be necessary to reduce the number of grid points (*e.g.*, to $N = 86$) or to use linear-scaling solvers rather than matrix inversion, as discussed in Section 11.2.5.

Note: The acceptable values for the number of Lebedev points per sphere are $N = 6$, 26, 38, 50, 86, 110, 146, 170, 194, 302, 350, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, 5294.

HeavyPoints

The number of Lebedev grid points to be placed non-hydrogen atoms in the QM system.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

194

OPTIONS:

Acceptable values are listed above.

RECOMMENDATION:

Use the default for geometry optimizations. For absolute solvation energies, the user may want to examine convergence with respect to N.

HPoints

The number of Lebedev grid points to be placed on H atoms in the QM system.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

110

OPTIONS:

Acceptable values are listed above.

RECOMMENDATION:

Use the default for geometry optimizations. For absolute solvation energies, the user may want to examine convergence with respect to N.

MMHeavyPoints

The number of Lebedev grid points to be placed on non-hydrogen atoms in the MM subsystem.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

110

OPTIONS:

Acceptable values are listed above.

RECOMMENDATION:

Use the default for geometry optimizations. For absolute solvation energies, the user may want to examine convergence with respect to N. This option applies only to MM/PCM or QM/MM/PCM calculations.

MMHPoints

The number of Lebedev grid points to be placed on H atoms in the MM subsystem.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

Acceptable values are listed above.

RECOMMENDATION:

Use the default for geometry optimizations. For absolute solvation energies, the user may want to examine convergence with respect to N. This option applies only to MM/PCM or QM/MM/PCM calculations.

Especially for complicated molecules, the user may want to visualize the cavity surface. This can be accomplished by setting **PrintLevel** ≥ 2, which will trigger the generation of several `.PQR` files that describe the cavity surface. (These are written to the Q-CHEM output file.) The `.PQR` format is similar to the common `.PDB` (Protein Data Bank) format, but also contains charge and radius information for each atom. One of the output `.PQR` files contains the charges computed in the PCM calculation and radii (in Å) that are half of the square root of the surface area represented by each surface grid point. Thus, in examining this representation of the surface, larger discretization points are associated with larger surface areas. A second `.PQR` file contains the solute's electrostatic potential (in atomic units), in place of the charge information, and uses uniform radii for the grid points. These `.PQR` files can be visualized using various third-party software, including the freely-available Visual Molecular Dynamics (VMD) program,[1,67] which is particularly useful for coloring the `.PQR` surface grid points according to their charge, and sizing them according to their contribution to the molecular surface area. (Examples of such visualizations can be found in Ref. 85.)

PrintLevel

Controls the printing level during PCM calculations.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Prints PCM energy and basic surface grid information. Minimal additional printing.

1 Level 0 plus PCM solute-solvent interaction energy components and Gauss' Law error.

2 Level 1 plus surface grid switching parameters and a `.PQR` file for visualization of the cavity surface apparent surface charges.

3 Level 2 plus a `.PQR` file for visualization of the electrostatic potential at the surface grid created by the converged solute.

4 Level 3 plus additional surface grid information, electrostatic potential and apparent surface charges on each SCF cycle.

5 Level 4 plus extensive debugging information.

RECOMMENDATION:

Use the default unless further information is desired.

Finally, note that setting **Method** to **Spherical** in the *$pcm* input selection requests the construction of a solute cavity consisting of a single, fixed sphere. This is generally not recommended but is occasionally useful for making contact with the results of Born models in the literature, or the Kirkwood-Onsager model discussed in Section 11.2.2. In this case, the cavity radius and its center must also be specified in the *$pcm* section. The keyword **HeavyPoints** controls the number of Lebedev grid points used to discretize the surface.

CavityRadius

Specifies the solute cavity radius.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

None

OPTIONS:

R Use a radius of R, in Ångstroms.

RECOMMENDATION:

None.

CavityCenter

Specifies the center of the spherical solute cavity.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

0.0 0.0 0.0

OPTIONS:

$x\ y\ z$ Coordinates of the cavity center, in Ångstroms.

RECOMMENDATION:

The format is **CavityCenter** followed by three floating-point values, delineated by spaces.

Uses the same coordinate system as the *$molecule* section.

Finally, the HetPCM method (Section 11.2.3.5) is activated by the appearance of a keyword **Het-PCM** in the *$pcm* section. Such calculations require an *$atomic_epsilon* input section that specifies the value of ε that is to be used for each atomic sphere. The format is simply a list of atom numbers with a double-precision value of ε for each:

```
$atomic_epsilon
1   epsilon_1
2   epsilon_2
3   epsilon_3
...
$end
```

Atoms can be omitted from this list and will be assigned $\varepsilon = 1$.

Het-PCM

If present, activates the heterogeneous PCM.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

If the keyword is present, HetPCM is turned on

RECOMMENDATION:

Requires an *$atomic_epsilon* section to provide a user-specified dielectric constant for each atom. Set PCM_PRINT ≥ 1 in *$rem* to confirm that the atomic values of ε are being used.

11.2.4.2 Examples

Example 11.3 A basic example of using the PCMs: optimization of trifluoroethanol in water. The solvent dielectric is specified in the *$solvent* section, which is described below.

```
$comment
2023/4/5 Add geom_opt_driver optimize per #2954, [39888/qchem]
$end

$molecule
   0 1
   C    -0.245826    -0.351674    -0.019873
   C     0.244003     0.376569     1.241371
   O     0.862012    -0.527016     2.143243
   F     0.776783    -0.909300    -0.666009
   F    -0.858739     0.511576    -0.827287
   F    -1.108290    -1.303001     0.339419
   H    -0.587975     0.878499     1.736246
   H     0.963047     1.147195     0.961639
   H     0.191283    -1.098089     2.489052
$end

$rem
   JOBTYPE         OPT
   BASIS           6-31G*
   METHOD          B3LYP
   SOLVENT_METHOD  PCM
   geom_opt_driver optimize
$end

$pcm
   Theory          CPCM
   Method          SWIG
   Solver          Inversion
   HeavyPoints     194
   HPoints         194
   Radii           Bondi
   vdwScale        1.2
$end

$solvent
   Dielectric 78.39
$end
```

Example 11.4 PCM with a single spherical cavity, applied to H_2O in acetonitrile. Compared to the Kirkwood-Onsager multipole expansion method, Example 11.2.2 on page 1150.

```
$molecule
   0 1
   O       0.00000000     0.00000000     0.11722303
   H      -0.75908339     0.00000000    -0.46889211
   H       0.75908339     0.00000000    -0.46889211
$end

$rem
   METHOD            HF
   BASIS             6-31g**
   SOLVENT_METHOD    pcm
$end

$pcm
   method        spherical   ! single spherical cavity with 590 discretization points
   HeavyPoints   590
   CavityRadius  1.8         ! Solute Radius, in Angstrom
   CavityCenter  0.0 0.0 0.0 ! Will be at center of Standard Nuclear Orientation
   Theory        SSVPE
$end

$solvent
   Dielectric        35.9   !   Acetonitrile
$end
```

Finally, we consider an example of a united-atom cavity. Note that a user-defined vdW radius is supplied only for carbon, so the hydrogen radius is taken to be zero and thus the hydrogen atoms are not used to construct the cavity

surface. (As mentioned above, the format for the *$van_der_waals* input section is discussion in Section 11.2.10).

Example 11.5 United-atom cavity construction for ethylene.

```
$comment
   Benzene (in benzene), with a united-atom cavity construction
   R = 2.28 A for carbon, R = 0 for hydrogen
$end

$molecule
   0 1
   C    1.38620    0.000000   0.000000
   C    0.69310    1.200484   0.000000
   C   -0.69310    1.200484   0.000000
   C   -1.38620    0.000000   0.000000
   C   -0.69310   -1.200484   0.000000
   C    0.69310   -1.200484   0.000000
   H    2.46180    0.000000   0.000000
   H    1.23090    2.131981   0.000000
   H   -1.23090    2.131981   0.000000
   H   -2.46180    0.000000   0.000000
   H   -1.23090   -2.131981   0.000000
   H    1.23090   -2.131981   0.000000
$end

$rem
   EXCHANGE        hf
   BASIS           6-31G*
   SOLVENT_METHOD  pcm
$end

$pcm
   theory          iefpcm   ! this is a synonym for ssvpe
   method          swig
   printlevel      1
   radii           read
$end

$solvent
   dielectric 2.27
$end

$van_der_waals
1
  6     2.28
  1     0.00
$end
```

Example 11.6 Heterogeneous (HetPCM) example, triggered by the **Het-PCM** flag in the *$pcm* section. This is a truncated model of protonated histidine, with the cationic aromatic moiety exposed to and environment where $\varepsilon = 78.4$, whereas $\varepsilon = 10.0$ for atoms in the side chain.

```
$molecule
1 1
  H   -4.9076613   -1.9352798   -1.3335875
  C   -2.3500449   -0.5754970    0.0756697
  N   -3.3660526    0.0772627    0.6735620
  H   -3.3085727    0.8201166    1.4041665
  C   -4.4907166   -0.4318030    0.1363201
  H   -5.4902154   -0.1170220    0.4064538
  N   -4.2187491   -1.3840482   -0.7759280
  C   -2.8779889   -1.4836771   -0.8202360
  H   -2.3293638   -2.1655889   -1.4569339
  C   -0.4271237    1.0622789   -0.0051732
  C   -0.8883153   -0.3558429    0.3532960
  H   -0.6778182   -0.5663962    1.4234430
  H   -0.2980823   -1.0704793   -0.2605079
  N   -1.0842830    2.0665662    0.8255888
  H   -0.6455886    1.2632695   -1.0772303
  H   -0.7421259    3.0116763    0.5376181
  H   -0.8094259    1.9175321    1.8239106
  H    0.6740052    1.1315080    0.1435944
$end

$rem
method            hf
basis             sto-3g
solvent_method    pcm
pcm_print         1
$end

$pcm
het-pcm
theory     cpcm
$end

$atomic_epsilon
1  78.4
2  78.4
3  78.4
4  78.4
5  78.4
6  78.4
7  78.4
8  78.4
9  78.4
10 10.0
11 10.0
12 10.0
13 10.0
14 10.0
15 10.0
16 10.0
17 10.0
18 10.0
$end
```

11.2.4.3 *$solvent* Section: Electrostatics Options

The solvent for PCM calculations is specified using the *$solvent* section, as documented in this section and the next one. Note that the *$solvent* section is also used to specify parameters for the Kirkwood-Onsager SCRF model (Section 11.2.2), and the particular options that can be listed in the *$solvent* section depend upon the value of SOLVENT_METHOD in the *$rem* section. A minimum requirement for a PCM calculation is that the *$solvent* section must specify the static dielectric constant (ε), and for nonequilibrium excited-state calculations the optical dielectric constant ($\varepsilon_{\text{opt}} \equiv \varepsilon_\infty$) must be specified as well. There are default values that are appropriate for water, but for any other solvent these constants must be specified. These can either be specified individually, using the variables **Dielectric** and **OpticalDielectric** in the *$solvent* section, or they can be set to predefined values by setting **SolventName** equal to one of the named solvents that is given in Table 11.4.

Unlike the SMx models (Section 11.2.9), which also employ named solvents (specified in the *$smx* section), use of **SolventName** with a PCM serves only to specify the two dielectric constants. Nonelectrostatic interactions are not included in an automatic way, although they can be added in an *ad hoc* way as described in Section 11.2.4.4. The dielectric constants listed in Table 11.4 correspond to values at 20°C. For water, this is slightly different than the default values of **Dielectric** and **OpticalDielectric**, which correspond to 25°C. The latter two values are fully tunable, whereas the named solvents are limited to the pairs of values (ε, ε_∞) listed in Table 11.4.

Dielectric

Specifies the static dielectric constant for the PCM solvent.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

ε Use a dielectric constant $\varepsilon > 0$.

RECOMMENDATION:

The static (*i.e.*, zero-frequency) dielectric constant is what is usually called "the" dielectric constant. The default corresponds to water at 25°C.

OpticalDielectric

Specifies the optical dielectric constant for the PCM solvent.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

1.78

OPTIONS:

ε_∞ Use an optical dielectric constant $\varepsilon_\infty > 0$.

RECOMMENDATION:

The default corresponds to water at 25°C. Note that $\varepsilon_\infty = n^2$, where n is the solvent's index of refraction.

SolventName

Specify both ε and ε_∞ for PCM calculations by naming the solvent.

INPUT SECTION: *$solvent*

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

name Use values ε and ε_∞ corresponding to solvent *name* in Table 11.4.

RECOMMENDATION:

The use of **SolventName** is incompatible with explicit specification of either **Dielectric** or **OpticalDielectric**.

11.2.4.4 *$solvent* Section: Nonelectrostatic Options

A PCM describes only electrostatic interactions with the solute, based on the static dielectric constant (for the initial state) and possibly the optical dielectric constant in the case of a nonequilibrium excitation or ionization process. Nonelectrostatic interaction terms,[62] describing cavitation (Pauli repulsion) and dispersion interactions, for example, can be added in an *ad hoc* way but have not been specifically parameterized for PCMs. (For solvation models with automatic incorporation of nonelectrostatic terms, which are needed to obtain absolute solvation energies, see the SMx models in Section 11.2.9 or the CMIRS approach in Section 11.2.7.

To add nonelectrostatic interactions to a PCM calculation, use the *$solvent* section whose general form is shown below.

```
$solvent
   NonEls  <Option>
   NSolventAtoms <Number unique of solvent atoms>
   SolventAtom <Number1> <Number2> <Number3> <SASradius>
   SolventAtom <Number1> <Number2> <Number3> <SASradius>
   . . .
   <Keyword>  <parameter/option>
   . . .
$end
```

The keyword **SolventAtom** requires multiple parameters, whereas all other keywords require only a single parameter, including the **Dielectric**, **OpticalDielectric**, and **SolventName** options that were discussed above. The keyword **SolventAtom** requires multiple parameters, whereas all other keywords require only a single parameter. Whereas the dielectric constant is required for any PCM calculation (with a default value representing water), the nonelectrostatic parameters are optional and have only infrequently been added to these models.[44,82]

The nonelectrostatic interactions currently available in Q-CHEM are based on the work of Cossi *et al.*,[40] and are computed outside of the SCF procedure used to determine the electrostatic interactions. The nonelectrostatic energy is highly dependent on the input parameters and can be extremely sensitive to the radii chosen to define the solute cavity. Accordingly, the inclusion of nonelectrostatic interactions is highly empirical and probably needs to be considered on a case-by-case basis. Following Ref. 40, the cavitation energy is computed using the same solute cavity that is used to compute the electrostatic energy, whereas the dispersion/repulsion energy is computed using a SAS cavity definition.

The following keywords (in the *$solvent* section) are used to define nonelectrostatic parameters for PCM calculations.

SolventName	ε	ε_∞	SolventName	ε	ε_∞	SolventName	ε	ε_∞
acetic_acid	6.2	1.88	dimethoxyethane	7.3	1.90	iodobenzene	4.7	2.63
acetone	21.1	1.85	1-2-dimethoxyethane			isobutanol	18.1	1.95
acetonitrile	36.6	1.81	glyme			2-methyl-1-propanol		
acetylacetone	26.5	2.11	dimethylacetamide	40.2	2.07	isopropanol	20.2	1.90
pentane-2-4-dione			dimethylaminobenzene	5.0	2.43	2-propanol		
2-4-pentanedione			n-n-dimethylaniline			mesitylene	2.3	2.25
aniline	7.1	2.52	dimethylformamide	38.2	2.05	1-3-5-trimethylbenzene		
benzenamine			N-N-dimethylformamide			methanol	33.0	1.77
anisole	4.4	2.30	dimethylsulfoxide	47.2	2.18	methyl_acetate	6.8	1.85
methoxybenzene			DMSO			acetic_acid_methyl_ester		
benzaldehyde	17.8	2.39	dioxane	2.2	2.02	methyl_ethanoate		
benzene	2.3	2.25	1-4-dioxane			nitrobenzene	35.6	2.41
benzonitrile	25.6	2.34	ethanol	25.3	1.85	nonane	2.0	1.98
benzyl_alcohol	13.7	2.38	ethanolamine	31.9	2.12	n-nonane		
phenyl_methanol			ethyl_acetate	6.1	1.88	octanol	10.1	2.05
bromobenzene	5.5	2.43	ethyl_ethanoate			1-octanol		
butanol	18.1	1.96	acetic_acid_ethyl_ester			p-xylene	2.3	2.24
1-butanol			ethyl_acetoacetate	14.0	2.02	para-xylene		
butanol	17.1	1.95	ethyl-3-oxobutanoate			1-4-dimethylbenzene		
2-butanol			acetoacetic_acid_ethyl_ester			pentane	1.8	1.84
butanone	18.6	1.90	ethyl_benzoate	6.2	2.27	n-pentane		
2-butanone			benzoic_acid_ethyl_ester			1-pentanol	15.1	1.99
butan-2-one			decane	2.0	1.99	2-pentanol	13.4	1.98
methyl_ethyl_ketone			n-decane			3-pentanol	14.0	2.00
carbon_tetrachloride	2.2	2.13	ethyl_formate	8.7	1.85	2-pentanone	15.4	1.94
tetrachloromethane			ethyl_methanoate			pentan-2-one		
chlorobenzene	5.7	2.33	formic_acid_methyl_ester			methyl_propyl_ketone		
chloroform	4.8	2.08	ethylene_glycol	42.8	2.05	phenol	13.4	2.37
trichloromethane			1-2-ethanediol			1-propanol	20.8	1.92
cyclohexane	2.0	2.03	fluorobenzene	5.5	2.23	propanal	17.4	1.86
cyclohexanol	16.4	2.15	formamide	111.8	2.10	propionaldehyde		
cyclohexanone	16.1	2.11	formic_acid	51.1	1.88	pyridine	13.3	2.28
1-1-dichloroethane	10.1	2.01	heptane	1.9	1.93	t-butanol	12.0	1.92
1-2-dichloroethane	10.4	2.09	n-heptane			tert-butanol		
dichloromethane	8.9	2.02	heptanol	11.9	2.03	2-methyl-2-propanol		
methylene_chloride			1-heptanol			tetrahydrofuran	7.5	1.97
diethylamine	3.7	1.92	hexane	1.9	1.89	THF		
diethyl_ether	4.3	1.83	n-hexane			toluene	2.4	2.24
diethyl_ketone	17.0	1.95	hexanol	13.6	2.01	triethylamine	2.4	1.96
3-pentanone			1-hexanol			n-n-diethylethanamine		
pentan-3-one						water	80.2	1.78

Table 11.4: Named solvents available for PCM calculations. (Indented names are synonyms.) Values for ε correspond to the most recent measurements at 20°C (from Ref. 171) and values of $\varepsilon_\infty = n^2$ correspond to the most recent measurement at 20°C and $\lambda = 589$ nm (from Ref. 172). Setting **SolventName** to one of these values specifies the two dielectric constants but does not stipulate any nonelectrostatic interactions.

NonEls

Specifies what type of nonelectrostatic contributions to include.

INPUT SECTION: *$solvent*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

Cav	Cavitation energy
Buck	Buckingham dispersion and repulsion energy from atomic number
LJ	Lennard-Jones dispersion and repulsion energy from force field
BuckCav	Buck + Cav
LJCav	LJ + Cav

RECOMMENDATION:

A very limited set of parameters for the Buckingham potential is available at present.

NSolventAtoms

The number of different types of atoms.

INPUT SECTION: *$solvent*

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

N	Specifies that there are N different types of atoms.

RECOMMENDATION:

This keyword is necessary when **NonEls = Buck**, **LJ**, **BuckCav**, or **LJCav**. Methanol (CH_3OH), for example, has three types of atoms (C, H, and O).

SolventAtom

Specifies a unique solvent atom.

INPUT SECTION: *$solvent*

TYPE:

Various

DEFAULT:

None.

OPTIONS:

Input (TYPE)	Description
Number1 (INTEGER):	The atomic number of the atom
Number2 (INTEGER):	How many of this atom are in a solvent molecule
Number3 (INTEGER):	Force field atom type
SASradius (FLOAT):	Probe radius (in Å) for defining the solvent accessible surface

RECOMMENDATION:

If not using **LJ** or **LJCav**, then **Number3** should be set to 0. The **SolventAtom** keyword is necessary when **NonEls = Buck**, **LJ**, **BuckCav**, or **LJCav**

Temperature

Specifies the solvent temperature.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

300.0

OPTIONS:

T Use a temperature of T, in Kelvin.

RECOMMENDATION:

Used only for the cavitation energy.

Pressure

Specifies the solvent pressure.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

1.0

OPTIONS:

P Use a pressure of P, in bar.

RECOMMENDATION:

Used only for the cavitation energy.

SolventRho

Specifies the solvent number density

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

Determined for water, based on temperature.

OPTIONS:

ρ Use a density of ρ, in molecules/Å^3.

RECOMMENDATION:

Used only for the cavitation energy.

SolventRadius

The radius of a solvent molecule of the PCM solvent.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

None

OPTIONS:

r Use a radius of r, in Å.

RECOMMENDATION:

Used only for the cavitation energy.

The following example illustrates the use of the nonelectrostatic interactions.

Example 11.7 Optimization of trifluoroethanol in water using both electrostatic and nonelectrostatic PCM interactions. OPLS-AA parameters are used in the Lennard-Jones potential for dispersion and repulsion.

```
$molecule
   0 1
   C    -0.245826    -0.351674    -0.019873     23
   C     0.244003     0.376569     1.241371     22
   O     0.862012    -0.527016     2.143243     24
   F     0.776783    -0.909300    -0.666009     26
   F    -0.858739     0.511576    -0.827287     26
   F    -1.108290    -1.303001     0.339419     26
   H    -0.587975     0.878499     1.736246     27
   H     0.963047     1.147195     0.961639     27
   H     0.191283    -1.098089     2.489052     25
$end

$rem
   JOBTYPE        OPT
   BASIS          6-31G*
   METHOD         B3LYP
   SOLVENT_METHOD PCM
   FORCE_FIELD    OPLSAA
$end

$pcm
   Theory      CPCM
   Method      SWIG
   Solver      Inversion
   HeavyPoints 194
   HPoints     194
   Radii       Bondi
   vdwScale    1.2
$end

$solvent
   NonEls        LJCav
   NSolventAtoms 2
   SolventAtom   8 1 186 1.30
   SolventAtom   1 2 187 0.01
   SolventRadius 1.35
   Temperature   298.15
   Pressure      1.0
   SolventRho    0.03333
   Dielectric    78.39
$end
```

11.2.4.5 Job Control and Examples for Non-Equilibrium Solvation

The **OpticalDielectric** keyword in the *$solvent* section is needed for nonequilibrium calculations, or it can be set to a predefined value by specifying **SolventName** instead. The LR-PCM excitation energy is automatically calculated whenever the CIS/TDDFT calculations are performed with PCM, but it is turned off while the perturbation scheme is employed.

ChargeSeparation

Partition fast and slow charges in solvent equilibrium state

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

No default.

OPTIONS:

Marcus Do slow-fast charge separation with the Marcus partition.

Pekar Do slow-fast charge separation with the Pekar partition.

RECOMMENDATION:

Charge separation is used in conjunction with the **StateSpecific** keyword in *$pcm*.

StateSpecific

Specifies which the state-specific PCM will be used.

INPUT SECTION: *$pcm*

TYPE:

Various

DEFAULT:

No default.

OPTIONS:

Perturb Perform ptSS and ptLR for vertical excitations.

External Perform self-consistent EI-SS-PCM to the excited state (for emission).

Internal Perform self-consistent II-SS-PCM to the excited state (for emission or excitation).

RECOMMENDATION:

Use for vertical excition energies ptSS-PCM or in very polar cases the nonequilibrium II-SS-PCM, and equilibrium EI-SS-PCM for emission energies of long-lived excited states.

Td_CEq_NonEq

Specify the self-consistent SS-PCM/TDDFT method to calculate the solvent effects on vertical absorption and emission in solution based on the constrained equilibrium principle for nonequilibrium solvation.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 Calculate nonequilibrium excited-state free energy in absorption based on the equilibrium ground-state reaction field via an RPA calculation at the ground-state geometry.

2 Calculate nonequilibrium excited-state free energy in absorption based on the nonequilibrium excited-state reaction field via an RPA calculation at the ground-state geometry.

3 Calculate equilibrium excited-state free energy in emission based on the equilibrium excited-state reaction field via an RPA calculation at the excited-state geometry.

4 Calculate nonequilibrium ground-state free energy in emission based on the nonequilibrium ground-state reaction field via a pure single-point energy calculation at the excited-state geometry.

RECOMMENDATION:

Option 2 and 3 need to be iterated until the excited-state reaction field converges. Option 1 is the first iteration of option 2. After the equilibrium excited state reaction field of option 3 converges, option 4 is executed.

Example 11.8 LR-TDDFT/LR-C-PCM low-lying vertical excitation energy

```
$molecule
   0 1
   C    0  0  0.0
   O    0  0  1.21
$end

$rem
   EXCHANGE            B3LYP
   CIS_N_ROOTS         10
   CIS_SINGLETS        TRUE
   CIS_TRIPLETS        TRUE
   RPA                 TRUE
   BASIS               6-31+G*
   SOLVENT_METHOD      PCM
$end

$pcm
   Theory      CPCM
   Method      SWIG
   Solver      Inversion
   Radii       Bondi
$end

$solvent
   Dielectric         78.39
   OpticalDielectric  1.777849
$end
```

Example 11.9 PCM solvation effects on the vertical excitation energies of planar DMABN using the ptSS and ptLR methods.

```
$molecule
   0  1
   C     0.000046   -0.000398    1.904953
   C     1.210027    0.000379    1.186051
   C     1.214640   -0.000065   -0.194515
   C     0.000164   -0.000616   -0.933832
   C    -1.214349   -0.001557   -0.194687
   C    -1.209753   -0.001846    1.185775
   H     2.151949    0.001377    1.722018
   H     2.164371    0.000481   -0.709640
   H    -2.164082   -0.002008   -0.709781
   H    -2.151763   -0.002287    1.721615
   C    -0.000227    0.001061    3.325302
   N    -0.000475    0.002405    4.484321
   N     0.000053   -0.000156   -2.297372
   C    -1.258656    0.001284   -3.036994
   H    -1.041042    0.001615   -4.102376
   H    -1.860897   -0.885647   -2.811117
   H    -1.859247    0.889133   -2.810237
   C     1.258563   -0.000660   -3.037285
   H     1.860651    0.886208   -2.810755
   H     1.859362   -0.888604   -2.811461
   H     1.040664   -0.000097   -4.102609
$end

$rem
   EXCHANGE              LRC-wPBEPBE
   OMEGA                 260
   BASIS                 6-31G*
   CIS_N_ROOTS           10
   RPA                   2
   CIS_SINGLETS          1
   CIS_TRIPLETS          0
   CIS_RELAXED_DENSITY   TRUE
   SOLVENT_METHOD        PCM
$end

$pcm
   NonEquilibrium
   Theory                IEFPCM
   StateSpecific         Perturb
$end

$solvent
   Dielectric            35.688000       ! Acetonitrile
   OpticalDielectric     1.806874
$end
```

Example 11.11.10 TDDFT with External Iteration SS-PCM for nitrate in acetonitrile.

View input online

Example 11.11.11 TDDFT with Internal Iteration SS-PCM and the additional ground state calculation for vertical emission energies of nitrate in acetonitrile.

View input online

Example 11.11.12 SS-PCM/TDDFT low-lying vertical excitation energy based on the constrained equilibrium principle.

View input online

Example 11.11.13 SS-PCM/TDDFT equilibrium excited-state free energy in emission based on the constrained equilibrium principle.

View input online

Example 11.14 SS-PCM/TDDFT nonequilibrium ground-state free energy in emission based on the constrained equilibrium principle.

```
$molecule
   0 1
   C     0.00000000   -0.57624800    0.00000000
   H    -0.95773500   -1.10813200    0.00000000
   H     0.95773500   -1.10813200    0.00000000
   O     0.00000000    0.70921900    0.00000000
$end

$rem
   METHOD           M062X
   BASIS            6-31+G(d)
   SOLVENT_METHOD   PCM
   POP_MULLIKEN     -1
   DFT_D            D3
$end

$pcm
   Theory            SSVPE
   Method            SWIG
   ChargeSeparation  Pekar
   StateSpecific     Pekar
   Td_CEq_NonEq      4
$end

$solvent
   Dielectric 78.5
   OpticalDielectric 1.778
$end
```

11.2.5 Linear-Scaling QM/MM/PCM Calculations

Recall that PCM electrostatics calculations require the solution of the set of linear equations given in Eq. (11.2), to determine the vector $\mathbf{q}$ of apparent surface charges. The precise forms of the matrices $\mathbf{K}$ and $\mathbf{R}$ depend upon the particular PCM (Table 11.3), but in any case they have dimension $N_{\text{grid}} \times N_{\text{grid}}$, where N_{grid} is the number of Lebedev grid points used to discretize the cavity surface. Construction of the matrix $\mathbf{K}^{-1}\mathbf{R}$ affords a numerically exact solution to Eq. (11.2), whose cost scales as $\mathcal{O}(N_{\text{grid}}^3)$ in CPU time and $\mathcal{O}(N_{\text{grid}}^2)$ in memory. This cost is exacerbated by smooth PCMs, which discard fewer interior grid points so that N_{grid} tends to be larger, for a given solute, as compared to traditional discretization schemes. For QM solutes, the cost of inverting $\mathbf{K}$ is usually negligible relative to the cost of the electronic structure calculation, but for the large values of N_{grid} that are encountered in MM/PCM or QM/MM/PCM jobs, the $\mathcal{O}(N_{\text{grid}}^3)$ cost of inverting $\mathbf{K}$ is often prohibitively expensive.

To avoid this bottleneck, Lange and Herbert[63] have developed an iterative conjugate gradient (CG) solver for Eq. (11.2) whose cost scales as $\mathcal{O}(N_{\text{grid}}^2)$ in CPU time and $\mathcal{O}(N_{\text{grid}})$ in memory. A number of other cost-saving options are

available, including efficient pre-conditioners and matrix factorizations that speed up convergence of the CG iterations, and a fast multipole algorithm for computing the electrostatic interactions.[97] Together, these features lend themselves to a solution of Eq. (11.2) whose cost scales as $\mathcal{O}(N_{\text{grid}})$ in both memory and CPU time, for sufficiently large systems.[63] Currently, these options are available only for C-PCM, not for SS(V)PE or IEF-PCM.

Set **Solver** to **CG** in the *$pcm* section in order to invoke the CG solver, as described in Section 11.2.4.1. Additional options for the CG solver are listed below, each of which should also be specified in the *$pcm* input section. Researchers who use this feature are asked to cite the original SwiG-PCM references[86,88] as well as the reference for the CG solver.[63]

CGThresh

The threshold for convergence of the conjugate gradient solver.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n Conjugate gradient converges when the maximum residual is less than 10^{-n}.

RECOMMENDATION:

The default typically affords PCM energies on par with the precision of matrix inversion for small systems. For systems that have difficulty with SCF convergence, one should increase n or try the matrix inversion solver. For well-behaved or very large systems, a smaller n might be permissible.

DComp

Controls decomposition of matrices to reduce the matrix norm for the CG Solver.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Turns off matrix decomposition

1 Turns on matrix decomposition

3 Option 1 plus only stores upper half of matrix and enhances gradient evaluation

RECOMMENDATION:

None

PreCond

Controls the use of the pre-conditioner for the CG solver.

INPUT SECTION: *$pcm*

TYPE:

None

DEFAULT:

Off

OPTIONS:

No options. Specify the keyword to enable pre-conditioning.

RECOMMENDATION:

A Jacobi block-diagonal pre-conditioner is applied during the conjugate gradient algorithm to improve the rate of convergence. This reduces the number of CG iterations, at the expense of some overhead. Pre-conditioning is generally recommended for large systems.

NoMatrix

Specifies whether PCM matrices should be explicitly constructed and stored.

INPUT SECTION: *$pcm*

TYPE:

None

DEFAULT:

Off

OPTIONS:

No options. Specify the keyword to avoid explicit construction of PCM matrices.

RECOMMENDATION:

Storing the PCM matrices requires $\mathcal{O}(N_{\text{grid}}^2)$ memory. If this is prohibitive, the **NoMatrix** option forgoes explicit construction of the PCM matrices, and instead constructs the matrix elements as needed, reducing the memory requirement to $\mathcal{O}(N_{\text{grid}})$ at the expense of additional computation.

UseMultipole

Controls the use of the adaptive fast multipole method in the CG solver.

INPUT SECTION: *$pcm*

TYPE:

None

DEFAULT:

Off

OPTIONS:

No options. Specify the keyword in order to enable the fast multipole method.

RECOMMENDATION:

The fast multipole approach formally reduces the CPU time to $\mathcal{O}(N_{\text{grid}})$, but is only beneficial for spatially extended systems with several thousand cavity grid points. Requires the use of **NoMatrix**.

MultipoleOrder

Specifies the highest multipole order to use in the FMM.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n The highest order multipole in the multipole expansion.

RECOMMENDATION:

Increasing the multipole order improves accuracy but also adds more computational expense. The default yields satisfactory performance in common QM/MM/PCM applications.

Theta

The multipole acceptance criterion.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

0.6

OPTIONS:

n A number between zero and one.

RECOMMENDATION:

The default is recommended for general usage. This variable determines when the use of a multipole expansion is valid. For a given grid point and box center in the FMM, a multipole expansion is accepted when $r/d \leq$ **Theta**, where d is the distance from the grid point to the box center and r is the radius of the box. Setting **Theta** to one will accept all multipole expansions, whereas setting it to zero will accept none. If not accepted, the grid point's interaction with each point inside the box is computed explicitly. A low **Theta** is more accurate but also more expensive than a higher **Theta**.

NBox

The FMM boxing threshold.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n The maximum number of grid points for a leaf box.

RECOMMENDATION:

The default is recommended. This option is for advanced users only. The adaptive FMM boxing algorithm divides space into smaller and smaller boxes until each box has less than or equal to **NBox** grid points. Modification of the threshold can lead to speedup or slowdown depending on the molecular system and other FMM variables.

A sample input file for the linear-scaling QM/MM/PCM methodology can be found in the `$QC/samples` directory, under the name `pcm_qmmm_crambin.in`. This sample involves a QM/MM description of a protein (crambin) in

which a single tyrosine side chain is taken to be the QM region. The entire protein is immersed in a dielectric using C-PCM with SwiG discretization.

11.2.6 Isodensity Implementation of SS(V)PE

11.2.6.1 Basic Job Control

As discussed above, results obtained various types of PCMs are quite sensitive to the details of the cavity construction. Q-CHEM's implementation of PCMs, using Lebedev grids, simplifies this construction somewhat, but leaves the radii of the atomic spheres as empirical parameters (albeit ones for which widely-used default values are provided). An alternative implementation of the SS(V)PE solvation model is also available,[32] which attempts to further eliminate empiricism associated with cavity construction by taking the cavity surface to be a specified iso-contour of the solute's electron density. [We call this the isodensity implementation of SS(V)PE in Table 11.3, and it is based on Chipman's "symmetrized" form of the $\mathbf{K}$ matrix,[32,88] although the difference between symmetric and asymmetric forms is essentially negligible when an isodensity cavity construction is used.[32]] In this case, the cavity surface is discretized by projecting a single-center Lebedev grid onto the iso-contour surface. Unlike the PCM implementation discussed in Section 11.2.3, for which point-group symmetry is disabled, this implementation of SS(V)PE supports full symmetry for all Abelian point groups. The larger and/or the less spherical the solute molecule is, the more points are needed to get satisfactory precision in the results. Further experience will be required to develop detailed recommendations for this parameter. Values as small as 110 points are usually sufficient for diatomic or triatomic molecules. The default value of 1202 points is adequate to converge the energy within 0.1 kcal/mol for solutes the size of mono-substituted benzenes.

Energy gradients are also not available for this implementation of SS(V)PE, although they are available for the implementation described in Section 11.2.3 in which the cavity is constructed from atom-centered spheres. As with the PCMs discussed in that section, the solute may be described using Hartree-Fock theory or DFT; post-Hartree–Fock correlated wave functions can also take advantage of molecular orbitals that are polarized using SS(V)PE. Researchers who use the isodensity SS(V)PE feature are asked to cite Ref. 29.

In related work, Pomogaeva and Chipman[128–131] recently introduced a "composite method for implicit representation of solvent" (CMIRS) that is based on SS(V)PE electrostatics but adds nonelectrostatic terms. This model is available in Q-CHEM[177] and is discussed in Section 11.2.7. In its current implementation, CMIRS requires an isodensity SS(V)PE calculation, However, the current implementation computes the nonelectrostatic interactions using the cavity and the solute's charge density generated from the isodensity SS(V)PE. To use the CMIRS model, an isodensity SS(V)PE calculation must be requested (as described below), and the keywords **IDEFESR** and **IPnRF** must both be set to 1 in the *$svp* input section. The CMIRS model is further described in Section 11.2.6.2.

An isodensity SS(V)PE calculation is requested by setting SOLVENT_METHOD = ISOSVP in the *$rem* section, in addition to normal job control variables for a single-point energy calculation. Whereas the other solvation models described in this chapter use specialized input sections (*e.g.*, *$pcm*) in lieu of a slew of *$rem* variables, the isodensity SS(V)PE code is an interface between Q-CHEM and a code written by Chipman,[32] so some *$rem* variables are used for job control of isodensity SS(V)PE calculations. These are listed below.

SVP_MEMORY

Specifies the amount of memory for use by the solvation module.

TYPE:

INTEGER

DEFAULT:

125

OPTIONS:

n corresponds to the amount of memory in MB.

RECOMMENDATION:

The default should be fine for medium size molecules with the default Lebedev grid, only increase if needed.

SVP_PATH

Specifies whether to run a gas phase computation prior to performing the solvation procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Run a gas-phase calculation and upon convergence run the SS(V)PE computation.

1 Do not run a gas-phase calculation.

RECOMMENDATION:

Running the gas-phase calculation provides a good guess to start the solvation stage and provides a more complete set of solvated properties.

SVP_CHARGE_CONV

Determines the convergence value for the charges on the cavity. When the change in charges fall below this value, if the electron density is converged, then the calculation is considered converged.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n Convergence threshold set to 10^{-n}.

RECOMMENDATION:

The default value unless convergence problems arise.

SVP_CAVITY_CONV

Determines the convergence value of the iterative isodensity cavity procedure.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Convergence threshold set to 10^{-n}.

RECOMMENDATION:

The default value unless convergence problems arise.

SVP_GUESS

Specifies how and if the SS(V)PE model will use a given guess for the charges and cavity points.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No guessing.

1 Read a guess from a previous isodensity SS(V)PE calculation.

2 Use a guess specified by the *$svpirf* section from the input.

RECOMMENDATION:

It is helpful to also set SCF_GUESS to READ when using a guess from a previous Q-CHEM run. Set **IPnRF** = 1 in the *$svp* section of the previous Q-CHEM job, in order to save the reaction field for subsequent reading (see Section 11.2.6.2).

This last *$rem* variable requires specification of a *$svpirf* input section, the format for which is the following:

```
$svpirf
   <# point> <x point>  <y point>  <z point>  <charge> <grid weight>
   <# point> <x normal> <y normal> <z normal>
$end
```

11.2.6.2 The *$svp* Input Section

More refined control over SS(V)PE jobs is obtained using a *$svp* input section. These are read directly by Chipman's SS(V)PE solvation module and therefore must be specified in the context of a Fortran namelist. The format is as follows:

```
$svp
   <KEYWORD>=<VALUE>, <KEYWORD>=<VALUE>,...
   <KEYWORD>=<VALUE>
$end
```

For example, the section may look like this:

```
$svp
   RHOISO=0.001, DIELST=78.39, NPTLEB=110
$end
```

The following keywords are supported in the *$svp* section:

DielSt

The static dielectric constant.

INPUT SECTION: *$svp*

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

real number specifying the constant.

RECOMMENDATION:

The default value 78.39 is appropriate for water solvent.

IDEFESR

Specifies whether to request a CMIRS calculation.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not invoke a CMIRS calculation.

1 Invoke a CMIRS calculation.

RECOMMENDATION:

None. Need to set **IPnRF** = 1 in the *$svp* section if CMIRS is desired.

IPnRF

Specifies whether to save the SS(V)PE reaction field.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not save the reaction field.

1 Save the reaction field.

RECOMMENDATION:

None. **IPnRF** = 1 is required for CMIRS (*i.e.*, if **IDEFESR** = 1).

IShape

A flag to set the shape of the cavity surface.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 use the electronic isodensity surface.

1 use a spherical cavity surface.

RECOMMENDATION:

Use the default surface.

RhoIso

Value of the electronic isodensity contour used to specify the cavity surface. (Only relevant for **IShape** = 0.)

INPUT SECTION: *$svp*

TYPE:

FLOAT

DEFAULT:

0.001

OPTIONS:

Real number specifying the density in electrons/bohr^3.

RECOMMENDATION:

The default value is optimal for most situations. Increasing the value produces a smaller cavity which ordinarily increases the magnitude of the solvation energy.

RadSph

Sphere radius used to specify the cavity surface (Only relevant for **IShape**=1.)

INPUT SECTION: *$svp*

TYPE:

FLOAT

DEFAULT:

Half the distance between the outermost atoms plus 1.4 Å.

OPTIONS:

Real number specifying the radius in Bohr (if positive) or in Å(if negative).

RECOMMENDATION:

Make sure that the cavity radius is larger than the length of the molecule.

IntCav

A flag to select the surface integration method.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Single center Lebedev integration.

1 Single center spherical polar integration.

RECOMMENDATION:

The Lebedev integration is by far the more efficient.

NPtLeb

The number of points used in the Lebedev grid for the single-center surface integration. (Only relevant if **IntCav** = 0.)

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

1202

OPTIONS:

Valid choices are: 6, 18, 26, 38, 50, 86, 110, 146, 170, 194, 302, 350, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, or 5294.

RECOMMENDATION:

The default value has been found adequate to obtain the energy to within 0.1 kcal/mol for solutes the size of mono-substituted benzenes.

NPtThe, NPtPhi

The number of (θ,ϕ) points used for single-centered surface integration (relevant only if **IntCav** = 1.)

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

8,16

OPTIONS:

θ,ϕ specifying the number of points.

RECOMMENDATION:

These should be multiples of 2 and 4 respectively, to provide symmetry sufficient for all Abelian point groups. Defaults are too small for all but the tiniest and simplest solutes.

LinEq

Flag to select the method for solving the linear equations that determine the apparent point charges on the cavity surface.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 use LU decomposition in memory if space permits, else switch to **LinEq** = 2

1 use conjugate gradient iterations in memory if space permits, else use **LinEq** = 2

2 use conjugate gradient iterations with the system matrix stored externally on disk.

RECOMMENDATION:

The default should be sufficient in most cases.

CvgLin

Convergence criterion for solving linear equations by the conjugate gradient iterative method (relevant if **LinEq** = 1 or 2.)

INPUT SECTION: *$svp*

TYPE:

FLOAT

DEFAULT:

1.0E-7

OPTIONS:

Real number specifying the actual criterion.

RECOMMENDATION:

The default value should be used unless convergence problems arise.

Note that the single-center surface integration approach that is used to find the isodensity surface may fail for certain very non-spherical solute molecules. The program will automatically check for this, aborting with a warning message if necessary. The single-center approach succeeds only for what is called a "star surface", meaning that an observer sitting at the center has an unobstructed view of the entire surface. Said another way, for a star surface any ray emanating out from the center will pass through the surface only once. Some cases of failure may be fixed by simply moving to a new center with the **ITrnGr** parameter described below. But some surfaces are inherently non-star surfaces and cannot be treated with this program until more sophisticated surface integration approaches are developed and implemented.

ITrnGr

Translation of the cavity surface integration grid.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 No translation (*i.e.*, center of the cavity at the origin of the atomic coordinate system)

1 Translate to the center of nuclear mass.

2 Translate to the center of nuclear charge.

3 Translate to the midpoint of the outermost atoms.

4 Translate to midpoint of the outermost non-hydrogen atoms.

5 Translate to user-specified coordinates in Bohr.

6 Translate to user-specified coordinates in Ångstroms.

RECOMMENDATION:

The default value is recommended unless the single-center integrations procedure fails. If cavity construction fails for a small, high-symmetry system, consider setting **ITrnGr** = 0.

TranX, TranY, TranZ

x, y, and z value of user-specified translation (only relevant if **ITrnGr** is set to 5 or 6).

INPUT SECTION: *$svp*

TYPE:

FLOAT

DEFAULT:

0, 0, 0

OPTIONS:

x, y, and z relative to the origin in the appropriate units.

RECOMMENDATION:

None.

IRotGr

Rotation of the cavity surface integration grid.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 No rotation.

1 Rotate initial xyz axes of the integration grid to coincide with principal moments of nuclear inertia (relevant if **ITrnGr** = 1)

2 Rotate initial xyz axes of integration grid to coincide with principal moments of nuclear charge (relevant if **ITrnGr** = 2)

3 Rotate initial xyz axes of the integration grid through user-specified Euler angles as defined by Wilson, Decius, and Cross.

RECOMMENDATION:

The default is recommended unless the knowledgeable user has good reason otherwise. If cavity construction fails for a small, high-symmetry system, consider setting **IRotGr** = 0.

RotThe RotPhi RotChi

Euler angles (θ, ϕ, χ) in degrees for user-specified rotation of the cavity surface (relevant if **IRotGr** = 3).

INPUT SECTION: *$svp*

TYPE:

FLOAT

DEFAULT:

0,0,0

OPTIONS:

θ, ϕ, χ in degrees

RECOMMENDATION:

None.

IOpPrd

Specifies the choice of system operator-product form.

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Symmetric form.

1 Non-symmetric form.

RECOMMENDATION:

The default uses more memory but is generally more efficient, we recommend its use unless there is shortage of memory available.

By default, Q-CHEM will check the validity of the single-center expansion by searching for the isodensity surface in two different ways: first, working inwards from a large distance, and next by working outwards from the origin. If the same result is obtained (within tolerances) using both procedures, then the cavity is accepted. If the two results do not

agree, then the program exits with an error message indicating that the inner isodensity surface is found to be too far from the outer isodensity surface.

Note: For some small molecules (such as atoms and diatomics), construction of the isodensity cavity may fail with the default parameters. In such cases, we recommend setting **ITrnGr** = 0 and also **IRotGr** = 0, with the molecule centered at the origin of the grid (*i.e.*, in the standard nuclear orientation).

Some molecules, for example C_{60}, can have a hole in the middle. Such molecules have two different "legal" isodensity surfaces, a small inner one inside the "hole", and a large outer one that is the desired surface for solvation. In such cases, the cavity check described in the preceding paragraph causes the program to exit. To avoid this, one can consider turning off the cavity check that works out from the origin, leaving only the outer cavity determined by working in from large distances.

ICvICk

Specifies whether to perform cavity check

INPUT SECTION: *$svp*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 no cavity check, use only the outer cavity

1 cavity check, generating both the inner and outer cavities and compare.

RECOMMENDATION:

Consider turning off cavity check only if the molecule has a hole and if a star (outer) surface is expected.

11.2.7 Composite Method for Implicit Representation of Solvent (CMIRS)

Whereas PCMs, including sophisticated ones like SS(V)PE and IEF-PCM, account for long-range solute–solvent interactions, an accurate model for free energies of solvation must also include a treatment of short-range, nonelectrostatic interactions. Various models decompose these interactions in different ways, but usually the nonelectrostatic terms attempt to model all or most of the following: solute–solvent dispersion (van der Waals) interactions, Pauli (exchange) repulsion between solute and solvent, the work associated with forming the solute cavity within the dielectric medium (the so-called "cavitation energy"), hydrogen-bonding and other specific interactions due to the molecular structure of the solvent, and changes in the structure (and therefore the entropy) of the neat solvent upon introduction of the solute

Pomogaeva and Chipman[128–131] have introduced an implicit solvation model that attempts to model these nonelectrostatic interactions alongside a PCM-style treatment of the bulk electrostatics. They call this approach the "composite method for implicit representation of solvent" (CMIRS), and it consists first of a self-consistent treatment of solute–continuum electrostatics using the SS(V)PE model (Section 11.2.6). To this electrostatics calculation, CMIRS adds a solute–solvent dispersion term that is modeled upon the non-local VV09 van der Waals dispersion density functional,[164] a Pauli repulsion contribution that depends upon the tail of the solute's electron density that extends beyond the solute cavity, and a hydrogen-bonding correction based on the maximum and minimum values of the normal component of the electric field generated by the solute at the cavity surface. The Gibbs free energy of solvation is thus modeled as

$$\Delta G_{\text{CMIRS}}(\rho_s) = \Delta G_{\text{SS(V)PE}}(\rho_s) + \Delta G_{\text{DEFESR}}(\rho_s) \,, \tag{11.22}$$

where $\Delta G_{\text{SS(V)PE}}$ is the continuum electrostatics contribution from the SS(V)PE model, which is based on a solute cavity defined as an isocontour ρ_s of the solute's charge density. The second term contains the short-range dispersion,

Solvent	SolvRho	A	B	C	D	Gamma
Benzene	0.0421	−0.00522	0.01294			
Cyclohexane	0.0396	−0.00938	0.03184			
DMSO	0.05279	−0.00951	0.044791		−162.07	4.1
CH_3CN	0.03764	−0.008178	0.045278		−0.33914	1.3
Water	0.05	−0.006736	0.032698	−1249.6	−21.405	3.7

Table 11.5: Optimized CMIRS parameters (from Ref. 177) using **RHOISO** = 0.0010 a.u. and, for ions, the proton reference ΔG value from Ref. 78.

Solvent	SolvRho	A	B	C	D	Gamma
Benzene	0.0421	−0.00572	0.01116			
Cyclohexane	0.0396	−0.00721	0.05618			
DMSO	0.05279	−0.002523	0.011757		−817.93	4.3
CH_3CN	0.03764	−0.003805	0.03223		−0.44492	1.2
Water	0.05	−0.006496	0.050833	−566.7	−30.503	3.2

Table 11.6: Optimized CMIRS parameters (from Ref. 177) using **RHOISO** = 0.0005 a.u. and, for ions, the proton reference ΔG value from Ref. 78.

exchange, and "field-extremum short-range" (DEFESR) interactions:

$$\Delta G_{\text{DEFESR}} = \Delta G_{\text{disp}} + \Delta G_{\text{exch}} + \Delta G_{\text{FESR}} \ . \tag{11.23}$$

These terms are evaluated only once, using a converged charge density for the solute from a SS(V)PE calculation.

The CMIRS approach was implemented in Q-CHEM by Zhi-Qiang You and John Herbert.[177] In the course of this work, a serious error was discovered in the the original implementation of ΔG_{disp} by Pomogaeva and Chipman, in the GAMESS program. Although reparameterization of a corrected version of the model leads to only small changes in the overall error statistics across a large database of experimental free energies of solvation, the apportionment between energy components in Eq. (11.23) changes significantly.[177] By request of Dan Chipman, the Q-CHEM implementation is termed "CMIRS v. 1.1", reserving v. 1.0 for the original GAMESS implementation of Pomogaeva and Chipman.

The CMIRS model is independently parametrized for each solvent of interest, but uses no more than five empirical parameters per solvent. It is presently available in Q-CHEM for water, acetonitrile, dimethyl sulfoxide, benzene, and cyclohexane. Error statistics for ΔG compare very favorably to those of the SMx models that are described in Section 11.2.9, *e.g.*, mean unsigned errors < 0.7 kcal/mol in benzene and cyclohexane and < 1.5 kcal/mol in water. The latter statistic includes challenging ionic solutes; errors for charge-neutral aqueous solutes are smaller still.[177]

The current implementation of CMIRS in Q-CHEM computes the electrostatic energy using Chipman's isodensity SS(V)PE module. The resulting isodensity cavity and the solute charge density are then employed in the calculation of the DEFESR interactions. To request a CMIRS calculation, users must set IDEFESR = 1 in an isodensity SS(V)PE calculation (see Section 11.2.6.2). The solvent-dependent empirical parameters **A**, **B**, **C**, **D**, **Gamma** in the CMIRS model need to be specified in the *$pcm_nonels* section. Three additional parameters are also required. One is the damping parameter **Delta** in the dispersion equation. We recommend the parameter fixed at $\delta = 7$ a.u. (about 3.7 Å), an optimized value that only considers dispersion at intermolecular distances larger than van der Waals contact distance. The second is solvent's average electron density **SolvRho**. The last one is the number of Gauss–Laguerre points **GauLag_N** for the integration over the solvent region in the exchange equation. We recommend 40 grid points for efficient integration with accuracy. Optimized parameters for the supported solvents are listed in Tables 11.5 and 11.6 for two different values of ρ_s.[177]

Example 11.15 CMIRS calculation for methane in the water solvent.

```
$molecule
   0  1
   C     0.000000     0.000005     0.000000
   H     0.748558     0.801458     0.000000
   H     0.506094    -0.972902     0.000000
   H    -0.627326     0.085706     0.895425
   H    -0.627326     0.085706    -0.895425
$end

$rem
   EXCHANGE        = b3lyp5
   BASIS           = 6-31+g*
   SCF_CONVERGENCE = 8
   MEM_TOTAL       = 4000
   MEM_STATIC      = 400
point_group_symmetry = False
   XC_GRID         = 000096000974
$end

@@@

$rem
   EXCHANGE        = b3lyp5
   BASIS           = 6-31+g*
   SCF_CONVERGENCE = 8
   MAX_SCF_CYCLES  = 100
   SOLVENT_METHOD  = isosvp
   SCF_GUESS       = read
   MEM_TOTAL       = 4000
   MEM_STATIC      = 400
   SVP_MEMORY      = 1000
point_group_symmetry = False
   XC_GRID         = 000096000974
$end

$molecule
   read
$end

$svp
   RHOISO=0.001, DIELST=78.36, NPTLEB=974,ITRNGR=2, IROTGR=2, IPNRF=1, IDEFESR=1
$end

$pcm_nonels
   A         -0.006736
   B          0.032698
   C      -1249.6
   D        -21.405
   Delta    7.0
   Gamma    3.7
   SolvRho  0.05
   GauLag_N 40
$end
```

11.2.8 COSMO

According to Table 11.3, COSMO and C-PCM appear to differ only in the dielectric screening factor, f_ε in Eq. (11.3). Indeed, surface charges in either model are computed according to

$$\mathbf{q} = -f_\varepsilon \mathbf{S}^{-1}\mathbf{v} \; . \tag{11.24}$$

As discussed in Section 11.2.4, the user can choose between various values of f_ε, including the original value $f_\varepsilon = (\varepsilon - 1)/(\varepsilon + 1/2)$ that was suggested by Klamt and co-workers,[80,81] or else $f_\varepsilon = (\varepsilon - 1)/\varepsilon$ as is typically used in C-PCM calculations.[41,88,162]. More importantly, however, COSMO differs from C-PCM in that the former includes an *ad hoc* correction for outlying charge that goes beyond Eq. (11.24), whereas C-PCM consists of nothing more than induced surface charges computed (self-consistently) according to Eq. (11.24). This correction, which is common to many implementations of COSMO,[62] involves the use of two separate solute cavities. It is worth noting that Eq. (11.24) was later shown to *implicitly* include an outlying charge correction,[28] by virtue of the fact that it is derivable from the SS(V)PE model,[62,87] and the latter was developed specifically with an eye towards the treatment of outlying charge. As such, there is little theoretical justification for the additional *explicit* correction for outlying charge, despite its success in practice.[83] See Ref. 62 for a discussion of these issues.

In any case, the nature of the *a posteriori* correction for the outlying charge proceeds as follows. Upon solution of Eq. (11.24), the outlying charge correction in COSMO[5,80] is obtained by first defining a larger cavity that is likely to contain essentially all of the solute's electron density; in practice, this typically means using atomic radii of $1.95R$, where R denotes the original atomic van der Waals radius that was used to compute $\mathbf{q}$. (Note that unlike the PCMs described in Sections 11.2.3 and 11.2.4, where the atomic radii have default values but a high degree of user-controllability is allowed, the COSMO atomic radii are parameterized for this model and are fixed.) A new set of charges, $\mathbf{q}' = -f_\varepsilon(\mathbf{S}')^{-1}\mathbf{v}'$, is then computed on this larger cavity surface, and the charges on the *original* cavity surface are adjusted to new values, $\mathbf{q}'' = \mathbf{q} + \mathbf{q}'$. Finally, a corrected electrostatic potential on the original surface is computed according to $\mathbf{v}'' = -f_\varepsilon \mathbf{S}\mathbf{q}''$. It is this potential that is used to compute the solute–continuum electrostatic interaction (polarization energy), $G_{\mathrm{pol}} = \frac{1}{2}\sum_i q_i'' v_i''$. (For comparison, when the C-PCM approach described in Section 11.2.3 is used, the electrostatic polarization energy is $G_{\mathrm{pol}} = \frac{1}{2}\sum_i q_i v_i$, computed using the original surface charges $\mathbf{q}$ and surface electrostatic potential $\mathbf{v}$.) With this outlying charge correction, Q-CHEM's implementation of COSMO resembles the one in TURBOMOLE.[140]

A COSMO calculation is requested by setting SOLVENT_METHOD = COSMO in the *$rem* section, in addition to normal job control variables. The keyword **Dielectric** in the *$solvent* section is used to set the solvent's static dielectric constant, as described above for other solvation models.

11.2.9 SMx Models

The SMx models were developed by Cramer, Truhlar, and coworkers at the University of Minnesota. Versions SM8,[108] SM12,[112] and SMD[109] are available in Q-CHEM. Of these, SMD is recommended because it is designed to work with arbitrary basis sets (unlike SM8) and it is implemented with both analytic gradients and Hessians (unlike SM8 or SM12).[14]

Each of these SMx methods is designed as a "universal" solvation model,[43] in the sense that it can be applied to any solvent for which a small of descriptors is known. The solvent descriptors are:

- dielectric constant
- refractive index
- bulk surface tension

- acidity on the Abraham scale
- basicity on the Abraham scale
- carbon aromaticity, which equals the fraction of non-hydrogenic solvent atoms that are aromatic carbon atoms
- electronegative halogenicity, which equals the fraction of non-hydrogenic solvent atoms that are F, Cl, or Br).

These models consist of a generalized Born treatment of continuum electrostatic interactions, along with nonelectrostatic interactions that are parameterized in terms of atomic surface tensions. The nonelectrostatic interactions include cavitation, dispersion, and changes in solvent structure, and the treatment of these nonelectrostatic effects is crucial to obtaining accurate (free) energies of solvation.

An SMx calculation is requested by setting SOLVENT_METHOD = SM8, SM12, or SMD. Some method-specific keywords are required (in a *$smx* input section) for some of these models, and these are discussed in the sections that follow. At a minimum, each of these models that the solvent be specified in the *$smx* section unless that solvent is water. Available solvents are listed in Table 11.7. These names should be given in the *$smx* section without spaces or hyphens, so that propanoic acid from Table 11.7 becomes `propanoicacid` and 1-hexanol becomes `1hexanol`.

Table 11.7: Solvents available for SMx calculations.[a]

1,1,1-trichloroethane	bromoethane	m-ethylbenzoate
1,1,2-trichloroethane	bromooctane	m-ethylethanoate
1,1-dichloroethane	butanal	m-ethylmethanoate
1,2,4-trimethylbenzene	butanoicacid	m-ethylphenylketone
1,4-dioxane	butanone	m-ethylpropanoate
1-bromo-2-methylpropane	butanonitrile	m-ethylbutanoate
1-bromopentane	butylethanoate	m-ethylcyclohexane
1-bromopropane	butylamine	m-ethylformamide
1-butanol	butylbenzene	m-xylene
1-chloropentane	carbon disulfide	heptane
1-chloropropane	carbon tetrachloride	hexadecane
1-decanol	chlorobenzene	hexane
1-fluorooctane	chlorotoluene	nitrobenzene
1-heptanol	*cis*-1,2-dimethylcyclohexane	nitroethane
1-hexanol	decalin	nitromethane
1-hexene	cyclohexane	methylaniline
1-hexyne	cyclohexanone	nonane
1-iodobutane	cyclopentane	octane
1-iodopentene	cyclopentanol	pentane
1-iodopropane	cyclopentanone	o-chlorotoluene
1-nitropropane	decane	o-cresol
1-nonanol	dibromomethane	o-dichlorobenzene
1-octanol	dibutyl ether	o-nitrotoluene
1-pentanol	dichloromethane	o-xylene
1-pentene	diethyl ether	pentadecane
1-pentyne	diethylsulfide	pentanal
1-propanol	diethylamine	pentanoic acid
2,2,2-trifluoroethanol	diiodomethane	pentylethanoate
2,2,4-trimethylpentane	dimethyldisulfide	pentylamine
2,4-dimethylpentane	dimethylacetamide	perfluorobenzene

2,4-dimethylpyridine	dimethylformamide	phenyl ether
2,6-dimethylpyridine	dimethylpyridine	propanal
2-bromopropane	DMSO	propanoic acid
2-chlorobutane	dipropylamine	propanonitrile
2-heptanone	dodecane	propylethanoate
2-hexanone	E-1,2-dichloroethene	propylamine
2-methylpentane	E-2-pentene	p-xylene
2-methylpyridine	ethanethiol	pyridine
2-nitropropane	ethanol	pyrrolidine
2-octanone	ethylethanoate	*sec*-butanol
2-pentanone	ethylmethanoate	t-butanol
2-propanol	ethylphenyl ether	t-butylbenzene
2-propen-1-ol	ethylbenzene	tetrachloroethene
3-methylpyridine	ethylene glycol	tetrahydrofuran
3-pentanone	fluorobenzene	tetrahyrothiophenedioxide
4-heptanone	formamide	tetralin
4-methyl-2-pentanone	formic acid	thiophene
4-methylpyridine	hexadecyliodide	thiophenol
5-nonanone	hexanoic acid	toluene
acetic acid	iodobenzene	*trans*-decalin
acetone	iodoethane	tribromomethane
acetonitrile	iodomethane	tributylphosphate
aniline	isobutanol	trichloroethene
anisole	isopropyl ether	trichloromethane
benzaldehyde	isopropylbenzene	triethylamine
benzene	isopropyltoluene	undecane
benzonitrile	m-cresol	water
benzyl alcohol	mesitylene	Z-1,2-dichloroethene
bromobenzene	methanol	other

[a] Solvent names should be specified without hyphens or spaces.

By setting **solvent = other**, the user may specify their own solvent. This requires specification of the following solvent descriptors:

- **Dielec**, the dielectric constant, ε, of the solvent
- **SolN** , the index of refraction at optical frequencies at 293 K, n_{20}^{D}
- **SolA** , Abraham's hydrogen bond acidity, $\sum \alpha_2^{\mathrm{H}}$
- **SolB** , Abraham's hydrogen bond basicity, $\sum \beta_2^{\mathrm{H}}$
- **SolG** , $\gamma = \gamma_m/\gamma^0$ where γ_m is the macroscopic surface tension at air/solvent interface at 298 K and γ^0 is 1 cal mol^{-1} Å^{-2} (1 dyne/cm = 1.43932 cal mol^{-1} Å^{-2})
- **SolC**, aromaticity (ϕ), equal to the fraction of non-hydrogen solvent atoms that are aromatic carbon atoms
- **SolH**, electronegative "halogenicity" (ψ), equal to the fraction of non-hydrogen solvent atoms that are F, Cl or Br

These parameters are specified in the *$smx* input section as suggested in Example 11.16. (Any solvent descriptors that are omitted from the *$smx* section default to zero.) Values can be derived from experiment or from interpolation or extrapolation of data available for other solvents. Solvent parameters for common organic solvents are tabulated in the Minnesota Solvent Descriptor Database. The latest version of this database is available at:

http://comp.chem.umn.edu/solvation/mnsddb.pdf

Job controls variables for the *$smx* section that are common to SM8, SM12, and SMD are given below.

Solvent

Sets the SMx solvent

INPUT SECTION: *$smx*

TYPE:

STRING

DEFAULT:

water

OPTIONS:

Any name from the list of solvents given in Table 11.7.

RECOMMENDATION:

NONE

Print

Controls extra printing for SMx calculations

INPUT SECTION: *$smx*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Minimal printing

1 Detailed breakdown of energy components (at SCF convergence)

2 Additional information, dependent on the specific SMx model

RECOMMENDATION:

The energy components obtained with **print** = 1 may be useful, although a summary is printed by default. The value **print** = 2 is probably only useful for diagnosing problems.

Example 11.16 SMD model with a user-defined solvent.

```
$comment
User-defined specification of pentane.
(Result should be equivalent to SOLVENT = PENTANE)
$end

$molecule
   0 1
   C    -0.361658    -0.986967     0.222366
   C    -1.331098     0.144597    -0.108363
   O    -0.592574     1.354183     0.036738
   C     0.798089     1.070899     0.136509
   C     0.964682    -0.396154    -0.256319
   H    -0.625676    -1.925862    -0.267011
   H    -0.333229    -1.158101     1.302753
   H    -1.697529     0.068518    -1.140448
   H    -2.193412     0.181620     0.562129
   H     1.130199     1.238399     1.169839
   H     1.348524     1.754318    -0.514697
   H     1.050613    -0.489646    -1.343151
   H     1.843065    -0.855802     0.199659
$end

$rem
   METHOD           b3lyp
   BASIS            6-31G*
   SOLVENT_METHOD   smd
$end

$smx
solvent other
epsilon 1.840
SolN    1.357
SolA    0.000  ! could be omitted
SolB    0.000  ! could be omitted
SolG   22.300
SolC    0.000  ! could be omitted
SolH    0.000  ! could be omitted
$end
```

11.2.9.1 The SM8 Model

The SM8 model is described in detail in Ref. 108. It may be employed in conjunction with density functional theory (with any density functional available in Q-CHEM) or with Hartree-Fock theory, but is intended for use only with the 6-31G*, 6-31+G*, and 6-31+G** basis sets, for reasons discussed below.

Bulk (continuum) electrostatic interactions in SM8 are described in terms of a generalized Born (GB) SCRF,[62] using a solute cavity constructed from atom-centered spheres. For the atoms H, C, N, O, F, Si, P, S, Cl, and Br, atomic radii have been specifically optimized for use with SM8, whereas for other atoms the Bondi radius is used,[12] or else a value of 2.0 Å for atoms not included in Bondi's paper. Geometry-dependent radii are computed from these "intrinsic" Coulomb radii via a de-screening approximation.[108]

In addition to GB electrostatics, there are several other contributions to the SM8 standard-state free energy of solvation. One is the electronic-nuclear-polarization (ENP) energy, which is a more descriptive (and precise) name for what is

often called the electrostatic solvation energy.[62] Another contribution to the free energy of solvation comes from short-range interactions with solvent molecules in the first solvation shell, and is sometimes called the cavitation–dispersion–solvent-structure (CDS) term. Collectively, the various CDS interactions represent the nonelectrostatic contributions to the solvation energy.

In the SMx models, this nonelectrostatic contribution is modeled as a sum of terms proportional to the solvent-accessible surface area (SASA) of each individual solute atoms. The proportionality constants, which are parameters of the model, have units of energy/(area)^2 so are sometimes called atomic surface tension parameters. The SASA of the solute molecule is the area of a surface generated by the center of a spherical effective solvent molecule rolling on the van der Waals surface of the solute molecule, as in the solvent-accessible surface that was mentioned in Section 11.2.3. The SASA is computed using the algorith described in Ref. 100 along with Bondi's values for the van der Waals radii,[12] or else a value of 2.0 Å if no Bondi radius is available. (As in the case of nonelectrostatic interactions in PCMs, this means that a different molecular surface is used for the bulk electrostatics as compared to the nonelectrostatic interactions.) The solvent probe radius used to generate the SASAs is set to 0.40 Å for all solvents. Note that the solvent-structure part of the CDS term includes many aspects of solvent structure that are not described by bulk electrostatics, for example, hydrogen bonding, exchange repulsion, and the variation of the effective dielectric constant in the first solvation shell, relative to its bulk value. The empirical nature of the CDS term also makes up for errors due to (i) assuming fixed and model-dependent values of the intrinsic Coulomb radii, and (ii) any systematic errors in the description of the solute–solvent electrostatic interactions using the GB approximation.

The final component of the SM8 solvation free energy is the concentration component. This is zero if the standard-state concentration of the solute is the same in the gas and solution phases (*e.g.*, if it is 1 mol/L in the gas phase as well as in the solution). Otherwise, this correction can be computed using ideal gas formulas, as discussed below.

SM8 does not require the user to assign molecular mechanics atom types to atoms or groups; all atomic surface tensions in the theory are unique and continuous functions of the solute geometry, defined by the model and calculated internally within Q-CHEM. In principle, SM8 can be used with any level of electronic structure theory so long as accurate partial charges can be computed, but Q-CHEM's implementation of SM8 specifically uses self-consistently polarized Charge Model 4 (CM4) class IV charges.[77] CM4 charges are obtained from Löwdin population analysis charges, via a mapping whose parameters depend on the basis set (and only on the basis set, not on the density functional or anything else). The supported basis sets in Q-CHEM are 6-31G*, 6-31+G*, and 6-31+G**; other basis sets should not be used in SM8 calculations. The charge mapping parameters are given in Ref. 77.

The SM8 solvation free energy is output at $T = 298$ K for a standard-state concentration of 1 M in both the gas and solution phase. However, solvation free energies in the literature are often tabulated using a standard state of $P = 1$ atm for the gas. To convert 1 M-to1 M solvation free energies at 298 K to a standard state consisting of $P = 1$ atm for the gas and a 1 M concentration in solution, add $+1.89$ kcal/mol to the computed solvation free energy.

Solution-phase geometry optimizations can be carried out, but basis sets that use spherical harmonic d functions, or angular momentum higher than d (f, g, *etc.*) are not supported. Since, by definition, the 6-31G*, 6-31+G*, and 6-31+G** basis sets have Cartesian d shells, they are examples of basis sets that may be used for geometry optimization with SM8. Solution-phase Hessian calculations can be carried out by numerical differentiation of analytical energy gradients or by double differentiation of energies, although the former procedure is both more stable and more economical. The analytic gradients of SM8 are based on the analytical derivatives of the polarization free energy and the analytical derivatives of the CDS terms derived in Ref. 182.

The SM8 test suite contains the following representative examples:

- single-point solvation energy and analytical gradient calculation for 2,2-dichloroethenyl dimethyl phosphate in water at the M06-2X/6-31G* level;
- single-point solvation energy calculation for 2,2-dichloroethenyl dimethyl phosphate in benzene at the M06-2X/

6-31G* level;

- single-point solvation energy calculation for 2,2-dichloroethenyl dimethyl phosphate in ethanol at the M06-2X/6-31G* level;
- single-point solvation energy calculation for 5-fluorouracil in water at the M06/6-31+G* level;
- single-point solvation energy calculation for 5-fluorouracil in octanol at the M06-L/6-31+G* level;
- single-point solvation energy and analytical gradient calculation for 5-fluorouracil in fluorobenzene at the M06-HF/6-31+G** level;
- geometry optimization for protonated methanol $CH_3OH_2^+$ in water at the B3LYP/6-31G* level;
- finite-difference frequency (with analytical gradient) calculation for protonated methanol $CH_3OH_2^+$ in water at the B3LYP/6-31G* level.

Users who wish to calculate solubilities can calculate them from the free energies of solvation by the method described in Ref. 158. The present model can also be used with confidence to calculate partition coefficients (*e.g.*, Henry's Law constants, octanol/water partition coefficients, *etc.*) by the method described in Ref. 42.

The user should note that the free energies of solvation calculated by the SM8 model in the current version of Q-CHEM are all what may be called equilibrium free energies of solvation. The nonequilibrium algorithm required for vertical excitation energies[96] is not yet available in Q-CHEM. (Nonequilibrium versions of PCMs are available instead; see Section 11.2.3.3.)

11.2.9.2 The SM12 Model

The SM12 model[112] is also available in Q-CHEM. Similar to SM8, it employs (a) the generalized Born approximation for the bulk electrostatic contribution to the free energy of solvation, and (b) the same formulas (with re-optimized parameters) for CDS contributions. SM12 holds several advantages over SM8, and perhaps foremost among these is that it uses CM5 charges,[111] which are based on Hirshfeld population analysis, or else charges derived from the electrostatic potential,[15,147] for the bulk electrostatics term. These charges are stable with respect to extension of the basis set, and thus SM12 can be used with larger basis sets whereas SM8 is limited to 6-31G*, 6-31+G*, and 6-31+G**, due to instabilities in the Löwdin charges in larger basis sets. In addition, SM12 is parameterized using a more diverse training set as compared to SM8, and is defined for the entire periodic table. However, the SM12 analytic gradient is not available in Q-CHEM at present.

An SM12 calculation is requested by setting SOLVENT_METHOD = SM12 in the *$rem* section. The manner in which the electrostatic term is computed is controlled by the **Charges** keyword in the *$smx* input section.

Charges

Sets the type of atomic charges for the SM12 electrostatic term.

INPUT SECTION: *$smx*

TYPE:

STRING

DEFAULT:

CM5

OPTIONS:

CM5	Charge Model 5 charges[111]
MK	Merz-Singh-Kollman charges[147]
CHELPG	ChElPG charges[15]

RECOMMENDATION:

None. Merz-Singh-Kollman and ChElPG charges are fit to reproduce the molecular electrostatic potential on the van der Waals surface or on a cubic grid, respectively, whereas CM5 is an empirical model based on Hirshfeld population analysis.

Example 11.17 SM12CM5 calculation of the solvation free energy of water in the 1-octanol solvent.

```
$molecule
   0 1
   O     0.000000    0.125787   0.000000
   H     0.758502   -0.503148   0.000000
   H    -0.758502   -0.503148   0.000000
$end

$rem
   METHOD               b3lyp
   BASIS                6-31G*
   SCF_GUESS            core
   SOLVENT_METHOD       sm12
   POINT_GROUP_SYMMETRY false
$end

$smx
   solvent 1octanol
   charges chelpg
$end
```

11.2.9.3 The SMD Model

The SMD model[109] is also available in Q-CHEM. Within the original verion of this model, the electrostatic contribution to the free energy solvation is described via the IEF-PCM model, where the CDS contributions follow the formulas as SM8 and SM12 with the parameters re-optimized to be compatible with the IEF-PCM electrostatics. Relative to SM8 or SM12, where the electrostatic interactions are defined in terms of atomic point charges that are sensitive to the choice of basis set (and therefore only certain basis sets are supported for use with these models), SMD can be used with any basis set.

The Q-CHEM implementation of the SMD model uses the simpler C-PCM model by default as for SOLVENT_METHOD = PCM. Based on the benchmark results shown in Table 4 of Ref. 62, the C-PCM results essentially have no difference from the IEF-PCM ones in terms of accuracy. An SMD energy, gradient, or Hessian calculation is requested by setting SOLVENT_METHOD = SMD in the *$rem* section. While Q-CHEM users can vary the parameters for the C-PCM or IEF-PCM part of the SMD calculation, this is not recommended because such modifications may be less compatible with the CDS parameters and thus lead to less accurate results. C-PCM is the preferred electrostatics model for SMD

calculations. Starting with Q-CHEM v. 6.3, seminumerical SMD Hessians are available, using the C-PCM analytic Hessian[102] combined with a finite-difference evaluation of the CDS contribution.

In Q-CHEM v. 5.2 and later, the default surface discretization method is changed from VTN to SwiG in order to ensure the smoothness of potential energy surface. In addition, the gas-phase SCF calculation that takes place before the SMx calculation is turned off by default. If one wants to obtain the solvation free energy, then the gas phase calculation is still required and it can be turned on by setting SMX_GAS_PHASE = TRUE. Setting this *$rem* variable to TRUE might also be helpful if directly converging SCF with the SMx models is difficult.

SMX_GAS_PHASE

Converge the gas-phase SCF first before doing calculations with SMx models

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Run SMx calculations directly

TRUE Run gas-phase calculation first

RECOMMENDATION:

Use the default unless solvation free energy is needed. Set it to TRUE if the SCF calculation fails to converge otherwise.

Example 11.18 SMD force calculation for tetrahydrofuran (THF) in the pentane solvent.

```
$molecule
   0 1
   C    -0.361658    -0.986967     0.222366
   C    -1.331098     0.144597    -0.108363
   O    -0.592574     1.354183     0.036738
   C     0.798089     1.070899     0.136509
   C     0.964682    -0.396154    -0.256319
   H    -0.625676    -1.925862    -0.267011
   H    -0.333229    -1.158101     1.302753
   H    -1.697529     0.068518    -1.140448
   H    -2.193412     0.181620     0.562129
   H     1.130199     1.238399     1.169839
   H     1.348524     1.754318    -0.514697
   H     1.050613    -0.489646    -1.343151
   H     1.843065    -0.855802     0.199659
$end

$rem
   JOBTYPE          force
   METHOD           b3lyp
   BASIS            6-31G*
   SOLVENT_METHOD   smd
$end

$smx
   solvent pentane
$end
```

11.2.10 Langevin Dipoles Model

Q-CHEM provides the option to calculate molecular properties in aqueous solution and the magnitudes of the hydration free energies by the Langevin dipoles (LD) solvation model developed by Florián and Warshel.[49,50] In this model, a

solute molecule is surrounded by a sphere of point dipoles, with centers on a cubic lattice. Each of these "Langevin" dipoles changes its size and orientation in the electrostatic field of the solute and the other Langevin dipoles. The electrostatic field from the solute is determined rigorously by the integration of its charge density, whereas for dipole–dipole interactions, a 12 Å cutoff is used. The Q-CHEM/CHEMSOL 1.0 implementation of the LD model is fully self-consistent in that the molecular quantum mechanical calculation takes into account solute–solvent interactions. Further details on the implementation and parameterization of this model can be found in the literature.[49,50]

The results of CHEMSOL calculations are printed in the standard output file. Below is a part of the output for a calculation on the methoxide anion (corresponding to the sample input given later on, and the sample file in the `$QC/samples` directory).

Energy Component	Value / kcal mol^{-1}
LD Electrostatic energy	−86.14
Hydrophobic energy	0.28
van der Waals energy	−1.95
Bulk correction	−10.07
Solvation free energy, ΔG(ILD)	−97.87

Table 11.8: Results of the iterative Langevin Dipoles (ILD) solvation model, for aqueous methoxide.

The total hydration free energy, ΔG(ILD) is calculated as a sum of several contributions. Note that the electrostatic part of ΔG is calculated by using the linear-response approximation[49] and contains contributions from the *polarization* of the solute charge distribution due to its interaction with the solvent. This results from the self-consistent implementation of the Langevin dipoles model within Q-CHEM.

To perform an LD calculation in Q-CHEM, specify normal job-control variables for a Hartree-Fock or DFT calculation, and set SOLVENT_METHOD = CHEM_SOL in the *$rem* section. Additional fine-tuning is accomplished using a set of keywords in a *$chem_sol* input section. The remainder of this section summarizes these keywords.

EField

Determines how the solute charge distribution is approximated in evaluating the electrostatic field of the solute.

INPUT SECTION: *$chem_sol*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Exact solute charge distribution is used.

0 Solute charge distribution is approximated by Mulliken atomic charges.

RECOMMENDATION:

None. The Mulliken-based procedure is faster but less rigorous.

NGrids

Sets the number of grids used to calculate the average hydration free energy.

INPUT SECTION: *$chem_sol*

TYPE:

INTEGER

DEFAULT:

5 ΔG_{hydr} will be averaged over 5 different grids.

OPTIONS:

n Use n different grids.

RECOMMENDATION:

None. The maximum allowed value of n is 20.

Print

Controls printing in the CHEMSOL part of the Q-CHEM output file.

INPUT SECTION: *$chem_sol*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Limited printing

1 Full printing

RECOMMENDATION:

None.

ReadRadii

Read user-defined atomic radii from section *$van_der_waals*.

INPUT SECTION: *$chem_sol*

TYPE:

None

DEFAULT:

Off

OPTIONS:

No options. Specify the keyword to use user-defined atomic radii.

RECOMMENDATION:

None.

Accurate calculations of hydration free energies require a judicious choice of the solute–solvent boundary in terms of atom-type dependent parameters. The default atomic van der Waals radii available in Q-CHEM were chosen to provide reasonable hydration free energies for most solutes and basis sets. These parameters basically coincide with the CHEMSOL 2.0 radii given in Ref. 50. The only difference between the Q-CHEM and CHEMSOL 2.0 atomic radii stems from the fact that Q-CHEM parameter set uses radii for carbon and oxygen that are independent of the atom's hybridization state. User-defined atomic radii can be specified by declaring the option **ReadRadii** in the *$chem_sol* input section, and then placing the radii in the *$van_der_waals* section. Two different (and mutually exclusive) formats

can be used, as shown below.

```
$van_der_waals
1
atomic_number   vdW_radius
...
$end

$van_der_waals
2
sequential_atom_number   vdW_radius
...
$end
```

The purpose of the second format is to permit the user to customize the radius of specific atoms, in the order that they appear in the *$molecule* section, rather than simply by atomic numbers as in format 1. The radii of atoms that are not listed in the *$van_der_waals* input will be assigned default values. The atomic radii that were used in the calculation are printed in the CHEMSOL part of the output file in the column denoted *rp*. All radii should be given in Ångstroms.

Example 11.19 A Langevin dipoles calculation on the methoxide anion. A customized value is specified for the radius of the carbon atom.

```
$molecule
  -1  1
   C    0.0000    0.0000  -0.5274
   O    0.0000    0.0000   0.7831
   H    0.0000    1.0140  -1.0335
   H    0.8782   -0.5070  -1.0335
   H   -0.8782   -0.5070  -1.0335
$end

$rem
   METHOD             hf
   BASIS              6-31G
   SCF_CONVERGENCE    6
   SOLVENT_METHOD     Chem_Sol
$end

$chem_sol
ReadRadii
$end

$van_der_waals
2
1 2.5
$end
```

11.2.11 Poisson Boundary Conditions

Each of the implicit solvation models described above is designed for isotropic, bulk solvation—conditions that can be qualitatively described using a scalar dielectric constant, ε. For an *anisotropic* environment, such as a liquid/vapor interface, a more general approach is to solve Poisson's equation with a spatially-varying dielectric *function*, $\varepsilon(\mathbf{r})$. Such an approach can be used, for example, to model the air/water interface by describing certain regions of space using $\varepsilon = 1$ (air) and other regions using $\varepsilon = 78$ (water).[35,36] The atomistic region is described at the SCF level. Construction of the dielectric function $\varepsilon(\mathbf{r})$ is based on a solute cavity, somewhat analogous to the solute cavity in a

PCM calculation but where the dielectric changes smoothly (rather than abruptly) from $\varepsilon = 1$ inside the cavity (in the atomistic QM region) to $\varepsilon = \varepsilon_{\text{solvent}}$ outside, in the continuum solvent. Options for modeling a liquid/vapor interface are also available; see Section 11.2.11.3. In contrast to PCMs, which must approximate the "volume polarization" due to penetration of the tails of the QM charge density beyond the solute cavity, this effect is described exactly under Poisson boundary conditions. The price to be paid for lifting this approximation is that charge densities must be discretized and integrated over three-dimensional space rather than simply a two-dimensional solute cavity surface.

Q-CHEM's implementation of the *Poisson Equation Solver* (PEqS) method was introduced in Ref. 36 and refined in Refs. 35 and 125. Support for dielectric media with finite ionic strength was added in Ref. 153. Users of this methodology are asked to cite Refs. 35 and 125.

11.2.11.1 Theory

The most general formulation of Poisson's equation, for a dielectric *function* rather than a dielectric *constant*, is

$$\hat{\boldsymbol{\nabla}} \cdot \left[\varepsilon(\mathbf{r}) \hat{\boldsymbol{\nabla}} \varphi_{\text{tot}}(\mathbf{r}) \right] = -4\pi \rho_{\text{sol}}(\mathbf{r}) \; . \tag{11.25}$$

The solute's charge density will be separated into nuclear and electronic components,

$$\rho_{\text{sol}}(\mathbf{r}) = \rho_{\text{nuc}}(\mathbf{r}) + \rho_{\text{elec}}(\mathbf{r}) \; . \tag{11.26}$$

The total electrostatic potential φ_{tot} is comprised of the solute's electrostatic potential $\varphi_{\text{elec}} + \varphi_{\text{nuc}}$, which is obtained from the density in Eq. (11.26), plus the polarization potential induced in the medium:

$$\varphi_{\text{tot}}(\mathbf{r}) = \varphi_{\text{elec}}(\mathbf{r}) + \varphi_{\text{nuc}}(\mathbf{r}) + \varphi_{\text{pol}}(\mathbf{r}) \; . \tag{11.27}$$

To compute ρ_{elec}, the electronic contribution to the electrostatic potential in vacuum is first evaluated on a Cartesian grid,

$$\varphi_{\text{elec}}(\mathbf{r}_i) = \sum_{\mu\nu}^{N_{\text{basis}}} P_{\mu\nu} \left\langle g_\mu(\mathbf{r}) \middle| \frac{1}{|\mathbf{r} - \mathbf{r}_i|} \middle| g_\nu(\mathbf{r}) \right\rangle \; . \tag{11.28}$$

Here $\mathbf{P}$ is the one-electron density matrix, g_μ and g_ν are atom-centered Gaussian basis functions, and the integration is over $\mathbf{r}$. The electronic part of the solute's charge density is then obtained from

$$\rho_{\text{elec}}(\mathbf{r}) = -\frac{1}{4\pi} \hat{\nabla}^2 \varphi_{\text{elec}}(\mathbf{r}) \; , \tag{11.29}$$

where in practice the Laplacian of φ_{elec} is evaluated using an eighth-order finite difference scheme.[35] The nuclear charge density ρ_{nuc} is described classically,

$$\rho_{\text{nuc}}(\mathbf{r}) = -\sum_{A}^{N_{\text{atoms}}} Z_A \, \delta\left(\mathbf{r} - \mathbf{R}_A\right) \; , \tag{11.30}$$

but to avoid numerical problems the nuclear charges are smeared out over one Cartesian grid voxel in practice. Operationally, this is achieved by adding Z_A/dV to the grid point nearest $\mathbf{R}_A$, where dV is the volume of a Cartesian voxel. The exact nuclear electrostatic potential in vacuum is computed according to

$$\varphi_{\text{nuc}}(\mathbf{r}) = -\frac{1}{4\pi} \sum_{A}^{N_{\text{atoms}}} \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \; . \tag{11.31}$$

The Dirac delta function in Eq. (11.30) has dimensions of (volume)$^{-1}$ and therefore incorporates the aforementioned factor of dV^{-1}

Having computed the electronic and nuclear charge densities and the electrostatic potentials in vacuum, one has in hand an exact solution of Eq. (11.25) for the case where $\varepsilon(\mathbf{r}) = 1$. The induced polarization charge density and

corresponding electrostatic potential are then obtained iteratively, following closely the procedure outlined in Refs. 47 and 4. Equation (11.25) is rewritten to appear as Poisson's equation in vacuum with an additional source charge density:

$$\begin{aligned}\hat{\nabla}^2\varphi_{\rm tot}(\mathbf{r}) &= -4\pi\left[\frac{\rho_{\rm sol}(\mathbf{r})}{\varepsilon(\mathbf{r})} + \frac{\hat{\boldsymbol{\nabla}}\ln\varepsilon(\mathbf{r})\cdot\hat{\boldsymbol{\nabla}}\varphi_{\rm tot}(\mathbf{r})}{4\pi}\right] \\ &= -4\pi\left[\rho_{\rm sol}(\mathbf{r}) + \rho_{\rm pol}(\mathbf{r})\right] ,\end{aligned} \tag{11.32}$$

where the polarization charge density is

$$\rho_{\rm pol}(\mathbf{r}) = \rho_{\rm iter}(\mathbf{r}) + \left(\frac{1-\varepsilon(\mathbf{r})}{\varepsilon(\mathbf{r})}\right)\rho_{\rm sol}(\mathbf{r}) . \tag{11.33}$$

The iterative charge density $\rho_{\rm iter}(\mathbf{r})$ takes the form

$$\rho_{\rm iter}^{(k+1)}(\mathbf{r}) = \frac{\hat{\boldsymbol{\nabla}}\ln\varepsilon(\mathbf{r})\cdot\hat{\boldsymbol{\nabla}}\varphi_{\rm tot}^{(k)}(\mathbf{r})}{4\pi} \tag{11.34}$$

at iteration $k+1$. This function is nonzero only in transition regions where the dielectric is being interpolated from vacuum to solvent. Note also that the second term in Eq. (11.33) vanishes wherever $\varepsilon(\mathbf{r}) = 1$. The quantity $\rho_{\rm iter}(\mathbf{r})$ is iterated to convergence by performing updates $\rho_{\rm iter}^{(k)}(\mathbf{r}) \rightarrow \rho_{\rm iter}^{(k+1)}(\mathbf{r})$ using Eq. (11.34), taking $\varphi_{\rm tot}^{(0)}(\mathbf{r})$ to be the solute's electrostatic potential in vacuum. Each time $\rho_{\rm iter}(\mathbf{r})$ is updated, a new polarization charge density is generated and Eq. (11.32) is solved to obtain a new total electrostatic potential. To stabilize the iterative updates of $\rho_{\rm iter}(\mathbf{r})$, we use a damping procedure

$$\begin{aligned}\rho_{\rm iter}^{(k+1)}(\mathbf{r}) &= \frac{\eta}{4\pi}\left[\hat{\boldsymbol{\nabla}}\ln\varepsilon(\mathbf{r})\cdot\hat{\boldsymbol{\nabla}}\varphi_{\rm tot}^{(k)}(\mathbf{r})\right] + (1-\eta)\rho_{\rm iter}^{(k)}(\mathbf{r}) \\ &= \eta\rho_{\rm iter}^{(k+1)} + (1-\eta)\rho_{\rm iter}^{(k)}(\mathbf{r})\end{aligned} \tag{11.35}$$

rather than using Eq. (11.34) as written. Following Refs. 47 and 4, we use $\eta = 0.6$.

Once the total charge density and electrostatic potential are iterated to self-consistency, the free energy of solvation,

$$G_{\rm pol} = \frac{1}{2}\int d\mathbf{r}\;\varphi_{\rm pol}(\mathbf{r})\;\rho_{\rm sol}(\mathbf{r}) , \tag{11.36}$$

is computed, where

$$\varphi_{\rm pol}(\mathbf{r}) = \varphi_{\rm tot}(\mathbf{r}) - \varphi_{\rm sol}(\mathbf{r}) . \tag{11.37}$$

Since $\varphi_{\rm tot}$ and $\rho_{\rm tot}$ are computed self-consistently at each SCF iteration, the polarization effects must be incorporated into the Fock matrix. This is accomplished by adding the correction term $\Delta\hat{F} = \delta G_{\rm pol}/\delta\rho_{\rm elec}$. The matrix form of this correction in the atomic orbital basis is

$$\Delta F_{\mu\nu} = \int d\mathbf{r}\;\varphi_{\rm pol}(\mathbf{r})\;g_\mu(\mathbf{r})\;g_\nu(\mathbf{r}) . \tag{11.38}$$

For additional details and a full description of the algorithm, see Ref. 35.

11.2.11.2 Nonequilibrium Solvation for Vertical Ionization

Nonequilibrium solvation within the framework of PCMs is discussed in Section 11.2.3.3, and that methodology [114,178] has been adapted for vertical ionization processes in conjunction with Poisson boundary conditions.[35,36] A state-specific approach has been implemented that requires solution of a nonlinear Schrödinger equation, Eq. (11.7). For the reference state, corresponding to $i = 0$ in Eq. (11.7), this is equivalent to solving Eq. (11.25) for the induced polarization potential and computing the solvation free energy and Fock matrix correction according to Eqs. (11.36) and (11.38), respectively.

For the ionized state, denoted by a subscript i, the solvent polarization is partitioned into fast and slow components. Within the Marcus partitioning scheme,[178] Eq. (11.32) becomes[35]

$$-\frac{1}{4\pi}\hat{\nabla}^2\varphi_{\mathrm{tot},i}(\mathbf{r}) = \rho_{\mathrm{sol},i}(\mathbf{r}) + \rho^{\mathrm{slow}}_{\mathrm{pol},0}(\mathbf{r}) + \rho^{\mathrm{fast}}_{\mathrm{pol},i}(\mathbf{r}) \ , \tag{11.39}$$

where the fast polarization charge density of the ionized state, $\rho^{\mathrm{fast}}_{\mathrm{pol},i}$, is given by

$$\rho^{\mathrm{fast}}_{\mathrm{pol},i}(\mathbf{r}) = \rho_{\mathrm{iter},i}(\mathbf{r}) + \left(\frac{1-\varepsilon_{\mathrm{opt}}(\mathbf{r})}{\varepsilon_{\mathrm{opt}}(\mathbf{r})}\right)\left[\rho_{\mathrm{sol},i}(\mathbf{r}) + \rho^{\mathrm{slow}}_{\mathrm{pol},0}(\mathbf{r})\right] \tag{11.40}$$

and the fast polarization potential, $\varphi^{\mathrm{fast}}_{\mathrm{pol},i}$, is

$$\varphi^{\mathrm{fast}}_{\mathrm{pol},i}(\mathbf{r}) = \varphi_{\mathrm{tot},i}(\mathbf{r}) - \varphi_{\mathrm{sol},i}(\mathbf{r}) - \varphi^{\mathrm{slow}}_{\mathrm{pol},0}(\mathbf{r}) \ . \tag{11.41}$$

Subscripts i and 0 refer to the ionized state and the reference state, respectively. The quantity $\varepsilon_{\mathrm{opt}}$ in Eq. (11.40) is the solvent's optical dielectric constant, and $\rho_{\mathrm{iter},i}$ is computed using Eq. (11.35) but with a dielectric function constructed using $\varepsilon_{\mathrm{opt}}$ and with $\varphi^{\mathrm{fast}}_{\mathrm{tot},i}$ substituted in place of φ_{tot}. For the Marcus partitioning scheme, the slow component of the induced solvent polarization charge density is computed as the three-dimensional (volume charge rather than surface charge) analogue of Eq. (11.9).

11.2.11.3 Basic Job Control

A calculation using Poisson boundary conditions is requested by setting SOLVENT_METHOD = PEQS in the *$rem* section. For computational efficiency, the Poisson Equation Solver (PEqS) is typically not engaged until the error in the vacuum SCF calculation falls below a specified threshold, taken to be 10^3 E_h by default and user-controllable by means of the *$rem* variable PEQS_SWITCH.

Note: When PEqS solvation is requested, symmetry and the standard nuclear orientation are disabled automatically, equivalent to POINT_GROUP_SYMMETRY = FALSE and NO_REORIENT = TRUE. Poisson's equation is solved on a three-dimensional Cartesian grid and the energy need not be rotationally invariant unless the grid spacing is very small.

PEQS_SWITCH

Inclusion of solvent effects begins when the SCF error falls below $10^{-\mathrm{PEQS\_SWITCH}}$.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

Use the default unless solvent effects need to be incorporated earlier in the SCF procedure.

Further job control for the Poisson equation solver (PEqS) routines is set up in two sections: *$peqs_grid* and *$peqs*. (An optional third section, *$epsilon*, is discussed later.) The *$peqs_grid* section is used to define the Cartesian discretization grid, as follows:

```
$peqs_grid
   DimX <Nx> <Xmin> <Xmax>
   DimY <Ny> <Ymin> <Ymax>
   DimZ <Nz> <Zmin> <Zmax>
$end
```

The Cartesian grid must be orthorhombic and `Nx`, `Ny`, and `Nz` refer to the number of grid points for the x, y, and z

axes.

Note: A fourth-order multigrid algorithm is used to accelerate convergence of the conjugate gradient PEqS routine that solves Eq. (11.32).[35] This requires that the number of grid points be odd, with the additional constraint that the quantities $N_x - 1$, $N_y - 1$, and $N_z - 1$ must all be divisible by 8.

Values in the last two columns of the *$peqs_grid* section specify the minimum and maximum values of x, y, and z, in units of Ångstroms. The length of the grid in the x direction is $L_x = x_{\max} - x_{\min}$ and the grid spacing is $\Delta x = L_x/(N_x - 1)$. The volume element discussed in the context of Eq. (11.30) is $dV = \Delta x \Delta y \Delta z$.

The *$peqs* section controls other aspects of a PEqS calculation, including construction of the solute cavity and other required parameters for interfacial or nonequilibrium solvation. The format is:

```
$peqs
   <Keyword>  <parameter/option>
$end
```

Available keywords are described below.

BatchSize

Evaluate electrostatic potential integrals in batches over small parts of the Cartesian grid, rather than computing all integrals in a single batch.

INPUT SECTION: *$peqs*

TYPE:

INTEGER

DEFAULT:

5000

OPTIONS:

n Corresponding to a batch size of n grid points.

RECOMMENDATION:

For large grids ($\geq 10^6$ grid points), a batch size $n \approx N_x N_y N_z/5$ is recommended.

MaxIter

Sets the maximum number of iterations used in the conjugate gradient solver.

INPUT SECTION: *$peqs*

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

User-defined.

RECOMMENDATION:

Use the default unless the calculation fails to converge.

OpticalDielectric

Sets the optical dielectric value of the solvent.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

1.7778

OPTIONS:

$\varepsilon_{\mathrm{opt}}$ Desired value of $\varepsilon_{\mathrm{opt}}$ (dimensionless).

RECOMMENDATION:

The optical dielectric is equal to the square of the index of refraction. The default value corresponds to water at 25°C.

PolarIterScale

Specifies the mixing parameter η that is used in Eq. (11.35) to stabilize iterative solution for the polarization charge density.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

0.6

OPTIONS:

η Desired value of the dimensionless mixing parameter.

RECOMMENDATION:

Use the default, which was tested in Refs. 47 and 4, unless the calculation proves difficult to converge.

Print

Specifies the level of printing in the output file.

INPUT SECTION: *$peqs*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Basic printing including some PEqS information at each SCF iteration.

1 Additional printing include atomic vdW radii used in cavity construction.

2 Debugging information.

RECOMMENDATION:

Use 0 or 1.

SoluteCavity

Specifies the type of solute cavity, which determines the form of the dielectric function $\varepsilon(\mathbf{r})$.

INPUT SECTION: *$peqs*

TYPE:

STRING

DEFAULT:

Vacuum

OPTIONS:

Vacuum	Perform calculation in vacuum, $\varepsilon(\mathbf{r}) \equiv 1$.
RigidVDW	Create a cavity using spherically symmetric error functions.
Spherical	Create a single spherical cavity around the solute.
Arbitrary	Create a user-defined dielectric cavity as obtained from the *$epsilon* input section.

RECOMMENDATION:

None.

SolventDielectric

Sets the dielectric value for the solvent.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

ε	Desired value of ε (dimensionless).

RECOMMENDATION:

The default value corresponds to water at 25°C.

SolverThresh

The electrostatic potential is considered converged when the error falls below $10^{-\mathrm{PEQSolverThresh}}$.

INPUT SECTION: *$peqs*

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n	Corresponding to 10^{-n}

RECOMMENDATION:

Use the default unless a tighter convergence criterion is desired, at greater computational cost.

11.2.11.4 Cavity Construction

In conventional PCMs the solute cavity is a rigid two-dimensional surface constructed from a union of atomic spheres, the radii of which are generally take to be equal to the van der Waals radius (r_{vdW}) scaled by a factor of 1.2; see Section 11.2.4.[63] This cavity is rigid in the sense that once the atomic coordinates are specified, it remains unchanged during the SCF cycles. (The isodensity cavity discussed in Section 11.2.6 is an exception, but is not available for

the PEqS method.) Furthermore, there is an abrupt and discontinuous change in the dielectric constant at the solute/continuum boundary. A three-dimensional analogue of this cavity, using continuous and differentiable spherically-symmetric error functions, has been implemented for PEqS calculations, following the procedure in Ref. 47. The permittivity function, which depends parametrically on the nuclear positions $\mathbf{R}_A$, is

$$\varepsilon(\mathbf{r};\{\mathbf{R}_A\}) = \left(\varepsilon_{\text{solvent}} - 1\right) \left\{ \prod_{A}^{N_{\text{atoms}}} h(d_A, \Delta; |\mathbf{r} - \mathbf{R}_A|) \right\} , \tag{11.42}$$

where

$$h(d_A, \Delta; |\mathbf{r} - \mathbf{R}_A|) = \frac{1}{2}\left[1 + \text{erf}\left(\frac{|\mathbf{r} - \mathbf{R}_A| - d_A}{\Delta}\right)\right] . \tag{11.43}$$

Equation (11.42) smoothly interpolates between $\varepsilon = 1$ and $\varepsilon = \varepsilon_{\text{solvent}}$, over a length scale of $\approx 4\Delta$. The value of Δ is specified using the **RigidScale** keyword. The quantity d_A in Eqs. (11.42) and (11.43) sets radius of the atomic sphere for atom A. By default, $d_A = 1.2\, r_{\text{vdW}}$ but this can be controlled as described below.

Radii

Sets the choice of atomic vdW radii, r_{vdW}.

INPUT SECTION: *$peqs*

TYPE:

STRING

DEFAULT:

Bondi

OPTIONS:

Bondi Use the modified Bondi set (Ref. 139).

UFF Universal Force Field radii (Ref. 133).

Read User-defined radii from a *$van_der_waals* input section.

RECOMMENDATION:

None. The **Read** option works in the same way as the corresponding keyword in the *$pcm* section, which is described in Section 11.2.4.1 with the format of the *$van_der_waals* input section described in Section 11.2.10.

RigidScale

Sets the length scale on which the error function employed in the rigid vdW cavity construction interpolates the dielectric value from vacuum to solvent.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

0.265 Å

OPTIONS:

Δ Specifies the desired value (in Å); see Eq. (11.43).

RECOMMENDATION:

Use the default value, which was tuned in Ref. 47 so that errors in small-molecule solvation energies were ≈ 1 kcal/mol.

VDWtype

Specifies details for the rigid vdW cavity construction.

INPUT SECTION: *$peqs*

TYPE:

STRING

DEFAULT:

Scaled

OPTIONS:

Unscaled $d_A = r_{\mathrm{vdW}}$

Scaled $d_A = r_{\mathrm{vdW}} \times scale$

Shifted $d_A = (r_{\mathrm{vdW}} + shift) \times scale$

RECOMMENDATION:

None. The values of $scale$ and $shift$ are set with the **VDWscale** and **VDWshift** keywords, respectively.

VDWscale

Sets the empirical scale factor applied to the atomic van der Waals radius.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

1.2

OPTIONS:

$scale$ Specifies the desired dimensionless scaling factor for the atomic vdW radii.

RECOMMENDATION:

Use the default value.

VDWshift

Adjusts the center of the spherically-symmetric error functions when constructing the rigid vdW cavity.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

0.0 Ångstroms

OPTIONS:

$shift$ Specifies the desired shift (in Å).

RECOMMENDATION:

If **VDWtype** is set to **Shifted**, the vdW scale factor is set to 1.0 by default. This can be adjusted using the **VDWScale** keyword.

Setting the **SoluteCavity** keyword to **Spherical** requests the construction of a spherical cavity around the solute, and the dielectric is smoothly interpolated from vacuum to solvent using a hyperbolic tangent function:[36]

$$\varepsilon(r) = \frac{1}{2}\Big\{(\varepsilon_{\mathrm{solvent}} + 1) + (\varepsilon_{\mathrm{solvent}} - 1)\,\tanh\big[\alpha(r - r_{\mathrm{mid}})\big]\Big\}\,. \tag{11.44}$$

Here, r_{mid} is the distance where the dielectric assumes the value $\varepsilon(r_{\mathrm{mid}}) = (\varepsilon_{\mathrm{solvent}} + 1)/2$. The value of r_{mid} is taken to be the sum of the sphere radius, R, and half of the interpolation length, L: $r_{\mathrm{mid}} = R + L/2$. The sphere radius and interpolation length are controlled with the **SphereRadius** and **InterpolLength** keywords described below. The parameter α controls the sharpness of the switching process, with $\alpha = 4/L$ by default.

SphereRadius

Sets the radius of the spherical solute cavity.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

No default.

OPTIONS:

R Desired spherical cavity radius (in Å).

RECOMMENDATION:

None. See the Supporting Information of Ref. 36 for more information.

InterpolLength

Sets the length scale on which the dielectric is smoothly interpolated from vacuum to solvent.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

No default.

OPTIONS:

L Desired interpolation length (in Å).

RECOMMENDATION:

None.

InterpolScale

For a given interpolation length (L), **InterpolScale** (α) sets the sharpness of the dielectric transition from vacuum to solvent.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

$\alpha = 4/L$

OPTIONS:

α Desired interpolation scale factor (in Å^{-1}).

RECOMMENDATION:

Use the default unless a broader (smaller α) or narrower (larger α) transition region is desired.

Interface

Perform a solvation calculation at a solvent/vacuum interface.

INPUT SECTION: *$peqs*

TYPE:

STRING

DEFAULT:

False

OPTIONS:

True Modify the dielectric function to simulate an interface between solvent and vacuum.

False Perform a solvation calculation in bulk solvent.

RECOMMENDATION:

The user will also need to specify the length scale on which the dielectric is smoothly interpolated from the bulk solvent value to vacuum, and the location of the Gibbs dividing surface (see below).

By setting **Interface = True**, the dielectric function on the Cartesian grid is further modified to mimic a solvent/vacuum interface. First, the solute cavity is constructed as specified by the **SoluteCavity** keyword as discussed above. Then, the dielectric function is smoothly interpolated in the z direction across the Gibbs dividing surface ($z \equiv z_{\mathrm{GDS}}$) using the following hyperbolic tangent switching function:[35,36]

$$\varepsilon(z) = \frac{1}{2}\Big\{(\varepsilon_{\mathrm{solvent}} + 1) + (1 - \varepsilon_{\mathrm{solvent}})\,\tanh\big[\beta(z - z_{\mathrm{GDS}})\big]\Big\}\,. \tag{11.45}$$

This interpolates the dielectric function over the interface length $L_{\mathrm{interface}}$, centered at the Gibbs dividing surface, $z \equiv z_{\mathrm{GDS}}$. Both of these values are controlled by the keywords **InterfaceLength** and **GibbsDS**, respectively. Similar to the parameter α used in the spherical cavity construction, the parameter $\beta = 4/L_{\mathrm{interface}}$ controls the sharpness of the interpolation across the Gibbs dividing surface.

InterfaceLength

Sets the length scale over which the dielectric function is smoothly transitioned from bulk solvent to vacuum in the z direction

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

None

OPTIONS:

$L_{\mathrm{interface}}$ Desired interface length (in Å).

RECOMMENDATION:

This sets the value $\beta = 4/L_{\mathrm{interface}}$ in Eq. (11.45). See the Supporting Information of Ref. 36 for a full description of the solvent/vacuum interface construction.

GibbsDS

Sets the location of the Gibbs dividing surface, in the z direction.

INPUT SECTION: *$peqs*

TYPE:

FLOAT

DEFAULT:

None

OPTIONS:

z_{GDS} Desired location of the Gibbs dividing surface (in Å).

RECOMMENDATION:

Consult the literature. One such way to determine this value is to compute a density profile of the solvent as a function of z and set this location to be where the solvent density has decreased to 50% of the bulk value. Usually $z_{\mathrm{GDS}} \approx L_{\mathrm{interface}}/2$.

NonequilJob

Obtain the nonequilibrium free energy of solvation for a vertical ionization process.

INPUT SECTION: *$peqs*

TYPE:

STRING

DEFAULT:

False

OPTIONS:

True Compute the nonequilibrium free energy for a vertical ionization process.

False Compute the equilibrium solvation free energy.

RECOMMENDATION:

None.

NonequilPartition

Specifies the manner in which the solvent response is partitioned into fast and slow components.

INPUT SECTION: *$peqs*

TYPE:

STRING

DEFAULT:

Marcus

OPTIONS:

Marcus Employ the Marcus partitioning scheme.

Pekar Employ the Pekar partitioning scheme.

RECOMMENDATION:

Use the default. Although the fast and slow solvation responses are different between the two approaches, the total solvation free energy is the same, 178 but the Pekar scheme is computationally more expensive than the Marcus scheme.

NonequilState

Specifies the state of interest for a nonequilibrium vertical ionization.

INPUT SECTION: *$peqs*

TYPE:

STRING

DEFAULT:

Reference

OPTIONS:

Reference The reference (initial) state, from which an electron will be removed.

Ionized The final (ionized) state.

RECOMMENDATION:

None. Both values will be needed in a compound input job, in order to compute the nonequilibrium response to vertical ionization; see Example 11.2.11.5.

Lastly, by setting **SoluteCavity** to **Arbitrary** in the *$peqs* input section, the user may choose to specify a completely user-defined permittivity function $\varepsilon(\mathbf{r})$,[3] which must be generated externally and input into Q-CHEM pointwise on the PEqS grid, using a *$epsilon* section in the input file. The format for that section consists of Cartesian grid points (x_i, y_i, z_i) following by $\varepsilon(x_i, y_i, z_i)$, as shown below.

```
$epsilon
x1  y1  z1    eps(x1,y1,z1)
x1  y1  z2    eps(x1,y1,z2)
. . .
x1  y1  zNz   eps(z1,y1,zNz)
x1  y2  z1    eps(x1,y2,z1)
. . .
x1  y2  zNz   eps(z1,y2,zNz)
. . .
. . .
xNx  yNy  zNz    eps(xNx,yNy,zNz)
$end
```

11.2.11.5 Examples

The following example computes the solvation free energy of a water molecule immersed in water. The Cartesian grid is cubic with a side length of 15.0 Å and 73 grid points in each direction. A rigid vdW cavity is used based on scaled

vdW radii for the atomic spheres. The dielectric is not set, so takes the default value of 78.39.

Example 11.20 Free energy of solvation of water in water.

```
$molecule
   0 1
   O     0.053004    -0.020947    -0.034784
   H     0.003424     0.185855     0.910594
   H    -0.844842     0.146674    -0.358413
$end

$rem
   EXCHANGE          wB97X-V
   BASIS             6-31+G*
   SCF_CONVERGENCE   5
   THRESH            14
   SOLVENT_METHOD    PEQS
   PEQS_SWITCH       0
$end

$peqs
   SOLUTECAVITY RIGIDVDW
$end

$peqs_grid
   DimX 73 -7.50 7.50
   DimY 73 -7.50 7.50
   DimZ 73 -7.50 7.50
$end
```

The next example illustrates calculation of the solvation free energy of a water molecule at a water/vacuum interface, with the Gibbs dividing surface placed at $z = 0.50$ Å. The length of the interface is set to $L_{\mathrm{interface}} = 2.75$ Å and the dielectric is interpolated from bulk solvent to vacuum in the positive z direction across the Gibbs dividing surface. The

Cartesian grid, solute cavity, and solvent dielectric are the same as in the previous example.

Example 11.21 Free energy of solvation of water at a water/vacuum interface.

```
$molecule
   0 1
   O     0.053004    -0.020947    -0.034784
   H     0.003424     0.185855     0.910594
   H    -0.844842     0.146674    -0.358413
$end

$rem
   SCF_CONVERGENCE   5
   THRESH            14
   EXCHANGE          wB97X-V
   BASIS             6-31+G*
   SOLVENT_METHOD    PEQS
   PEQS_SWITCH       0
$end

$peqs
   SOLUTECAVITY RIGIDVDW
   INTERFACE TRUE
   INTERFACELENGTH 2.75
   GIBBSDS 0.50
   INTERFACEDIRECTION POSITIVE
$end

$peqs_grid
   DimX 73 -7.50 7.50
   DimY 73 -7.50 7.50
   DimZ 73 -7.50 7.50
$end
```

The final example illustrates a nonequilibrium (**NonEquilJob = True**) solvation calculation for the vertical ionization of H_2O^- in bulk water. This is a compound job that first calculates the equilibrium solvation free energy of the anionic state (**NonEquilState = Reference**), then computes the nonequilibrium energy correction for the ionized state (**NonEquilState = Ionized**). The two jobs are separated by "`@@@`". The Cartesian grid and solvent dielectric are the same as the previous examples, but the solute cavity is chosen to be spherical with a radius of 2.0 Å. The Marcus scheme (**NonEquilPartition = Marcus**) is used to partition the solvent response into fast and slow components. Since

this is the default method, the **NonEquilPartition** keyword is omitted.

Example 11.22 Nonequilibrium free energy of solvation for the vertical ionization of H_2O^-

```
$molecule
   -1 2
   O     0.053004    -0.020947    -0.034784
   H     0.003424     0.185855     0.910594
   H    -0.844842     0.146674    -0.358413
$end

$rem
   SCF_CONVERGENCE   5
   THRESH            14
   EXCHANGE          HF
   BASIS             6-31++G*
   SOLVENT_METHOD    PEQS
   PEQS_SWITCH       0
$end

$peqs
   SOLUTECAVITY   SPHERICAL
   SPHERERADIUS   2.00
   NONEQUILJOB    TRUE
   NONEQUILSTATE  REFERENCE
$end

$peqs_grid
   DimX 73 -7.50 7.50
   DimY 73 -7.50 7.50
   DimZ 73 -7.50 7.50
$end

@@@

$molecule
   0 1
   O     0.053004    -0.020947    -0.034784
   H     0.003424     0.185855     0.910594
   H    -0.844842     0.146674    -0.358413
$end

$rem
   SCF_CONVERGENCE   5
   THRESH            14
   EXCHANGE          HF
   BASIS             6-31++G*
   SOLVENT_METHOD    PEQS
   PEQS_SWITCH       0
$end

$peqs
   SOLUTECAVITY   SPHERICAL
   SPHERERADIUS   2.00
   NONEQUILJOB    TRUE
   NONEQUILSTATE  IONIZED
$end

$peqs_grid
   DimX 73 -7.50 7.50
   DimY 73 -7.50 7.50
   DimZ 73 -7.50 7.50
$end
```

11.3 Stand-Alone QM/MM Calculations

Q-CHEM can perform hybrid quantum mechanics/molecular mechanics (QM/MM) calculations either as a stand-alone program, which is described in this section, or in conjunction with the CHARMM package.[68] See Section 11.4 for a description of a latter approach.

11.3.1 Available QM/MM Methods and Features

Three modes of operation are available:

- MM calculations only (no QM)
- QM/MM calculations using a two-layer ONIOM model with mechanical embedding
- QM/MM calculations using the Janus model for electronic embedding

Q-CHEM can carry out purely MM calculations, wherein the entire molecular system is described by a MM force field and no electronic structure calculation is performed. The MM force fields available at present are AMBER,[166] CHARMM,[51] and OPLSAA.[74]

As implemented in Q-CHEM, the ONIOM model[163] is a mechanical embedding scheme that partitions a molecular system into two subsystems (layers): an MM subsystem and a QM subsystem. The total energy of an ONIOM system is given by

$$E_{\text{total}} = E_{\text{total}}^{\text{MM}} - E_{\text{QM}}^{\text{MM}} + E_{\text{QM}}^{\text{QM}} \tag{11.46}$$

where $E_{\text{total}}^{\text{MM}}$ is the MM energy of the total system (*i.e.*, QM + MM subsystems), $E_{\text{QM}}^{\text{MM}}$ is the MM energy of the QM subsystem, and $E_{\text{QM}}^{\text{QM}}$ is the QM energy of the QM subsystem. MM energies are computed via a specified MM force field, and QM energies are computed via a specified electronic structure calculation.

The advantage of the ONIOM model is its simplicity, which allows for straightforward application to a wide variety of systems. A disadvantage of this approach, however, is that QM subsystem does not interact directly with the MM subsystem. Instead, such interactions are incorporated indirectly, in the E_{total}^{MM} contribution to the total energy. As a result, the QM electron density is not polarized by the electrostatic charges of the MM subsystem.

If the QM/MM interface partitions the two subsystems across a chemical bond, a link atom (hydrogen) must be introduced to act as a cap for the QM subsystem. Currently, Q-CHEM supports only carbon link atoms, of atom type 26, 35, and 47 in the CHARMM27 force field.

The Janus model[144] is an electronic embedding scheme that also partitions the system into MM and QM subsystems, but is more versatile than the ONIOM model. The Janus model in Q-CHEM is based upon the "YinYang atom" model of Shao and Kong.[145] In this approach, the total energy of the system is simply the sum of the subsystem energies,

$$E_{\text{total}} = E_{\text{MM}} + E_{\text{QM}} \tag{11.47}$$

The MM subsystem energy, E_{MM}, includes van der Waals interactions between QM and MM atoms but not QM/MM Coulomb interactions. Rather, E_{QM} includes the direct Coulomb potential between QM atoms and MM atoms as external charges during the QM calculation, thus allowing the QM electron density to be polarized by the MM atoms. Because of this, Janus is particularly well suited (as compared to ONIOM) for carrying out excited-state QM/MM calculations, for excited states of a QM model system embedded within the electrostatic environment of the MM system. Within a Janus calculation, Q-CHEM first computes E_{MM} with the specified force field and then computes E_{QM} with the specified electronic structure theory.

When the Janus QM/MM partition cuts across a chemical bond, a YinYang atom[145] is automatically introduced by Q-CHEM. This atom acts as a hydrogen cap in the QM calculation, yet also participates in MM interactions. To retain charge neutrality of the total system, the YinYang atom has a single electron and a modified nuclear charge in the QM calculation, equal to $q_{nuclear} = 1 + q_{\mathrm{MM}}$ (*i.e.*, the charge of a proton plus the charge on the YinYang atom in the MM subsystem).

Because this modified charge will affect the bond containing the YinYang atom, an additional repulsive Coulomb potential is applied between the YinYang atom and its connecting QM atom to maintain a desirable bond length. The additional repulsive Coulomb energy is added to E_{MM}. The YinYang atom can be an atom of any kind, but it is highly recommended to use carbon atoms as YinYang atoms.

Q-CHEM's stand-alone QM/MM capabilities also include the following features:

- Analytic QM/MM gradients are available for QM subsystems described with density functional theory (DFT) or Hartree-Fock (HF) electronic structure theory, allowing for geometry optimizations and QM/MM molecular dynamics.
- Single-point QM/MM energy evaluations are available for QM subsystems described with most post-HF correlated wave functions.
- Single-point QM/MM calculations are available for excited states of the QM subsystem, where the latter may be described using CIS, TDDFT, or correlated wave function models. Analytic gradients for excited states are available for QM/MM calculations if the QM subsystem is described using CIS.
- Single-point MM or QM/MM energy evaluations and analytic gradients are available using periodic boundary conditions with Ewald summation.
- Implicit solvation for both Janus QM/MM calculations as well as MM-only calculations is available using the Polarizable Continuum Models (PCMs) discussed in Section 11.2.3.
- Gaussian blurring of MM external charges is available for Janus QM/MM calculations.
- User-defined MM atoms types, MM parameters, and force fields.

11.3.2 Using the Stand-Alone QM/MM Features

11.3.2.1 *$molecule* section

To perform QM/MM calculations, the user must assign MM atom types for each atom in the *$molecule* section. The format for this specification is modeled upon that used by the TINKER molecular modeling package,[135] although the TINKER program is *not* required to perform QM/MM calculations using Q-CHEM. Force field parameters and MM atom type numbers used within Q-CHEM are identical to those used TINKER for the AMBER99, CHARMM27, and OPLSAA force fields, and the format of the force field parameters files is also the same.

The *$molecule* section must use Cartesian coordinates to define the molecular geometry for internal QM/MM calculations; the *Z*-matrix format is not valid. MM atom types are specified in the *$molecule* section immediately after the Cartesian coordinates on a line so that the general format for the *$molecule* section is

```
$molecule
  <Charge> <Multiplicity>
  <Atom> <X> <Y> <Z> <MM atom type>
  . . .
$end
```

For example, one can define a TIP3P water molecule using AMBER99 atom types, as follows:

```
$molecule
   0 1
   O        -0.790909    1.149780    0.907453  2001
   H        -1.628044    1.245320    1.376372  2002
   H        -0.669346    1.913705    0.331002  2002
$end
```

When the input is specified as above, the MM bond connectivity, in principle, can be determined based on the distances between atoms; if two atoms are sufficiently close, they are considered to be bonded. Occasionally, this approach can lead to problems when non-bonded atoms are in close proximity of one another, leading to incorrect classification of bonded atom pairs, regardless of whether the corresponding MM bond parameters are available. To avoid such a scenario, Q-CHEM() requires the user to specify the bonds explicitly by setting the *$rem* variable USER_CONNECT = TRUE; the *$molecule* section must have the following format

```
$molecule
  <Charge> <Multiplicity>
  <Atom> <X> <Y> <Z> <MM atom type> <Bond 1> <Bond 2> <Bond 3> <Bond 4>
  . . .
$end
```

Each `<Bond #>` is the index of an atom to which `<Atom>` is bonded. Four bonds must be specified for each atom, even if that atom is connected to fewer than four other atoms. (For non-existent bonds, use zero as a placeholder.) Currently, Q-CHEM supports no more than four MM bonds per atom.

After setting USER_CONNECT = TRUE, a TIP3P water molecule in the AMBER99 force field could be specified as follows:

```
$molecule
   0 1
   O        -0.790909    1.149780    0.907453  2001  2 3 0 0
   H        -1.628044    1.245320    1.376372  2002  1 0 0 0
   H        -0.669346    1.913705    0.331002  2002  1 0 0 0
$end
```

Explicitly defining the bonds in this way is highly recommended and a requirement.

11.3.2.2 *$force_field_params* section

In many cases, all atoms types (within both the QM and MM subsystems) will be defined by a given force field. In certain cases, however, a particular atom type may not be defined in a given force field. For example, a QM/MM calculation on the propoxide anion might consist of a QM subsystem containing an alkoxide functional group, for

which MM parameters do not exist. Even though the alkoxide moiety is described using quantum mechanics, van der Waals parameters are nominally required for atoms within the QM subsystem, which interact with the MM atoms via Lennard-Jones-type interactions.

In such cases, there are four possible options, the choice of which is left to the user's discretion:

1. Use a similar MM atom type as a substitute for the missing atom type.
2. Ignore the interactions associated with the missing atom type.
3. Define a new MM atom type and associated parameters.
4. Define a new force field.

These options should be applied with care. Option 1 involves selecting an atom type that closely resembles the undefined MM atom. For example, the oxygen atom of an alkoxide moiety could perhaps use the MM atom type corresponding to the oxygen atom of a neutral hydroxyl group. Alternatively, the atom type could be ignored altogether (option 2) by specifying MM atom type 0 (zero). Setting the atom type to zero should be accompanied with setting all four explicit bond connections to placeholders if USER_CONNECT = TRUE. An atom type of zero will cause all MM energies involving that atom to be zero.

The third option in the list above requires the user to specify a *$force_field_params* section in the Q-CHEM input file. This input section can be used to add new MM atom type definitions to one of Q-CHEM's built-in force fields. At a minimum, the user must specify the atomic charge and two Lennard-Jones parameters (radius and well depth, ϵ), for each new MM atom type. Bond, angle, and torsion parameters for stretches, bends, and torsions involving the new atom type may also be specified, if desired. The format for the *$force_field_params* input section is

```
$force_field_params
NumAtomTypes <n>
AtomType -1 <Charge> <LJ Radius> <LJ Epsilon>
AtomType -2 <Charge> <LJ Radius> <LJ Epsilon>
. . .
AtomType -n <Charge> <LJ Radius> <LJ Epsilon>
Bond <a> <b> <Force constant> <Equilibrium Distance>
. . .
Angle <a> <b> <c> <Force constant> <Equilibrium Angle>
. . .
Torsion <a> <b> <c> <d> <Force constant> <Phase Angle> <Multiplicity>
. . .
$end
```

The first line in this input section specifies how many new MM atom types appear in this section (<n>). These are specified on the following lines labeled with the AtomType tag. The atom type numbers are required to be negative and to appear in the order $-1, -2, -3, \ldots, -n$. The *$molecule* section for a water molecule, with user-defined MM parameters for both oxygen and hydrogen, might appear as follows:

```
$molecule
   0 1
   O        -0.790909    1.149780    0.907453        -1  2 3 0 0
   H        -1.628044    1.245320    1.376372        -2  1 0 0 0
   H        -0.669346    1.913705    0.331002        -2  1 0 0 0
$end
```

The remainder of each AtomType line in the *$force_field_params* section consists of a charge (in elementary charge units), a Lennard-Jones radius (in Å), and a Lennard-Jones well depth (ϵ, in kcal/mol).

Each (optional) `Bond` line in the *$force_field_params* section defines bond-stretching parameters for a bond that contains a new MM atom type. The bond may consist of both atoms `<a>` and `<b>` defined an `AtomType` line, or else `<a>` may be defined with an `AtomType` line and `<b>` defined as a regular atom type for the force field. In the latter case, the label for `<b>` should be the number of its general van der Waals type. For example, the atom type for a TIP3P oxygen in AMBER99 is 2001, but its van der Waals type is 21, so the latter would be specified in the `Bond` line. The remaining entries of each `Bond` line are the harmonic force constant, in kcal/mol/Å^2, and the equilibrium distance, in Ångstrom.

Similar to the `Bond` lines, each (optional) `Angle` line consists of one or more new atom types along with existing van der Waals types. The central atom of the angle is `<b>`. The harmonic force constant and equilibrium bond angle are the final entries in each `Bond` line. The bond angle is given in degrees and the force constant in kcal/mol/(radian)^2.[103,168]

Each (optional) `Torsion` line consists of one or more new MM atom types along with regular van der Waals types. The connectivity of the four atoms that constitute the dihedral angle is `<a>`–`<b>`–`<c>`–`<d>`, and the torsional potential energy function is

$$E_{\mathrm{torsion}}(\theta) = k_{\mathrm{torsion}}[1 + \cos(m\theta - \phi)] \tag{11.48}$$

The force constant (k_{torsion}) is specified in kcal/mol and the phase angle (ϕ) in degrees. The multiplicity (m) is an integer.

11.3.2.3 User-Defined Force Fields

Option 4 in the list on page 1226 is the most versatile, and allows the user to define a completely new force field. This option is selected by setting FORCE_FIELD = READ, which tells Q-CHEM to read force field parameters from a text file whose name is specified in the *$force_field_params* section as follows:

```
$force_field_params
    Filename <path/filename>
$end
```

Here, `<path/filename>` is the full (absolute) path and name of the file that Q-CHEM will attempt to read for the MM force field. E.g., if the user has a file named `MyForceField.prm` that resides in the path `/Users/me/parameters/`, then this would be specified as

```
$force_field_params
   Filename /Users/me/parameters/MyForceField.prm
$end
```

Within the force field file, the user should first declare various rules that the force field will use, including how van der Waals interactions will be treated, scaling of certain interactions, and the type of improper torsion potential. The rules are declared in the file as follows:

```
RadiusRule <option>
EpsilonRule <option>
RadiusSize <option>
ImptorType <option>
vdw-14-scale <x>
chg-14-scale <x>
torsion-scale <x>
```

Currently, only a Lennard-Jones potential is available for van der Waals interactions. `RadiusRule` and `EpsilonRule` control how to average σ and ϵ, respectively, between atoms A and B in their Lennard-Jones potential. The options available for both of these rules are `Arithmetic` [*e.g.*, $\sigma_{AB} = (\sigma_A + \sigma_B)/2$] or `Geometric` [*e.g.*, $\sigma_{AB} =$

$(\sigma_A \sigma_B)^{1/2}$]. `RadiusSize` has options `Radius` or `Diameter`, which specify whether the parameter σ is the van der Waals radius or diameter in the Lennard-Jones potential.

`ImptorType` controls the type of potential to be used for improper torsion (out-of-plane bending) energies, and has two options: `Trigonometric` or `Harmonic`. These options are described in more detail below.

The scaling rules takes a floating point argument <x>. The `vdw-14-scale` and `chg-14-scale` rules only affect van der Waals and Coulomb interactions, respectively, between atoms that are separated by three consecutive bonds (atoms 1 and 4 in the chain of bonds). These interaction energies will be scaled by <x>. Similarly, `torsion-scale` scales dihedral angle torsion energies.

After declaring the force field rules, the number of MM atom types and van der Waals types in the force field must be specified using:

```
NAtom <n>
Nvdw <n>
```

where <n> is a positive integer.

Next, the atom types, van der Waals types, bonds, angles, dihedral angle torsion, improper torsions, and Urey-Bradley parameters can be declared in the following format:

```
Atom 1 <Charge> <vdw Type index> <Optional description>
Atom 2 <Charge> <vdw Type index> <Optional description>
. . .
Atom <NAtom> <Charge> <vdw Type index> <Optional description>
. . .
vdw 1 <Sigma> <Epsilon> <Optional description>
vdw 2 <Sigma> <Epsilon> <Optional description>
. . .
vdw <Nvdw> <Sigma> <Epsilon> <Optional description>
. . .
Bond <a> <b> <Force constant> <Equilibrium Distance>
. . .
Angle <a> <b> <c> <Force constant> <Equilibrium Angle>
. . .
Torsion <a> <b> <c> <d> <Force constant 1> <Phase Angle 1> <Multiplicity 1>
. . .
Improper <a> <b> <c> <d> <Force constant> <Equilibrium Angle> <Multiplicity>
. . .
UreyBrad <a> <b> <c>  <Force constant> <Equilibrium Distance>
```

The parameters provided in the force field parameter file correspond to a basic MM energy functional of the form

$$E_{\mathrm{MM}} = E_{\mathrm{Coul}} + E_{\mathrm{vdW}} + E_{\mathrm{bond}} + E_{\mathrm{angle}} + E_{\mathrm{torsion}} + E_{\mathrm{imptor}} + E_{\mathrm{UreyBrad}} \tag{11.49}$$

Coulomb and van der Waals interactions are computed for all non-bonded pairs of atoms that are at least three consecutive bonds apart (*i.e.*, 1–4 pairs and more distant pairs). The Coulomb energy between atom types `1` and `2` is simply

$$E_{\mathrm{Coul}} = f_{\mathrm{scale}} \frac{q_1 q_2}{r_{12}} \tag{11.50}$$

where q_1 and q_2 are the respective charges on the atoms (specified with `<Charge>` in elementary charge units) and r_{12} is the distance between the two atoms. For 1–4 pairs, f_{scale} is defined with `chg-14-scale` but is unity for all

other valid pairs. The van der Waals energy between two atoms with van der Waals types `a` and `b`, and separated by distance r_{ab}, is given by a "6-12" Lennard-Jones potential:

$$E_{\mathrm{vdW}}(r_{ab}) = f_{\mathrm{scale}}\,\epsilon_{ab}\left[\left(\frac{\sigma_{ab}}{r_{ab}}\right)^{12} - 2\left(\frac{\sigma_{ab}}{r_{ab}}\right)^{6}\right] \tag{11.51}$$

Here, f_{scale} is the scaling factor for 1–4 interactions defined with `vdw-14-scale` and is unity for other valid interactions. The quantities ϵ_{ab} and σ_{ab} are the averages of the parameters of atoms `a` and `b` as defined with `EpsilonRule` and `RadiusRule`, respectively (see above). The units of `<Sigma>` are Å , and the units of `<Epsilon>` are kcal/mol. Hereafter, we refer to atoms' van der Waals types with `a, b, c, ...` and atoms' charges with `1,2, 3, ...`.

The bond energy is a harmonic potential,

$$E_{\mathrm{bond}}(r_{ab}) = k_{\mathrm{bond}}(r_{ab} - r_{eq})^2 \tag{11.52}$$

where k_{bond} is provided by `<Force Constant>` in kcal/mol/Å^2 and r_{eq} by `<Equilibrium Distance>` in Å. Note that `<a>` and `<b>` in the `Bond` definition correspond to the van der Waals type indices from the `vdw` definitions, *not* the `Atom` indices.

The bending potential between two adjacent bonds connecting three different atoms (`<a>-<b>-<c>`) is also taken to be harmonic,

$$E_{\mathrm{angle}}(\theta_{abc}) = k_{\mathrm{angle}}(\theta_{abc} - \theta_{eq})^2 \tag{11.53}$$

Here, k_{angle} is provided by `<Force Constant>` in kcal/mol/degrees and θ_{eq} by `<Equilibrium Angle>` in degrees. Again, `<a>`, `<b>`, and `<c>` correspond to van der Waals types defined with `vdw`.

The energy dependence of the `<a>-<b>-<c>-<d>` dihedral torsion angle, where `<a>`, `<b>`, `<c>`, and `<d>` are van der Waals types, is defined by

$$E_{\mathrm{torsion}}(\theta_{abcd}) = f_{\mathrm{scale}}\sum_{m} k_{abcd}[1 + \cos(m\theta_{abcd} - \phi)] \tag{11.54}$$

Here, f_{scale} is the scaling factor defined by `torsion-scale`. The force constant k_{abcd} is defined with `<Force constant>` in kcal/mol, and the phase angle ϕ is defined with `<Phase Angle>` in degrees. The summation is over multiplicities, m, and Q-CHEM supports up to three different values of m per dihedral angle. The force constants and phase angles may depend on m, so if more than one multiplicity is used, then `<Force constant> <Phase Angle> <Multiplicity>` should be specified for each multiplicity. For example, to specify a dihedral torsion between van der Waals types 2–1–1–2, with multiplicities $m = 2$ and $m = 3$, we might have:

```
Torsion  2  1  1  2  2.500  180.0  2 1.500 60.0 3
```

Improper torsion angle energies for four atoms `<a>-<b>-<c>-<d>`, where `<c>` is the central atom, can be computed in one of two ways, as controlled by the `ImptorType` rule. If `ImptorType` is set to `Trigonometric`, then the improper torsion energy has a functional form similar to that used for dihedral angle torsions:

$$E_{\mathrm{imptor}}(\theta_{abcd}) = \frac{k_{abcd}}{N_{\mathrm{equiv}}}[1 + \cos(m\,\theta_{abcd} - \phi)] \tag{11.55}$$

Here, θ_{abcd} is the out-of-plane angle of atom `<c>`, in degrees, and k_{abcd} is the force constant defined with `<Force Constant>`, in kcal/mol. The phase ϕ and multiplicity m need to be specified in the `Improper` declaration, although the definition of an improper torsion suggests that these values should be set to $\phi = 0$ and $m = 2$. The quantity N_{equiv} accounts for the number of equivalent permutations of atoms `<a>`, `<b>`, and `<d>`, so that the improper torsion angle is only computed once. If `ImptorType` is set to `Harmonic`, then in place of Eq. (11.55), the following energy function is used:

$$E_{\mathrm{imptor}}(\theta_{abcd}) = \frac{k_{abcd}}{N_{\mathrm{equiv}}}\theta_{abcd}^2 \tag{11.56}$$

The Urey-Bradley energy, which accounts for a non-bonded interaction between atoms <a> and <c> that are separated by two bonds (*i.e.*, a 1-3 interaction through <a>-<b>-<c>), is given by

$$E_{\mathrm{UreyBrad}}(r_{ac}) = k_{abc}(r_{ac} - r_{eq})^2 \tag{11.57}$$

The distance in Å between atoms <a> and <c> is r_{ac}, the equilibrium distance r_{eq} is provided by <Equilibrium Distance> in Å, and the force constant k_{abc} is provided by <Force Constant> in kcal/mol/Å^2.

A short example of a valid text-only file defining a force field for a flexible TIP3P water could be as follows:

```
//-- Force Field Example --//

// -- Rules -- //
RadiusRule Geometric
RadiusSize Radius
EpsilonRule Geometric
ImptorType Trigonometric
vdw-14-scale 1.0
chg-14-scale 0.8
torsion-scale 0.5

// -- Number of atoms and vdw to expect -- //
NAtom 2
Nvdw 2

// -- Atoms -- //
Atom 1 -0.8340  2  TIP3P Oxygen
Atom 2  0.4170  1  TIP3P Hydrogen

// -- vdw -- //
vdw 1  0.0000  0.0000 H parameters
vdw 2  1.7682  0.1521 O parameters

// -- Bond -- //
Bond 1 2  553.0  0.9572

// -- Angle -- //
Angle 1 2 1  100.0  104.52
```

Lines that do not begin with one of the keywords will be ignored, and have been used here as comments.

11.3.2.4 *$qm_atoms* and *$forceman* sections

For QM/MM calculations (but not for purely MM calculations) the user must specify the QM subsystem using a *$qm_atoms* input section, which assumes the following format:

```
$qm_atoms
   <QM atom 1 index> <QM atom 2 index> . . .
   . . .
   <QM atom n index>
$end
```

Multiple indices can appear on a single line and the input can be split across multiple lines. Each index is an integer corresponding to one of the atoms in the *$molecule* section, beginning at 1 for the first atom in the *$molecule* section.

Link atoms for the ONIOM model and YinYang atoms for the Janus model are not specified in the *$qm_atoms* section, as these are inserted automatically whenever a bond connects a QM atom and an MM atom.

Q-CHEM 4.2.2 and later versions also support, for example

```
$qm_atoms
   18:31 35
$end
```

which specifies 15 QM atoms (atoms 18 through 31; atom 35).

For Janus QM/MM calculations, there are several ways of dealing with van der Waals interactions between the QM and MM atoms. By default, van der Waals interactions are computed for all QM–MM and MM–MM atom pairs but not for QM–QM atom pairs. In some cases, the user may prefer not to neglect the van der Waals interactions between QM–QM atoms, or the user may prefer to neglect any van der Waals interaction that involves a QM atom. Q-CHEM allows the user this control via two options in the *$forceman* section. To turn on QM–QM atom van der Waals interactions, the user should include the following in their input:

```
$forceman
   QM-QMvdw
$end
```

Similarly, to turn off all van der Waals interactions with QM atoms, the following should be included:

```
$forceman
   NoQM-QMorQM-MMvdw
$end
```

11.3.2.5 Periodic Boundary Conditions

Periodic boundary conditions (using Ewald summation for the long-range Coulomb interactions) can be used in conjunction with both MM-only calculations and QM/MM calculations. The approach is based off of the work of Nam *et al.*[118] and (independently) Riccardi *et al.*,[137] as implemented in both the AMBER[165] and CHARMM[16,137] programs. These approaches use Mulliken charges to represent the periodic images of the wave function, and while suitable for semi-empirical calculations with minimal basis sets, instabilities in the Mulliken charges for extended basis sets lead to SCF convergence failure in the QM/MM-Ewald calculations.[64] The implementation in Q-CHEM thus allows for the use of ChElPG charges to represent the image wave functions, affording an algorithm that is stable in extended basis sets.[64,65]

The efficiency of the Ewald summation is governed by the parameter, α, that controls the partition of the Coulomb potential into short- and long-range components, and in the QM/MM-Ewald method there are separate values of α for the QM and MM portions of the calculations. Improper selection of α_{MM} and/or α_{QM} can greatly increase the computational time, and the choices that are optimal for MM calculations need not be optimal for QM/MM calculations.[64] The cost of the MM Ewald summation scales as $\mathcal{O}(N_{\mathrm{recip}}N_{\mathrm{atoms}})$, where N_{recip} is the number of reciprocal-space lattice vectors that is used for the k-space sum. The QM portion of the calculation scales as $\mathcal{O}(N_{\mathrm{recip}}N_{\mathrm{QM}}N_{\mathrm{atoms}})$, where N_{QM} is the number of QM atoms (whereas $N_{\mathrm{atoms}} = N_{\mathrm{QM}} + N_{\mathrm{MM}}$ is the total number of atoms). The MM Ewald parameter is thus selected to minimize the amount of work that is done in real space. The optimal value, which is typically $\alpha_{\mathrm{MM}} \approx 0.5$ Å^{-1}, can be found by solving the equation[64]

$$\alpha_{\mathrm{MM}} = 2C/L \tag{11.58}$$

where

$$C = \left[-\ln\left(10^{-\text{SCF\_CONVERGENCE}}\right)\right]^{1/2} \tag{11.59}$$

and L is the length of the simulation cell. (Only cubic simulation cells are available at present.) In contrast, the parameter α_{QM} should be selected to minimize the total number of vectors in both real and reciprocal space. The optimal value, which is often $\alpha_{\mathrm{QM}} \approx 0.1$ Å^{-1}, is determined by solving the equation[64]

$$\frac{2CL^3\alpha_{\mathrm{QM}}^3}{\pi^{3/2}} + \frac{\alpha_{\mathrm{QM}}^2 L^2}{\pi^{1/2}} - \alpha_{\mathrm{QM}} L - 2C = 0 \; . \tag{11.60}$$

To perform an MM- or QM/MM-Ewald job, one must set MM_SUBTRACTIVE = TRUE and EWALD_ON = TRUE in the *$rem* section, but otherwise job control is largely done through the *$forceman* section. The following variables must be set for every type of Ewald calculation.

- The keyword **Ewald** will turn on Ewald summation.
- The keyword **alpha** should be followed by a value for the MM Ewald parameter and then the QM Ewald parameter. (The latter must be set even for MM-only jobs.)
- **Box_length** specifies the side length of the cubic simulation cell, in Å.

The following parameters are optional for further job control.

- **Dielectric** specifies a dielectric constant for the surrounding medium, which appears in the "dipole term" of Ewald summation (E_{dipole} in Ref. 64). If no value is set, the dielectric constant is set to infinity, corresponding to "tin foil" boundary conditions.
- The keyword **Ewald_SCF_thresh_on**, followed by a real number, causes Q-CHEM to wait until the DIIS error falls below the specified value before adding the Ewald correction to the Fock matrix, thus obviating the sometimes-costly Ewald correction in early SCF cycles. (The default value is 1.0, which turns on Ewald summation immediately in most cases)

A short example of a *$forceman* section using Ewald summation could be as follows:

```
$forceman
   ewald
   alpha                0.35   0.1
   box_length           15.00
   dielectric           88.0
   mm_read_scratch
   ewald_scf_thresh_on  0.0001
$end
```

11.3.2.6 L-BFGS for QM/MM optimization

A QM/MM geometry optimization job using L-BFGS algorithm can be requested by using the following option in the *$forceman* section:

```
$forceman
QMMM-LBFGS
$end
```

The additional job controls for this job are described below. These options are also need to be specified in the *$forceman* section.

- **LBFGS_M**: Curvature information from the last M steps will be used to construct the Hessian approximation. The default value is 10.
- **MaxSteps**: Maximum number of optimization steps. The default value is 2000.
- **ConvG**: Convergence on the maximun gradient components. The default tolerance value is 0.0003.
- **ConvD**: Convergence on the maximum atomic displacement. The default tolerance value is 0.0012.
- **ConvE**: Convergence on maximum (absolute) energy change. The default tolerance value is 0.000001.

A sample *$forceman* section for a QM/MM optimization job using all the above options with their default values looks like:

```
$forceman
QMMM-LBFGS
LBFGS_M 10
MaxSteps 2000
ConvG  0.0003
ConvD  0.0012
ConvE  0.000001
$end
```

Example 11.23 QM/MM geometry optimization of water dimer using L-BFGS.

```
$molecule
   0 1
   O     1.3584299158    0.1073418692   -0.2758823010   101 2 3 0 0
   H     0.3884464033   -0.0182409613   -0.1252830261   88 1 0 0 0
   H     1.7255084907   -0.4596377323    0.4452211001   88 1 0 0 0
   O    -1.3286731559   -0.2344591932    0.1344130752   101 5 6 0 0
   H    -1.7784222051    0.6341523014    0.2664866218   88 4 0 0 0
   H    -1.7690342253   -0.5495694315   -0.6907120483   88 4 0 0 0
$end

$rem
   JOBTYPE           opt
   EXCHANGE          hf
   BASIS             sto-3g
   QM_MM_INTERFACE   janus
   FORCE_FIELD       charmm27
   USER_CONNECT      true
   NO_REORIENT       true
   INTEGRAL_SYMMETRY false
$end

$forceman
QMMM-LBFGS
LBFGS_M 10
$end

$qm_atoms
1 2 3
$end
```

11.3.3 Additional Job Control Variables

A QM/MM job is requested by setting the *$rem* variables QM_MM_INTERFACE and FORCE_FIELD. Also required are a *$qm_atoms* input section and appropriate modifications to the *$molecule* section, as described above. Additional job control variables are detailed here.

QM_MM_INTERFACE

Enables internal QM/MM calculations.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

MM	Molecular mechanics calculation (*i.e.*, no QM region)
ONIOM	QM/MM calculation using two-layer mechanical embedding
JANUS	QM/MM calculation using electronic embedding

RECOMMENDATION:

The ONIOM model and Janus models are described above. Choosing MM leads to no electronic structure calculation. However, when using MM, one still needs to define the *$rem* variables BASIS and EXCHANGE in order for Q-CHEM to proceed smoothly.

FORCE_FIELD

Specifies the force field for MM energies in QM/MM calculations.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

AMBER99	AMBER99 force field
CHARMM27	CHARMM27 force field
OPLSAA	OPLSAA force field

RECOMMENDATION:

None.

CHARGE_CHARGE_REPULSION

The repulsive Coulomb interaction parameter for YinYang atoms.

TYPE:

INTEGER

DEFAULT:

550

OPTIONS:

n Use Q = $n \times 10^{-3}$

RECOMMENDATION:

The repulsive Coulomb potential maintains bond lengths involving YinYang atoms with the potential $V(r) = Q/r$. The default is parameterized for carbon atoms.

FORCEMAN_PRINT

Controls printing level for MM (and thus QM/MM) jobs.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Minimal printing, as befits a large MM job that runs for many steps.

1 Additional information including MM gradient.

2 Print individual terms in the MM energy expression.

3 Print connectivity information.

4 Print individual terms in the MM gradient.

RECOMMENDATION:

Use the default unless trying to diagnose a problem.

GAUSSIAN_BLUR

Enables the use of Gaussian-delocalized external charges in a QM/MM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Delocalizes external charges with Gaussian functions.

FALSE Point charges

RECOMMENDATION:

None

GAUSS_BLUR_WIDTH

Delocalization width for external MM Gaussian charges in a Janus calculations.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Use a width of $n \times 10^{-4}$ Å.

RECOMMENDATION:

Blur all MM external charges in a QM/MM calculation with the specified width. Gaussian blurring is currently incompatible with PCM calculations. Values of 1.0–2.0 Å are recommended in Ref. 45.

MM_SUBTRACTIVE

Specifies whether a subtractive scheme is used in the E_{Coul}, Eq. (11.50), portion of the calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Only pairs that are not 1-2, 1-3, or 1-4 pairs are used.

TRUE All pairs are calculated, and then the pairs that are double counted (1-2, 1-3, and 1-4) are subtracted out.

RECOMMENDATION:

When running QM/MM or MM calculations there is not recommendation. When running a QM/MM-Ewald calculation the value must be set to TRUE.

MODEL_SYSTEM_CHARGE

Specifies the QM subsystem charge if different from the *$molecule* section.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n The charge of the QM subsystem.

RECOMMENDATION:

This option only needs to be used if the QM subsystem (model system) has a charge that is different from the total system charge.

MODEL_SYSTEM_MULT

Specifies the QM subsystem multiplicity if different from the *$molecule* section.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n The multiplicity of the QM subsystem.

RECOMMENDATION:

This option only needs to be used if the QM subsystem (model system) has a multiplicity that is different from the total system multiplicity. ONIOM calculations must be closed shell.

PRINT_INPUT

Specifies whether to echo the input file in the output.

TYPE:

INTEGER

DEFAULT:

TRUE

OPTIONS:

TRUE Print the Q-CHEM input file as part of the Q-CHEM output file.

FALSE Do not repeat the input file.

RECOMMENDATION:

This is useful for record-keeping and defaults to TRUE except for QM/MM jobs, where the default value is FALSE due to the size of some input files.

USER_CONNECT

Enables explicitly defined bonds.

TYPE:

STRING

DEFAULT:

FALSE

OPTIONS:

TRUE Bond connectivity is read from the *$molecule* section

FALSE Bond connectivity is determined by atom proximity

RECOMMENDATION:

Set to TRUE if bond connectivity is known, in which case this connectivity must be specified in the *$molecule* section. This greatly accelerates MM calculations.

11.3.4 QM/MM Examples

Features of this job:

- Geometry optimization using ONIOM mechanical embedding.
- MM region (water 1) described using OPLSAA.
- QM region (water 2) described using PBE0/6-31G*.
- *$molecule* input section contains user-defined MM bonds. A zero is used as a placeholder if there are no more connections.

Example 11.24 ONIOM optimization of water dimer.

```
$molecule
   0 1
   O   -0.790909    1.149780    0.907453  186  2  3  0  0
   H   -1.628044    1.245320    1.376372  187  1  0  0  0
   H   -0.669346    1.913705    0.331002  187  1  0  0  0
   O    1.178001   -0.686227    0.841306  186  5  6  0  0
   H    0.870001   -1.337091    1.468215  187  4  0  0  0
   H    0.472696   -0.008397    0.851892  187  4  0  0  0
$end

$rem
   JOBTYPE           opt
   METHOD            pbe0
   BASIS             6-31G*
   QM_MM_INTERFACE   oniom
   FORCE_FIELD       oplsaa
   USER_CONNECT      true
   MOLDEN_FORMAT     true
$end

$qm_atoms
   4 5 6
$end
```

Features of this job:

- Janus electronic embedding with a YinYang link atom (the glycosidic carbon at the C1′ position of the deoxyribose).

- MM region (deoxyribose) is described using AMBER99.
- QM region (adenine) is described using HF/6-31G*.
- The first 5 electronically excited states are computed with CIS. MM energy interactions between a QM atom and an MM atom (*e.g.*, van der Waals interactions, as well as angles involving a single QM atom) are assumed to be the same in the excited states as in the ground state.
- *$molecule* input section contains user-defined MM bonds.
- Gaussian-blurred charges are used on all MM atoms, with a width set to 1.5 Å.

Example 11.25 Excited-state single-point QM/MM calculation on deoxyadenosine.

```
$molecule
   0 1
   O     0.000000     0.000000     0.000000   1244      2       9       0       0
   C     0.000000     0.000000     1.440000   1118      1       3      10      11
   C     1.427423     0.000000     1.962363   1121      2       4       6      12
   O     1.924453    -1.372676     1.980293   1123      3       5       0       0
   C     2.866758    -1.556753     0.934073   1124      4       7      13      18
   C     2.435730     0.816736     1.151710   1126      3       7       8      14
   C     2.832568    -0.159062     0.042099   1128      5       6      15      16
   O     3.554295     1.211441     1.932365   1249      6      17       0       0
   H    -0.918053     0.000000    -0.280677   1245      1       0       0       0
   H    -0.520597    -0.885828     1.803849   1119      2       0       0       0
   H    -0.520597     0.885828     1.803849   1120      2       0       0       0
   H     1.435560     0.337148     2.998879   1122      3       0       0       0
   H     3.838325    -1.808062     1.359516   1125      5       0       0       0
   H     1.936098     1.681209     0.714498   1127      6       0       0       0
   H     2.031585    -0.217259    -0.694882   1129      7       0       0       0
   H     3.838626     0.075227    -0.305832   1130      7       0       0       0
   H     4.214443     1.727289     1.463640   1250      8       0       0       0
   N     2.474231    -2.760890     0.168322   1132      5      19      27       0
   C     1.538394    -2.869204    -0.826353   1136     18      20      28       0
   N     1.421481    -4.070993    -1.308051   1135     19      21       0       0
   C     2.344666    -4.815233    -0.582836   1134     20      22      27       0
   C     2.704630    -6.167666    -0.619591   1140     21      23      24       0
   N     2.152150    -7.057611    -1.455273   1142     22      29      30       0
   N     3.660941    -6.579606     0.239638   1139     22      25       0       0
   C     4.205243    -5.691308     1.066416   1138     24      26      31       0
   N     3.949915    -4.402308     1.191662   1137     25      27       0       0
   C     2.991769    -4.014545     0.323275   1133     18      21      26       0
   H     0.951862    -2.033257    -1.177884   1145     19       0       0       0
   H     2.449361    -8.012246    -1.436882   1143     23       0       0       0
   H     1.442640    -6.767115    -2.097307   1144     23       0       0       0
   H     4.963977    -6.079842     1.729564   1141     25       0       0       0
$end

$rem
   METHOD             cis
   BASIS              6-31G*
   QM_MM_INTERFACE    janus
   USER_CONNECT       true
   FORCE_FIELD        amber99
   GAUSSIAN_BLUR      true
   GAUSS_BLUR_WIDTH   15000
   CIS_N_ROOTS        5
   CIS_TRIPLETS       false
   MOLDEN_FORMAT      true
   PRINT_ORBITALS     true
$end

$qm_atoms
   18 19 20 21 22 23 24 25 26 27 28 29 30 31
$end
```

Features of this job:

- An MM-only calculation. BASIS and EXCHANGE need to be defined, in order to prevent a crash, but no electronic structure calculation is actually performed.

- All atom types and MM interactions are defined in *$force_field_params* using the CHARMM27 force field. Atomic charges, equilibrium bond distances, and equilibrium angles have been extracted from a HF/6-31G* calculation, but the force constants and van der Waals parameters are fictitious values invented for this example.
- Molecular dynamics is propagated for 10 steps within a microcanonical ensemble (NVE), which is the only ensemble available at present. Initial velocities are sampled from a Boltzmann distribution at 400 K.

Example 11.26 MM molecular dynamics with user-defined MM parameters.

```
$molecule
  -2 1
   C    0.803090    0.000000    0.000000  -1  2  3  6  0
   C   -0.803090    0.000000    0.000000  -1  1  4  5  0
   H    1.386121    0.930755    0.000000  -2  1  0  0  0
   H   -1.386121   -0.930755    0.000000  -2  2  0  0  0
   H   -1.386121    0.930755    0.000000  -2  2  0  0  0
   H    1.386121   -0.930755    0.000000  -2  1  0  0  0
$end

$rem
   METHOD            hf
   BASIS             sto-3g
   QM_MM_INTERFACE   MM
   FORCE_FIELD       charmm27
   USER_CONNECT      true
   JOBTYPE           aimd
   TIME_STEP         42
   AIMD_STEPS        10
   AIMD_INIT_VELOC   thermal
   AIMD_TEMP         400
$end

$force_field_params
   NumAtomTypes 2
   AtomType -1  -0.687157  2.0000  0.1100
   AtomType -2  -0.156422  1.3200  0.0220
   Bond     -1  -1  250.00  1.606180
   Bond     -1  -2  300.00  1.098286
   Angle    -2  -1  -2  50.00  115.870
   Angle    -2  -1  -1  80.00  122.065
   Torsion  -2  -1  -1  -2  2.500  180.0  2
$end
```

Further examples of QM/MM calculations can be found in the `$QC/samples` directory, including a QM/MM/PCM example, `pcm_qmmm_crambin.in`. This calculation consists of a protein molecule (crambin) described using a force field, but with one tyrosine side chain described using electronic structure theory. The entire QM/MM system is placed within an implicit solvent model, of the sort described in Section 11.2.3.

11.4 Q-CHEM/CHARMM Interface

Q-CHEM can be used a QM back-end for QM/MM calculations using the CHARMM package.[68] In this case, both software packages are required to perform the calculations, but all the code required for communication between the programs is incorporated in the released versions. Stand-alone QM/MM calculations are described in Section 11.3.

QM/MM jobs that use the CHARMM interface are controlled using the following *$rem* keywords:

QM_MM

Turns on the Q-CHEM/CHARMM interface.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do QM/MM calculation through the Q-CHEM/CHARMM interface.

FALSE Turn this feature off.

RECOMMENDATION:

Use the default unless running calculations with CHARMM.

QMMM_PRINT

Controls the amount of output printed from a QM/MM job.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Limit molecule, point charge, and analysis printing.

FALSE Normal printing.

RECOMMENDATION:

Use the default unless running calculations with CHARMM.

QMMM_CHARGES

Controls the printing of QM charges to file.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Writes a charges.dat file with the Mulliken charges from the QM region.

FALSE No file written.

RECOMMENDATION:

Use the default unless running calculations with CHARMM where charges on the QM region need to be saved.

ESP_EFIELD

Triggers the calculation of the electrostatic potential (ESP) and/or the electric field at the positions of the MM charges.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Computes ESP only.

1 Computes ESP and electric field.

2 Computes electric field only.

RECOMMENDATION:

None.

GEOM_PRINT

Controls the amount of geometric information printed at each step.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Prints out all geometric information; bond distances, angles, torsions.

FALSE Normal printing of distance matrix.

RECOMMENDATION:

Use if you want to be able to quickly examine geometric parameters at the beginning and end of optimizations. Only prints in the beginning of single point energy calculations.

QMMM_FULL_HESSIAN

Trigger the evaluation of the full QM/MM Hessian.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Evaluates full Hessian.

FALSE Hessian for QM-QM block only.

RECOMMENDATION:

None

LINK_ATOM_PROJECTION

Controls whether to perform a link-atom projection

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Performs the projection

FALSE No projection

RECOMMENDATION:

Necessary in a full QM/MM Hessian evaluation on a system with link atoms

HESS_AND_GRAD

Enables the evaluation of both analytical gradient and Hessian in a single job

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Evaluates both gradient and Hessian.

FALSE Evaluates Hessian only.

RECOMMENDATION:

Use only in a frequency (and thus Hessian) evaluation.

GAUSSIAN_BLUR

Enables the use of Gaussian-delocalized external charges in a QM/MM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Delocalizes external charges with Gaussian functions.

FALSE Point charges

RECOMMENDATION:

None

SKIP_CHARGE_SELF_INTERACT

Ignores the electrostatic interactions among external charges in a QM/MM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE No electrostatic interactions among external charges.

FALSE Computes the electrostatic interactions among external charges.

RECOMMENDATION:

None

Example 11.27 Do a basic QM/MM optimization of the water dimer. You need CHARMM to do this but this is the Q-CHEM file that is needed to test the QM/MM functionality. These are the bare necessities for a Q-CHEM/CHARMM QM/MM calculation.

```
$molecule
   0 1
   O    -0.91126   1.09227   1.02007
   H    -1.75684   1.51867   1.28260
   H    -0.55929   1.74495   0.36940
$end

$rem
   METHOD        hf        ! HF Exchange
   BASIS         cc-pvdz   ! Correlation Consistent Basis
   QM_MM         true      ! Turn on QM/MM calculation
   JOBTYPE       force     ! Need this for QM/MM optimizations
$end

$external_charges
   1.20426      -0.64330   0.79922  -0.83400
   1.01723      -1.36906   1.39217   0.41700
   0.43830      -0.06644   0.91277   0.41700
$end
```

The Q-CHEM/CHARMM interface is unique in that:

- The external point charges can be replaced with Gaussian-delocalized charges with a finite width.[45] This is an empirical way to include the delocalized character of the electron density of atoms in the MM region. This can be important for the electrostatic interaction of the QM region with nearby atoms in the MM region.

- We allow the evaluation of the full QM/MM Hessian.[174] When link atoms are inserted to saturate the QM region, all Hessian elements associated with link atoms are automatically projected onto their QM and MM host atoms.
- For systems with a large number of MM atoms, one can define blocks consisting of multiple MM atoms (*i.e.*, mobile blocks) and efficiently evaluate the corresponding mobile-block Hessian (MBH) for normal mode analysis.

11.5 Effective Fragment Potential Method

11.5.1 Introduction

The Effective Fragment Potential (EFP) method is a computationally inexpensive way of modeling intermolecular interactions in non-covalent systems. The EFP approach can be viewed as a polarizable QM/MM scheme with no empirical parameters. EFP was originally developed by Prof. Mark Gordon's group[46,55] and implemented in GAMESS.[142] A review of the EFP theory and applications can be found in Refs. 53 and 56.

A new implementation of the EFP method based on the LIBEFP library by Dr. Ilya Kaliman (see https://libefp.github.io) has been added to Q-CHEM.[75,76] The new EFP module is called EFPMAN2. EFPMAN2 can run calculations in parallel on shared memory multi-core computers and clusters of computers. EFPMAN2 is interfaced with the CCMAN and CCMAN2 modules to allow coupled cluster and EOM-CC calculations with EFP and with ADCMAN module which allows ADC/EFP calculations. CIS and TDDFT calculations with EFP are also available.

11.5.2 Theoretical Background

The total energy of the system consists of the interaction energy of the effective fragments ($E^{\text{ef-ef}}$) and the energy of the *ab initio* (*i.e.*, QM) region in the field of the fragments. The former includes electrostatics, polarization, dispersion and exchange-repulsion contributions (the charge transfer term, which might be important for description of the ionic and highly polar species, is omitted in the current implementation):

$$E^{\text{ef-ef}} = E_{\text{elec}} + E_{\text{pol}} + E_{\text{disp}} + E_{\text{ex-rep}} \; . \tag{11.61}$$

The QM-EF interactions are computed as follows. The electrostatics (Coulomb) and polarization parts of the EFP potential contribute to the quantum Hamiltonian via one-electron terms,

$$H'_{pq} = H_{pq} + \langle p|\hat{V}^{\text{Coul}} + \hat{V}^{\text{pol}}|q\rangle \tag{11.62}$$

whereas dispersion and exchange-repulsion QM-EF interactions are treated as additive corrections to the total energy.

The electrostatic component of the EFP energy accounts for Coulomb interactions. In molecular systems with hydrogen bonds or polar molecules, this is the leading contribution to the total intermolecular interaction energy.[17] An accurate representation of the electrostatic potential is achieved by using multipole expansion (obtained from the Stone's distributed multipole analysis) around atomic centers and bond midpoints (*i.e.*, the points with high electronic density) and truncating this expansion at octupoles.[46,55,155,156] The fragment-fragment electrostatic interactions consist of charge–charge, charge–dipole, charge–quadrupole, charge–octupole, dipole–dipole, dipole–quadrupole, and quadrupole–quadrupole terms, as well as terms describing interactions of electronic multipoles with the nuclei and nuclear repulsion energy.

The multipole representation of the electrostatic density of a fragment breaks down when the fragments are too close. The multipole interactions become too repulsive due to significant overlap of the electronic densities and the charge-penetration effect. The magnitude of the charge-penetration effect is usually around 15% of the total electrostatic

energy in polar systems, however, it can be as large as 200% in systems with weak electrostatic interactions.[151] To account for the charge-penetration effect, the simple exponential damping of the charge-charge term is used.[52,151] The charge-charge screened energy between the expansion points k and l is given by the following expression, where α_k and α_l are the damping parameters associated with the corresponding expansion points:

$$E_{kl}^{\text{ch-ch}} = \begin{cases} \left[1-(1+\alpha_k R_{kl}/2)e^{-\alpha_k R_{kl}}\right]\frac{q^k q^l}{R_{kl}} & \text{if } \alpha_k = \alpha_l \\ \left[1-\left(\frac{\alpha_l^2}{\alpha_l^2-\alpha_k^2}\right)e^{-\alpha_k R_{kl}}-\left(\frac{\alpha_k^2}{\alpha_k^2-\alpha_l^2}\right)e^{-\alpha_l R_{kl}}\right]\frac{q^k q^l}{R_{kl}} & \text{if } \alpha_k \neq \alpha_l \end{cases} . \tag{11.63}$$

Alternatively, one can obtain the short-range charge-penetration energy using the spherical Gaussian overlap (SGO) approximation:[152]

$$E_{kl}^{\text{pen}} = -2\left(\frac{1}{-2ln|S_{kl}|}\right)^{1/2}\frac{S_{kl}^2}{R_{kl}} \tag{11.64}$$

where S_{kl} is the overlap integral between localized MOs k and l, calculated for the exchange-repulsion term, Eq. (11.76). This charge-penetration energy is calculated and printed separately from the rest of the electrostatic energy. Using overlap-based damping generally results in a more balanced description of intermolecular interactions and is recommended, except possibly where short-range repulsive interactions are of interest.[25]

Electrostatic interaction between an effective fragment and the QM part is described by perturbation $\hat{V}^{\text{Coul}}$ of the *ab initio* Hamiltonian [see Eq. (11.62)]. The perturbation enters the one-electron part of the Hamiltonian as a sum of contributions from the expansion points of the effective fragments. Contribution from each expansion point consists of four terms originating from the electrostatic potential of the corresponding multipole (charge, dipole, quadrupole, and octupole):

$$V_k^{\text{Coul}}(x) = q_k T(r_{kx}) - \sum_a^{x,y,z}\mu_a^k T_a(r_{kx}) + \frac{1}{3}\sum_{ab}^{x,y,z}\Theta_{ab}^k T_{ab}(r_{kx}) - \frac{1}{15}\sum_{abc}^{x,y,z}\Omega_{abc}^k T_{abc}(r_{kx}) , \tag{11.65}$$

where q, μ, Θ and Ω are the net charge, dipole, quadrupole and octupole, respectively, located at multipole expansion points k (all atoms and bond midpoints). T are the electrostatic tensors of ranks zero to three. Interaction of the QM electronic density with multipole charges can be also augmented by gaussian-type damping function,[59,138] such that Eq. (11.65) becomes

$$\begin{aligned} V_k^{\text{Coul}}(x) = \left[q_k^{\text{nuc}} + q_k^{\text{ele}}(1-\exp\left(-\beta_k r_{kx}^2\right)\right]T(r_{kx}) - \sum_a^{x,y,z}\mu_a^k\, T_a(r_{kx}) + \frac{1}{3}\sum_{ab}^{x,y,z}\Theta_{ab}^k\, T_{ab}(r_{kx}) \\ - \frac{1}{15}\sum_{abc}^{x,y,z}\Omega_{abc}^k\, T_{abc}(r_{kx}) , \end{aligned} \tag{11.66}$$

where q_k^{nuc} is the nuclear charge and q_k^{ele} is the electronic charge on multipole point k, respectively. Thus, only the electronic charges are damped (smeared) by gaussians. Damping parameters α and β in Eqs. (11.63) and (11.66) are determined by minimizing the difference between the electrostatic potentials from the damped multipole expansion and the electronic wave function in the parameter-generating calculation for each fragment. (See Section 11.5.7.)

Polarization accounts for the intramolecular charge redistribution in response to external electric field. It is the major component of many-body interactions responsible for cooperative molecular behavior. EFP employs distributed polarizabilities placed at the centers of valence LMOs. Unlike the isotropic total molecular polarizability tensor, the distributed polarizability tensors are anisotropic.

The polarization energy of a system consisting of an *ab initio* and effective fragment regions is[46]

$$E^{\text{pol}} = -\frac{1}{2}\sum_k\sum_a^{x,y,z}\mu_a^k\left(F_a^{\text{mult},k}+F_a^{\text{ai,nuc},k}\right)+\frac{1}{2}\sum_k\sum_a^{x,y,z}\bar{\mu}_a^k\, F_a^{\text{ai,elec},k} \tag{11.67}$$

where μ^k and $\bar{\mu}^k$ are the induced dipole and the conjugated induced dipole at the distributed point k; $F^{\text{mult},k}$ is the external field due to static multipoles and nuclei of other fragments, and $F^{\text{ai,elec},k}$ and $F^{\text{ai,nuc},k}$ are the fields due to the electronic density and nuclei of the *ab initio* part, respectively.

The induced dipoles at each polarizability point k are computed as

$$\mu^k = \alpha^k F^{\text{total},k} \tag{11.68}$$

where α^k is the distributed polarizability tensor at k. The total field $F^{\text{total},k}$ comprises from the static field and the field due to other induced dipoles, F_k^{ind}, as well as the field due to nuclei and electronic density of the *ab initio* region:

$$F^{\text{ai,total},k} = F^{\text{mult},k} + F^{\text{ind},k} + F^{\text{ai,elec},k} + F^{\text{ai,nuc},k} \ . \tag{11.69}$$

As follows from the above equation, the induced dipoles on a particular fragment depend on the values of the induced dipoles of all other fragments. Moreover, the induced dipoles on the effective fragments depend on the *ab initio* electronic density, which, in turn, is affected by the field created by these induced dipoles through a one electron contribution to the Hamiltonian:

$$\hat{V}_k^{\text{pol}}(x) = \frac{1}{2} \sum_a^{x,y,z} \left(\mu_a^k + \bar{\mu}_a^k\right) T_a(r_{kx}) \tag{11.70}$$

where k are the polarizability expansion points. The total polarization contribution is computed self-consistently using a two level iterative procedure. The objectives of the higher and lower levels are to converge the wave function and induced dipoles for a given fixed wave function, respectively. In the absence of the *ab initio* region, the induced dipoles of the EF system are iterated until self-consistency with each other.

Self-consistent treatment of polarization accounts for many-body interaction effects. Polarization energy between EFP fragments is augmented by gaussian-like damping functions with default parameter $\alpha = \beta = 0.6$, applied to electric field F:[152]

$$F = F_0 f^{\text{damp}} \tag{11.71}$$

$$f^{\text{damp}} = 1.0 - \left(1 + \sqrt{\alpha\beta} r^2\right) e^{-(\alpha\beta)^{1/2} r^2} \tag{11.72}$$

Dispersion provides a leading contribution to van der Waals and π-stacking interactions,[24,57] and its leading-order contribution is

$$E^{\text{disp}} = \sum_n \frac{C_6}{R^6} \tag{11.73}$$

where coefficients C_6 are derived from the frequency-dependent distributed polarizabilities with expansion points located at the LMO centroids, *i.e.*, at the same centers as the polarization expansion points. The higher-order dispersion terms (induced dipole-induced quadrupole, induced quadrupole/induced quadrupole, *etc.*) are approximated as $1/3$ of the C_6 term.[2]

For small distances between effective fragments, dispersion interactions are corrected for charge penetration and electronic density overlap effect either with the Tang-Toennies damping formula[157] with parameter $b = 1.5$,

$$C_6^{kl} \to \left(1 - e^{-bR} \sum_{k=0}^{6} \frac{(bR)^k}{k!}\right) C_6^{kl} \ , \tag{11.74}$$

or else using interfragment overlap (so-called overlap-based damping):[152]

$$C_6^{kl} \to \left(1 - S_{kl}^2 \left(1 - 2\log|S_{kl}| + 2\log^2|S_{kl}|\right)\right) C_6^{kl} \ . \tag{11.75}$$

Note: QM/EFP dispersion interactions are currently disabled.

Exchange-repulsion originates from the Pauli exclusion principle, which states that the wave function of two identical fermions must be anti-symmetric. In traditional classical force fields, exchange-repulsion is introduced as a positive (repulsive) term, *e.g.*, R^{-12} in the Lennard-Jones potential. In contrast, EFP uses a wave function-based formalism to account for this inherently quantum-mechanical effect.

The exchange-repulsion interaction is derived as an expansion in the intermolecular overlap, truncated at the quadratic term,[72,73] which requires that each effective fragment carries a basis set that is used to calculate overlap and kinetic one-electron integrals for each interacting pair of fragments. The exchange-repulsion contribution from each pair of localized orbitals i and j belonging to fragments A and B, respectively, is:

$$\begin{aligned} E_{ij}^{\text{exch}} = & -4\sqrt{\frac{-2\ln|S_{ij}|}{\pi}}\frac{S_{ij}^2}{R_{ij}} \\ & - 2S_{ij}\left(\sum_{k\in A} F_{ik}^{A} S_{kj} + \sum_{l\in B} F_{jl}^{B} S_{il} - 2T_{ij}\right) \\ & + 2S_{ij}^2\left(\sum_{J\in B} -Z_J R_{iJ}^{-1} + 2\sum_{l\in B} R_{il}^{-1} + \sum_{I\in A} -Z_I R_{Ij}^{-1} + 2\sum_{k\in A} R_{kj}^{-1} - R_{ij}^{-1}\right) \end{aligned} \tag{11.76}$$

where i, j, k and l are the LMOs, I and J are the nuclei, S and T are the intermolecular overlap and kinetic energy integrals, and F is the Fock matrix element.

The expression for the E_{ij}^{exch} involves overlap and kinetic energy integrals between pairs of localized orbitals. In addition, since Eq. (11.76) is derived within an infinite basis set approximation, it requires a reasonably large basis set to be accurate. [The 6-311++G(3df,2p) basis set is recommended and 6-31+G is considered to be the smallest acceptable basis set.] These factors make exchange-repulsion the most computationally expensive part of an EFP energy calculation for systems of moderate size.

Large systems require additional considerations. Since total exchange-repulsion energy is given by a sum of terms in Eq. (11.76) over all the fragment pairs, its computational cost formally scales as $\mathcal{O}(N^2)$ with the number of effective fragments N. However, exchange-repulsion is a short-range interaction; the overlap and kinetic energy integrals decay exponentially with the inter-fragment distance. Therefore, by employing a distance-based screening, the number of overlap and kinetic energy integrals scales as $\mathcal{O}(N)$. Consequently, for large systems exchange-repulsion may become less computationally expensive than the long-range components of EFP (such as Coulomb interactions).

Note: The QM/EFP exchange-repulsion energy is currently disabled.

11.5.3 Excited-State Calculations with EFP

Interface of EFP with EOM-CCSD (both via CCMAN and CCMAN2), CIS, CIS(D), TDDFT and ADC has been developed.[84,143,149]

In the excited state calculations, the induced dipoles of the fragments are frozen at their ground state (HF or DFT) values. The resulting excitation energies account for a zero-order response of the polarizable environment. Additionally, perturbative state-specific polarization corrections are computed according to

$$\Delta E_{\text{pol}} = \frac{1}{2}\sum_{k}\sum_{a}^{x,y,z}\left[-(\mu_{\text{ex},a}^{k}-\mu_{\text{gr},a}^{k})(F_{a}^{\text{mult},k}+F_{a}^{\text{nuc},k})+(\tilde{\mu}_{\text{ex},a}^{k}F_{\text{ex},a}^{\text{ai},k}-\tilde{\mu}_{\text{gr},a}^{k}F_{\text{gr},a}^{\text{ai},k})-(\mu_{\text{ex},a}^{k}-\mu_{\text{gr},a}^{k}+\tilde{\mu}_{\text{ex},a}^{k}-\tilde{\mu}_{\text{gr},a}^{k})F_{\text{ex},a}^{\text{ai},k}\right] \tag{11.77}$$

where $F_{\text{gr}}^{\text{ai}}$ and $F_{\text{ex}}^{\text{ai}}$ are the fields due to the reference state and excited-state electronic densities, respectively. μ_{gr}^{k} and $\tilde{\mu}_{\text{gr}}^{k}$ are the induced dipole and conjugated induced dipole at the distributed polarizability point k consistent with the reference-state density, while μ_{ex}^{k} and $\tilde{\mu}_{\text{ex}}^{k}$ are the induced dipoles corresponding to the excited state density.

The first two terms in Eq. (11.77) provide a difference of the polarization energy of the QM/EFP system in the excited and ground electronic states; the last term is the leading correction to the interaction of the ground-state-optimized induced dipoles with the wave function of the excited state. Thus, the excited states have both direct and indirect polarization contributions. The indirect term comes from the orbital relaxation of the solute in the field due to induced dipoles of the solvent. The direct term given by Eq. (11.77) is the response of the polarizable environment to the change

in solute's electronic density upon excitation. Note that the direct polarization contribution can be very large (tenths of eV) in EOM-IP/EFP since the electronic densities of the neutral and the ionized species are very different.

An important advantage of the perturbative QM/EFP scheme is that it does not compromise multi-state nature of single-referenced excited state calculations and that the electronic wave functions of the target states remain orthogonal to each other since they are obtained with the same (reference-state) field of the polarizable environment. For example, transition properties between these states can be calculated.

In the EOM-CCSD/EFP calculations, the reference-state CCSD equations for the T cluster amplitudes are solved with the HF Hamiltonian modified by the electrostatic and polarization contributions due to the effective fragments, Eq. (11.62). In the coupled-cluster calculation, the induced dipoles of the fragments are frozen at their HF values.

EOM-CC/EFP scheme works with any type of the EOM excitation operator $\hat{R}_k$ currently supported in Q-CHEM, *i.e.*, spin-flipping (SF), excitation energies (EE), ionization potential (IP), electron affinity (EA) (see Section 7.10.17 for details). However, direct polarization correction requires calculation of one-electron density of the excited state, and will be computed only for the methods with implemented one-electron properties.

11.5.4 Pairwise Fragment Energy Decomposition

Decomposition of the interaction energy of the QM and EFP regions in the energy components and in the contributions of individual solvent molecules is available for the ground and excited states.[150] The ground state QM/EFP energy is decomposed as:

$$\begin{aligned} E_{\text{QM–EF, gr}} &= E^{(1)}_{\text{elec}} + E^{(0)}_{\text{pol-solute}} + E^{(1)}_{\text{pol-solute}} + E_{\text{pol-frag}} + E^{\text{disp}}_{\text{QM–EFP}} + E^{\text{ex-rep}}_{\text{QM–EF}} \\ &= \langle \Psi^0_{\text{gr}} | \hat{V}^{\text{Coul}} | \Psi^0_{\text{gr}} \rangle + \left[\langle \Psi^{\text{sol}}_{\text{gr}} | \hat{H}^{\text{QM}} | \Psi^{\text{sol}}_{\text{gr}} \rangle - \langle \Psi^0_{\text{gr}} | \hat{H}^{\text{QM}} | \Psi^0_{\text{gr}} \rangle \right] \\ &\quad + \left[\langle \Psi^{\text{sol}}_{\text{gr}} | \hat{V}^{\text{Coul}} | \Psi^{\text{sol}}_{\text{gr}} \rangle - \langle \Psi^0_{\text{gr}} | \hat{V}^{\text{Coul}} | \Psi^0_{\text{gr}} \rangle \right] + \left[E^{\text{pol}}_{\text{QM–EF, gr}} + \langle \Psi^{\text{sol}}_{\text{gr}} | \hat{V}^{\text{pol}} | \Psi^{\text{sol}}_{\text{gr}} \rangle \right] \\ &\quad + E^{\text{disp}}_{\text{QM-EF}} + E^{\text{ex-rep}}_{\text{QM–EF}} \end{aligned} \tag{11.78}$$

where the terms (from left to right) mean the first-order electrostatic energy, solute polarization energy of the zero- and first orders, solvent polarization energy, and additive dispersion and exchange-repulsion terms. Superscripts "sol" and "0" denote QM wavefunction optimized in a solvent and gas phase, respectively. Each of the integrals involving $\hat{V}^{\text{Coul}}$ and $\hat{V}^{\text{pol}}$ operators can be decomposed into individual fragment contributions, *e.g.*,

$$E^{(1)}_{\text{elec}} = \langle \Psi^0_{\text{gr}} | \hat{V}^{\text{Coul}} | \Psi^0_{\text{gr}} \rangle = \sum_{A}^{\text{fragments}} \langle \Psi^0_{\text{gr}} | \sum_{k \in A} \hat{V}^{\text{Coul}}_k | \Psi^0_{\text{gr}} \rangle \tag{11.79}$$

and similarly for the other terms. Polarization energy can be approximately decomposed into individual fragment contributions as:

$$E^{\text{pol}}_{\text{QM–EF, gr}} = \frac{1}{2} \sum_{A}^{\text{fragments}} \sum_{p \in A} (-\mu^p F^{\text{ai,nuc,p}} + \bar{\mu}^p F^{\text{ai,elec},p}) \tag{11.80}$$

where p are polarizability expansion points. Dispersion and exchange-repulsion terms are also pairwise-additive.

The only term that cannot be similarly split into fragment contributions is the zero-order solute polarization energy:

$$E^{(0)}_{\text{pol-solute}} = \langle \Psi^{\text{sol}}_{\text{gr}} | \hat{H}^{\text{QM}} | \Psi^{\text{sol}}_{\text{gr}} \rangle - \langle \Psi^0_{\text{gr}} | \hat{H}^{\text{QM}} | \Psi^0_{\text{gr}} \rangle \, . \tag{11.81}$$

This term is referred to as "non-separable term" in the output printout. From perturbation theory, this term is expected to be about twice smaller and of the opposite sign than the first-order solute polarization term:

$$E^{(1)}_{\text{pol-solute}} = \langle \Psi^{\text{sol}}_{\text{gr}} | \hat{V}^{\text{Coul}} | \Psi^{\text{sol}}_{\text{gr}} \rangle - \langle \Psi^0_{\text{gr}} | \hat{V}^{\text{Coul}} | \Psi^0_{\text{gr}} \rangle \, . \tag{11.82}$$

Table 11.9: Notation for energy decomposition terms

EFP_ORDER = 1	
(0) `ELEC ENERGY <Psi_0|Vcoul|Psi_0>`	$\langle\Psi^0_{\rm gr/ex}\|\hat{V}^{{\rm Coul},A}\|\Psi^0_{\rm gr/ex}\rangle$
`TOTAL QM-EFP ELECTROSTATIC ENERGY`	$E^{\rm elec(1)}_{\rm gr/ex} = \sum_A^{\rm fragments} \mathbf{(0)}_A$
`NON-SEPARABLE TERM <Psi_0|H0|Psi_0>`	$\langle\Psi^0_{\rm gr/ex}\|\hat{H}_{\rm QM}\|\Psi^0_{\rm gr/ex}\rangle$
EFP_ORDER = 2	
(1) `ELEC + SOLUTE POL ENERGY <Psi_sol|Vcoul|Psi_sol>`	$\langle\Psi^{\rm sol}_{\rm gr/ex}\|\hat{V}^{{\rm Coul},A}\|\Psi^{\rm sol}_{\rm gr/ex}\rangle$
(2) `SOLVENT POL ENERGY Epol`	$\frac{1}{2}\sum_{p\in A}(-\mu^p_{\rm gr}F^{\rm ai,nuc,p} + \bar{\mu}^p_{\rm gr}F^{\rm ai,elec}_{\rm gr},p)$
(3) `SOLVENT POL ENERGY <Psi_sol|Vpol|Psi_sol>`	$\langle\Psi^{\rm sol}_{\rm gr/ex}\|\hat{V}^{{\rm pol},A_{\rm gr}}\|\Psi^{\rm sol}_{\rm gr/ex}\rangle$
(4) `SOLVENT POL ENERGY Epol_corr`	for excited states only, see Eq. (11.88)
(5) `SOLVENT POL ENERGY TOTAL`	**(2)+(3)**
(6) `PAIRWISE TOTAL ENERGY`	**(1) + (2) + (3)**
`QM-EFP TOTAL ENERGY`	$\sum_A^{\rm fragments} \mathbf{(6)}_A$
`NON-SEPARABLE TERM <Psi_sol|H0|Psi_sol>`	$\langle\Psi^{\rm sol}_{\rm gr/ex}\|\hat{H}_{\rm QM}\|\Psi^{\rm sol}_{\rm gr/ex}\rangle$

Application of the energy decomposition analysis to the electronically excited states is described below. The zero-order total solvatochromic shift can be represented as:

$$E^{\rm QM/EFP}_{\rm solv} = \sum_A^{\rm fragments} \left(\Delta E^{\rm elec(1),A}_{\rm ex/gr} + \Delta E^{\rm pol\text{-}solute(1),A}_{\rm ex/gr} + \Delta E^{\rm pol\text{-}frag(1),A}_{\rm ex/gr}\right) + \Delta E^{\rm pol\text{-}solute(0),A}_{\rm ex/gr} \,. \tag{11.83}$$

The various terms are defined as

$$\Delta E^{\rm elec(1),A}_{\rm ex/gr} = \sum_{k\in A}\left(\langle\Psi^0_{\rm ex}|\hat{V}^{\rm Coul}_k|\Psi^0_{\rm ex}\rangle - \langle\Psi^0_{\rm gr}|\hat{V}^{\rm Coul}_k|\Psi^0_{\rm gr}\rangle\right) \tag{11.84}$$

$$\Delta E^{\rm pol\text{-}solute(1),A}_{\rm ex/gr} = \sum_{k\in A}\left(\langle\Psi^{\rm sol}_{\rm ex}|\hat{V}^{\rm Coul}_k|\Psi^{\rm sol}_{\rm ex}\rangle - \langle\Psi^0_{\rm ex}|\hat{V}^{\rm Coul}_k|\Psi^0_{\rm ex}\rangle - \langle\Psi^{\rm sol}_{\rm gr}|\hat{V}^{\rm Coul}_k|\Psi^{\rm sol}_{\rm gr}\rangle + \langle\Psi^0_{\rm gr}|\hat{V}^{\rm Coul}_k|\Psi^0_{\rm gr}\rangle\right) \tag{11.85}$$

$$\Delta E^{\rm pol\text{-}frag(1),A}_{\rm ex/gr} = \sum_{p\in A}\left(\langle\Psi^{\rm sol}_{\rm ex}|\hat{V}^{\rm pol}_{p,\rm gr}|\Psi^{\rm sol}_{\rm ex}\rangle - \langle\Psi^{\rm sol}_{\rm gr}|\hat{V}^{\rm pol}_{p,\rm gr}|\Psi^{\rm sol}_{\rm gr}\rangle\right) \tag{11.86}$$

$$\Delta E^{\rm pol\text{-}solute(0),A}_{\rm ex/gr} = \langle\Psi^{\rm sol}_{\rm ex}|\hat{H}_{\rm QM}|\Psi^{\rm sol}_{\rm ex}\rangle - \langle\Psi^0_{\rm ex}|\hat{H}_{\rm QM}|\Psi^0_{\rm ex}\rangle - \langle\Psi^{\rm sol}_{\rm gr}|\hat{H}_{\rm QM}|\Psi^{\rm sol}_{\rm gr}\rangle + \langle\Psi^0_{\rm gr}|\hat{H}_{\rm QM}|\Psi^0_{\rm gr}\rangle \,. \tag{11.87}$$

Fragment contribution of the perturbative polarization correction to the excited states [Eq. (11.77)] can be obtained as follows:

$$\Delta E_{{\rm pol},A} = \frac{1}{2}\sum_{p\in A}\Big[-(\mu^p_{\rm ex} - \mu^p_{\rm gr})(F^{{\rm mult},p} + F^{{\rm nuc},p}) + (\tilde{\mu}^p_{\rm ex}F^{{\rm ai},p}_{\rm ex} - \tilde{\mu}^p_{\rm gr}F^{{\rm ai},p}_{\rm gr}) - (\mu^p_{\rm ex} - \mu^p_{\rm gr} + \tilde{\mu}^p_{\rm ex} - \tilde{\mu}^p_{\rm gr})F^{{\rm ai},p}_{\rm ex}\Big] \tag{11.88}$$

where A is a fragment of interest.

The energy is decomposed separately for all computed excited states. The excited state analysis is implemented for CIS/TD-DFT and EOM-CCSD methods both in ccman and ccman2. Energy decomposition analysis is activated by keyword EFP_PAIRWISE. Both ground and excited state energy decompositions are conducted in two steps, controlled by keyword EFP_ORDER. In the first step (EFP_ORDER = 1), the first-order electrostatic energy and $\langle\Psi^0_{\rm gr}|\hat{H}^{\rm QM}|\Psi^0_{\rm gr}\rangle$ (or $\langle\Psi^0_{\rm ex}|\hat{H}^{\rm QM}|\Psi^0_{\rm ex}\rangle$ for the excited states) part of the non-separable term are computed and printed. In the second step (EFP_ORDER = 2), the remaining terms are evaluated. Thus, for a complete analysis, the user is required to conduct two consequent simulations with EFP_ORDER set to 1 and 2, respectively. Table 11.9 shows notations used in the output to denote various terms in Eqs. (11.78)–(11.88).

11.5.5 Extension to Macromolecules: Fragmented EFP Scheme

Macromolecules such as proteins or DNA present a large number of electronic structure problems (photochemistry, redox chemistry, reactivity) that can be described within QM/EFP framework. EFP has been extended to deal with

such complex systems via the so-called fragmented EFP scheme (fEFP). The current Q-CHEM implementation allows one to (i) compute interaction energy between a ligand and a macromolecule (both represented by EFP) and (ii) to calculate the excitation energies, ionization potentials, electronic affinities of a QM moiety interacting with a fEFP macromolecule using QM/EFP scheme (see Section 11.5.3). In the present implementation, the ligand cannot be covalently bound to the macromolecule.

There are multiple ways to cut a large molecule into units depending on the position of the cut between two covalently bound residues. An obvious way to cut a protein is to cut through peptide bonds such that each fragment represents one amino acid. Alternatively, one can cut bonds between two atoms of the same nature (carbonyl and carbon-α or carbon-α and the first carbon of the side chain). The user can choose the most appropriate way to cut.

Consider a protein (P) consisting of N amino acids, $A_1A_2\ldots A_N$, and is split into N fragments (A_i). The fragments can be saturated by either "hydrogen link atom" (HLA),[148] or by mono-valent groups of atoms from the neighboring fragment(s), called "cap link atom" (CLA) hereafter. If fragments are capped using the HLA scheme, the hydrogen is located along the peptide bond axis and at the distance corresponding to the equilibrium bond length of a CH bond:

$$P = A_1H + \sum_{i=2}^{N-1} HA_iH + HA_N \tag{11.89}$$

In the CLA scheme, the cap has exactly the same geometry as the respective neighboring group. If the cuts are made through peptide bonds (one fragment is one amino acid), the caps (C^i) are either an aldehyde to saturate the -N(H) end of the fragment, or an amine to saturate the -C(=O) extremity of the fragment.

$$P = A_1C^2 + \sum_{i=2}^{N-1} C^{i-1}A_iC^{i+1} + C^{N-1}A_N \tag{11.90}$$

Q-CHEM provides a two-step script, `prefefp.pl`, located in `$QC/bin` which takes a PDB file and breaks it into capped fragments in the GAMESS format, such that the EFP parameters for these capped fragments can be generated, as explained in Section 11.5.9. As the EFP parameters are generated for each capped fragment, the neighboring fragments have duplicated parameter points (overlapping areas) in both the HLA and CLA schemes due to the overlapping caps. Since multipole expansion points and polarizability expansion points are computed on each capped residue by the standard procedure, the multipole (and damping terms) and polarizabilities need to be removed ($C^\emptyset$) from the overlapping areas.

Equations (11.89) and (11.90) become:

$$P = A_1C^\emptyset + \sum_{i=2}^{N-1} C^\emptyset A_iC^\emptyset + C^\emptyset A_N \tag{11.91}$$

The details concerning this removing procedure are presented in Section 11.5.9.

Once these duplicate parameters are removed from the EFP parameters of the capped fragments, the EFP-EFP and QM-EFP calculations can be conducted as usual.

Currently, fEFP includes electrostatic and polarization contributions, which appear in EFP(ligand)/fEFP(macromolecule) and in QM/fEFP calculations (note that the QM part is not covalently bound to the macromolecule). Consequently, the total interaction energy (E^{tot}) between a ligand (L) and a protein (P) divided into fragments is:

$$E^{\text{tot}}(P-L) = E^{\text{elec}}(P-L) + E^{\text{pol}}(P-L) \tag{11.92}$$

The electrostatics is an additive term; its contribution to fragment-fragment and ligand-fragment interaction is computed as follows:

$$E^{\text{elec}}(P-L) = \sum_{i}^{N} E^{\text{elec}}\left(C^\emptyset A_{i=1}C^\emptyset - L\right). \tag{11.93}$$

The polarization contribution in an EFP system (no QM) is:

$$E^{\text{pol}}(P-L) = -\frac{1}{2}\sum_{k\in P,L}\mu^k F^{\text{mult},k} + \frac{1}{2}\sum_{k\in P}\mu^k F^{\text{mult},k}\,. \tag{11.94}$$

The first term is the polarization energy obtained upon convergence of the induced dipoles of the ligand ($\mu^k_{\text{efp}}(L)$) and all fragments ($\mu^k_{\text{fefp}}(A_i)$). The system is thus fully polarized, all fragments (A_i or L) are polarizing each other until self-consistency:

$$\mu^k_{\text{efp}}(L) = \sum_{k\in A_i}\alpha^k(F^{\text{mult},k} + F^{\text{ind},k}) \tag{11.95}$$

$$\mu^k_{\text{fefp}}(A_i) = \sum_{j\neq i}\sum_{k\in L,A_j}\alpha^k(F^{\text{mult},k} + F^{\text{ind},k}) \tag{11.96}$$

The second term of Eq. (11.94) is the polarization of the protein by itself; this value has to be subtracted once the induced dipoles [Eq. (11.95)] converged.

The LA scheme is available to perform QM/fEFP job. In this situation the fEFP has to include a macromolecule (covalent bond between fragments). This scheme is not able yet to perform QM/fEFP/EFP in which a macromolecule and solvent molecules would be described at the EFP level of theory.

In addition to the HLA and CLA schemes, Q-CHEM also features the *molecular fragmentation with conjugated caps* (MFCC) approach,[60,180,181] which avoids the issue of overlapping of saturated fragments. The MFCC procedure consists of a summation over the interactions between a ligand and capped residues (CLA scheme) and a subtraction over the interactions of merged caps ($C^{i+1}C^{i-1}$) with the ligand.[61] (These are the so-called so-called conjugated caps or "concaps".[60]) $N-1$ concap fragments are actually used to subtract the overlapping effect.

$$P = A_1C^2 + \sum_{i=2}^{N-1} C^{i-1}A_iC^{i+1} + C^{N-1}A_N - \sum_{i}^{N-1} C^{i+1}C^{i-1} \tag{11.97}$$

In this scheme the contributions due to overlapping caps simply cancel out and the EFP parameters do not need any modifications, in contrast to the HLA or CLA procedures. However, the number of parameters that need to be generated is larger, namely, N capped fragments and $N-1$ concaps.

The MFCC electrostatic interaction energy is given as the sum of the interaction energy between each capped fragment ($C^{i-1}A_iC^{i+1}$) and the ligand minus the interaction energy between each concap ($C^{i-1}C^{i+1}$) and the ligand:

$$E^{\text{elec}}(P-L) = \sum_{i}^{N} E^{\text{elec}}\left(C^{i-1}A_iC^{i+1} - L\right) - \sum_{i}^{N-1} E^{\text{elec}}\left(C^{i-1}C^{i+1} - L\right) \tag{11.98}$$

The main advantage of MFCC is that the multipole expansion obtained on each capped residue or concap are kept during the $E^{\text{elec}}(P-L)$ calculation. In the present implementation, there are no polarization contributions. The MFCC scheme is not yet available for QM/fEFP.

11.5.6 Running EFP Jobs

The current version supports single point calculations in systems consisting of (i) *ab initio* and EFP regions (QM/MM); or (ii) EFP region only. The *ab initio* region can be described by conventional quantum methods like HF, DFT, or correlated methods including methods for the excited states [CIS, CIS(D), TDDFT, ADC, EOM-CCSD methods]. Theoretical details on the interface of EFP with EOM-CCSD and CIS(D) can be found in Refs. 149 and 84. ADC/EFP models are described in Ref. [143].

Note: EFP provides both implicit (through orbital response) and explicit (as instantaneous response of the polarizable EFP fragments) corrections to the electronic excited states. EFP-modified excitation energies are printed in the property section of the output.

Electrostatic, polarization, exchange-repulsion, and dispersion contributions are calculated between EFs; only electrostatic and polarization terms are evaluated between *ab initio* and EF regions.

The *ab initio* region is specified by regular Q-CHEM input using *$molecule* and *$rem* sections. In calculations with no QM part, the *$molecule* section should contain a dummy atom (for example, helium).

Positions of EFs are specified in the *$efp_fragments* section. Two geometry formats, controlled by EFP_COORD_XYZ keyword, are available for fragments, the Euler angle format and the XYZ format. In the Euler angle format, each line in this section contains the information on an individual fragment: fragment's name and position, specified by center-of-mass coordinates (x, y, z) and the Euler rotation angles (α, β, γ) relative to the fragment frame, *i.e.*, the coordinates of the standard fragment provided in the fragment library. In the XYZ format, the name of the fragment is provided on the first line followed by three lines specifying names and (x, y, z) coordinates of the first three atoms of the fragment.

11.5.7 Library of Fragments

The effective fragments are rigid and their potentials are generated from a set of *ab initio* calculations on each unique isolated fragment. The EFP includes: (i) multipoles (produced by the Stone's Distributed Multipolar Analysis) for Coulomb and polarization terms; (ii) static polarizability tensors centered at localized molecular orbital (LMO) centroids (obtained from coupled-perturbed Hartree-Fock calculations), which are used for calculations of polarization; (iii) dynamic polarizability tensors centered on the LMOs that are generated by time-dependent HF calculations and used for calculations of dispersion; and (iv) the Fock matrix, basis set, and localized orbitals needed for the exchange-repulsion term. Additionally, the EF potential contains coordinates of atoms, coordinates of the points of multi-polar expansion (typically, atoms and bond mid-points), coordinates of the LMO centroids, electrostatic and polarization screening parameters, and atomic labels of the EF atoms.

Q-CHEM provides a library of standard fragments with precomputed effective fragment potentials. Currently the library includes common organic solvents, nucleobases, and molecules from S22 and S66 datasets for non-covalent interactions; see Table 11.10. The fragment library is located in `$QCAUX/fraglib` directory. GAMESS format of EFPs is used.

Note: The fragments from Q-CHEM fragment library have "_L" added to their names to distinguish them from user-defined fragments.

The parameters for the standard fragments were computed as follows. The geometries of the solvent molecules were optimized with MP2/cc-pVTZ; geometries of nucleobases were optimized with RI-MP2/cc-pVTZ. Geometries of molecules from S22 and S66 datasets are discussed in Ref. 48. The EFP parameters were obtained in GAMESS. To generate the electrostatic multipoles and electrostatic screening parameters, analytic DMA procedure was used, with 6-31+G* basis for non-aromatic compounds and 6-31G* for aromatic compounds and nucleobases. The rest of the potential, *i.e.*, static and dynamic polarizability tensors, wave function, Fock matrix, *etc.*, were obtained using 6-311++G(3df,2p) basis set.

11.5.8 Calculation of User-Defined EFP Potentials

User-defined EFP parameters can be generated in "MAKEFP" job in GAMESS; see the GAMESS manual for details. The user-defined parameter (`.efp`) files should be located in the working directory; the name of the `.efp` file should match exactly the name of the fragment, for example, beginning of the `cl_ion.efp` parameter file should look like:

```
$cl_ion
Comment line
COORDINATES (BOHR)
```

Table 11.10: Standard fragments available in Q-CHEM

acetone	ACETONE_L
acetonitrile	ACETONITRILE_L
adenine	ADENINE_L
ammonia	AMMONIA_L
benzene	BENZENE_L
carbon tetrachloride	CCL4_L
cytosine C1	CYTOSINE_C1_L
cytosine C2a	CYTOSINE_C2A_L
cytosine C2b	CYTOSINE_C2B_L
cytosine C3a	CYTOSINE_C3A_L
cytosine C3b	CYTOSINE_C3B_L
dichloromethane	DCM_L
dimethyl sulfoxide	DMSO_L
guanine enol N7	GUANINE_EN7_L
guanine enol N9	GUANINE_EN9_L
guanine enol N9RN7	GUANINE_EN9RN7_L
guanine keton N7	GUANINE_KN7_L
guanine keton N9	GUANINE_KN9_L
methane	METHANE_L
methanol	METHANOL_L
phenol	PHENOL_L
thymine	THYMINE_L
toluene	TOLUENE_L
water	WATER_L
acetamide, S66, gas phase	ACETAMIDE_L
acetamide, S66, H-bonded dimer	ACETAMIDE_HB_L
acetic acid, S66, gas phase	ACETICAC_L
acetic acid, S66, H-bonded dimer	ACETICAC_HB_L
adenine, S22 stack dimer	ADENINE_L
adenine, S22 WC dimer	ADENINE_WC_L
2-aminopyridine, S22	AMINOPYRIDINE_L
cyclopentane, S66	CPENTANE_L
ethylene	ETHENE_L
acetylene	ETHYNE_L
formic acid, S22 H-bonded dimer	FORMICAC_HB_L
formamide, S22 dimer	FORMID_L
hydrogen cyanide	HCN_L
indole, S22	INDOLE_L
methylamine, S66	MENH2_L
neopentane, S66	NEOPENTANE_L
O_2	O2_L
pentane, S66	PENTANE_L
peptide, S66	PEPTIDE_L
pyrazine	PYRAZINE_L
pyridine, S66	PYRIDINE_L
2-pyridoxine, S22	PYRIDOXINE_L
thymine, S22 stack dimer	THYMINE_L
thymine, S22 WC dimer	THYMINE_WC_L
uracil, S66, gas phase	URACIL_L
uracil, S66, H-bonded dimer	URACIL_HB_L

```
 ....
```

The EFP potential generation begins by determining an accurate structure for the fragment (EFP is the frozen-geometry potential, so the fragment geometry will remain the same in all subsequent calculations). We recommend MP2/cc-pVTZ level of theory.

11.5.8.1 Generating EFP Parameters Using GAMESS

EFP parameters can be generated in GAMESS using a "MAKEFP" job (`RUNTYP = MAKEFP` in GAMESS). For EFP parameters calculations, 6-311++G(3df,2p) basis set is recommended. Originally Stone's distributed multipole analysis is recommended for non-aromatic compounds (`bigexp = 0` in the group `$stone`); optionally, one may decrease the basis set to 6-31G* or 6-31+G* for generation of electrostatic multipoles and screening parameters. (To prepare such a "mixed" potential, one has to run two separate MAKEFP calculations in larger and smaller bases, and combine the corresponding parts of the potential). In aromatic compounds, one must either use numerical grid for generation of multipoles (`bigexp = 4.0`) or use 6-31G* basis with standard analytic DMA, which is recommended. The MAKEFP job produces (usually in the scratch directory) the `.efp` file containing all the necessary EFP parameters. See the GAMESS manual for further details. Below are examples of a GAMESS input file for RUNTYP = MAKEFP, for water and for benzene.

```
GAMESS input example for water.

 $contrl units=angs local=boys runtyp=makefp coord=cart icut=11 $end
 $system timlim=99999 mwords=200 $end
 $scf soscf=.f. diis=.t. conv=1.0d-06 $end
 $basis gbasis=n311 ngauss=6 npfunc=2 ndfunc=3 nffunc=1
   diffs=.t. diffsp=.t. $end
 $stone
   bigexp=0.0
 $end
 $damp ifttyp(1)=3,2 iftfix(1)=1,1 thrsh=500.0 $end
 $dampgs
h3=h2
bo31=bo21
 $end
 $makefp chtr=.f. disp7=.f. $end
 $data
 water h2o (geometry: mp2/cc-pvtz)
 c1
o1 8.0  0.0000  0.0000  0.1187
h2 1.0  0.0000  0.7532 -0.4749
h3 1.0  0.0000 -0.7532 -0.4749
 $end

GAMESS input example for benzene.

 $contrl units=bohr local=boys runtyp=makefp coord=cart icut=11 $end
 $system timlim=99999 mwords=200 $end
 $scf soscf=.f. diis=.t. conv=1.0d-06  $end
```

```
 $basis gbasis=n311 ngauss=6 npfunc=2 ndfunc=3 nffunc=1
   diffs=.t. diffsp=.t. $end
 $stone
   bigexp=4.0
 $end
 $damp ifttyp(1)=3,2 iftfix(1)=1,1 thrsh=500.0 $end
 $dampgs
c6=c5
c2=c1
c3=c1
c4=c1
c5=c1
c6=c1
h8=h7
h9=h7
h10=h7
h11=h7
h12=h7
bo32=bo21
bo43=bo21
bo54=bo21
bo61=bo21
bo65=bo21
bo82=bo71
bo93=bo71
bo104=bo71
bo115=bo71
bo126=bo71
 $end
 $makefp chtr=.f. disp7=.f. $end
 $data
benzene c6h6 (geometry: mp2/cc-pvtz)
c1
c1     6.0    1.3168      -2.2807       0.0000
c2     6.0    2.6336       0.0000       0.0000
c3     6.0    1.3168       2.2807       0.0000
c4     6.0   -1.3168       2.2807       0.0000
c5     6.0   -2.6336      -0.0000       0.0000
c6     6.0   -1.3168      -2.2807       0.0000
h7     1.0    2.3386      -4.0506       0.0000
h8     1.0    4.6772       0.0000       0.0000
h9     1.0    2.3386       4.0506       0.0000
h10    1.0   -2.3386       4.0506       0.0000
h11    1.0   -4.6772       0.0000       0.0000
h12    1.0   -2.3386      -4.0506       0.0000
 $end
```

11.5.9 fEFP Input Structure

A two-step script, `prefefp.pl` located in `$QC/bin`, allows users to break molecular structures from a PDB file into the capped fragments in the GAMESS format, such that parameters for fEFP calculations can be generated.

To use the `prefefp.pl` scripts you need a PDB file, a MAP file, and a directory with all your `.efp` parameter files. Run the following commands to: (1) obtain the N input file generating the N EFP parameters for the N capped fragments, and (2) create the EFP input file in XYZ format.

```
perl prefp.pl 1 <PDB file> <MAP file>
perl prefp.pl 2 <PDB file> <.efp path> <MAP file> <GMS input file name>
```

At the first step the script splits the biomolecule (PDB format) into N fragments generating N GAMESS MAKEFP input files with the help of a MAP file.

At the second step the `.efp` file from GAMESS MAKEFP is analyzed and is auto-edited using the same MAP file to create the final EFP input (XYZ format).

The MAP file is required as an input for the script. It defines groups of atoms belonging to each EFP fragment both for the MAKEFP calculation and for the consequent EFP jobs. Here is a description of the MAP file: Each fragment described using section $RESIDUE followed by closing $end In this example the Lys2 is extracted cutting through the peptide bond, the cut bond is saturated with hydrogen atom. The explanation of each variable is given below.

```
$Residue
Name = lys2
PreAtoms = 14-35
NH = 14,12
CH = 34,36
PostAtoms = 14-35
Rescharge = +1
USEFP = lys2
$end
```

The four first lines are required for the first step of the script (GAMESS MAKEFP job); the next ones are necessary for the actual EFP job:

- `Name`: Residue name
- `PreAtoms`: Atoms which belongs to the residue for GAMESS MAKEFP calculation.
- `CH, NH, or OH`: In the case of broken bonds a hydrogen atom is added so that in X–Y bond (where X belongs to the Lys2 residue and Y belongs to the previous or next residue) the Y atom is replaced by H along the X–Y axis. The default equilibrium distance for the X-H bond is set to 1.08 Å for a C–H bond, to 1.00 Å for a N–H bond, and to 0.94 Å for a O–H bond. It required to specify the atom number of the X and Y atoms.
- `PostAtoms`: Atoms which belong to the residue after removing the overlapping fragment atoms or caps when the HLA or the CLA scheme is used. This important step removes multipoles and polarizability expansion points of those atoms according to the cutoff procedure (set by default to 1.3 Å and 1.2 Å for multipoles and polarizability expansion points, respectively). Multipole expansion at duplicated points are eliminated but to maintain the net integer charge on each amino acid the monopole expansion of the caps is redistributed on the natural fragment. This method is called Expand-Remove-Redistribute. Concerning the polarizability expansion

points, only one polarizability expansion point is removed when a hydrogen atom saturates the dangling bond, whereas 6 or 5 polarizability points are removed when the cap is an amine or an aldehyde, respectively.

- `ResCharge`: The net charge of the residue after removing the overlapping fragment atoms (*cf.* the LA scheme).
- `USEFP`: Name of the EFP fragment (and `.efp` file) to use with this fragment in the actual EFP calculation.

Note:

- In the MFCC scheme, the two first letters of the concap fragment have to be "CC".
- If the *PostAtoms* keyword is not present, the second script will generate an EFP job file without any modification of the parameters, which is useful for the MFCC scheme.
- Q-CHEM's implementation of EFP currently does not support dipole–quadrupole polarizabilities. These may be present in a `.efp` generated using GAMESS and should be removed if that file is to be read by Q-CHEM.

11.5.10 Input Keywords

EFP_COORD_XYZ

Use coordinates of three atoms instead of Euler angles to specify position and orientation of the fragments

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

None

EFP_DIRECT_POLARIZATION_DRIVER

Use direct solver for EFP polarization

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

Direct polarization solver provides stable convergence of induced dipoles which may otherwise become problematic in case of closely lying or highly polar or charged fragments. The computational cost of direct polarization versus iterative polarization becomes higher for systems containing more than 10000 polarizable points.

EFP_ENABLE_LINKS

Enable fragment links in EFP region

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

None

EFP

Specifies that EFP calculation is requested

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

The keyword should be present if excited state calculation is requested

EFP_FRAGMENTS_ONLY

Specifies whether there is a QM part

TYPE:

LOGICAL

DEFAULT:

FALSE QM part is present

OPTIONS:

TRUE Only MM part is present: all fragments are treated by EFP

FALSE QM part is present: do QM/MM EFP calculation

RECOMMENDATION:

None

EFP_INPUT

Specifies the format of EFP input

TYPE:

LOGICAL

DEFAULT:

FALSE Dummy atom (*e.g.*, He) in *$molecule* section should be present

OPTIONS:

TRUE A format without dummy atom in *$molecule* section

FALSE A format with dummy atom in *$molecule* section

RECOMMENDATION:

None

FEFP_EFP

Specifies that fEFP_EFP calculation is requested to compute the total interaction energies between a ligand (the last fragment in the $efp_fragments section) and the protein (represented by fEFP)

TYPE:

STRING

DEFAULT:

OFF

OPTIONS:

OFF disables fEFP

LA enables fEFP with the Link Atom (HLA or CLA) scheme (only electrostatics and polarization)

MFCC enables fEFP with MFCC (only electrostatics)

RECOMMENDATION:

The keyword should be invoked if EFP/fEFP is requested (interaction energy calculations). This keyword has to be employed with EFP_FRAGMENT_ONLY = TRUE. To switch on/off electrostatics or polarzation interactions, the usual EFP controls are employed.

FEFP_QM

Specifies that fEFP_QM calculation is requested to perform a QM/fEFPcompute computation. The fEFP part is a fractionated macromolecule.

TYPE:

STRING

DEFAULT:

OFF

OPTIONS:

OFF disables fEFP_QM and performs a QM/EFP calculation

LA enables fEFP_QM with the Link Atom scheme

RECOMMENDATION:

The keyword should be invoked if QM/fEFP is requested. This keyword has to be employed with efp_fragment_only false. Only electrostatics is available.

EFP_ELEC

Controls fragment-fragment electrostatics in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on electrostatics

FALSE switch off electrostatics

RECOMMENDATION:

None

EFP_POL

Controls fragment-fragment polarization in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on polarization

FALSE switch off polarization

RECOMMENDATION:

None

EFP_DISP

Controls fragment-fragment dispersion in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on dispersion

FALSE switch off dispersion

RECOMMENDATION:

None

EFP_EXREP

Controls fragment-fragment exchange repulsion in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on exchange repulsion

FALSE switch off exchange repulsion

RECOMMENDATION:

None

EFP_QM_ELEC

Controls QM-EFP electrostatics

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on QM-EFP electrostatics

FALSE switch off QM-EFP electrostatics

RECOMMENDATION:

None

EFP_QM_POL

Controls QM-EFP polarization

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on QM-EFP polarization

FALSE switch off QM-EFP polarization

RECOMMENDATION:

None

EFP_QM_DISP

Controls QM-EFP dispersion

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE switch on QM-EFP dispersion

FALSE switch off QM-EFP dispersion

RECOMMENDATION:

None

EFP_QM_EXREP

Controls QM-EFP exchange-repulsion

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE switch on QM-EFP exchange-repulsion

FALSE switch off QM-EFP exchange-repulsion

RECOMMENDATION:

None

EFP_ELEC_DAMP

Controls fragment-fragment electrostatic screening in EFP

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 switch off electrostatic screening

1 use overlap-based damping correction

2 use exponential damping correction if SCREEN2 screening parameters are provided in the EFP potential

RECOMMENDATION:

Overlap-based damping is recommended

EFP_DISP_DAMP

Controls fragment-fragment dispersion screening in EFP

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 switch off dispersion screening

1 use Tang-Toennies screening, with fixed parameter $b = 1.5$

2 use overlap-based damping

RECOMMENDATION:

None

EFP_POL_DAMP

Controls fragment-fragment polarization screening in EFP

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 switch off polarization screening

1 use Tang-Toennies screening

RECOMMENDATION:

None

EFP_POL_DAMP_TT_VALUE

Controls the value of the fragment–fragment polarization Tang–Toennies screening factor in EFP

TYPE:

INTEGER

DEFAULT:

600

OPTIONS:

n corresponds to n/1000.

RECOMMENDATION:

Use the default value of 0.6 for this screening factor, which was hard coded in the original EFP implementation.

EFP_QM_ELEC_DAMP

Controls QM-EFP electrostatics screening in EFP

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 switch off electrostatic screening

1 use QM-EFP electrostatic damping if SCREEN screening parameters are provided in the EFP potential

RECOMMENDATION:

None

EFP_PAIRWISE

Controls QM-EFP pairwise fragment energy decomposition analysis

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 energy decomposition is turned off

1 energy decomposition is turned on

RECOMMENDATION:

None

EFP_ORDER

Controls QM-EFP pairwise fragment energy decomposition analysis

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 the first step of energy decomposition is performed

2 the second step of energy decomposition is performed

RECOMMENDATION:

The EFP_PAIRWISE keyword should be turned on to activate the energy decomposition analysis.

11.5.11 Examples

Example 11.28 Basic EFP-only calculation of benzene dimer in XYZ input format. EFP parameters are read from the fragment library (`$QCAUX/fraglib`).

```
$comment
   Pure EFP energy computation on benzene dimer
$end

$molecule
   0 1
   He       5.0    5.0    5.0
$end

$rem
   METHOD               hf
   BASIS                6-31G(d)
   EFP_FRAGMENTS_ONLY   true
   EFP_DISP_DAMP        1
   EFP_COORD_XYZ        1
   PURECART             2222
$end

$efp_fragments
BENZENE_L
   A01C   -0.07088   -2.35729    1.06421
   A02C    0.75298   -3.00688    0.16337
   A03C    0.51391   -2.89905   -1.19436
BENZENE_L
   A01C   -1.72945    1.38131   -0.01219
   A02C   -0.47330    1.37787   -0.59037
   A03C    0.65547    1.37017    0.20840
$end
```

Example 11.29 Basic EFP-only calculation of benzene dimer in Euler angle input format with new EFPMAN2 module. EFP parameters are read from the fragment library (`$QCAUX/fraglib`).

```
$molecule
   0 1
   He       5.0    5.0    5.0
$end

$rem
   METHOD               hf
   BASIS                6-31G(d)
   EFP_FRAGMENTS_ONLY   true
   EFP_DISP_DAMP        1
   PURECART             2222
$end

$efp_fragments
   BENZENE_L -0.30448173 -2.24210052 -0.29383131 -0.642499 1.534222 -0.568147
   BENZENE_L -0.60075437  1.36443336  0.78647823  3.137879 1.557344 -2.568550
$end
```

Example 11.30 QM/MM computation of one water molecule in QM part and one water + two ammonia molecules in EFP part. EFP parameters are read from the fragment library (`$QCAUX/fraglib`).

```
$molecule
   0 1
   O   0.0000      0.0000      0.2243
   H  -1.4233      0.0000     -0.8973
   H   1.4233      0.0000     -0.8973
$end

$rem
   METHOD          hf
   BASIS           6-31G(d)
   EFP_DISP_DAMP   1
   PURECART        2222
$end

$efp_fragments
   WATER_L   -2.12417561  1.22597097 -0.95332054 -2.902133 1.734999 -1.953647
   AMMONIA_L  1.04358758  1.90477190  2.88279926 -1.105309 2.033306 -1.488582
   AMMONIA_L -4.16795656 -0.98129149 -1.27785935  2.526442 1.658262 -2.742084
$end
```

Example 11.31 EOM-IP-CCSD/EFP calculation; CN radical hydrated by 6 waters.

```
$comment
   EOM-IP/EFP; CN radical hydrated by 6 waters
   all active orbitals and frozen core are tested
$end

$molecule
   -1 1
    C   1.004122   2.504092  -0.325463
    N   0.816221   2.319773   0.780625
$end

$rem
   METHOD               eom-ccsd
   BASIS                6-31+G*
   EFP_FRAGMENTS_ONLY   false
   PURECART             2222
   SCF_CONVERGENCE      8
   IP_STATES            4
   EFP                  1
   EOM_FAKE_IPEA        true
   CCMAN2               false
   EFP_EXREP            0
$end

$efp_fragments
   WATER_L 1.12736608 -1.43556954 -0.73517708 -1.45590530  2.99520330  0.11722720
   WATER_L 1.25577919  0.62068648 -2.69876653  2.56168924  1.26470722  0.33910203
   WATER_L 3.76006184 -1.03358049  0.45980636 -1.53852111  2.58787281 -1.98107746
   WATER_L 4.81593067  2.87535152 -0.24524178 -1.86802100  0.73283467 -2.17837806
   WATER_L 4.07402278  0.74020006 -1.92695949  2.21177738  1.69303397 -2.30505848
   WATER_L 3.60104027  1.35547341  1.88776964  0.43895304  1.25442317  1.07742578
$end
```

Example 11.32 Excited states of formaldehyde with 6 EFP water molecules by CIS(D).

```
$molecule
   0 1
   C1    1.063245   2.026797   0.433887
   O2    1.115445   1.079872   1.154242
   H3    1.094466   3.039490   0.836046
   H4    0.983660   1.924177  -0.645223
$end

$rem
   METHOD               cis
   BASIS                6-31+G*
   EFP_FRAGMENTS_ONLY   false
   PURECART             2222
   UNRESTRICTED         true
   SCF_CONVERGENCE      8
   CIS_N_ROOTS          4
   EFP                  1
   MEM_STATIC           256
   AO2MO_DISK           1000
$end

$efp_fragments
   WATER_L 1.45117729 -1.31271387 -0.39790305 -1.075756  2.378141  1.029199
   WATER_L 1.38370965  0.22282733 -2.74327999  2.787663  1.446660  0.168420
   WATER_L 4.35992117 -1.31285676  0.15919381 -1.674869  2.547933 -2.254831
   WATER_L 4.06184149  2.79536141  0.05055916 -1.444143  0.750463 -2.291224
   WATER_L 4.09898096  0.83731430 -1.93049301  2.518412  1.592607 -2.199818
   WATER_L 3.96160175  0.71581837  2.05653146  0.825946  1.414384  0.966187
$end
```

Example 11.33 Fragment pairwise excitation energy decomposition analysis of the first excited state of formaldehyde solvated by two water molecules. TDDFT/EFP

```
$comment
fragment pairwise excitation energy decomposition analysis
efp_pairwise=1 initiates the analysis
efp_order=1 computes electrostatic contribution to solvatochromic shift
efp_order=2 computes solute and solvent polarization components to solvatochromic shift
results of efp_order=1 job are needed for a proper analysis of efp_order=2 results
$end

$molecule
0 1
          C       0.524512     0.000804     0.000092
          O      -0.666413     0.000723    -0.000022
          H       1.085163    -0.934036    -0.000186
          H       1.099066     0.923427    -0.000188
$end

$rem
exchange = wb97x
basis 6-31+G*
cis_n_roots 1
cis_triplets = false
jobtype sp
efp_pairwise = 1
efp_order = 1
efp_coord_xyz = 1
point_group_symmetry False
$end

$efp_fragments
water_1
A01O1  -1.815220   2.663988  -0.023113
A02H2  -1.617480   1.725461   0.001936
A03H3  -2.476026   2.736906  -0.714908
water_1
A01O1   0.990542   3.342755  -0.406837
A02H2   1.243548   4.121277   0.093615
A03H3   0.036378   3.313023  -0.310676
$end

@@@

$comment
efp_order=2 computes solute and solvent polarization components to solvatochromic shift
$end

$molecule
READ
$end

$rem
exchange = wb97x
basis 6-31+G*
jobtype sp
cis_n_roots 1
cis_triplets = false
efp_pairwise = 1
efp_order = 2
efp_coord_xyz = 1
point_group_symmetry False
$end

$efp_fragments
water_1
A01O1  -1.815220   2.663988  -0.023113
```

Example 11.34 Fragment pairwise excitation energy decomposition analysis of the first excited state of formaldehyde solvated by two water molecules. EOM-CCSD/EFP via ccman2

```
$comment
fragment pairwise excitation energy decomposition analysis
efp_pairwise=1 initiates the analysis
efp_order=1 computes electrostatic contribution to solvatochromic shift
$end

$molecule
0 1
          C       0.524512     0.000804     0.000092
          O      -0.666413     0.000723    -0.000022
          H       1.085163    -0.934036    -0.000186
          H       1.099066     0.923427    -0.000188
$end

$rem
method eom-ccsd
ee_singlets  2
ccman2 true
exchange = hf
basis 6-31+G*
jobtype sp
efp_pairwise = 1
efp_order = 1
efp_coord_xyz = 1
point_group_symmetry False
$end

$efp_fragments
water_1
A01O1  -1.815220   2.663988  -0.023113
A02H2  -1.617480   1.725461   0.001936
A03H3  -2.476026   2.736906  -0.714908
water_1
A01O1   0.990542   3.342755  -0.406837
A02H2   1.243548   4.121277   0.093615
A03H3   0.036378   3.313023  -0.310676
$end

@@@

$comment
efp_order=2 computes polarization contributions to solvatochromic shift
$end

$molecule
READ
$end

$rem
method eom-ccsd
ee_singlets  2
ccman2 true
exchange = hf
basis 6-31+G*
jobtype sp
efp_pairwise = 1
efp_order = 2
efp_coord_xyz = 1
point_group_symmetry False
$end

$efp_fragments
water_1
A01O1  -1.815220   2.663988  -0.023113
```

11.6 Projection-Based Density Embedding

11.6.1 Introduction & Theory

The formally exact density embedding method developed by Manby, Miller and coworkers[93,104] affords a step further than electrostatic embedding in the description of chemical environment. This embedding scheme allows for the partition of a system into two interacting subsystems that are treated at two different levels of QM theories, *e.g.*, coupled cluster embedded in DFT. This type of embedding fully accounts for polarization as well as quantum mechanical exchange, as calculated from the super-molecular embedding density and the exchange correlation functional used. The goal of this embedding theory is to perform a higher-level QM (DFT or WFT) calculation in an environment described by a lower-level QM theory.

11.6.1.1 Working Equations

For a system whose total density can be represented as $\boldsymbol{\gamma} = \boldsymbol{\gamma}^A + \boldsymbol{\gamma}^B$, *i.e.*, the molecular orbitals on fragments A and B are orthogonal to each other, its total energy can be expressed as

$$\begin{aligned} E_{\text{tot}}[\boldsymbol{\gamma}^A + \boldsymbol{\gamma}^B] = &\underbrace{\text{tr}(\boldsymbol{\gamma}^A \mathbf{h}) + J[\boldsymbol{\gamma}^A] + E_{\text{xc}}[\boldsymbol{\gamma}^A]}_{\text{Energy of A}} + \underbrace{\text{tr}(\boldsymbol{\gamma}^B \mathbf{h}) + J[\boldsymbol{\gamma}^B] + E_{\text{xc}}[\boldsymbol{\gamma}^B]}_{\text{Energy of B}} \\ &+ \underbrace{J[\boldsymbol{\gamma}^A, \boldsymbol{\gamma}^B] + E_{\text{xc}}[\boldsymbol{\gamma}^A, \boldsymbol{\gamma}^B]}_{\text{Non-additive terms}} \end{aligned} \tag{11.99}$$

where $\mathbf{h}$ is the core-Hamiltonian matrix, J and E_{xc} are the Coulomb and XC energies (exact exchange K will also be present if a hybrid functional is employed). In an embedding calculation where the electron density of A (denoted as $\tilde{\gamma}^A$) is optimized at a high-level theory in the presence of γ^B (which is obtained at a low-level theory and then fixed), the total energy can be expressed as

$$E_{\text{tot}}[\tilde{\gamma}^A, \gamma^B] = E^{\text{high}}[\tilde{\gamma}^A] + E^{\text{low}}[\tilde{\gamma}^A + \gamma^B] - E^{\text{low}}[\tilde{\gamma}^A]. \tag{11.100}$$

Differentiating Eq. (11.100) with respect to $\tilde{\gamma}^A$ gives the Fock matrix for the embedding calculation:

$$\mathbf{f}^{\text{A-in-B}} = \frac{\delta}{\delta \tilde{\gamma}^A} E_{\text{tot}}[\tilde{\gamma}^A, \gamma^B] = \mathbf{F}^{\text{high}}[\tilde{\gamma}^A] + \mathbf{v}_{\text{emb}}[\tilde{\gamma}^A, \gamma^B], \tag{11.101}$$

where the embedding potential

$$\begin{aligned} \mathbf{v}_{\text{emb}}[\tilde{\gamma}^A, \gamma^B] &= \frac{\partial}{\partial \tilde{\gamma}^A} \left(E^{\text{low}}[\tilde{\gamma}^A + \gamma^B] - E^{\text{low}}[\tilde{\gamma}^A] \right) \\ &= \mathbf{F}^{\text{low}}[\tilde{\gamma}^A + \gamma^B] - \mathbf{F}^{\text{low}}[\tilde{\gamma}^A] \\ &= (\mathbf{J}[\boldsymbol{\gamma}^A + \boldsymbol{\gamma}^B] - \mathbf{J}[\boldsymbol{\gamma}^A]) + (\mathbf{v}_{\text{xc}}^{\text{low}}[\boldsymbol{\gamma}^A + \boldsymbol{\gamma}^B] - \mathbf{v}_{\text{xc}}^{\text{low}}[\boldsymbol{\gamma}^A]) + \mu \mathbf{P}^B, \end{aligned} \tag{11.102}$$

where $\mathbf{P}^B = \mathbf{S}\boldsymbol{\gamma}^B\mathbf{S}$ is the projector formed by B's orbitals and μ is a large enough constant (*e.g.* 10^6 a.u.) that effectively elevates the energies of B's orbitals and thereby enforces the orthogonality between MOs on A and B. In the Q-CHEM implementation (with GEN_SCFMAN_EMBED = TRUE), an alternative approach is employed, where one diagonalizes a modified Fock matrix from which the variational degrees of freedom spanned by B's orbitals are projected out:

$$\tilde{\mathbf{f}}^{\text{A-in-B}} = (\mathbf{I} - \mathbf{S}\boldsymbol{\gamma}^B)\mathbf{f}^{\text{A-in-B}}(\mathbf{I} - \boldsymbol{\gamma}^B\mathbf{S}), \tag{11.103}$$

where $\mathbf{S}$ is the AO overlap matrix.

One disadvantage of using the energy expression given by Eq. (11.100) is that the embedding potential [Eq. (11.102)] depends on $\tilde{\gamma}^A$, which needs to be updated in every iteration of the embedding calculation. Thus this scheme affords

no savings in computational cost for DFT-in-DFT calculations. To address this limitation, a linearized approximation was suggested,[93] which is based on the following expansion (to the first order of $\tilde{\gamma}^A - \gamma^A$):

$$E^{\text{low}}[\tilde{\gamma}^A + \gamma^B] - E^{\text{low}}[\tilde{\gamma}^A] \approx E^{\text{low}}[\gamma^A + \gamma^B] - E^{\text{low}}[\gamma^A] + \text{Tr}\left((\tilde{\gamma}^A - \gamma^A)\mathbf{v}_{\text{emb}}[\gamma^A, \gamma^B]\right), \tag{11.104}$$

where

$$\mathbf{v}_{\text{emb}}[\gamma^A, \gamma^B] = \frac{\delta}{\delta\gamma^A}\left(E^{\text{low}}[\gamma^A + \gamma^B] - E^{\text{low}}[\gamma^A]\right) = \mathbf{F}^{\text{low}}[\gamma^A + \gamma^B] - \mathbf{F}[\gamma^A] \tag{11.105}$$

Based on this linearized approximation, the total energy of the entire system is approximately given by

$$E_{\text{tot}}[\tilde{\gamma}^A; \gamma^A, \gamma^B] = E^{\text{high}}[\tilde{\gamma}^A] + E^{\text{low}}[\gamma^A + \gamma^B] - E^{\text{low}}[\gamma^A] + \text{Tr}\left((\tilde{\gamma}^A - \gamma^A)\mathbf{v}_{\text{emb}}[\gamma^A, \gamma^B]\right), \tag{11.106}$$

and the Fock matrix for embedding calculation:

$$\mathbf{f}^{\text{A-in-B}} = \frac{\delta}{\delta\tilde{\gamma}^A}E_{\text{tot}}[\tilde{\gamma}^A; \gamma^A, \gamma^B] = \mathbf{F}^{\text{high}}[\tilde{\gamma}^A] + \mathbf{v}_{\text{emb}}[\gamma^A, \gamma^B] \tag{11.107}$$

where the embedding potential stays unchanged during the embedding calculation. In Q-CHEM 5.4.1 and versions after, this linearized approximation is used by default in projection-based embedding calculations.

For WFT-in-DFT calculations, one can absorb the embedding potential $\mathbf{v}_{\text{emb}}[\gamma^A, \gamma^B]$ into the Hamiltonian that is used for the correlated WFT calculation, converting Eq. (11.106) into

$$E[\Psi^A; \gamma^A, \gamma^B] = \langle\Psi^A|\hat{H}^{\text{A-in-B}}|\Psi^A\rangle + E^{\text{low}}[\gamma^A + \gamma^B] - E^{\text{low}}[\gamma^A] - \text{Tr}\left(\gamma^A\mathbf{v}_{\text{emb}}[\gamma^A, \gamma^B]\right). \tag{11.108}$$

11.6.1.2 Partition of the Occupied Space

An embedding calculation usually starts from an SCF calculation of the full system at the lower level of theory, which yields canonical MOs. Therefore, it is necessary to partition the occupied space and assign orbitals to fragments A and B (without losing generality, assuming A is the embedded fragment). In the original work by Manby *et al.*,[104] this was achieved by

- Performing a Pipek-Mezey localization[126] of the canonical occupied orbitals;
- Assigning a PM-localized orbital to the "active" fragment A if its Mulliken population on A is greater than 0.4.

This approach has been adopted as the default occupied space partition method in Q-CHEM.

Recently a parameter-free and more robust partition scheme was proposed by Claudino and Mayhall, which is known as the Subsystem Projected AO Decomposition (SPADE) procedure.[33] In this approach, one first transforms the occupied orbitals into the symmetrically orthogonalized AO basis:

$$\bar{\mathbf{C}}_{\text{occ}} = \mathbf{S}^{1/2}\mathbf{C}_{\text{occ}} \tag{11.109}$$

and then denotes the rows in $\bar{\mathbf{C}}_{\text{occ}}$ that correspond to fragment A as $\bar{\mathbf{C}}^A_{\text{occ}}$. A singular value decomposition (SVD) is then applied to $\bar{\mathbf{C}}^A_{\text{occ}}$: $\bar{\mathbf{C}}^A_{\text{occ}} = \mathbf{U}^A\Sigma^A(\mathbf{V}^A)^T$, and the SPADE orbitals are then obtained by rotating the original $\mathbf{C}_{\text{occ}}$:

$$\mathbf{C}^{\text{SPADE}}_{\text{occ}} = \mathbf{C}_{\text{occ}}\mathbf{V}^A. \tag{11.110}$$

The largest gap in the singular value spectrum determines the most appropriate occupied orbital partition under the given fragmentation.

11.6.1.3 Truncation of the Virtual Space Using Concentric Localization

A WFT-in-DFT calculation requires not only the occupied orbitals on the "active" fragment but also the virtual orbitals. Unlike the occupied orbitals, the virtual orbitals obtained from a projection-based embedding calculation are not assigned to fragments but stay delocalized. If the full virtual space is used in the post-SCF calculation, the savings on computational cost will be rather limited since only the number of occupied orbitals is reduced. Therefore, it is desirable to further truncate the virtual space so that one can significantly reduce the computational cost of WFT-in-DFT calculations.

Claudino and Mayhall recently proposed a simple and efficient approach to truncate the virtual space based on concentric localization (CL),[34] which shares the same spirit as the SPADE partition scheme for occupied space. As the first step, the original set of delocalized virtual orbitals ($\mathbf{C}_{\text{vir}}$) represented in the working basis (WB) are projected onto the embedded fragment A in a user-specified projection basis (PB):

$$ {}^{A}\mathbf{C}'_{\text{vir}} = (\mathbf{S}^{-1}_{\text{PB},A})({}^{A}\mathbf{S}_{\text{PB,WB}})\mathbf{C}_{\text{vir}} \tag{11.111} $$

where the superscript "A" indicates that only the rows corresponding to fragment A's basis functions are included and $\mathbf{S}_{\text{PB},A}$ denotes the overlap matrix for PB functions on fragment A only. One can choose PB to be the same as WB or even a smaller basis set. A particular set of virtual orbitals denoted as $\mathbf{C}_0$ can then be selected by performing an SVD on the overlap between ${}^{A}\mathbf{C}'_{\text{vir}}$ and $\mathbf{C}_{\text{vir}}$:

$$ ({}^{A}\mathbf{C}'_{\text{vir}})^{T}({}^{A}\mathbf{S}_{\text{PB,WB}})\mathbf{C}_{\text{vir}} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{T} \tag{11.112} $$

with

$$ \mathbf{V} = [\mathbf{V}_{\text{span}}|\mathbf{V}_{\text{null}}] \tag{11.113a} $$

$$ \mathbf{C}_0 = \mathbf{C}_{\text{vir}}\mathbf{V}_{\text{span}} \tag{11.113b} $$

$$ \mathbf{C}_{\text{null},0} = \mathbf{C}_{\text{vir}}\mathbf{V}_{\text{null}} \tag{11.113c} $$

By construction, $\mathbf{C}_0$ should consist of the virtual valence shell of the WB. In order to achieve higher accuracy for the embedded correlated method, one can select more virtual orbitals from $\mathbf{C}_{\text{null},0}$ in a stepwise fashion. A recommended way[34] is to singular value decompose the matrix $\mathbf{C}^{T}\mathbf{F}\mathbf{C}_{\text{null},0}$, *i.e.*, the coupling between $\mathbf{C}_0$ and $\mathbf{C}_{\text{null},0}$ through the Fock operator

$$ \mathbf{C}_0^{T}\mathbf{F}\mathbf{C}_{\text{null},0} = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^{T} \tag{11.114} $$

where

$$ \mathbf{V}_0 = [\mathbf{V}_{\text{span},0}|\mathbf{V}_{\text{null},0}] \tag{11.115a} $$

$$ \mathbf{C}_1 = \mathbf{C}_{\text{null},0}\mathbf{V}_{\text{span},0} \tag{11.115b} $$

$$ \mathbf{C}_{\text{null},1} = \mathbf{C}_{\text{null},0}\mathbf{V}_{\text{null},0} \tag{11.115c} $$

$$ \mathbf{C}_{\text{vir, act}} = [\mathbf{C}_0|\mathbf{C}_1] \tag{11.115d} $$

As the size of $\mathbf{C}_1$ is the same as $\mathbf{C}_0$, going through the procedure given by Eq. (11.114) doubles the number of active virtual orbitals. This procedure can be carried on iteratively, rendering the accuracy of this method tunable:

$$ \mathbf{C}_n^{T}\mathbf{F}\mathbf{C}_{\text{null},n} = \mathbf{U}_n\mathbf{\Sigma}_n\mathbf{V}_n^{T} \tag{11.116} $$

where

$$ \mathbf{V}_n = [\mathbf{V}_{\text{span},n}|\mathbf{V}_{\text{null},n}] \tag{11.117a} $$

$$ \mathbf{C}_{n+1} = \mathbf{C}_{\text{null},n}\mathbf{V}_{\text{span},n} \tag{11.117b} $$

$$ \mathbf{C}_{\text{null},n+1} = \mathbf{C}_{\text{null},n}\mathbf{V}_{\text{null},n} \tag{11.117c} $$

$$ \mathbf{C}_{\text{vir, act}} = [\mathbf{C}_0|\mathbf{C}_1|\cdots|\mathbf{C}_{n+1}] \tag{11.117d} $$

The virtual orbitals that span the null space, $\mathbf{C}_{\text{null},n+1}$, will remain inactive in the post-SCF calculations. In practice, one is often able to obtain sub-kcal/mol accuracy by only including $\mathbf{C}_0$ and $\mathbf{C}_1$, which is known as the "double-ζ" CL shell model.

11.6.2 Job Control for DFT-in-DFT and WFT-in-DFT Calculations

The DFT-in-DFT and WFT-in-DFT calculations are implemented in Q-CHEM under the framework of its default SCF engine GEN_SCFMAN, which is triggered by setting GEN_SCFMAN_EMBED = TRUE. The embedding job requires specification of fragments through the *$molecule* section (see Section 12.2). Unless otherwise specified, the first fragment is regarded as the embedded ("active") fragment. The user is allowed to select the embedded fragment through the *$embed_fragment* section.

The high-level electronic structure method is given by *$rem* variable METHOD, and the low-level method can be specified via ENV_METHOD. Note that one should choose a mean-field method as the low-level theory, such as pure or hybrid density functionals. Unlike the previous implementation of this embedding theory based on "EmbedMan" (Sec. 11.6.3), there is no specific limitation to the high-level methods. Currently supported options include hybrid DFT (*e.g.*, ωB97X-V) and correlated wavefunction methods (*e.g.*, MP2, CCSD(T), etc.) for ground-state calculations, as well as TDDFT and WFT-based methods such as EOM-CCSD for excited states.

The calculation starts with an SCF calculation at the low-level theory, and then it performs an embedded SCF calculation with the higher-level DFT (for DFT-in-DFT cases) or Hartree-Fock (for WFT-in-DFT cases). To make post-SCF correlated methods directly applicable, the molecular orbitals are stored on disk in the "[Occ(environment) | Occ(active) | Vir (active) | Vir(inactive)]" order, and the environment occupied orbitals and the excluded inactive virtual orbitals are treated as frozen core and frozen virtual orbitals in the correlated WFT method, respectively. Note that the virtual orbitals are not truncated unless the concentric localization method is invoked by setting CONCENTRIC_VIRTS = TRUE since that is the only virtual space truncation scheme for projection-based embedding theory that is currently available in Q-CHEM.

In Q-CHEM 5.3, the use of projection-based embedding theory is limited to overall closed-shell systems ("restricted-in-restricted" embedding). The extension to open-shell systems based on unrestricted SCF calculations has been enabled in Q-CHEM 5.4.1 and future releases.

From Q-CHEM 5.4.2, projection-based embedding is extended to work with complex basis functions. This enables the possibility to combine CBF high-level methods, namely CBF-(EOM-)CCSD and CBF-(EOM-)CC2 with complex HF (theory and functioning of these methods are described in Sections 7.10.9 and 4.9.5, respectively). The keyword to trigger this feature is GEN_SCFMAN_NH_EMBED. SPADE partition is not implemented in this framework, and the orbitals are partitioned via Pipek-Mezey localization and Mulliken population analysis of the real part.

GEN_SCFMAN_EMBED

Run a projection-based embedding calculation using the implementation based on-GEN_SCFMAN

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a projection-based embedding calculation

FALSE Do not perform an embedding calculation

RECOMMENDATION:

None

ENV_METHOD

Specify the low-level theory in a projection-based embedding calculation

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Parsed in the same way as *$rem* variable "METHOD"

RECOMMENDATION:

A mean-field method (pure or hybrid density functional) should be chosen.

FIXING_V_EMBED

Invoke the linearized approximation for the energy functional used for embedding calculations

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Use the linearized approximation for energy functional [Eq. (11.106)]

FALSE Use the original energy functional [Eq. (11.100)]

RECOMMENDATION:

Use the default to achieve savings in computational costs

SPADE_PARTITION

Use the SPADE approach to determine the initial set of embedded (active) orbitals

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use SPADE to partition the occupied space

FALSE Use the Pipek-Mezey localization + Mulliken population to assign occupied orbitals

RECOMMENDATION:

Use SPADE if a significant gap in the spectrum of singular values can be detected.

CONCENTRIC_VIRTS

Use the concentric localization (CL) scheme to truncate the virtual space

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use the CL scheme to truncate the virtual space

FALSE Leave the virtual space untruncated

RECOMMENDATION:

Use CL truncation for WFT-in-DFT calculations.

CONCENTRIC_VIRTS_ZETA

Specify the size of the truncated virtual space

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

m The total number of the CL-truncated virtuals is $m \times n_{\text{occ}}^{\text{active}}$

RECOMMENDATION:

Use the default; set it to a larger value if higher accuracy is requested.

CONCENTRIC_REF_BASIS

Specify the projection basis (PB) in the concentric localization procedure

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Parsed in the same way as BASIS; if unspecified, the working basis (WB) will be used as PB.

RECOMMENDATION:

WB is usually a good choice; a smaller basis can chosen with caution to further reduce the computational cost.

EMBEDDING_EARLY_STOP

Terminate the embedding calculation once the system partition is done (skip the embedded SCF)

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Terminate the embedding calculation once the system partition is done (skip the embedded SCF)

FALSE Doing a normal embedding calculation

RECOMMENDATION:

Turn it on for environment ESP/E-field calculations (see Section 10.6)

Example 11.35 DFT-in-DFT embedding calculation: ωB97X-V (water 2) in the presence of water molecules 1 and 3 described by PBE. The embedded ("active") fragment is specified by user through the *$embed_fragment* section.

```
$molecule
   0 1
   --
   0 1
   O    -0.00000   -0.00000   0.00000
   H     0.56704    0.41003   0.65488
   H     0.16427   -0.95671   0.05856
   --
   0 1
   O    -0.45701   -2.75857   0.16997
   H    -0.36918   -3.38806  -0.54777
   H    -1.36971   -2.42607   0.12617
   --
   0 1
   O    -2.62421   -0.99554  -0.00046
   H    -3.25789   -0.73194   0.66864
   H    -1.89698   -0.35254   0.04416
$end

$rem
   METHOD             wb97x-v
   BASIS              def2-svpd
   GEN_SCFMAN_EMBED   true
   THRESH             14
   SCF_CONVERGENCE    8
   ENV_METHOD         pbe
$end

$embed_fragment
   2
$end
```

Example 11.36 RI-MP2 (fragment 1) embedded in the environment described by B3LYP. The SPADE scheme is employed to partition the occupied space and the concentric localization (CL) method to truncate virtual space. The working basis cc-pVDZ is used as the default projection basis for CL. The same setup works with other correlated methods such as MP2, CCSD, etc.

```
$molecule
   0 1
   ---
   -1 1
   O       -1.1867 -0.2472  0.0000
   H       -1.9237  0.3850  0.0000
   ---
   1 1
   H       -0.0227  1.1812  0.8852
   C        0.0000  0.5526  0.0000
   H       -0.0227  1.1812 -0.8852
   C        1.1879 -0.3829  0.0000
   H        2.0985  0.2306  0.0000
   H        1.1184 -1.0093  0.8869
   H        1.1184 -1.0093 -0.8869
$end

$rem
   METHOD             rimp2
   BASIS              cc-pvdz
   AUX_BASIS          rimp2-cc-pvdz
   GEN_SCFMAN_EMBED   true
   THRESH             14
   SCF_CONVERGENCE    8
   ENV_METHOD         b3lyp
   SPADE_PARTITION    true
   CONCENTRIC_VIRTS   true
   N_FROZEN_CORE      0   ! no extra FC orbitals
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

Example 11.37 Embedded TDDFT-in-DFT calculation for HCHO in the environment of 5 water molecules. The environment occupied orbitals and the truncated virtuals remain frozen in the TDDFT calculation.

```
$molecule
0 1
--
0 1
C  1.331746  0.495076 -0.018037
O  1.023200 -0.680562  0.005977
H  0.567546  1.281583 -0.032606
H  2.385108  0.805133 -0.025252
--
0 1
O -1.943531  0.122291  0.090528
H -1.162581 -0.446371  0.092679
H -2.417905 -0.144628 -0.698637
O  1.944863 -3.376107 -1.905899
H  1.789904 -2.545078 -1.472450
H  2.563515 -3.252302 -2.616140
O  4.733862 -2.927159 -4.945061
H  4.171657 -2.306053 -5.393004
H  5.533115 -3.047999 -5.444154
O -1.551695  4.433654  1.051722
H -2.112970  4.877077  1.676899
H -2.009816  4.341491  0.224599
O  4.162091  1.378891  2.507658
H  3.250771  1.110644  2.513900
H  4.466676  1.486386  3.401063
$end

$rem
METHOD            CAM-B3LYP
BASIS             6-31+G(d)
SCF_ALGORITHM     DIIS
GEN_SCFMAN_EMBED  TRUE
ENV_METHOD        PBE0
SPADE_PARTITION   TRUE
CONCENTRIC_VIRTS  TRUE
CIS_N_ROOTS       2
CIS_TRIPLETS      FALSE
INTEGRAL_SYMMETRY FALSE
POINT_GROUP_SYMMETRY FALSE
$end
```

Example 11.38 Unrestricted projection-based embedding calculation: PBE0 (hydroxyl radical) in PBE (water molecule).

```
$molecule
   0 2
   --
   0 2
   O    -1.62521   0.00950    0.00000
   H    -0.64565  -0.05426    0.00000
   --
   0 1
   O    1.26276   -0.06380    0.00000
   H    1.70602    0.31170   -0.76717
   H    1.70602    0.31170    0.76717
$end

$rem
   jobtype            sp
   method             pbe0
   env_method         pbe
   basis              6-31g(d)
   unrestricted       true
   gen_scfman_embed   true
   thresh             14
   scf_convergence    8
   scf_algorithm      diis
   integral_symmetry    false
   point_group_symmetry false
$end
```

Example 11.39 Unrestricted projection-based embedding calculation: B3LYP in PBE Closed-shell supersystem with two open-shell fragments, giving a different result from R-in-R embedding. Note the use of a negative multiplicity on the second fragment to indicate an exess of beta electrons.

```
$molecule
0  1
---
0  2
O          1.232230   -0.273189    -0.127612
H         -2.080142    0.432972     0.072281
---
0 -2
C         -1.212970    -0.229528    -0.009715
H         -1.265591    -0.953985     0.809795
C          0.084975     0.559038     0.051054
H          0.150613     1.120024     0.994301
H          0.131609     1.284180    -0.764522
H         -1.273754    -0.774862    -0.954058
H          1.247387    -0.899873     0.615068
$end

$rem
  METHOD             b3lyp
  ENV_METHOD         pbe
  BASIS              6-31g*
  UNRESTRICTED       false
  GEN_SCFMAN_EMBED   true
  SCF_CONVERGENCE    8
$end
```

Example 11.40 CBF-EOM-CC in CBF-HF.

```
$molecule
0 1
---
0 1
 C     1.331746     0.495076    -0.018037
 O     1.023200    -0.680562     0.005977
 H     0.567546     1.281583    -0.032606
 H     2.385108     0.805133    -0.025252
---
0 1
 O    -1.943531     0.122291     0.090528
 H    -1.162581    -0.446371     0.092679
 H    -2.417905    -0.144628    -0.698637
$end

$rem
jobtype               sp
method                ccsd
complex_exponents     true
complex_scf           1
complex_scf_guess     1
basis                 6-31G  ! specify basis set explicitly, basis=gen
complex_basis         6-31G* !      "       "                complex_basis=zbasis_general
complex_theta         50
complex_ccman         true
gen_scfman_nh_embed   1
embed_method          1
env_method            hf
scf_convergence       12
cc_convergence        12
n_frozen_core         0
cs_ccsd               1
complex_metscf        1
complex_n_electrons   0
complex_spin_state    1
$end

$complex_ccman
cs_alpha 1000
cs_theta 0
$end
```

11.6.3 Previous Implementation Based on "EmbedMan"

The original implementation of the projection-based embedding in Q-CHEM was through the "EmbedMan" module, which was based on the old SCF module that is no longer the default. This implementation applies a level-shift projection operator to enforce the orthogonality between orbitals belonging to fragments 1 and 2. For the 1-in-2 case, the Fock matrix can be written as

$$\mathbf{f}^{(1)} = \mathbf{h} + \mathbf{J}[\gamma^{(1)} + \gamma^{(2)}] + \nu_{xc}[\gamma^{(1)} + \gamma^{(2)}] + \mu\mathbf{P}^{(2)} \tag{11.118}$$

where $\mathbf{P}^{(2)}$ is the level-shift projection operator constructed as:

$$\mathbf{P}^{(2)}_{\alpha\beta} = [\mathbf{S}\gamma^{(2)}\mathbf{S}]_{\alpha\beta} \tag{11.119}$$

where $\gamma^{(2)}$ is the localized density of fragment 2, and $\mathbf{S}$ is the AO overlap matrix. Upon convergence, an energy correction term is added to the final energy to account for the level-shift projection operator contribution to the Fock matrix energy. The correction term is calculated as the following:

$$E_{\text{correction}} = \mu * \text{tr}(\gamma^{(1)}\mathbf{P}^{(2)}) \tag{11.120}$$

Once the KS-DFT energy of fragment 1-in-2 is computed, a post-SCF method can be applied to this converged density to obtain the high-level QM additive energy of fragment 1. The same procedure can be repeated for fragment 2-in-1, without continuing to a post-SCF method to yield the low-level QM additive energy of the fragment 2. These energies are then summed to yield the total energy.

To run an embedding calculation through the "EmbedMan" module, one must split the super-molecular system into two fragments indexed 1 and 2, and set EMBEDMAN to 1. This is done through the standard Q-CHEM fragment input syntax. Two separate jobs must be run to find the total energy of fragment 1-in-2 at a high level QM theory, and fragment 2-in-1 at a low level QM theory. The order of the fragments in the *$molecule* section determines which fragment will undergo the high level QM. The user must submit a separate job for the 2-in-1 low-level QM calculation, with the order of the fragments reversed and EMBED_THEORY set to 0, which is the default value. The user must then add the final energies of the calculations to determine the total QM/QM embedded energy.

For the current Q-CHEM implementation of density embedding, it is necessary to specify the basis as MIXED, which requires to define the basis for each individual atom. When using CCSD(T), one should specify CCMAN2 as TRUE, for Q-CHEM's most updated coupled-cluster code. The current implementation of density embedding only works in combination with the following settings: SCF_ALGORITHM = DIIS, INCFOCK = 0, and PURECART = 222. It is also recommended that users disable symmetry for calculations with INTEGRAL_SYMMETRY = FALSE, and POINT_GROUP_SYMMETRY = FALSE. Refer to the sample input for correct job settings.

EMBEDMAN

Turns density embedding on.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use density embedding.

1 Turn on density embedding.

RECOMMENDATION:

Use EMBEDMAN for QM/QM density embedded calculations.

EMBED_THEORY

Specifies post-DFT method performed on fragment one.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No post HF method, only DFT on fragment one.

1 Perform CCSD(T) calculation on fragment one.

2 Perform MP2 calculation on fragment one.

RECOMMENDATION:

This should be 1 or 2 for the high-level QM calculation of fragment 1-in-2, and 0 for fragment 2-in-1 low-level QM calculation.

EMBED_MU

Specifies exponent value of projection operator scaling factor, μ [Eqs. (11.118) and (11.120)].

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n $\mu = 10^n$.

RECOMMENDATION:

Values of 2 - 7 are recommended. A higher value of μ leads to better orthogonality of the fragment MOs but $\mu > 10^7$ introduces numerical noise. $\mu < 10^2$ results in non-additive terms becoming too large. Energy corrections are fairly insensitive to changes in μ within the range of $10^2 - 10^7$.

EMBED_THRESH

Specifies threshold cutoff for AO contribution used to determine which MOs belong to which fragments

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Threshold $= n/1000$

RECOMMENDATION:

Acceptable values range from 0 to 1000. Should only need to be tuned for non-highly localized MOs

Example 11.11.41 Input for a MP2/PBE density embedding calculation of He-in-HF. The sum of the final energies for these two jobs will yield the total QM/QM energy.

View input online

11.7 Frozen-Density Embedding Theory

11.7.1 Introduction

Frozen-Density Embedding Theory [169,170] (FDET) provides a formal framework in which the whole system is described by means of two independent quantities: the embedded wave function (interacting or not) and the density associated with the environment.

The total energy equation in frozen density embedding theory for a wave function in state I embedded in a environment density $\rho_B(\mathbf{r})$ reads (for definitions see Table 11.11):

$$E_{AB}^{\mathrm{tot}}[\Psi_A^I, \rho_B] = \langle \Psi_A^I | \hat{H}_A | \Psi_A^I \rangle + V_B^{\mathrm{nuc}}[\rho_A^I] + J_{\mathrm{int}}[\rho_A^I, \rho_B] + E_{xc}^{\mathrm{nad}}[\rho_A^I, \rho_B] + T_s^{\mathrm{nad}}[\rho_A^I, \rho_B] + E_{v_B}^{\mathrm{HK}}[\rho_B] + V_A^{\mathrm{nuc}}[\rho_B] \tag{11.121}$$

The embedding operator $\hat{v}_{\mathrm{emb}}$, which is added to the Hamiltonian of subsystem A $\left(\hat{H}_A\right)$, is given in the form of a

potential:

$$v_{\mathrm{emb}}[\rho_A^I, \rho_B, v_B](\mathbf{r}) = v_B(\mathbf{r}) + \int \frac{\rho_B(\mathbf{r}')}{|\mathbf{r} - \mathbf{r}'|}\mathrm{d}\mathbf{r} + \frac{\delta E_{xc,T}^{\mathrm{nad}}[\rho_A^I, \rho_B]}{\delta \rho_A^I(\mathbf{r})} \tag{11.122}$$

The last term (non-electrostatic component) in equation 11.122 causes the embedding potential to be ρ_A-dependent, which in return induces an inconsistency between the potential and the energy. In the canonical form of FDET (*conventional* FDET) this inconsistency is addressed by performing macrocycles in which the embedding potential is repeatedly constructed using the current (embedded) density $\rho_A^{\mathrm{curr}}(\mathbf{r})$ after each cycle until self-consistency is reached. Each calculation performed with FDE-Man accounts for just *one cycle*, so self-consistency can only be reached by running multiple calculations, where the densities are updated using the importing options for the density matrices described in Section 11.7.3.2. Self-consistent macrocycles include the mutual polarization procedure: *Freeze-and-Thaw*.

However, in *linearized FDET* the non-additive energy functionals (for abbreviation denoted as $E_{xc,T}^{\mathrm{nad}}[\rho_A^I, \rho_B]$) are each approximated by a functional which is linear in $\rho_A(\mathbf{r})$. The approximation is constructed as a Taylor expansion of the non-additive energy functional at a reference density $\rho_A^{\mathrm{ref}}(\mathbf{r})$ with the series being truncated after the linear term.

$$E_{xc,T}^{\mathrm{nad}}[\rho_A^I, \rho_B] \approx E_{xc,T}^{\mathrm{nad}}[\rho_A^{\mathrm{ref}}, \rho_B] + \int \left(\rho_A^I(\mathbf{r}) - \rho_A^{\mathrm{ref}}(\mathbf{r})\right) \frac{\delta E_{xc,T}^{nad}[\rho_A^{\mathrm{ref}}, \rho_B]}{\delta \rho_A^{\mathrm{ref}}(\mathbf{r})}\mathrm{d}\mathbf{r} \tag{11.123}$$

In contrast to conventional FDET, the embedding potential then becomes ρ_A-independent and macrocycles are no longer necessary. Another consequence of the linearization is that orthogonality between states is maintained since the same potential is used for all states.

11.7.2 FDE-Man

Several FDET-based methods are available in the FDE-Man module of Q-CHEM. The FDE-Man job control is accomplished in two sections, *$rem* and *$fde*.

The fragments are specified via the fragment descriptors (see Section 12) in the *$molecule* section, whereas the first fragment corresponds to the embedded species (A), the second fragment represents the environment (B).

Note: The current implementation allows only for closed shell fragments.

FDE

Turns density embedding on.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Perform an FDET calculation.

False Don't perform FDET calculation.

RECOMMENDATION:

Set the *$rem* variable FDE to TRUE to start a FDET calculation.

Enabling FDE-Man, the specification of the embedding method and other job control parameters (thresholds, max. iterations *etc.*) should be set in the *$rem* section.

METHOD
Determines which FDET-based method should be used for the embedded wavefunction if FDE = TRUE.
TYPE:
STRING
DEFAULT:
None
OPTIONS:
NAME Use METHOD = *NAME*, where *NAME* is either HF for Hartree-Fock theory or else one of the DFT methods listed in Section 5.3.5.
RECOMMENDATION:
None

Other DFT functionals can also be requested with the EXCHANGE and CORRELATION keywords as described in Section 5.4.

Note: The current implementation is restricted to mean-field methods that are solved with the SCFMAN module.

The standard capabilities to use customized basis sets are also possible with the FDE-Man module.

11.7.2.1 FDE-ADC

FDE-ADC[132] is a density embedding method based on the combination of the Algebraic Diagrammatic Construction scheme for the polarization propagator (ADC, Section 7.11) and Frozen-Density Embedding Theory (FDET). In this particular variant, the subsystem A is represented by a wave function whereas subsystem B is described by a density. The FDE-ADC method uses the linearized FDET approximation.[179]

Similar to the other methods, the ADC specifications have to be done inside the *$rem* section, as shown below.

METHOD
Determines which FDET-based method should be used if FDE = TRUE.
TYPE:
STRING
DEFAULT:
None
OPTIONS:
adc(2) Perform an FDE-ADC(2)-s calculation.
adc(2)-x Perform an FDE-ADC(2)-x calculation.
adc(3) Perform an FDE-ADC(3) calculation (potential constructed with MP(2) density).
cvs-adc(2) Perform an FDE-ADC(2)-s calculation of core excitations.
cvs-adc(2)-x Perform an FDE-ADC(2)-x calculation of core excitations.
cvs-adc(3) Perform an FDE-ADC(3) calculation of core excitations.
RECOMMENDATION:
None

FDE-ADC also supports the excited state analysis (STATE_ANALYSIS) carried out by the LIBWFA module.

11.7.3 FDE-Man Job Control

The FDE-Man job control with respect to embedding parameters is accomplished via options in the *$fde* input section. The format of the *$fde* section requires key and value pairs separated by a space character:

```
$fde
   <Keyword>  <parameter>
$end
```

Note: The following job control variables belong *only* in the *$fde* section. Do not place them in the *$rem* section.

The supermolecular expansion (SE) uses the full basis set of the supersystem for calculations on each fragment. Because of the computational cost this option should only be used for small to medium sized supersystems. Note that for visualization of orbitals or densities SE only supports the generation of volumetric data via MAKE_CUBE_FILES (MOLDEN files are not supported, *i.e.* MOLDEN_FORMAT should be avoided). On the other hand, the monomer expansion (ME) only uses the basis set of each individual fragment. This choice of basis expansion is recommended for daily basis embedding calculations.

Analogous to a regular DFT calculation in Q-CHEM (by using METHOD) the exchange-correlation functional combination can either be selected with one keyword **XC_Func**,or by defining **X_Func** and **C_Func** (similar to EXCHANGE and CORRELATION). To include only the electrostatic terms in the embedding potential one can set all the functional keywords to **None**.

It is recommended to employ the same level of approximation for the non-additive kinetic and exchange-correlation functional, for instance TF/SVWN or PW91k/PW91.

The *minimal specifications required* to successfully run a FDET calculation are:

- Functionals for embedding potential: **T_Func**, **XC_Func** (or **X_Func** and **C_Func**),
- Basis expansion: **Expansion**,
- Method for environment density: **rhoB_method**

T_Func

Kinetic energy functional used for the construction of the embedding potential.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

TF	Use Thomas-Fermi kinetic energy functional.
PW91k / LC94	Use kinetic energy functional based on PW91.
None	Do not include any kinetic energy functional.

RECOMMENDATION:

Use the same level of approximation as for the non-additive exchange-correlation energy functional.

XC_Func

Exchange-Correlation functional used for the construction of the embedding potential.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

NAME *NAME* can be any of the LDA or GGA exchange-correlation functionals available in Q-CHEM.

None Do not include any exchange-correlation energy functional.

RECOMMENDATION:

Only use LDA or GGA-type functionals.

X_Func

Exchange functional used for the construction of the embedding potential.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

NAME *NAME* can be any of the LDA or GGA exchange functionals available in Q-CHEM.

None Do not include any exchange energy functional.

RECOMMENDATION:

Only use LDA or GGA-type functionals. **XC_Func** and **X_Func** are mutually exclusive.

C_Func

Exchange-Correlation functional used for the construction of the embedding potential.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

NAME *NAME* can be any of the LDA or GGA correlation functionals available in Q-CHEM.

None Do not include any exchange energy functional.

RECOMMENDATION:

Only use LDA or GGA-type functionals. **XC_Func** and **C_Func** are mutually exclusive.

Expansion

Specifies which basis set expansion should be used.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

SE/super/supermolecular	Supermolecular basis is used for both System A and B.
ME/mono/monomer	Monomer expansion basis is used on each System A and B.

RECOMMENDATION:

SE should be used for testing purposes only, since it is more expensive. Use the ME as standard choice, particularly for large systems.

rhoA_method

Method to calculate reference density $\rho_A^{\mathrm{ref}}(\mathbf{r})$ of the core fragment (A).

If DFT is requested, the respective exchange-correlation functional is the same as defined for the embedding calculation, *i.e.* with the keywords **XC_FUNC** or **X_FUNC** and **C_FUNC**.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

HF

OPTIONS:

HF	Use Hartree-Fock method.
DFT	Use Density Functional Theory.
MP	Use Second-Order Møller-Plesset perturbation theory.

RECOMMENDATION:

Use either HF or DFT.

rhoB_method

Method to calculate the environment density (B).

If DFT is requested, the respective exchange-correlation functional has to be defined using the keyword **XC_FUNC_B** or **X_FUNC_B** and **C_FUNC_B**.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

HF	Use Hartree-Fock method.
DFT	Use Density Functional Theory.
MP	Use Second-Order Moller-Pleset method.

RECOMMENDATION:

Use either HF or DFT.

XC_Func_B

Exchange-Correlation functional used for the environment DFT calculation.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

NAME *NAME* can be any of the LDA, GGA, or global hybrid-GGA exchange-correlation functionals available in Q-CHEM.

RECOMMENDATION:

None

X_Func_B

Exchange functional used for the environment DFT calculation.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

NAME *NAME* can be any of the LDA or GGA exchange functionals available in Q-CHEM.

RECOMMENDATION:

XC_Func_B and **X_Func_B** are mutually exclusive.

C_Func_B

Correlation functional used for the environment DFT calculation.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

NAME *NAME* can be any of the LDA or GGA correlation functionals available in Q-CHEM.

RECOMMENDATION:

XC_Func_B and **C_Func_B** are mutually exclusive.

PrintLevel

Print level for FDE-Man output.

INPUT SECTION: *$fde*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 minimum print level

1 extended print level

2 maximum print level

3 maximum print level and additional text files (densities, etc.)

RECOMMENDATION:

Use minimal print level.

11.7.3.1 Superposition of Molecular Densities

When the selected environment comprises many atoms and molecules, a sensible approximation is to construct the total environment density as the sum of the densities of the individual molecules. This can be done with the keywords **Superposition_B** or **Import_Superposition_B**. In the first case, the molecular densities are computed in the same calculation, and assembled into one single density matrix. The molecular densities and the superposed density matrices are saved into text files.

Superposition_B

Compute the density of fragment B as a superposition of molecular densities.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Do a superposition of molecular densities to build the density of fragment B.

FALSE Don't do a superposition, use the whole fragment B.

RECOMMENDATION:

The molecular densities are calculated with the method specified with the keyword: **rhoB_method**.

In the second case, the individual density matrices are expected to be already calculated independently and are required to be present in the same folder of the calculation.

Import_Superposition_B

Assemble the density of fragment B from molecular densities.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Read and superpose the molecular densities to build the density of fragment B.

FALSE Don't do a superposition, use the whole fragment B.

RECOMMENDATION:

The files with the molecular density matrices should be `.txt` files, numerated from 0, with the prefix `SCF_Dens_B_Subfrag_`.

Note: No embedding calculation is performed if these functions are selected. In both cases, only the geometry of fragment B has to be given, and each molecule needs to be specified as independent fragment in the MOLECULE section.

When using the *superposition* options, the file of the superposed density matrix of the environment (B) is saved with the name `Densmat_B.txt`, and can be used with the **import_rhoB** option to subsequently perform an embedding calculation (see Sec. 11.7.3.2 below).

11.7.3.2 Import Densities

In FDE-Man the user has the option to import the density matrices used to generate the embedding potential. This alternative is particularly useful if the desired method to generate the densities for the embedding potential is not available in the current version, for instance, or when the superposition of densities is used to generate $\rho_B(\mathbf{r})$. The density matrices must be given *in AO basis* in a text file with the following format:

```
<nspin> <nbas> <nbas>
<value>
<value>
...
```

where `<nspin>` has two possible values (1 or 2) depending on the number of spin densities contained in the file. The first line should also contain the dimensions of the density matrix (`<nbas>`).

import_rhoA

Import density of subsystem A used for the embedding potential.

The file must be named `Densmat_A.txt` and located in the calculation folder.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Read the density for subsystem A from file.

FALSE Do not use density from file.

RECOMMENDATION:

None

import_rhoB

Import density of subsystem B used for the embedding potential.

The file must be named `Densmat_B.txt` and located in the calculation folder.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Read the environment density from file.

FALSE Do not use density from file.

RECOMMENDATION:

None

11.7.3.3 Pre-Polarization of the Environment Density

Formally the exactness of FDET equations requires the condition $\forall \mathbf{r}\,(\rho_0(\mathbf{r}) \leq \rho_B(\mathbf{r}))$ to be satisfied, whereby $\rho_0(\mathbf{r})$ denotes the exact density of the supersystem.[119,169,170] This condition can, however, not be assured *a priori*. In practice, deficiencies with the initial choice of $\rho_B(\mathbf{r})$ can be cured to some extent by pre-polarizing the environment density.[136] Pre-polarization schemes available in FDE-Man can be turn on with the **prepol** keyword. The different pre-polarization options are described below.

prepol

Pre-polarize $\rho_B(\mathbf{r})$ in the field of the embedded species represented by point charges or a charge density.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Turn on pre-polarization of $\rho_B(\mathbf{r})$.

FALSE Do not pre-polarize the environment density.

RECOMMENDATION:

See Ref. 136 for a review of FDET polarization schemes.

prepol_type

Type of pre-polarization scheme.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

Mulliken

OPTIONS:

Mulliken	Use Mulliken charges of system A to pre-polarize $\rho_B(\mathbf{r})$.
CHELPG	Use ChElPG charges of system A to pre-polarize $\rho_B(\mathbf{r})$.
elstat / Coulomb	Use electrostatic embedding potential (using density and nuclear charges of A) to pre-polarize $\rho_B(\mathbf{r})$.

RECOMMENDATION:

None

root_Aref

Defines which electronic state is considered for $\rho_A^{\text{ref}}(\mathbf{r})$ in the pre-polarization scheme.

INPUT SECTION: *$fde*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n	Use *n*th state of A for pre-polarization scheme (ground state = 0).

RECOMMENDATION:

None

11.7.3.4 Miscellaneous Options

Debug

Request additional printings as well as saving the separate components of the embedding potential (in AO basis) to disk.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Print extra options and save components to disk.
FALSE	Do not print extra info.

RECOMMENDATION:

This option creates multiple files and additional details in the output. Use only for debugging purposes.

rhoB_basis

Use a different basis set for fragment B than the one for the embedded system A, specified in the *$rem* section.

Basis sets can be requested in the same way as in the BASIS section. However, contrary to the standard definition inside the input file, the user-defined basis and mixed basis must be defined without `$basis` and `$end`, in a text file called `fde_genbasB` and located in the calculation folder.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

General, Gen	User-defined. See Section 8.4.
Symbol	Use standard basis sets as in the tables in Section 8.3.
Mixed	Use a combination of different built-in basis sets (see Section 8.5).

RECOMMENDATION:

Smaller basis for fragment B can be used to improve the performance of embedding calculations.

Note: All basis used within the FDE-Man module use PURECART = 2111.

print_props

Print molecular properties (multipole moments, Mulliken charges) associated with the reference fragment densities used to construct the embedding potential.

INPUT SECTION: *$fde*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0	Do not print molecular properties.
1	Print molecular properties only for fragment A.
2	Print molecular properties for both fragments A and B.

RECOMMENDATION:

None

If the user desires to *only* save the embedding potential without performing an embedding calculation, it can be done by setting to *TRUE* simultaneously the variables **debug** and **Vemb_only**. In this case, the individual components of the embedding potential are saved to disk into text files as mentioned above in the **debug**.

Vemb_only

Only generate the embedding potential and save its components to disk.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Generate embedding potential and exit FDE-Man.

FALSE Perform regular embedding calculation.

RECOMMENDATION:

None

11.7.4 Single-Fragment Calculations

FDE-Man allows the user to perform the embedding calculation by importing the embedding potential in its atomic-orbital matrix representation. Consequently, all the specifications needed to construct the embedding potential are no longer required. Additionally, the single-fragment mode is turned on when only *one molecule* is given in the MOLECULE section.

import_vmat

Import the embedding potential in AO matrix representation.

The expected file name is `FDE_vembmat.txt` and does not have a header. All the keywords related to fragment B and the embedding potential will cause an error.

INPUT SECTION: *$fde*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Read and add the embedding potential to the Fock matrix.

FALSE Don't read the embedding potential, use other options.

RECOMMENDATION:

None

Note: This functionality is constructed as a separate module, only the **import_rhoA** and **print_props**=1 options can be used with the single-fragment mode.

One can also use the single-fragment mode to obtain Mulliken and ChElPG charges from a density matrix. In order to do that, one needs to import the density of fragment A, and set **charges** to one of the options below. Importing the density matrix of fragment A is done as described previously.

charges

Compute Mulliken or ChElPG charges from a density matrix in AO matrix representation.

The **import_rhoA** option must be set to *TRUE*.

INPUT SECTION: *$fde*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

Mulliken Compute Mulliken charges from system A density matrix.

CHELPG Compute ChElPG charges from system A density matrix.

RECOMMENDATION:

None

11.7.5 Read an External Potential From a File

Users can introduce an external potential affecting the active molecule in a DFT calculation, representing an embedding environment. Any functional can be used. To invoke the reading of an external potential, users must set the keyword **PROJ_CAP = -10** in the $rem section, and the file named **external_potential.grid** must be present at the same directory. The external potential must be expressed at the same exact grid points. The potential file includes only the potential values in *atomic coordinates*. As a preparation step the grid points can be printed using libqints by setting the flag **use_libqints = true**. A related published work[26] can also be read in order to see its usage in different DFT functional.

11.7.6 Examples

Example 11.42 Input for a FDE-HF/cc-pVDZ calculation with monomer expansion of CO embedded in one water molecule, using LDA functionals for the embedding potential.

```
$molecule
   0 1
--
   0 1
   C    -3.618090    1.376803   -0.020795
   O    -4.735683    1.525556    0.115023
--
   0 1
   O    -7.956372    1.485406    0.116792
   H    -6.992316    1.421133    0.177470
   H    -8.105846    2.442220    0.111599
$end

$rem
point_group_symmetry = False
   METHOD       = hf
   BASIS        = cc-pvdz
   FDE          = true
   MEM_STATIC   = 1024
   MEM_TOTAL    = 2200
$end

$fde
   T_Func       TF
   X_Func       Slater
   C_Func       VWN5
   expansion    mono
   rhoB_method  HF
$end
```

Example 11.43 Input for a superposition of molecular densities with HF/cc-pVDZ of 5 water molecules.

```
$rem
   METHOD         = HF
   BASIS          = cc-pvdz
   FDE            = true
   MEM_STATIC     = 1024
   MEM_TOTAL      = 4000
point_group_symmetry = False
$end

$molecule
  0 1
--
  0 1
  O       8.67400    -66.68200    -88.40400
  H       8.90400    -65.96300    -89.01800
  H       9.36700    -67.38600    -88.67300
--
  0 1
  O      10.22000    -68.60400    -89.71700
  H      10.40700    -68.14800    -90.58500
  H       9.58700    -69.25100    -90.08700
--
  0 1
  O       8.20100    -70.34800    -90.56300
  H       8.40300    -71.24600    -90.20800
  H       8.75700    -70.28300    -91.32900
--
  0 1
  O       4.44400    -62.52200    -85.97300
  H       5.18300    -62.60300    -86.58400
  H       4.88700    -62.11200    -85.22800
--
  0 1
  O       5.71200    -65.65900    -83.58300
  H       5.98900    -65.59500    -84.50500
  H       4.99900    -66.31400    -83.64900
$end

$fde
   T_Func           TF
   XC_Func          PBE
   expansion        mono
   Superposition_B  true
   rhoB_method      HF
$end
```

Example 11.44 Input for a FDE-ADC(2)/cc-pVDZ calculation in supermolecular expansion of CO embedded in one water molecule. For the embedding potential, the selected functionals are: Thomas-Fermi kinetic energy, and PBE for the exchange-correlation term.

```
$molecule
   0 1
--
   0 1
   C    -3.618090    1.376803   -0.020795
   O    -4.735683    1.525556    0.115023
--
   0 1
   O    -7.956372    1.485406    0.116792
   H    -6.992316    1.421133    0.177470
   H    -8.105846    2.442220    0.111599
$end

$rem
   METHOD               = adc(2)
   BASIS                = cc-pvdz
   EE_STATES            = 2
   FDE                  = true
   MEM_STATIC           = 1024
   MEM_TOTAL            = 16000
   ADC_DAVIDSON_MAXITER = 100
   ADC_DAVIDSON_CONV    = 5
point_group_symmetry = False
$end

$fde
   T_Func       TF
   XC_Func      PBE
   expansion    super
   rhoB_method  HF
$end
```

Example 11.45 Input for a Externally generated potential file to be read in any DFT calculation.

```
$comment
First: Generates the DFTGRID for the the projection of the potential indicating
use_libqints     true // This command will generate the DFTGRIDS.txt
SYM_IGNORE       true //
no_reorient      true // Orientation should not change in all job runs
Warning: basis set used to generate the dftgrid must be same when external
potential is read.
Second: Run first inital job without PROJ_CAP and then with PROJ_CAP
$end

$molecule
0 1
        O         0.0000000000    0.0000000000    0.1210550360
        H        -0.7527950783    0.0000000000   -0.4706429710
        H         0.7527950783    0.0000000000   -0.4706429710
$end

$rem
jobtype               sp
exchange              b3lyp
correlation           none
basis                 6-31G*
scf_convergence       8
max_scf_cycles        5000
mem_static            4000
mem_total             42000
molden_format         false
SYM_IGNORE            true
no_reorient           true
$end

@@@

$molecule
  read
$end

$rem
jobtype               sp
exchange              b3lyp
correlation           none
basis                 6-31G*
scf_convergence       8
max_scf_cycles        5000
mem_static            4000
mem_total             42000
molden_format         false
PROJ_CAP              -10
SYM_IGNORE            true
no_reorient           true
$end
```

11.7.7 FDE-Man output

In general the FDE-Man output indicates all important stages of the embedding calculation, which are:

1. Generation of $\rho_A^{\mathrm{ref}}(\mathbf{r})$,

2. Generation of $\rho_B(\mathbf{r})$,
3. Construction of the embedding potential,
4. Start of FDE-Man embedding calculation and
5. Final FDE-Man summary.

In the following table definitions of the terms printed to the output are collected. These quantities are printed for every state, *i.e.* for every $\rho_A^I(\mathbf{r})$. In addition, the non-electrostatic interactions with respect to the reference density $\rho_A^{\mathrm{ref}}(\mathbf{r})$ are printed at the top of the FDE-Man summary.

Subsystem Energies	
`Embedded system (A)`	$\langle \Psi_A^I \vert \hat{H}_A + v_{\mathrm{emb}}^{\mathrm{lin}} \vert \Psi_A^I \rangle$
`Environment (B)`	$E_B = E_{v_B}^{\mathrm{HK}}[\rho_B]$ or E_{HF}
Electrostatic Interactions	
`rho_A <-> rho_B`	$J_{\mathrm{int}}[\rho_A, \rho_B] = \int\int \frac{\rho_A(\mathbf{r})\rho_B(\mathbf{r}')}{\vert \mathbf{r}-\mathbf{r}' \vert} \mathrm{d}\mathbf{r}\mathrm{d}\mathbf{r}'$
`rho_A <-> Nuc_B`	$V_B^{\mathrm{nuc}}[\rho_A] = \int \rho_A(\mathbf{r}) v_B(\mathbf{r}) \mathrm{d}\mathbf{r}$
`rho_B <-> Nuc_A`	$V_A^{\mathrm{nuc}}[\rho_B] = \int \rho_B(\mathbf{r}) v_A(\mathbf{r}) \mathrm{d}\mathbf{r}$
`Nuc_A <-> Nuc_B`	$V_{N_A N_B} = \sum_i \sum_j \frac{Z_i Z_j}{\vert R_i - R_j \vert}$
Non-Electrostatic Interactions	
`non-additive E_xc`	$E_{xc}^{\mathrm{nad}}[\rho_A^I, \rho_B]$
`non-additive T_s`	$T_s^{\mathrm{nad}}[\rho_A^I, \rho_B]$
`integrated v_xc nad`	$\int \rho_A^I(\mathbf{r}) v_{xc}^{\mathrm{nad}}(\mathbf{r}) \mathrm{d}\mathbf{r}$
`integrated v_T nad`	$\int \rho_A^I(\mathbf{r}) v_T^{\mathrm{nad}}(\mathbf{r}) \mathrm{d}\mathbf{r}$
Final FDET energies	
`Delta_Lin`	$\int (\rho_A^I(\mathbf{r}) - \rho_A^{\mathrm{ref}}(\mathbf{r})) v_{xc,T}^{\mathrm{nad}}(\mathbf{r}) \mathrm{d}\mathbf{r}$
`Final Energy (A)`	$E_A^{\mathrm{emb}}[\Psi_A^I, \rho_B] = \langle \Psi_A^I \vert \hat{H}_A \vert \Psi_A^I \rangle + J_{\mathrm{int}}[\rho_A^I, \rho_B] + V_B^{\mathrm{nuc}}[\rho_A^I]$ $+E_{xc,T}^{\mathrm{nad}}[\rho_A^{\mathrm{ref}}, \rho_B] + \Delta^{\mathrm{lin}}[\rho_A^I, \rho_A^{\mathrm{ref}}, \rho_B] + V_A^{\mathrm{nuc}}[\rho_B] + V_{N_A N_B}$
`Final Energy (A+B)`	$E_A^{\mathrm{emb}}[\Psi_A^I, \rho_B] + E_B$

Table 11.11: Definition of output terms.

11.8 Polarizable Embedding Model

11.8.1 Introduction

The polarizable embedding (PE) model is a fragment-based quantum-classical explicit embedding scheme to model molecular properties in complex heterogeneous environments[120,121]. The theory is explained thoroughly in literature[101,120,121]. In essence, the environment is represented by a multi-center multipole expansion to model electrostatic interactions, whereas polarization is taken into account by dipole-dipole polarizabilities placed at the expansion points. Polarization effects can thus be treated fully self-consistently by mutual polarization of the environment and the quantum region.

A recent tutorial review on how to prepare PE calculations in general (creating embedding potentials) is also available[154]. For automated generation of embedding potentials, please refer to the PyFraME tool [1] which is also explained in the aforementioned review.

PE can be used for Hartree-Fock and density-functional theory ground-state SCF methods. In addition, PE has been combined with the algebraic-diagrammatic construction for the polarization propagator (ADC)[141], explained in the subsequent section.

The combined scheme of the PE model and ADC (PE-ADC)[141] is built on top of a PE-HF ground-state calculation and takes into account perturbative corrections of the excitation energies in a density-driven manner. That is, after the Hartree-Fock ground-state calculations, the induced dipole moments in the environment are kept frozen and an ADC calculation is performed as usual. Thereafter, perturbative corrections of the electronic excitation energies are calculated based on i) the transition density (perturbative linear-response-type correction, *ptLR*), and ii) the difference density (perturbative state-specific correction, *ptSS*) for each excited state.

11.8.2 Job Control

The PE job control is accomplished in two sections, *$rem* and *$pe*. To enabling PE-ADC, specification of the ADC method and other ADC job control parameters (thresholds, max. iterations *etc.*) should be set in the *$rem* section. PE-ADC supports the excited state analysis (STATE_ANALYSIS) carried out by the LIBWFA module.

PE

Turns PE on.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Perform a PE calculation.

False Don't perform a PE calculation.

RECOMMENDATION:

Set the *$rem* variable PE to TRUE to start a PE calculation.

Note: Turning PE on disables symmetry by setting POINT_GROUP_SYMMETRY to FALSE.

Note: Setting the REM variables USE_LIBQINTS and GEN_SCFMAN to TRUE is required to run PE.

The PE-specific options can be set in the *$pe* input section. The format of the *$pe* section requires key and value pairs separated by a space character:

```
$pe
   <keyword>  <parameter>
$end
```

Note: The following job control variables belong *only* in the *$pe* section. Do not place them in the *$rem* section.

[1] https://gitlab.com/FraME-projects/PyFraME

POTFILE

Path of the potential file.

INPUT SECTION: *$pe*

TYPE:

STRING

DEFAULT:

potfile.pot

OPTIONS:

Provide the path/name of the potential file.

RECOMMENDATION:

None

DIIS

Use DIIS acceleration to obtain induced moments.

INPUT SECTION: *$pe*

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Turns DIIS acceleration on.

FALSE Turns DIIS acceleration off (normal Jacobi solver is used).

RECOMMENDATION:

TRUE

CONVERGENCE_INDUCED

Threshold for induced moments convergence.

Converge induced moments to a residual norm of $10^{-\text{CONVERGENCE\_INDUCED}}$.

INPUT SECTION: *$pe*

TYPE:

INTEGER

DEFAULT:

8 Corresponding to 10^{-8}

OPTIONS:

$n \leq 12$ Corresponding to 10^{-n}

RECOMMENDATION:

Use the default unless higher accuracy is desired.

MAXITER

Maximum number of iterations for induced moments.

INPUT SECTION: *$pe*

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

$n \geq 1$

RECOMMENDATION:

Use the default. If more iterations are required to converge the induced moments, there might be an error in the system setup.

BORDER

Activate border redistribution/removal options for sites in proximity to the QM/MM border.

INPUT SECTION: *$pe*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Enable border options.

FALSE Disable border options.

RECOMMENDATION:

None

BORDER_TYPE

Remove or redistribute multipole moments/polarizabilities.

INPUT SECTION: *$pe*

TYPE:

STRING

DEFAULT:

REMOVE

OPTIONS:

REMOVE remove multipole moments/polarizabilities.

REDIST redistribute multipole moments/polarizabilities.

RECOMMENDATION:

None

BORDER_RMIN

Minimum distance from QM atoms to MM sites to be taken into account for removal/redistribution

INPUT SECTION: *$pe*

TYPE:

FLOAT

DEFAULT:

2.2 (AU)

OPTIONS:

$r > 0$ (Unit depends on BORDER_RMIN_UNIT)

RECOMMENDATION:

None

BORDER_RMIN_UNIT

Unit of BORDER_RMIN, default is atomic units (AU)

INPUT SECTION: *$pe*

TYPE:

STRING

DEFAULT:

AU

OPTIONS:

AU Use atomic units.

AA Use Angstrom.

RECOMMENDATION:

None

BORDER_REDIST_ORDER

Order from which on moments are removed. For example, if set to 1 (default), only charges are redistributed and all higher order moments are removed.

INPUT SECTION: *$pe*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

$n = 0, 1, 2, 3, ...$

RECOMMENDATION:

None

BORDER_N_REDIST

Number of neighbor sites to redistribute multipole moments/polarizabilities to. The default (-1) redistributes to all sites which are not in the border region.

INPUT SECTION: *$pe*

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

$n = -1, 1, 2, 3, ...,$ number of MM sites

RECOMMENDATION:

Use the default value.

BORDER_REDIST_POL

Redistribute polarizabilities? If set to FALSE, polarizabilities are removed.

INPUT SECTION: *$pe*

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Redistribute polarizabilities.

FALSE Remove polarizabilities.

RECOMMENDATION:

None

Example 11.46 Input for a PE-HF calculation of 4-Nitroaniline in presence of six water molecules

```
$comment
The potential file \texttt{gen\_scfman\_pe\_potfile.pot} can be
found in the samples folder.
$end

$molecule
   0 1
   C      8.64800        1.07500       -1.71100
   C      9.48200        0.43000       -0.80800
   C      9.39600        0.75000        0.53800
   C      8.48200        1.71200        0.99500
   C      7.65300        2.34500        0.05500
   C      7.73200        2.03100       -1.29200
   H     10.18300       -0.30900       -1.16400
   H     10.04400        0.25200        1.24700
   H      6.94200        3.08900        0.38900
   H      7.09700        2.51500       -2.01800
   N      8.40100        2.02500        2.32500
   N      8.73400        0.74100       -3.12900
   O      7.98000        1.33100       -3.90100
   O      9.55600       -0.11000       -3.46600
   H      7.74900        2.71100        2.65200
   H      8.99100        1.57500        2.99500
$end

$rem
   METHOD               HF
   BASIS                STO-3G
   PE                   TRUE
point_group_symmetry False
   USE_LIBQINTS         TRUE
$end

$pe
   potfile  gen_scfman_pe_potfile.pot
$end
```

11.8.3 Interpreting the Output

After SCF convergence, the PE module prints a summary of PE energy contributions, for example:

```
-----------------------------------------------------------------------
Polarizable Embedding Summary:

   Electrostatics:
      Electronic:                  0.30901227399981
      Nuclear:                     -0.32134940137969
      Multipole:                   0.00000000000000
      Total:                       -0.01233712737988

   Polarization:
      Electronic:                  -0.01817189734325
      Nuclear:                     0.01717961961137
```

```
      Multipole:                    -0.02091890381649
      Total:                        -0.02191118154837

   Total Energy:                    -0.03424830892825
-----------------------------------------------------------------------
```

If a PE-ADC calculation is carried out, the perturbative corrections are printed together with the excitation energies:

```
  Excited state 1 (singlet, A)                                  [converged]
  ------------------------------------------------------------------------
  Term symbol:  2 (1) A                                R^2 =  1.84764e-13

  Total energy:                                  -483.3704138865 a.u.
  Excitation energy:                                    3.906651 eV
  ----------------------------------------
  PE ptSS energy correction:                           -0.001804 eV
  Corrected Excitation Energy (ptSS):                   3.904847 eV
  ----------------------------------------
  PE ptLR energy correction:                           -0.000096 eV
  Corrected Excitation Energy (ptLR):                   3.906554 eV
  ----------------------------------------
```

11.9 Atomic Interactions Represented By Empirical Dispersion (AIRBED)

11.9.1 Introduction

The properties of a molecule can be influenced by its physical environment, this can occur as a direct consequence of the molecule-environment interaction, or indirectly through the geometrical constraints imposed by the environment modifying the molecular structure. Even when there is no chemical bonding between a molecule and its environment, and the interaction is dominated by relatively weak intermolecular forces the effects of this interaction can be significant.

Recently the Atomic Interactions Represented By Empirical Dispersion (AIRBED) approach was introduced.[113] Within this approach, the empirical dispersion correction commonly used in DFT calculations was modified to capture the repulsion at short inter-nuclear distances, in addition to the attractive dispersion interaction with point charges included to account for electrostatic effects. This allows the important components of the interaction between the molecule and environment to be described without the electronic structure of the environment atoms being included with the DFT calculation, and can be viewed as a quantum mechanics/molecular mechanics approach integrated within the DFT calculation that will provide a more consistent treatment of the non-bonded interactions.

The AIRBED approach is based upon the DFT-D2 method[58]

$$E_{\mathrm{DISP}} = -s_6 \sum_{A}^{N} \sum_{B<A}^{N} \frac{C_6^{AB}}{R_{AB}^6} f_{\mathrm{dmp}}(R_{AB}) \tag{11.124}$$

$$C_6^{AB} = (C_6^A C_6^B)^{1/2} \tag{11.125}$$

$$f_{\mathrm{dmp}}(R_{AB}) = [1 + e^{-d(R_{AB}/R_{AB}^0 - 1)}]^{-1} \tag{11.126}$$

where R_{AB} and R^0_{AB} are the internuclear separation and sum of the van der Waals radii of atoms A and B respectively, C_6^{AB} is the dispersion coefficient for atom pair AB, s_6 is a scaling factor and $f_{\rm dmp}(R_{AB})$ is the damping function. In the AIRBED approach, $E_{\rm DISP}$ is replaced by $E_{\rm vdW}$, where $E_{\rm vdW}$ describes the repulsion at short inter-nuclear separations in addition to the dispersion interaction through a modification of the nature of the damping function.

$$E_{\rm vdW} = -s_6 \sum_A^{N_e} \sum_B^{N_m} \frac{C_6^{AB}}{R_{AB}^6} f_{\rm dmp}^{r+d}(R_{AB}) \tag{11.127}$$

$$f_{\rm dmp}^{r+d}(R_{AB}) = 1 - e^{[-d(R_{AB}/R^0_{AB}-1)+\alpha]}. \tag{11.128}$$

Here the system is partitioned into the “molecule" and “environment", with N_e environment atoms and N_m molecule atoms. The R^0_{AB} values used for this contribution to the energy are derived from the experimental van der Waals radii,[12] which tend to be larger than the values used in the standard DFT-D2 corrections. Values of d=20.0 and s_6 are used which are unchanged from DFT-D2. The additional parameter α is introduced to allow some additional flexibility to tune the environment-molecule interaction. Note that this modified interaction is only applied for the interaction between the atoms of the environment and the molecule, and the original, unmodified dispersion correction is used for the interaction between the atoms of the molecule. Furthermore, the atoms of the molecular environment can be assigned an arbitrary charge. The atoms of the environment specified in the *$airbed* block and charges can be assigned using the *$external_charges* block (Section B.1.1.7). Gradients and second-derivatives have been implemented for this model allowing the optimisation of structures and the calculation of harmonic vibrational frequencies.

11.9.2 AIRBED Job Control

AIRBED

Perform an AIRBED calculation.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Perform an AIRBED calculation.

False Don’t perform an AIRBED calculation.

RECOMMENDATION:

Set the *$rem* variable DFT_D to EMPIRICAL_GRIMME.

AIRBED_ALPHA

Sets the value of α.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $\alpha = n/1000$

RECOMMENDATION:

0 or -1200 for hBN surface

Example 11.47 AIRBED geometry optimisation calculation for a CO_2 molecule between two benzene molecules.

```
$molecule
   0 1
   C    -0.0000157292   -0.0001328162   -0.0004579916
   O     1.1694414838   -0.0000944371   -0.0002871420
   O    -1.1694729115   -0.0001468838   -0.0005255697
$end

$rem
   JOBTYPE    opt
   METHOD     b3lyp
   BASIS      6-31G*
   DFT_D      empirical_grimme
   AIRBED     true
$end

$airbed
   C         4.3491833901   -1.1078655146    0.8509307628
   C        -4.3495814364   -1.2912463365   -0.5342957195
   C        -4.3494173551   -0.1830539959   -1.3852617935
   C         4.3489888330    0.1830906311    1.3853863667
   C         4.3492973035    1.2912820357    0.5344194022
   C        -4.3493286755    1.1079015860   -0.8508049578
   C        -4.3491474478   -1.1078704423    0.8509142888
   C         4.3495268718   -1.2912427486   -0.5342791223
   C         4.3493258802   -0.1830511599   -1.3852462591
   C        -4.3488991732    0.1830852343    1.3853711150
   C        -4.3492443156    1.2912773945    0.5344051306
   C         4.3492909042    1.1079049825   -0.8507905472
$end

$external_charges
    4.3491833901   -1.1078655146    0.8509307628   -0.12
   -4.3495814364   -1.2912463365   -0.5342957195   -0.12
   -4.3494173551   -0.1830539959   -1.3852617935   -0.12
    4.3489888330    0.1830906311    1.3853863667   -0.12
    4.3492973035    1.2912820357    0.5344194022   -0.12
   -4.3493286755    1.1079015860   -0.8508049578   -0.12
   -4.3491474478   -1.1078704423    0.8509142888   -0.12
    4.3495268718   -1.2912427486   -0.5342791223   -0.12
    4.3493258802   -0.1830511599   -1.3852462591   -0.12
   -4.3488991732    0.1830852343    1.3853711150   -0.12
   -4.3492443156    1.2912773945    0.5344051306   -0.12
    4.3492909042    1.1079049825   -0.8507905472   -0.12
$end
```

References and Further Reading

[1] The VMD program may be downloaded from http://www.ks.uiuc.edu/Research/vmd.

[2] I. Adamovic and M. S. Gordon. *Mol. Phys.*, 103:379, 2005. DOI: 10.1080/00268970512331317246.

[3] H. Aksu, S. K. Paul, J. M. Herbert, and B. D. Dunietz. *J. Phys. Chem. B*, 124:6998, 2020. DOI: 10.1021/acs.jpcb.0c04032.

[4] O. Andreussi, I. Dabo, and N. Marzari. *J. Chem. Phys.*, 136:064102, 2012. DOI: 10.1063/1.3676407.

[5] K. Baldridge and A. Klamt. *J. Chem. Phys.*, 106:6622, 1997. DOI: 10.1063/1.473662.

[6] V. Barone and M. Cossi. *J. Phys. Chem. A*, 102:1995, 1998. DOI: 10.1021/jp9716997.

[7] V. Barone, M. Cossi, and J. Tomasi. *J. Chem. Phys.*, 107:3210, 1997. DOI: 10.1063/1.474671.

[8] T. J. Bi, L. K. Xu, F. Wang, M. J. Ming, and X. Y. Li. *Phys. Chem. Chem. Phys.*, 19:32242, 2017. DOI: 10.1039/c7cp05673g.

[9] T. J. Bi, L. K. Xu, F. Wang, and X. Y. Li. *Phys. Chem. Chem. Phys.*, 20:13178, 2018. DOI: 10.1039/c8cp00930a.

[10] R. L. Blackbourn and J. T. Hupp. *J. Phys. Chem.*, 94:1788, 1990. DOI: 10.1021/j100368a016.

[11] R. Bonaccorsi, P. Palla, and J. Tomasi. *J. Am. Chem. Soc.*, 106:1945, 1984. DOI: 10.1021/ja00319a008.

[12] A. Bondi. *J. Phys. Chem.*, 68:441, 1964. DOI: 10.1021/j100785a001.

[13] P. E. Bowling, M. Gray, S. K. Paul, and J. M. Herbert. *J. Chem. Theory Comput.*, 21:1722, 2025. DOI: 10.1021/acs.jctc.4c01665.

[14] C. Brandt and J. M. Herbert. Impact of vibrational entropy corrections on enzyme thermochemistry and barrier heights using the SMD solvation model with semi-analytic second derivatives. (in preparation), 2025.

[15] C. M. Breneman and K. B. Wiberg. *J. Comput. Chem.*, 11:361, 1990. DOI: 10.1002/jcc.540110311.

[16] B. R. Brooks, C. L. Brooks III, A. D. Mackerell, Jr., L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, C. Qui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York, and M. Karplus. *J. Comput. Chem.*, 30:1545, 2009. DOI: 10.1002/jcc.21287.

[17] A. D. Buckingham. *Q. Rev. Chem. Soc.*, 13:183, 1959. DOI: 10.1039/QR9591300183.

[18] R. Cammi and B. Mennucci. *J. Chem. Phys.*, 110:9877, 1999. DOI: 10.1063/1.478861.

[19] R. Cammi and J. Tomasi. *Int. J. Quantum Chem. Symp.*, 29:465, 1995. DOI: 10.1002/qua.560560850.

[20] R. Cammi, S. Corni, B. Mennucci, and J. Tomasi. *J. Chem. Phys.*, 122:104513, 2005. DOI: 10.1063/1.1867373.

[21] E. Cancès and B. Mennucci. *J. Chem. Phys.*, 114:4744, 2001. DOI: 10.1063/1.1349091.

[22] E. Cancès, B. Mennucci, and J. Tomasi. *J. Chem. Phys.*, 107:3032, 1997. DOI: 10.1063/1.474659.

[23] M. Caricato, B. Mennucci, J. Tomasi, F. Ingrosso, R. Cammi, S. Corni, and G. Scalmani. *J. Chem. Phys.*, 124: 124520, 2006. DOI: 10.1063/1.2183309.

[24] K. Carter-Fenk and J. M. Herbert. *Phys. Chem. Chem. Phys.*, 22:24870, 2020. DOI: 10.1039/D0CP05039C.

[25] K. Carter-Fenk and J. M. Herbert. *Mol. Phys.*, 121:e2055504, 2023. DOI: 10.1080/00268976.2022.2055504.

[26] C. Chakravarty., H. Aksu, J. A. Martinez B., P. Ramos, M. Pavanello, and B. D. Dunietz. *J. Phys. Chem. Lett.*, 13:4849, 2022. DOI: 10.1021/acs.jpclett.2c00982.

[27] F. Chen and D. M. Chipman. *J. Chem. Phys.*, 119:10289, 2003. DOI: 10.1063/1.1615232.

[28] D. M. Chipman. *J. Chem. Phys.*, 110:8012, 1999. DOI: 10.1063/1.478729.

[29] D. M. Chipman. *J. Chem. Phys.*, 112:5558, 2000. DOI: 10.1063/1.481133.

[30] D. M. Chipman. *Theor. Chem. Acc.*, 107:80, 2002. DOI: 10.1007/s00214-001-0302-1.

[31] D. M. Chipman. *J. Chem. Phys.*, 124:224111, 2006. DOI: 10.1063/1.2203068.

[32] D. M. Chipman and M. Dupuis. *Theor. Chem. Acc.*, 107:90, 2002. DOI: 10.1007/s00214-001-0303-0.

[33] D. Claudino and N. J. Mayhall. *J. Chem. Theory Comput.*, 15:1053, 2019. DOI: 10.1021/acs.jctc.8b01112.

[34] D. Claudino and N. J. Mayhall. *J. Chem. Theory Comput.*, 15:6085, 2019. DOI: 10.1021/acs.jctc.9b00682.

[35] M. P. Coons and J. M. Herbert. *J. Chem. Phys.*, 148:222834, 2018. DOI: 10.1063/1.5023916.

[36] M. P. Coons, Z.-Q. You, and J. M. Herbert. *J. Am. Chem. Soc.*, 138:10879, 2016. DOI: 10.1021/jacs.6b06715.

[37] M. Cossi and V. Barone. *J. Phys. Chem. A*, 104:10614, 2000. DOI: 10.1021/jp000997s.

[38] M. Cossi and V. Barone. *J. Chem. Phys.*, 112:2427, 2000. DOI: 10.1063/1.480808.

[39] M. Cossi and V. Barone. *J. Chem. Phys.*, 115:4708, 2001. DOI: 10.1063/1.1394921.

[40] M. Cossi, B. Mennucci, and R. Cammi. *J. Comput. Chem.*, 17:57, 1996. DOI: 10.1002/(SICI)1096-987X(19960115)17:1<57::AID-JCC6>3.0.CO;2-#.

[41] M. Cossi, N. Rega, G. Scalmani, and V. Barone. *J. Comput. Chem.*, 24:669, 2003. DOI: 10.1002/jcc.10189.

[42] C. J. Cramer and D. G. Truhlar. In M. R. Reddy and M. D. Erion, editors, *Free Energy Calculations and Rational Drug Design*, pages 63–96. Kluwer Academic/Plenum, New York, 2001.

[43] C. J. Cramer and D. G. Truhlar. *Acc. Chem. Res.*, 41:760, 2008. DOI: 10.1021/ar800019z.

[44] C. J. Cramer and D. G. Truhlar. *Acc. Chem. Res.*, 42:493, 2009. DOI: 10.1021/ar900004j.

[45] D. Das, K. P. Eurenius, E. M. Billings, P. Sherwood, D. C. Chatfield, M. Hodoscek, and B. R. Brooks. *J. Chem. Phys.*, 117:10534, 2002. DOI: 10.1063/1.1520134.

[46] P. N. Day, J. H. Jensen, M. S. Gordon, S. P. Webb, W. J. Stevens, M. Krauss, D. Garmer, H. Basch, and D. Cohen. *J. Chem. Phys.*, 105:1968, 1996. DOI: 10.1063/1.472045.

[47] G. Fisicaro, L. Genovese, O. Andreussi, N. Marzari, and S. Goedecker. *J. Chem. Phys.*, 144:014103, 2016. DOI: 10.1063/1.4939125.

[48] J. C. Flick, D. Kosenkov, E. G. Hohenstein, C. D. Sherrill, and L. V. Slipchenko. *J. Chem. Theory Comput.*, 8: 2835, 2012. DOI: 10.1021/ct200673a.

[49] J. Florián and A. Warshel. *J. Phys. Chem. B*, 101:5583, 1997. DOI: 10.1021/jp9705075.

[50] J. Florián and A. Warshel. *J. Phys. Chem. B*, 103:10282, 1999. DOI: 10.1021/jp992041r.

[51] N. Foloppe and A. D. MacKerell. *J. Comput. Chem.*, 21:86, 2000. DOI: 10.1002/(SICI)1096-987X(20000130)21:2<86::AID-JCC2>3.0.CO;2-G.

[52] M. A. Freitag, M. S. Gordon, J. H. Jensen, and W. J. Stevens. *J. Chem. Phys.*, 112:7300, 2000. DOI: 10.1063/1.481370.

[53] D. Ghosh, D. Kosenkov, V. Vanovschi, C. F. Williams, J. M. Herbert, M. S. Gordon, M. W. Schmidt, L. V. Slipchenko, and A. I. Krylov. *J. Phys. Chem. A*, 114:12739, 2010. DOI: 10.1021/jp107557p.

[54] P. J. S. Gomes, C. Serpa, R. M. D. Nunes, L. G. Arnaut, and S. J. Formosinho. *J. Phys. Chem. A*, 114:2778, 2010. DOI: 10.1021/jp9108255.

[55] M. S. Gordon, M. A. Freitag, P. Bandyopadhyay, J. H. Jensen, V. Kairys, and W. J. Stevens. *J. Phys. Chem. A*, 105:293, 2001. DOI: 10.1021/jp002747h.

[56] M. S. Gordon, D. G. Fedorov, S. R. Pruitt, and L. V. Slipchenko. *Chem. Rev.*, 112:632, 2012. DOI: 10.1021/cr200093j.

[57] M. Gray and J. M. Herbert. *J. Phys. Chem. C*, 127:2675, 2023. DOI: 10.1021/acs.jpcc.2c08413.

[58] S. Grimme. *J. Comput. Chem.*, 27:1787, 2006. DOI: 10.1002/jcc.20495.

[59] P. K. Gurunathan, A. Acharya, D. Ghosh, D. Kosenkov, I. Kaliman, Y. Shao, A. I. Krylov, and L. V. Slipchenko. *J. Phys. Chem. B*, 120:6562, 2016. DOI: 10.1021/acs.jpcb.6b04166.

[60] X. He, T. Zhu, X. W. Wang, J. F. Liu, and J. Z. H. Zhang. *Acc. Chem. Res.*, 47:2748, 2014. DOI: 10.1021/ar500077t.

[61] J. M. Herbert. *J. Chem. Phys.*, 151:170901, 2019. DOI: 10.1063/1.5126216.

[62] J. M. Herbert. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 11:e1519, 2021. DOI: 10.1002/wcms.1519.

[63] J. M. Herbert and A. W. Lange. The polarizable continuum model for (bio)molecular electrostatics: Basic theory and recent advances for macromolecules and simulations. In Q. Cui, P. Ren, and M. Meuwly, editors, *Many-Body Effects and Electrostatics in Multi-Scale Computations of Biomolecules*, chapter 11, pages 363–416. Pan Stanford, 2016.

[64] Z. C. Holden, R. M. Richard, and J. M. Herbert. *J. Chem. Phys.*, 139:244108, 2013. DOI: 10.1063/1.4850655.

[65] Z. C. Holden, B. Rana, and J. M. Herbert. *J. Chem. Phys.*, 150:144115, 2019. DOI: 10.1063/1.5089673.

[66] C.-P. Hsu, G. R. Fleming, M. Head-Gordon, and T. Head-Gordon. *J. Chem. Phys.*, 114:3065, 2001. DOI: 10.1063/1.1338531.

[67] W. Humphrey, A. Dalke, and K. Schulten. *J. Molec. Graphics*, 14:33, 1996. DOI: 10.1016/0263-7855(96)00018-5.

[68] H. L. Woodcock III, M. Hodoscek, A. T. B. Gilbert, P. M. W. Gill, H. F. Schaefer III, and B. R. Brooks. *J. Comput. Chem.*, 28:1485, 2007. DOI: 10.1002/jcc.20587.

[69] R. Improta, V. Barone, G. Scalmani, and M. J. Frisch. *J. Chem. Phys.*, 125:054103, 2006. DOI: 10.1063/1.2222364.

[70] R. Improta, G. Scalmani, M. J. Frisch, and V. Barone. *J. Chem. Phys.*, 127:074504, 2007. DOI: 10.1063/1.2757168.

[71] M. F. Iozzi, M. Cossi, R. Improta, N. Rega, and V. Barone. *J. Chem. Phys.*, 124:184103, 2006. DOI: 10.1063/1.2188392.

[72] J. H. Jensen and M. S. Gordon. *Mol. Phys.*, 89:1313, 1996. DOI: 10.1080/00268979609482543.

[73] J. H. Jensen and M. S. Gordon. *J. Chem. Phys.*, 108:4772, 1998. DOI: 10.1063/1.475888.

[74] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives. *J. Am. Chem. Soc.*, 118:11225, 1996. DOI: 10.1021/ja9621760.

[75] I. A. Kaliman and L. V. Slipchenko. *J. Comput. Chem.*, 34:2284, 2013. DOI: 10.1002/jcc.23375.

[76] I. A. Kaliman and L. V. Slipchenko. *J. Comput. Chem.*, 36:129, 2015. DOI: 10.1002/jcc.23772.

[77] C. P. Kelly, C. J. Cramer, and D. G. Truhlar. *J. Chem. Theory Comput.*, 1:1133, 2005. DOI: 10.1021/ct050164b.

[78] C. P. Kelly, C. J. Cramer, and D. G. Truhlar. *J. Phys. Chem. B*, 111:408, 2007. DOI: 10.1021/jp065403l.

[79] J. G. Kirkwood. *J. Chem. Phys.*, 2:767, 1934. DOI: 10.1063/1.1749393.

[80] A. Klamt and V. Jonas. *J. Chem. Phys.*, 105:9972, 1996. DOI: 10.1063/1.472829.

[81] A. Klamt and G. Schüürmann. *J. Chem. Soc. Perkin Trans. 2*, page 799, 1993. DOI: 10.1039/P29930000799.

[82] A. Klamt, B. Mennucci, J. Tomasi, V. Barone, C. Curutchet, M. Orozco, and F. J. Luque. *Acc. Chem. Res.*, 42: 489, 2009. DOI: 10.1021/ar800187p.

[83] A. Klamt, C. Moya, and J. Palomar. *J. Chem. Theory Comput.*, 11:4220, 2015. DOI: 10.1021/acs.jctc.5b00601.

[84] D. Kosenkov and L. V. Slipchenko. *J. Phys. Chem. A*, 115:392, 2011. DOI: 10.1021/jp110026c.

[85] A. W. Lange and J. M. Herbert. *J. Phys. Chem. Lett.*, 1:556, 2010. DOI: 10.1021/jz900282c.

[86] A. W. Lange and J. M. Herbert. *J. Chem. Phys.*, 133:244111, 2010. DOI: 10.1063/1.3511297.

[87] A. W. Lange and J. M. Herbert. *J. Chem. Phys.*, 134:204110, 2011. DOI: 10.1063/1.3592372.

[88] A. W. Lange and J. M. Herbert. *Chem. Phys. Lett.*, 509:77, 2011. DOI: 10.1016/j.cplett.2011.04.092.

[89] A. W. Lange, J. M. Herbert, B. J. Albrecht, and Z.-Q. You. *Mol. Phys.*, 118:e1644384, 2020. DOI: 10.1080/00268976.2019.1644384.

[90] V. I. Lebedev. *Zh. Vychisl. Mat. Mat. Fix.*, 16:293, 1976. DOI: 10.1016/0041-5553(76)90100-2.

[91] V. I. Lebedev. *Sibirsk. Mat. Zh.*, 18:132, 1977.

[92] V. I. Lebedev and D. N. Laikov. *Dokl. Math.*, 366:741, 1999.

[93] S. J. R. Lee, M. Welborn, F. R. Manby, and T. F. Miller III. *Acc. Chem. Res.*, 52:1359, 2019. DOI: 10.1021/acs.accounts.8b00672.

[94] M. A. Leontovich. *Introduction to Thermodynamics. Statistical Physics*, volume 28. Nauka, Moscow, 1983.

[95] H. Li and J. H. Jensen. *J. Comput. Chem.*, 25:1449, 2004. DOI: 10.1002/jcc.20072.

[96] J. Li, C. J. Cramer, and D. G. Truhlar. *Int. J. Quantum Chem.*, 77:264, 2000. DOI: 10.1002/(SICI)1097-461X(2000)77:1<264::AID-QUA24>3.0.CO;2-J.

[97] P. Li, H. Johnston, and R. Krasny. *J. Comput. Phys.*, 228:3858, 2009. DOI: 10.1016/j.jcp.2009.02.022.

[98] X. Y. Li. *Int. J. Quantum Chem.*, 115:700, 2015. DOI: 10.1002/qua.24901.

[99] X. Y. Li, Q. Zhu, F. C. He, and K. X. Fu. Extension of classical thermodynamics to nonequilibrium polarization. In M. Tadashi, editor, *Thermodynamics*, pages 205–232. InTech, 2011. DOI: 10.5772/13695.

[100] D. A. Liotard, G. D. Hawkins, G. C. Lynch, C. J. Cramer, and D. G. Truhlar. *J. Comput. Chem.*, 16:422, 1995. DOI: 10.1002/jcc.540160405.

[101] N. H. List, J. M. H. Olsen, and J. Kongsted. *Phys. Chem. Chem. Phys.*, 18:20234, 2016. DOI: 10.1039/C6CP03834D.

[102] J. Liu and W. Liang. *J. Chem. Phys.*, 138:024101, 2013. DOI: 10.1063/1.4773397.

[103] A. D. Mackerell, Jr., D. Bashford, M. Bellott, R. L. Dunbrack, Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph-McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher, III, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Wantanabe, J. Wiórkiewicz-Kuczera, D. Yin, and M. Karplus. *J. Phys. Chem. B*, 102:3586, 1998. DOI: 10.1021/jp973084f.

[104] F. R. Manby, M. Stella, J. D. Goodpaster, and T. F. Miller III. *J. Chem. Theory Comput.*, 8:2564, 2012. DOI: 10.1021/ct300544e.

[105] M. Mantina, A. C. Chamberlin, R. Valero, C. J. Cramer, and D. G. Truhlar. *J. Phys. Chem. A*, 113:5806, 2009. DOI: 10.1021/jp8111556.

[106] R. A. Marcus. *J. Chem. Phys.*, 24:966, 1956. DOI: 10.1063/1.1742723.

[107] R. A. Marcus. *J. Chem. Phys.*, 24:979, 1956. DOI: 10.1063/1.1742724.

[108] A. V. Marenich, R. M. Olson, C. P. Kelly, C. J. Cramer, and D. G. Truhlar. *J. Chem. Theory Comput.*, 3:2011, 2007. DOI: 10.1021/ct7001418.

[109] A. V. Marenich, C. J. Cramer, and D. G. Truhlar. *J. Phys. Chem. B*, 113:6378, 2009. DOI: 10.1021/jp810292n.

[110] A. V. Marenich, C. J. Cramer, D. G. Truhlar, C. A. Guido, B. Mennucci, G. Scalmani, and M. J. Frisch. *Chem. Sci.*, 2:2143, 2011. DOI: 10.1039/c1sc00313e.

[111] A. V. Marenich, S. V. Jerome, C. J. Cramer, and D. G. Truhlar. *J. Chem. Theory Comput.*, 8:527, 2012. DOI: 10.1021/ct200866d.

[112] A. V. Marenich, C. J. Cramer, and D. G. Truhlar. *J. Chem. Theory Comput.*, 9:609, 2013. DOI: 10.1021/ct300900e.

[113] S. E. Mason, P. H. Beton, and N. A. Besley. *J. Chem. Theory Comput.*, 15:5628, 2018. DOI: 10.1021/acs.jctc.9b00576.

[114] J.-M. Mewes, Z.-Q. You, M. Wormit, T. Kriesche, J. M. Herbert, and A. Dreuw. *J. Phys. Chem. A*, 119:5446, 2015. DOI: 10.1021/jp511163y.

[115] J.-M. Mewes, J. M. Herbert, and A. Dreuw. *Phys. Chem. Chem. Phys.*, 19:1644, 2017. DOI: 10.1039/C6CP05986D.

[116] S. Miertuš, E. Scrocco, and J. Tomasi. *Chem. Phys.*, 55:117, 1981. DOI: 10.1016/0301-0104(81)85090-2.

[117] K. V. Mikkelsen, H. Ågren, H. J. A. Jensen, and T. Helgaker. *J. Chem. Phys.*, 89:3086, 1988. DOI: 10.1063/1.454965.

[118] K. Nam, J. Gao, and D. M. York. *J. Chem. Theory Comput.*, 1:2, 2005. DOI: 10.1021/ct049941i.

[119] J. Neugebauer. *Phys. Rep.*, 489:1, 2010. DOI: 10.1016/j.physrep.2009.12.001.

[120] J. M. H. Olsen and J. Kongsted. *Adv. Quantum Chem.*, 61:107, 2011. DOI: 10.1016/B978-0-12-386013-2.00003-6.

[121] J. M. H. Olsen, K. Aidas, and J. Kongsted. *J. Chem. Theory Comput.*, 6:3721, 2010. DOI: 10.1021/ct1003803.

[122] R. M. Olson, A. V. Marenich, C. J. Cramer, and D. G. Truhlar. *J. Chem. Theory Comput.*, 3:2046, 2007. DOI: 10.1021/ct7001607.

[123] J. L. Pascual-Ahuir, E. Silla, and I. Tu non. *J. Comput. Chem.*, 15:1127, 1994. DOI: 10.1002/jcc.540151009.

[124] S. K. Paul and J. M. Herbert. *J. Am. Chem. Soc.*, 143:10189, 2021. DOI: 10.26434/chemrxiv.14273651.v1.

[125] S. K. Paul, M. P. Coons, and J. M. Herbert. *J. Chem. Phys.*, 151:189901, 2019. DOI: 10.1063/1.5132808.

[126] J. Pipek and P. G. Mezey. *J. Chem. Phys.*, 90:4916, 1989. DOI: 10.1063/1.456588.

[127] J. R. Pliego Jr. and J. M. Riveros. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 10:e1440, 2020. DOI: 10.1002/wcms.1440.

[128] A. Pomogaeva and D. M. Chipman. *J. Chem. Theory Comput.*, 7:3952, 2011. DOI: 10.1021/ct200575c.

[129] A. Pomogaeva and D. M. Chipman. *J. Phys. Chem. A*, 117:5812, 2013. DOI: 10.1021/jp404624x.

[130] A. Pomogaeva and D. M. Chipman. *J. Chem. Theory Comput.*, 10:211, 2014. DOI: 10.1021/ct400894j.

[131] A. Pomogaeva and D. M. Chipman. *J. Phys. Chem. A*, 119:5173, 2015. DOI: 10.1021/jp5098519.

[132] S. Prager, A. Zech, F. Aquilante, A. Dreuw, and T. A. Wesolowski. *J. Chem. Phys.*, 144:204103, 2016. DOI: 10.1063/1.4948741.

[133] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard III, and W. M. Skiff. *J. Am. Chem. Soc.*, 114:10024, 1992. DOI: 10.1021/ja00051a040.

[134] H. S. Ren, M. J. Ming, J. Y. Ma, and X. Y. Li. *J. Phys. Chem. A*, 117:8017, 2013. DOI: 10.1021/jp4046935.

[135] P. Ren and J. W. Ponder. *J. Phys. Chem. B*, 107:5933, 2003. DOI: 10.1021/jp027815+.

[136] N. Ricardi, A. Zech, Y. Gimbal-Zofka, and T. A. Wesolowski. *Phys. Chem. Chem. Phys.*, 20:26053, 2018. DOI: 10.1039/c8cp05634j.

[137] D. Riccardi, P. Schaefer, and Q. Cui. *J. Phys. Chem. B*, 109:17715, 2005. DOI: 10.1021/jp0517192.

[138] C. I. V. Rojas and L. V. Slipchenko. *J. Chem. Theory Comput.*, 16:6408, 2020. DOI: 10.1021/acs.jctc.9b01156.

[139] R. S. Rowland and R. Taylor. *J. Phys. Chem.*, 100:7384, 1996. DOI: 10.1021/jp953141+.

[140] A. Schäfer, A. Klamt, D. Sattle, J. C. W. Lohrenz, and F. Eckert. *Phys. Chem. Chem. Phys.*, 2:2187, 2000. DOI: 10.1039/B000184H.

[141] M. Scheurer, M. F. Herbst, P. Reinholdt, J. M. H. Olsen, A. Dreuw, and J. Kongsted. *J. Chem. Theory Comput.*, 14:4870, 2018. DOI: 10.1021/acs.jctc.8b00576.

[142] M. W. Schmidt, K. K. Baldridge, J. A. Boatz, S. T. Elbert, M. S. Gordon, J. H. Jensen, S. Koseki, N. Matsunaga, K. A. Nguyen, S. Su, T. L. Windus, M. Dupuis, and J. A. Montgomery, Jr. *J. Comput. Chem.*, 14:1347, 1983. DOI: 10.1002/jcc.540141112.

[143] R. Sen, A. Dreuw, and S. Faraji. *Phys. Chem. Chem. Phys.*, 21:3683, 2019. DOI: 10.1039/C8CP06527F.

[144] H. M. Senn and W. Thiel. QM/MM methods for biological systems. In M. Reiher, editor, *Atomistic Approaches in Modern Biology: From Quantum Chemistry to Molecular Simulations*, volume 268 of *Topics in Current Chemistry*, pages 173–290. Springer-Verlag, Berlin, 2007. DOI: 10.1007/128_2006_084.

[145] Y. Shao and J. Kong. *J. Phys. Chem. A*, 111:3661, 2007. DOI: 10.1021/jp067307q.

[146] D. Si and H. Li. *J. Chem. Phys.*, 131:044123, 2009. DOI: 10.1063/1.3187527.

[147] U. C. Singh and P. A. Kollman. *J. Comput. Chem.*, 5:129, 1984. DOI: 10.1002/jcc.540050204.

[148] U. C. Singh and P. A. Kollman. *J. Comput. Chem.*, 7:718, 1986. DOI: 10.1002/jcc.540070604.

[149] L. V. Slipchenko. *J. Phys. Chem. A*, 114:8824, 2010. DOI: 10.1021/jp101797a.

[150] L. V. Slipchenko. *J. Phys. Chem. A*, 128:656, 2022. DOI: 10.1021/acs.jpca.3c06194.

[151] L. V. Slipchenko and M. S. Gordon. *J. Comput. Chem.*, 28:276, 2007. DOI: 10.1002/jcc.20520.

[152] L. V. Slipchenko and M. S. Gordon. *Mol. Phys.*, 107:999, 2009. DOI: 10.1080/00268970802712449.

[153] C. J. Stein, J. M. Herbert, and M. Head-Gordon. *J. Chem. Phys.*, 151:224111, 2019. DOI: 10.1063/1.5131020.

[154] C. Steinmann, P. Reinholdt, M. S. Nørby, J. Kongsted, and J. M. H. Olsen. *Int. J. Quantum Chem.*, 119:1, 2019. DOI: 10.1002/qua.25717.

[155] A. J. Stone. *Chem. Phys. Lett.*, 83:233, 1981. DOI: 10.1016/0009-2614(81)85452-8.

[156] A. J. Stone and M. Alderton. *Mol. Phys.*, 56:1047, 1985. DOI: 10.1021/ct050190+.

[157] K. T. Tang and J. P. Toennies. *J. Chem. Phys.*, 80:3726, 1984. DOI: 10.1063/1.447150.

[158] J. D. Thompson, C. J. Cramer, and D. G. Truhlar. *J. Chem. Phys.*, 119:1661, 2003. DOI: 10.1063/1.1579474.

[159] J. Tomasi and M. Persico. *Chem. Rev.*, 94:2027, 1994. DOI: 10.1021/cr00031a013.

[160] J. Tomasi, B. Mennucci, and E. Cancès. *J. Mol. Struct. (Theochem)*, 464:211, 1999. DOI: 10.1016/S0166-1280(98)00553-3.

[161] J. Tomasi, B. Mennucci, and R. Cammi. *Chem. Rev.*, 106:2999, 2005. DOI: 10.1021/cr9904009.

[162] T. N. Truong and E. V. Stefanovich. *Chem. Phys. Lett.*, 240:253, 1995. DOI: 10.1016/0009-2614(95)00541-B.

[163] T. Vreven and K. Morokuma. *Annu. Rep. Comput. Chem.*, 2:35, 2006. DOI: 10.1016/S1574-1400(06)02003-2.

[164] O. A. Vydrov and T. Van Voorhis. *Phys. Rev. Lett.*, 103:063004, 2009. DOI: 10.1103/PhysRevLett.103.063004.

[165] R. C. Walker, M. F. Crowley, and D. A. Case. *J. Comput. Chem.*, 29:1019, 2008. DOI: 10.1002/jcc.20857.

[166] J. Wang, P. Cieplak, and P. A. Kollman. *J. Comput. Chem.*, 21:1049, 2000. DOI: 10.1002/1096-987X(200009)21:12<1049::AID-JCC3>3.0.CO;2-F.

[167] J.-B. Wang, J.-Y. Ma, and X.-Y. Li. *Phys. Chem. Chem. Phys.*, 12:207, 2010. DOI: 10.1039/B914652K.

[168] S. J. Weiner, P. A. Kollman, D. T. Nguyen, and D. A. Case. *J. Comput. Chem.*, 7:230, 1986. DOI: 10.1002/jcc.540070216.

[169] T. A. Wesolowski. *Phys. Rev. A*, 77:012504, 2008. DOI: 10.1103/PhysRevA.77.012504.

[170] T. A. Wesolowski and A. Warshel. *J. Phys. Chem.*, 97:8050, 1993. DOI: 10.1021/j100132a040.

[171] C. Wohlfarth. volume 6 of *Landolt-Börnstein, New Series IV*. Springer Science + Business Media, 1991. DOI: 10.1007/b44266.

[172] C. Wohlfarth and B. Wohlfarth. *Refractive Indices of Organic Liquids*, volume 38B of *Group III Condensed Matter*. Landolt-Börnstein, Springer-Verlag, Berlin, 1996. DOI: 10.1007/b85533.

[173] M. W. Wong, M. J. Frisch, and K. B. Wiberg. *J. Am. Chem. Soc.*, 113:4776, 1991. DOI: 10.1021/ja00013a010.

[174] H. L. Woodcock, W. Zheng, A. Ghysels, Y. Shao, J. Kong, and B. R. Brooks. *J. Chem. Phys.*, 129:214109, 2008. DOI: 10.1063/1.3013558.

[175] H. W. Wu, H. S. Ren, Q. Zhu, J. Y. Ma, and X. Y. Li. *Phys. Chem. Chem. Phys.*, 14:5538, 2012. DOI: 10.1039/c2cp23759h.

[176] D. M. York and M. Karplus. *J. Phys. Chem. A*, 103:11060, 1999. DOI: 10.1021/jp9920971.

[177] Z.-Q. You and J. M. Herbert. *J. Chem. Theory Comput.*, 12:4338, 2016. DOI: 10.1021/acs.jctc.6b00644.

[178] Z.-Q. You, J.-M. Mewes, A. Dreuw, and J. M. Herbert. *J. Chem. Phys.*, 143:204104, 2015. DOI: 10.1063/1.4936357.

[179] A. Zech, F. Aquilante, and T. A. Wesolowski. *J. Chem. Phys.*, 143:164106, 2015. DOI: 10.1063/1.4933372.

[180] D. W. Zhang and J. Z. H. Zhang. *J. Chem. Phys.*, 119:3599, 2003. DOI: 10.1063/1.1591727.

[181] D. W. Zhang, X. H. Chen, and J. Z. H. Zhang. *J. Comput. Chem.*, 24:1846, 2003. DOI: 10.1002/jcc.10346.

[182] T. Zhu, J. Li, D. A. Liotard, C. J. Cramer, and D. G. Truhlar. *J. Chem. Phys.*, 110:5503, 1999. DOI: 10.1063/1.478447.

Chapter 12

Fragment-Based Methods

12.1 Introduction

Molecular complexes and molecular clusters represent a broad class of systems with interesting chemical and physical properties. Such systems can be naturally partitioned into fragments each representing a molecule or several molecules. Q-CHEM contains a set of methods designed to use such partitioning either for physical or computational advantage. Some of these methods (*e.g.* the ALMO-EDA method and its most recent updates/extensions) were developed and implemented by Dr. Rustam Z. Khaliullin, Dr. Paul R. Horn, Dr. Yuezhi Mao, Dr. Jonathan Thirman, Dr. Daniel S. Levine, Dr. Qinghui Ge, and Matthias Loipersberger working with Prof. Martin Head-Gordon at the University of California–Berkeley. Other methods [*e.g.*, the XSAPT family of methods and TDDFT(MI)] were developed by Drs. Leif Jacobson, Ka Un Lao, and Jie Liu working with Prof. John Herbert at Ohio State University.

The list of methods that use partitioning includes:

- Initial guess at the MOs as a superposition of the converged MOs on the isolated fragments (FRAGMO guess).[52]
- Constrained (locally-projected) SCF methods for molecular interactions (SCF MI methods) between both closed-shell[52] and open-shell[40] fragments.
- Single Roothaan-step (RS) correction methods that improve FRAGMO and SCF MI description of molecular systems.[40,52]
- Automated calculation of the BSSE with counterpoise correction method (full SCF and RS implementation).
- The original version the ALMO-EDA method (energy decomposition analysis based on absolutely localized molecular orbitals), including the associated charge transfer analysis,[40,53,54] and the analysis of intermolecular bonding in terms of complementary occupied-virtual pairs (COVPs).[40,54,55]
- The second-generation ALMO-EDA method,[38,41,42,81,83] including its extension to single-bond interactions[67–69] and the ALMO-EDA(solv) scheme[84] for the inclusion of implicit solvents in EDA calculation.
- The adiabatic ALMO-EDA method that analyzes the effects intermolecular interactions on molecular properties.[78,80]
- An extension of the ALMO-EDA to RI-MP2.[77,110,111]
- An extension of the ALMO-EDA to intermolecular interactions involving excited-state molecules (calculated by CIS or TDDFT/TDA).[20,22]

- The variational explicit polarization (XPol) method, a self-consistent, charge-embedded, monomer-based SCF calculation.[31,45,118]
- Symmetry-adapted perturbation theory (SAPT), a monomer-based method for computing intermolecular interaction energies and decomposing them into physically-meaningful components.[48,109]
- XPol+SAPT (XSAPT), which extends the SAPT methodology to systems consisting of more than two monomers.[31,45,46]
- Closed- and open-shell AO-XSAPT(KS)+D, a dispersion-corrected version of XSAPT in atomic orbital basis that affords accurate intermolecular interaction energies at very low cost.[59,60,62]
- A stable and physically-motivated energy decomposition approach, SAPT/cDFT, in which cDFT is used to define the charge-transfer component of the interaction energy and SAPT defines the electrostatic, polarization, Pauli repulsion, and van der Waals contributions.[63]
- The electrostatically-embedded many-body expansion[17,66,101,102] and the fragment molecular orbital method,[18,56] for decomposing large clusters into small numbers of monomers, facilitating larger calculations.
- The *Ab Initio* Frenkel Davydov Model,[90,93] a low-order scaling, highly parallelizable approach to computing excited state properties of liquids, crystals, and aggregates.
- TDDFT for molecular interactions [TDDFT(MI)], an excited-state extension of SCF MI that offers a reduced-cost way to compute excited states in molecular clusters, crystals, and aggregates.[32,72,73]
- The ALMO-CIS and ALMO-CIS+CT models (also applicable to TDDFT/TDA) for computing a substantial number of excited states in large molecular clusters.[15,21]

Other fragment-based approaches in Q-CHEM include:

- The Effective Fragment Potential (EFP) method[23] developed by Prof. Lyudmila Slipchenko at Purdue University and Prof. Anna Krylov at USC (see Section 11.5)
- Fragment-based approaches to diabatic states and electronic couplings (see Section 10.14.3)

12.2 Specifying Fragments in the *$molecule* Section

To request any of the methods mentioned above one must specify how system is partitioned into fragments. All atoms and all electrons in the systems should be assigned to a fragment. Each fragment must contain an integer number of electrons. In the current implementation, both open and closed-shell fragments are allowed. In order to specify fragments, the fragment descriptors must be inserted into the *$molecule* section of the Q-CHEM input file. A fragment descriptor consists of two lines: the first line must start with two hyphens followed by optional comments, the second line must contain the charge and the multiplicity of the fragment. At least two fragments must be specified. Fragment

descriptors in the *$molecule* section does not affect jobs that are not designed to use fragmentation.

Example 12.12.1 Fragment descriptors in the *$molecule* section.

```
$molecule
   0 1
-- water molecule - proton donor
   0 1
   O1
   H2 O1 0.96
   H3 O1 0.96 H2 105.4
-- water molecule - proton acceptor
   0 1
   O4 O1 ROO  H2 105.4 H3   0.0
   X5 O4 2.00 O1 120.0 H2 180.0
   H6 O4 0.96 X5  55.6 O1  90.0
   H7 O4 0.96 X5  55.6 01 -90.0

   ROO = 2.4
$end
```

Open shell systems must have a number of alpha electrons greater than the number of beta electrons. However, individual fragments in the system can be made to contain excess beta electrons by specifying a negative multiplicity. For instance, a multiplicity of -2 indicates one excess beta electron, as in the second fragment of the following example.

Example 12.12.2 Open shell fragment descriptors in the *$molecule* section. The second fragment is made with a negative multiplicity, so that overall the number of alpha and beta electrons match, yielding an approximate singlet state.

```
$molecule
   0 1
--  An alpha spin H atom
   0 2
   H1
--  A beta spin H atom
   0 -2
   H2 H1  1.50
$end
```

12.3 FRAGMO Initial Guess for SCF Methods

An accurate initial guess can be generated for molecular systems by superimposing converged molecular orbitals on isolated fragments. This initial guess is requested by specifying FRAGMO option for SCF_GUESS keyword and can be used for both the conventional SCF methods and the locally-projected SCF methods. The number of SCF iterations can be greatly reduced when FRAGMO is used instead of SAD. This can lead to significant time savings for jobs on multi-fragment systems with large basis sets.[53] Unlike the SAD guess, the FRAGMO guess is idempotent.

To converge molecular orbitals on isolated fragments, a child Q-CHEM job is executed for each fragment. *$rem* variables of the child jobs are inherited from the *$rem* section of the parent job. If SCF_PRINT_FRGM is set to TRUE the output of the child jobs is redirected to the output file of the parent job. Otherwise, the output is suppressed.

Additional keywords that control child Q-CHEM processes can be set in the *$rem_frgm* section of the parent input file. This section has the same structure as the *$rem* section. Options in the *$rem_frgm* section override options of the parent job. *$rem_frgm* is intended to specify keywords that control the SCF routine on isolated fragments. Please be careful with the keywords in *$rem_frgm* section. *$rem* variables FRGM_METHOD, FRGM_LPCORR, JOBTYPE, BASIS,

PURECART, ECP are not allowed in *$rem_frgm* and will be ignored. *$rem* variables FRGM_METHOD, FRGM_LPCORR, JOBTYPE, and SCF_GUESS are not inherited from the parent job.

The use of FRAGMO guess is also supported when GEN_SCFMAN = TRUE. It is extended to support more SCF orbital types (R/U/RO/G). Meanwhile, users are allowed to read in the previously generated FRAGMO guess instead of recalculating them if there is no difference between these jobs on the fragment level. This can be particularly useful for scenarios such as scanning a potential energy curve for an intermolecular complex, or for restarting an EDA job. This is controlled by the *$rem* variable FRAGMO_GUESS_MODE.

FRAGMO_GUESS_MODE

Decide what to do regarding the FRAGMO guess in the present job (for gen_scfman only)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Spawn fragment jobs sequentially and collect the results as the FRAGMO guess at the end.

1 Generate fragment inputs in folders “FrgX" under the scratch directory of the present job and then terminate. Users can then take advantage of a queuing system to run these jobs simultaneously using “FrgX" as their scratch folders (should be handled with scripting).

2 Read in the available fragment data.

RECOMMENDATION:

Consider using “1" if the fragment calculations are evenly expensive. Use “2" when FRAGMO guess is pre-computed.

Example 12.3 FRAGMO guess can be used with the conventional SCF calculations. *$rem_frgm* keywords in this example specify that the SCF on isolated fragments does not have to be converged tightly. See also Example 12.2 for an open-shell fragment example.

```
$molecule
0 1
--
0 1
O          -0.106357    0.087598    0.127176
H           0.851108    0.072355    0.136719
H          -0.337031    1.005310    0.106947
--
0 1
O           2.701100   -0.077292   -0.273980
H           3.278147   -0.563291    0.297560
H           2.693451   -0.568936   -1.095771
--
0 1
O           2.271787   -1.668771   -2.587410
H           1.328156   -1.800266   -2.490761
H           2.384794   -1.339543   -3.467573
--
0 1
O          -0.518887   -1.685783   -2.053795
H          -0.969013   -2.442055   -1.705471
H          -0.524180   -1.044938   -1.342263
$end

$rem
   METHOD          EDF1
   BASIS           6-31(2+,2+)g(df,pd)
   SCF_GUESS       FRAGMO
   SCF_PRINT_FRGM  FALSE
$end

$rem_frgm
   SCF_CONVERGENCE 2
$end
```

Example 12.12.4 FRAGMO guess for ROSCF calculation in GEN_SCFMAN. The first fragment is RO and the second fragment is close-shell, while the super-system is computed with RO as well. The complex in the second job has a modified inter-fragment distance so it can make use of the FRAGMO guess generated by the first job. Note that ROSCF = TRUE is needed to treat the fragments and the supersystem consistently.

View input online

12.4 Automated Evaluation of Counterpoise Correction

Evaluation of the basis set superposition error (BSSE) can be automated in Q-CHEM. To calculate BSSE-corrected binding energies, specify fragments in the *$molecule* section and set JOBTYPE to BSSE. The BSSE jobs are not limited to the SCF energies and can be evaluated for multi-fragment systems at any level of theory. Q-CHEM separates the system into fragments as specified in the *$molecule* section and performs a series of jobs on (a) each isolated fragment, (b) each isolated fragment with the remaining atoms in the system replaced by the ghost atoms, and (c) the entire system. Q-CHEM saves all calculated energies and prints out the uncorrected and the counterpoise (CP)-corrected

interaction energies. The *$rem_frgm* section can be used to control calculations on fragments, however, make sure that the fragments and the entire system are treated equally. It means that all numerical methods and convergence thresholds that affect the final energies (such as SCF_CONVERGENCE, THRESH, PURECART, XC_GRID) should be the same for the fragments and for the entire system. Avoid using *$rem_frgm* in the BSSE jobs unless absolutely necessary.

Example 12.5 Evaluation of counterpoise-corrected intermolecular interaction energy for the water trimer at the (i) B3LYP-D3(BJ)/6-31+G(d), (ii) MP2/6-31++G(d,p) levels

```
$molecule
 0 1
 --
 0 1
 O          -0.089523    0.063946    0.086866
 H           0.864783    0.058339    0.103755
 H          -0.329829    0.979459    0.078369
 --
 0 1
 O           2.632273   -0.313504   -0.750376
 H           3.268182   -0.937310   -0.431464
 H           2.184198   -0.753305   -1.469059
 --
 0 1
 O           0.475471   -1.428200   -2.307836
 H          -0.011373   -0.970411   -1.626285
 H           0.151826   -2.317118   -2.289289
 $end

 $rem
 JOBTYPE          BSSE
 METHOD           B3LYP
 DFT_D            D3_BJ
 BASIS            6-31+G(d)
 SCF_CONVERGENCE 8
 INTEGRAL_SYMMETRY    FALSE
 POINT_GROUP_SYMMETRY FALSE
$end

@@@

$molecule
 read
$end

$rem
 JOBTYPE          BSSE
 METHOD           MP2
 BASIS            6-31++G(d,p)
 SCF_CONVERGENCE 8
integral_symmetry FALSE
point_group_symmetry False
$end
```

12.5 Locally-Projected SCF and First-Generation ALMO-EDA Methods

12.5.1 Locally-Projected SCF

Constrained locally-projected SCF is an efficient method for removing the SCF diagonalization bottleneck in calculations for systems of weakly interacting components such as molecular clusters and molecular complexes.[40,52] The

method is based on the equations of the locally-projected SCF for molecular interactions (SCF-MI).[24,40,52,94,108] In the SCF-MI method, the occupied molecular orbitals on a fragment can be expanded only in terms of the atomic orbitals of the same fragment. Such constraints produce non-orthogonal MOs that are localized on fragments and are called absolutely-localized molecular orbitals (ALMOs). The ALMO approximation excludes charge-transfer from one fragment to another. It also prevents electrons on one fragment from borrowing the atomic orbitals of other fragments to compensate for incompleteness of their own AOs and, therefore, removes the BSSE from the interfragment binding energies. The locally-projected SCF methods perform an iterative minimization of the SCF energy with respect to the ALMOs coefficients. The convergence of the algorithm is accelerated with the locally-projected modification of the DIIS extrapolation method.[52]

The ALMO approximation significantly reduces the number of variational degrees of freedom of the wave function. The computational advantage of the locally-projected SCF methods over the conventional SCF method grows with both basis set size and number of fragments. Although still cubic scaling, SCF-MI effectively removes the diagonalization step as a bottleneck in these calculations, because it contains such a small prefactor. In the current implementation, the SCF-MI methods do not speed up the evaluation of the Fock matrix and, therefore, do not perform significantly better than the conventional SCF in the calculations dominated by the Fock build.

Two locally-projected schemes are implemented. One is based on the locally-projected equations of Stoll *et al.*,[108] the other uses the locally-projected equations of Gianinetti *et al.*.[24] These methods have comparable performance. The Stoll iteration is only slightly faster than the Gianinetti iteration but the Stoll equations might be a little bit harder to converge. The Stoll equations also produce ALMOs that are orthogonal within a fragment. The type of the locally-projected SCF calculations is requested by specifying either STOLL or GIA for the FRGM_METHOD keyword.

Example 12.6 Locally-projected SCF method of Stoll

```
$molecule
   0 1
   --
   -1 1
   B        0.068635     0.164710     0.123580
   F       -1.197609     0.568437    -0.412655
   F        0.139421    -1.260255    -0.022586
   F        1.118151     0.800969    -0.486494
   F        0.017532     0.431309     1.531508
   --
   +1 1
   N       -2.132381    -1.230625     1.436633
   H       -1.523820    -1.918931     0.977471
   H       -2.381590    -0.543695     0.713005
   H       -1.541511    -0.726505     2.109346
   H       -2.948798    -1.657993     1.873482
$end

$rem
   METHOD           BP86
   BASIS            6-31(+,+)G(d,p)
   FRGM_METHOD      STOLL
$end

$rem_frgm
   SCF_CONVERGENCE 2
   THRESH           5
$end
```

12.5.2 Roothaan-Step Corrections and FRAGMO Initial Guess

Locally-projected SCF cannot quantitatively reproduce the full SCF intermolecular interaction energies for systems with significant charge-transfer between the fragments (*e.g.*, hydrogen bonding energies in water clusters). Good accuracy in the intermolecular binding energies can be achieved if the locally-projected SCF-MI iteration scheme is combined with a charge-transfer perturbative correction.[52] To account for charge-transfer, one diagonalization of the full Fock matrix is performed after the locally-projected SCF equations are converged and the final energy is calculated as infinite-order perturbative correction to the locally-projected SCF energy. This procedure is known as single Roothaan-step (RS) correction.[52,70,71] It is performed if FRGM_LPCORR is set to RS. To speed up evaluation of the charge-transfer correction, second-order perturbative correction to the energy can be evaluated by solving the linearized single-excitation amplitude equations. This algorithm is called the approximate Roothaan-step correction and can be requested by setting FRGM_LPCORR to ARS.

Both ARS and RS corrected energies are very close to the full SCF energy for systems of weakly interacting fragments but are less computationally expensive than the full SCF calculations. To test the accuracy of the ARS and RS methods, the full SCF calculation can be done in the same job with the perturbative correction by setting FRGM_LPCORR to RS_EXACT_SCF or to ARS_EXACT_SCF. It is also possible to evaluate only the full SCF correction by setting FRGM_LPCORR to EXACT_SCF.

The iterative solution of the linear single-excitation amplitude equations in the ARS method is controlled by a set of NVO keywords described below.

Restrictions. Only single point HF and DFT energies can be evaluated with the locally-projected methods. Geometry optimization can be performed using numerical gradients. Wave function correlation methods (MP2, CC, *etc.*.) are not implemented for the absolutely-localized molecular orbitals. SCF_ALGORITHM cannot be set to anything but DIIS, however, all SCF convergence algorithms can be used on isolated fragments (set SCF_ALGORITHM in the *$rem_frgm*

section).

Example 12.7 Comparison between the RS corrected energies and the conventional SCF energies can be made by calculating both energies in a single run.

```
$molecule
   0 1
   --
   0 1
   O      -1.56875     0.11876     0.00000
   H      -1.90909    -0.78106     0.00000
   H      -0.60363     0.02937     0.00000
   --
   0 1
   O       1.33393    -0.05433     0.00000
   H       1.77383     0.32710    -0.76814
   H       1.77383     0.32710     0.76814
$end

$rem
   METHOD           HF
   BASIS            AUG-CC-PVTZ
   FRGM_METHOD      GIA
   FRGM_LPCORR      RS_EXACT_SCF
$end

$rem_frgm
   SCF_CONVERGENCE 2
   THRESH          5
$end
```

For some systems good accuracy for the intermolecular interaction energies can be achieved without converging SCF-MI calculations and applying either the RS or ARS charge-transfer correction directly to the FRAGMO initial guess. Set FRGM_METHOD to NOSCF_RS or NOSCF_ARS to request the single Roothaan correction or approximate Roothaan correction, respectively. To get a somewhat better energy estimate set FRGM_METHOD to NOSCF_DRS and NOSCF_RS_FOCK. In the case of NOSCF_RS_FOCK, the same steps as in the NOSCF_RS method are performed followed by one more Fock build and calculation of the proper SCF energy. In the case of the double Roothaan-step correction, NOSCF_DRS, the same steps as in NOSCF_RS_FOCK are performed followed by one more diagonalization. The final energy in the NOSCF_DRS method is evaluated as a perturbative correction, similar to the single Roothaan-step correction.

Charge-transfer corrections applied directly to the FRAGMO guess are included in Q-CHEM to test accuracy and performance of the locally-projected SCF methods. However, for some systems they give a reasonable estimate of the binding energies at a cost of one (or two) SCF step(s).

12.5.3 First-Generation ALMO-EDA and Perturbative Charge-Transfer Analysis

The strength of intermolecular binding is inextricably connected to the fundamental nature of interactions between the molecules. Intermolecular complexes can be stabilized through weak dispersive forces, electrostatic effects (*e.g.*, charge–charge, charge–dipole, and charge–induced dipole interactions) and donor-acceptor type orbital interactions such as forward and back-donation of electron density between the molecules. Depending on the extent of these interactions, the intermolecular binding could vary in strength from just several kJ/mol (van der Waals complexes) to several hundred kJ/mol (metal–ligand bonds in metal complexes). Understanding the contributions of various interaction modes enables one to tune the strength of the intermolecular binding to the ideal range by designing materials that promote desirable effects. One of the most powerful techniques that modern first principles electronic structure

methods provide to study and analyze the nature of intermolecular interactions is the decomposition of the total molecular binding energy into the physically meaningful components such as dispersion, electrostatic, polarization, charge transfer, and geometry relaxation terms.

Energy decomposition analysis based on absolutely-localized molecular orbitals (ALMO-EDA) is implemented in Q-CHEM,[53] including the open shell generalization.[40] In ALMO-EDA, the total intermolecular binding energy is decomposed into the "frozen density" component (FRZ), the polarization (POL) term, and the charge-transfer (CT) term. The "frozen density" term is defined as the energy change that corresponds to bringing infinitely separated fragments together without *any* relaxation of their MOs. The FRZ term is calculated as a difference between the FRAGMO guess energy and the sum of the converged SCF energies on isolated fragments. The polarization (POL) energy term is defined as the energy lowering due to the *intrafragment* relaxation of the frozen occupied MOs on the fragments. The POL term is calculated as a difference between the converged SCF-MI energy and the FRAGMO guess energy. Finally, the charge-transfer (CT) energy term is due to further *interfragment* relaxation of the MOs. It is calculated as a difference between the fully converged SCF energy and the converged SCF-MI energy.

The total charge-transfer term includes the energy lowering due to electron transfer from the occupied orbitals on one molecule (more precisely, occupied in the converged SCF-MI state) to the virtual orbitals of another molecule as well as the further energy change caused by induction that accompanies such an occupied/virtual mixing. The energy lowering of the occupied-virtual electron transfer can be described with a single non-iterative Roothaan-step correction starting from the converged SCF-MI solution. Most importantly, the mathematical form of the SCF-MI(RS) energy expression allows one to decompose the occupied-virtual mixing term into bonding and back-bonding components for each pair of molecules in the complex. The remaining charge-transfer energy term (*i.e.*, the difference between SCF-MI(RS) energy and the full SCF energy) includes all induction effects that accompany occupied-virtual charge transfer and is generally small. This last term is called higher order (HO) relaxation. Unlike the RS contribution, the higher order term cannot be divided naturally into forward and back-donation terms. The BSSE associated with each charge-transfer term (forward donation, back-bonding, and higher order effects) can be corrected individually.

To perform energy decomposition analysis, specify fragments in the *$molecule* section and set JOBTYPE to EDA. For a complete EDA job, Q-CHEM

- performs the SCF on isolated fragments (use the *$rem_frgm* section if convergence issues arise but make sure that keywords in this section do not affect the final energies of the fragments),
- generates the FRAGMO guess to obtain the FRZ term,
- converges the SCF-MI equations to evaluate the POL term,
- performs evaluation of the perturbative (RS or ARS) variational correction to calculate the forward donation and back-bonding components of the CT term for each pair of molecules in the system,
- converges the full SCF procedure to evaluate the higher order relaxation component of the CT term.

The FRGM_LPCORR keyword controls evaluation of the CT term in an EDA job. To evaluate all of the CT components mentioned above set this keyword to RS_EXACT_SCF or ARS_EXACT_SCF. If the HO term in not important then the final step (*i.e.*, the SCF calculation) can be skipped by setting FRGM_LPCORR to RS or ARS. If only the total CT term is required then set FRGM_LPCORR to EXACT_SCF.

ALMO charge transfer analysis (ALMO-CTA) is performed together with ALMO EDA.[54] The ALMO charge transfer scale, ΔQ, provides a measure of the distortion of the electronic clouds upon formation of an intermolecular bond and is such that all CT terms (*i.e.*, forward-donation, back-donation, and higher order relaxation) have well defined energetic effects (*i.e.*, ALMO-CTA is consistent with ALMO-EDA).

To remove the BSSE from the CT term (both on the energy and charge scales), set EDA_BSSE to TRUE. Q-CHEM generates an input file for each fragment with MIXED basis set to perform the BSSE correction. As with all jobs with MIXED basis set and d or higher angular momentum basis functions on atoms, the PURECART keyword needs to be initiated. If EDA_BSSE = TRUE then general basis sets (BASIS = GEN) cannot be used in the current implementation.

Please note that the energy of the geometric distortion of the fragments is not included into the total binding energy calculated in an EDA job. The geometry optimization of isolated fragments must be performed to account for this term.

In the 5.2 release and after, the "EDA2" driver (see Section 12.6) will be employed by default when JOBTYPE = EDA is set, which covers almost all features of the first-generation ALMO-EDA/CTA while including many new features (such as further decomposition of the frozen term). The original implementation of the first-generation ALMO-EDA is still accessible by setting EDA2 = FALSE.

Example 12.8 Energy decomposition analysis of the binding energy between the water molecules in a tetramer. ALMO-CTA results are also printed out.

```
$molecule
0 1
--
0 1
O          -0.106357    0.087598    0.127176
H           0.851108    0.072355    0.136719
H          -0.337031    1.005310    0.106947
--
0 1
O           2.701100   -0.077292   -0.273980
H           3.278147   -0.563291    0.297560
H           2.693451   -0.568936   -1.095771
--
0 1
O           2.271787   -1.668771   -2.587410
H           1.328156   -1.800266   -2.490761
H           2.384794   -1.339543   -3.467573
--
0 1
O          -0.518887   -1.685783   -2.053795
H          -0.969013   -2.442055   -1.705471
H          -0.524180   -1.044938   -1.342263
$end

$rem
   JOBTYPE          EDA
   EDA2             FALSE
   METHOD           EDF1
   BASIS            6-31(+,+)g(d,p)
   PURECART         1112
   FRGM_METHOD      GIA
   FRGM_LPCORR      RS_EXACT_SCF
   EDA_BSSE         TRUE
$end
```

Example 12.9 An open-shell EDA example of Na^+ interacting with the methyl radical.

```
$molecule
   1 2
--
   0 2
   C       -1.447596  -0.000023   0.000019
   H       -1.562749   0.330361  -1.023835
   H       -1.561982   0.721445   0.798205
   H       -1.561187  -1.052067   0.225866
--
   1 1
   Na       1.215591   0.000036  -0.000032
$end

$rem
   JOBTYPE              EDA
   EDA2                 FALSE
   METHOD               B3LYP
   BASIS                6-31G*
   UNRESTRICTED         TRUE
   SCF_GUESS            FRAGMO
   FRGM_METHOD          STOLL
   FRGM_LPCORR          RS_EXACT_SCF
   EDA_BSSE             TRUE
   DIIS_SEPARATE_ERRVEC 1
$end
```

12.5.4 Perturbative Charge-Transfer Analysis Usinig Complementary Occupied/Virtual Pairs

In addition to quantifying the amount and energetics of intermolecular charge transfer, it is often useful to have a simple description of orbital interactions in intermolecular complexes. The polarized ALMOs obtained from the SCF-MI procedure and used as a reference basis set in the decomposition analysis do not directly show which occupied-virtual orbital pairs are of most importance in forming intermolecular bonds. By performing rotations of the polarized ALMOs within a molecule, it is possible to find a "chemist's basis set" that represents bonding between molecules in terms of just a few localized orbitals called complementary occupied-virtual pairs (COVPs). This orbital interaction model validates existing conceptual descriptions of intermolecular bonding. For example, in the modified ALMO basis, hydrogen bonding in water dimer is represented as an electron pair localized on an oxygen atom donating electrons to the O–H σ-antibonding orbital on the other molecule,[55] and the description of synergic bonding in metal complexes agrees well with simple Dewar-Chatt-Duncanson model.[16,54,76]

Set EDA_COVP to TRUE to perform the COVP analysis of the CT term in an EDA job. COVP analysis is currently implemented only for systems of two fragments. Set EDA_PRINT_COVP to TRUE to print out localized orbitals that form occupied-virtual pairs. In this case, MOs obtained in the end of the run (SCF-MI orbitals, SCF-MI(RA) orbitals, converged SCF orbitals) are replaced by the orbitals of COVPs. Each orbital is printed with the corresponding CT energy term in kJ/mol (instead of the energy eigenvalues in hartrees). These energy labels make it easy to find correspondence between an occupied orbital on one molecule and the virtual orbital on the other molecule. The examples below show how to print COVP orbitals. One way is to set *$rem* variable PRINT_ORBITALS, the other is to set IANLTY to 200 and use the *$plots* section in the Q-CHEM input. In the first case, the orbitals can be visualized using MOLDEN (set MOLDEN_FORMAT to TRUE). In the second case use Visual Molecular Dynamics (VMD)[1,43] or a similar third

party program capable of making 3D plots.

Example 12.10 COVP analysis of the CT term. The COVP orbitals are printed in the Q-CHEM and MOLDEN formats.

```
$molecule
0 1
--
0 1
O          -1.521720    0.129941    0.000000
H          -1.924536   -0.737533    0.000000
H          -0.571766   -0.039961    0.000000
--
0 1
O           1.362840   -0.099704    0.000000
H           1.727645    0.357101   -0.759281
H           1.727645    0.357101    0.759281
$end

$rem
   JOBTYPE         EDA
   EDA2            FALSE
   BASIS           6-31G
   PURECART        1112
   METHOD          B3LYP
   FRGM_METHOD     GIA
   FRGM_LPCORR     RS_EXACT_SCF
   EDA_COVP        TRUE
   EDA_PRINT_COVP  TRUE
   PRINT_ORBITALS  16
   MOLDEN_FORMAT   TRUE
$end
```

Example 12.11 COVP analysis of the CT term. Note that it is not necessary to run a full EDA job. It is suffice to set FRGM_LPCORR to RS or ARS and EDA_COVP = TRUE to perform the COVP analysis. The orbitals of the most significant occupied-virtual pair are printed into an ASCII file called `plot.mo` which can be converted into a cube file and visualized in VMD.

```
$molecule
0 1
--
0 1
O          -1.521720    0.129941    0.000000
H          -1.924536   -0.737533    0.000000
H          -0.571766   -0.039961    0.000000
--
0 1
O           1.362840   -0.099704    0.000000
H           1.727645    0.357101   -0.759281
H           1.727645    0.357101    0.759281
$end

$rem
   BASIS           6-31G
   PURECART        1112
   METHOD          B3LYP
   FRGM_METHOD     GIA
   FRGM_LPCORR     RS
   IANLTY          200
   EDA_COVP        TRUE
   EDA_PRINT_COVP  TRUE
$end

$plots
MOs
   80  -4.0  4.0
   60  -3.0  3.0
   60  -3.0  3.0
   2   0  0  0
   6  11
$end
```

12.5.5 Job Control Options

FRGM_METHOD

Specifies a locally-projected method.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

STOLL	Locally-projected SCF equations of Stoll are solved.
GIA	Locally-projected SCF equations of Gianinetti are solved.
NOSCF_RS	Single Roothaan-step correction to the FRAGMO initial guess.
NOSCF_ARS	Approximate single Roothaan-step correction to the FRAGMO initial guess.
NOSCF_DRS	Double Roothaan-step correction to the FRAGMO initial guess.
NOSCF_RS_FOCK	Non-converged SCF energy of the single Roothaan-step MOs.

RECOMMENDATION:

STOLL and GIA are for variational optimization of the ALMOs. NOSCF options are for computationally fast corrections of the FRAGMO initial guess.

FRGM_LPCORR

Specifies a correction method performed after the locally-projected equations are converged.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

ARS	Approximate Roothaan-step perturbative correction.
RS	Single Roothaan-step perturbative correction.
EXACT_SCF	Full SCF variational correction.
ARS_EXACT_SCF	Both ARS and EXACT_SCF in a single job.
RS_EXACT_SCF	Both RS and EXACT_SCF in a single job.

RECOMMENDATION:

For large basis sets use ARS, use RS if ARS fails.

SCF_PRINT_FRGM

Controls the output of Q-CHEM jobs on isolated fragments.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	The output is printed to the parent job output file.
FALSE	The output is not printed.

RECOMMENDATION:

Use TRUE if details about isolated fragments are important.

EDA_BSSE

Calculates the BSSE correction when performing the energy decomposition analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set to TRUE unless a very large basis set is used.

EDA_COVP

Perform COVP analysis when evaluating the RS or ARS charge-transfer correction. COVP analysis is currently implemented only for systems of two fragments.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set to TRUE to perform COVP analysis in an EDA or SCF-MI(RS) job.

EDA_PRINT_COVP

Replace the final MOs with the CVOP orbitals in the end of the run.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set to TRUE to print COVP orbitals instead of conventional MOs.

NVO_LIN_MAX_ITE

Maximum number of iterations in the preconditioned conjugate gradient solver of the single-excitation amplitude equations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User–defined number of iterations.

RECOMMENDATION:

None.

NVO_LIN_CONVERGENCE

Target error factor in the preconditioned conjugate gradient solver of the single-excitation amplitude equations.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n User–defined number.

RECOMMENDATION:

Solution of the single-excitation amplitude equations is considered converged if the maximum residual is less than 10^{-n} multiplied by the current DIIS error. For the ARS correction, n is automatically set to 1 since the locally-projected DIIS error is normally several orders of magnitude smaller than the full DIIS error.

NVO_METHOD

Sets method to be used to converge solution of the single-excitation amplitude equations.

TYPE:

INTEGER

DEFAULT:

9

OPTIONS:

n User–defined number.

RECOMMENDATION:

This is an experimental option. Use the default.

NVO_UVV_PRECISION

Controls convergence of the Taylor series when calculating the U_{vv} block from the single-excitation amplitudes. Series is considered converged when the maximum element of the term is less than 10^{-n}.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

n User–defined number.

RECOMMENDATION:

NVO_UVV_PRECISION must be the same as or larger than THRESH.

NVO_UVV_MAXPWR

Controls convergence of the Taylor series when calculating the U_{vv} block from the single-excitation amplitudes. If the series is not converged at the nth term, more expensive direct inversion is used to calculate the U_{vv} block.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n User–defined number.

RECOMMENDATION:

None.

NVO_TRUNCATE_DIST

Specifies which atomic blocks of the Fock matrix are used to construct the preconditioner.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

$n > 0$ If distance between a pair of atoms is more than n Ångstroms do not include the atomic block.

-2 Do not use distance threshold, use NVO_TRUNCATE_PRECOND instead.

-1 Include all blocks.

0 Include diagonal blocks only.

RECOMMENDATION:

This option does not affect the final result. However, it affects the rate of the PCG algorithm convergence. For small systems, use the default.

NVO_TRUNCATE_PRECOND

Specifies which atomic blocks of the Fock matrix are used to construct the preconditioner. This variable is used only if NVO_TRUNCATE_DIST is set to -2.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n If the maximum element in an atomic block is less than 10^{-n} do not include the block.

RECOMMENDATION:

Use the default. Increasing n improves convergence of the PCG algorithm but overall may slow down calculations.

12.6 Second-Generation ALMO-EDA Method

12.6.1 Introduction

The ALMO-EDA method introduced in Section 12.5.3 is a very useful tool for unraveling the nature of intermolecular interactions. Nevertheless, it has two major shortcomings: (i) Although the polarization (POL) energy is variationally

evaluated, it does not have a meaningful basis set limit. As the employed basis set becomes larger, the POL term starts to be contaminated by charge transfer (CT) and loses its intended meaning. (ii) The frozen (FRZ) interaction is a monolithic term in the original ALMO-EDA scheme. In practice, further decomposition of the FRZ term is often desired. For example, if one wants to use ALMO-EDA as a tool for the development of empirical force fields, the separation of the FRZ term into contributions from permanent electrostatics, Pauli repulsion and dispersion will be helpful since they are usually modeled by distinct functional forms in classical force fields. These drawbacks have been addressed recently, defining the second generation of the ALMO-EDA method (also referred to as "EDA2" in the text below).[38,41,42]

12.6.2 Generalized SCF-MI Calculations and Additional Features

The original definition of the ALMOs used in SCF-MI calculations is based on the fragment-blocking structure of the AO-to-MO transformation matrix, *i.e.*, for a given fragment, the associated MOs can only be expanded by the AO basis functions centered on the atoms that belong to the same fragment. Here we propose a generalized definition for SCF-MI calculations: given a set of basis vectors ($\mathbf{G}$) in which each of them is tagged to a fragment but is allowed to be spanned by any AO basis function, it defines the working basis of the SCF-MI problem. Then, within this basis, the locally projected SCF equations can be solved in a similar way, with the constraint that the MO coefficient matrix in the working basis ($\mathbf{G}$) is fragment-block-diagonal, while the MO coefficient matrix in the AO basis does not necessarily retain the blocking structure. The basis vectors in $\mathbf{G}$ can be either non-orthogonal or orthogonal between fragments. More details on the generalized SCF-MI equations are available in Ref. 38.

This generalized SCF-MI scheme is implemented in GEN_SCFMAN (the original AO-block based scheme is available in GEN_SCFMAN as well). It is used for the variational optimization of the polarized (but CT-forbidden) intermediate state in "EDA2" (see Section 12.6.3). Another preferable feature of this generalized scheme is that the interfragment linear dependency in $\mathbf{G}$ can be properly handled. Therefore, this scheme can be used to replace the original AO-block based SCF-MI without becoming ill-defined when interfragment linear dependency occurs. In contrast, the original ALMO-EDA method that employs the AO-block based approach fails when the sum of the number of orbitals on each fragment is not equal to the number of orbitals for the super-system (the latter is determined by the total number of AO basis functions and BASIS_LIN_DEP_THRESH), which often happens when substantially large basis sets are used or when the super-system comprises a large number of fragments.

SCF-MI calculations based on the GEN_SCFMAN implementation are triggered by setting GEN_SCFMAN = TRUE and FRGM_METHOD = STOLL or GIA (the other options of FRGM_METHOD are not allowed). A subset of supported algorithms in GEN_SCFMAN are available for restricted (R) and unrestricted (U) SCF-MI, including DIIS, GDM, GDM_LS, and NEWTON_CG. While the DIIS algorithm iteratively solves for the locally-projected SCF equations, the latter two methods use the energy derivatives with respect to the on-fragment orbital rotations to minimize the energy until the gradient reaches zero. As for standard calculations using GEN_SCFMAN, internal stability analysis is also available for R- and U-SCF-MI, and one can set FD_MAT_VEC_PROD to TRUE if the analytical Hessian is not available for the employed density functional (note: for functionals containing non-local correlation, one can always use FD_MAT_VEC_PROD = FALSE).

As in the original implementation, perturbative corrections can be applied on top of the SCF-MI solution to approach the full SCF result, and this is still controlled by FRGM_LPCORR. Note that among the options introduced in Section 12.5.5, only ARS and RS are allowed here since the exact SCF calculation is actually beyond the scope of SCF-MI.

In addition, with this more general implementation users are allowed to specify some fragments to be frozen during the SCF-MI calculation, *i.e.*, intrafragment relaxation does not occur on these fragments. This is achieved by specifying the *$rem* variable SCFMI_FREEZE_SS. Such a calculation can be interpreted as an active fragment being embedded in a frozen environment where the interaction between them is treated quantum mechanically.

SCFMI_MODE

Determine whether generalized SCF-MI is used and also the property of the working basis.

TYPE:

INTEGER

DEFAULT:

0 (“1" is used by basic “EDA2" calculations).

OPTIONS:

0 AO-block based SCF-MI (the original definition of ALMOs).

1 Generalized SCF-MI with basis vectors that are non-orthogonal between fragments.

2 Generalized SCF-MI with basis vectors that are orthogonal between fragments.

RECOMMENDATION:

None

SCFMI_FREEZE_SS

Keep the first several fragments unrelaxed in an SCFMI calculation.

TYPE:

INTEGER

DEFAULT:

0 (all fragments are active)

OPTIONS:

n Freeze the first n fragments.

RECOMMENDATION:

None

Example 12.12 Generalized SCF-MI calculation for the water dimer with single Roothaan-step perturbative correction. For this specific case, the result is identical to that given by AO-block based SCF-MI (SCFMI_MODE = 0).

```
$molecule
0 1
--
0 1
O  -1.551007  -0.114520   0.000000
H  -1.934259   0.762503   0.000000
H  -0.599677   0.040712   0.000000
--
0 1
O   1.350625   0.111469   0.000000
H   1.680398  -0.373741  -0.758561
H   1.680398  -0.373741   0.758561
$end

$rem
   METHOD           b3lyp
   GEN_SCFMAN       true
   BASIS            6-31+G(d)
   GEN_SCFMAN       true
   SCF_CONVERGENCE  8
   THRESH           14
   FRGM_METHOD      stoll
   FRGM_LPCORR      rs
   SCFMI_MODE       1 !gen scfmi (non-orthogonal)
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
$end
```

12.6.3 Polarization Energy with a Well-Defined Basis Set Limit

The definition of polarization energy lowering in the original ALMO-EDA used the full AO space of each fragment as the variational degrees of freedom. This is based on the assumption that the AO basis functions are fragment-ascribable based on their atomic centers. However, this assumption becomes inappropriate when very large basis sets are used, especially those with diffuse functions (*e.g.* def2-QZVPPD). In such scenarios, basis functions on a given fragment tend to describe other fragments so that the "absolute localization" constraint becomes weaker and finally gets effectively removed. This is why the original ALMO-EDA scheme does not have a well-defined basis set limit for its polarization energy.

To overcome this problem, Horn and Head-Gordon[38] proposed a new definition for the POL term in the ALMO-EDA method based on fragment electrical response functions (FERFs). FERFs on a given fragment are prepared by solving CPSCF equations after its SCF solution is found:

$$H_{ai,bj}(\Delta_\mu)_{bj} = 2(M_\mu)_{ai}, \tag{12.1}$$

where $\mathbf{H}$ is SCF orbital Hessian and $\mathbf{M}_\mu$ is a component (μ) of a multipole matrix with a certain order. The resulting fragment response matrices ($\{\Delta_\mu\}$) are a set of $n_\mathrm{v} \times n_\mathrm{o}$ matrices. Then, a singular value decomposition (SVD) is performed on Δ_μ:

$$(\Delta_\mu)_{ai} = (L_\mu)_{ab}(d_\mu)_{bj}(R_\mu^T)_{ji}, \tag{12.2}$$

and the left vectors (not including the null vectors) will be used to construct a truncated virtual space, which is used to define the variational degrees of freedom for the SCF-MI problem:

$$\mathbf{V}_\mu = \mathbf{C}_\mathrm{vir}\mathbf{L}_\mu, \tag{12.3}$$

where $\mathbf{C}_\mathrm{vir}$ denotes the original virtual orbitals of the given fragment.

The basic spirit of using FERFs is to obtain a subset of virtuals that is most pertinent to the electrical polarization of a given fragment, while the redundant variational degrees of freedom (which might be CT-like) are excluded. This scheme is shown to give a well-defined basis set limit for the polarization energy that relies on the SCF-MI calculation. The multipole orders dipole (D), quadrupole (Q), and octopole (O)m included on the RHS of Eq. (12.2), decide the span of FERFs on each fragment. Numerical experiments suggest that the inclusion of dipole- and quadrupole-type responses is able to long-range induced electrostatics correctly and also gives a well-defined basis set limit, which is thus recommended as the working basis of the SCF-MI problem. The full span of the polarization subspace of fragment A is thus:

$$\mathbf{O}_A \oplus \mathrm{span}\{\mathbf{V}_{\mu x}, \mathbf{V}_{\mu y}, \mathbf{V}_{\mu z}\} \oplus \mathrm{span}\{\mathbf{V}_{Q2,-2}, \mathbf{V}_{Q2,-1}, \mathbf{V}_{Q2,0}, \mathbf{V}_{Q2,1}, \mathbf{V}_{Q2,2}\}. \tag{12.4}$$

Therefore, each occupied orbital will be paired with eight virtual orbitals (if the employed AO basis is large enough).

The polarization subspaces constructed as in Eq. (12.4) are non-orthogonal between fragments. Therefore, it is named as the "nDQ" model for polarization. There is another version of this method which enforces interfragment orthogonality between the polarization subspaces and it is correspondingly termed as "oDQ" (or with other multipole orders). The preparation of orthogonal FERFs is more complicated (see Ref. 38 for the details) and usually gives less favorable polarization energies. For most general cases, we recommend the use of the "nDQ" model. Calculations using FERFs are performed using the generalized SCF-MI procedure introduced in Section 12.6.2.

However, solving the CPSCF equation requires inverting the large orbital Hessian matrix $\mathbf{H}$ of dimension $ov \times ov$, which is the time demanding step in solving for FERFs. For example, in Hartree-Fock theory, in the canonical representation, it has matrix elements

$$H_{ai,bj} = 2\left[(\epsilon_a - \epsilon_i)\delta_{ij}\delta_{ab} + \langle aj||ib\rangle + \langle ab||ij\rangle\right]. \tag{12.5}$$

Recently, Aldossary and coauthors[4] proposed an uncoupled-FERFs (uFERFs) to improve the conventional FERFs. The idea of uFERFs is simply to adopt the solution for uncoupled CPSCF equation, where the orbital Hessian is

approximated with

$$H_{ai,bj} = 2(\epsilon_a - \epsilon_i)\delta_{ij}\delta_{ab}. \tag{12.6}$$

With this approximation, the orbital Hessian is easily inverted, and the fragment response matrices are simply

$$\left(\Delta_\mu^{\text{uFERF}}\right)_{ai} = \frac{(M_\mu)_{ai}}{\epsilon_a - \epsilon_i}. \tag{12.7}$$

Benchmark calculations on the S22, S66 and Ionic43 datasets shows that the POL energy is hardly changed using uFERFs, while a general 5-10 times speedup for FERF creation and 20% average speedup for the whole EDA calculation is achieved. In addition, the importance of monopole uFERFs (response to scaled nuclear charges) for intermolecular interactions is recognized for strong ion-neutral interactions. Therefore, it is recommended to use MDQ-uFERFs for the calculation of POL energy in ALMO-EDA calculations.

CHILD_MP

Compute FERFs for fragments and use them as the basis for SCF-MI calculations.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute FERFs (use the full AO span of each fragment).

TRUE Compute fragment FERFs.

RECOMMENDATION:

Use FERFs to compute polarization energy when large basis sets are used. In an “EDA2" calculation, this *$rem* variable is set based on the given option automatically.

CHILD_MP_ORDERS

The multipole orders included in the prepared FERFs. The last digit specifies how many multipoles to compute, and the digits in the front specify the multipole orders: 2: dipole (D); 3: quadrupole (Q); 4: octopole (O). Multipole order 1 is reserved for monopole FERFs which can be used to separate the effect of orbital contraction.[67]

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

21 D

232 DQ

2343 DQO

RECOMMENDATION:

Use 232 (DQ) when FERF is needed.

FRAG_CPSCF_MAXITER

The maximum number of iterations executed by the conjugate-gradient solver before switching to the MINRES algorithm. The maximum number of MINRES iterations is set to twice of the value of FRAG_CPSCF_MAXITER. Note that this rem variable is also used for the CPSCF equations involved in the adiabatic EDA (Sec. 12.7.3) and force decomposition analysis (Sec. 12.7.4) calculations.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

RECOMMENDATION:

Use the default

FRAG_CPSCF_CONV

The convergence threshold for the CPSCF equation (using the RMS error of the residual vector)

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n Convergence is reached when the RMS error is below 10^{-n}

RECOMMENDATION:

Use the default

EDA_UFERF

Using uncoupled-FERFs (uFERFs)[4] instead of FERFs

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Use uFERFs

FALSE Use FERFs

RECOMMENDATION:

Use the default uFERFs, use FALSE when FERFs are desired.

Example 12.13 Generalized SCF-MI calculation for the water dimer using nDQ FERFs.

```
$molecule
0 1
--
0 1
O  -1.551007  -0.114520   0.000000
H  -1.934259   0.762503   0.000000
H  -0.599677   0.040712   0.000000
--
0 1
O   1.350625   0.111469   0.000000
H   1.680398  -0.373741  -0.758561
H   1.680398  -0.373741   0.758561
$end

$rem
   METHOD             wb97x-v
   GEN_SCFMAN         true
   BASIS              6-31+G(d)
   GEN_SCFMAN         true
   SCF_ALGORITHM      diis
   SCF_CONVERGENCE    8
   THRESH             14
   SCF_FINAL_PRINT    1
   EDA_UFERF          false
   FRGM_METHOD        stoll
   SCFMI_MODE         1 !nonortho gen scfmi
   CHILD_MP           true
   CHILD_MP_ORDERS    232 !DQ
   FD_MAT_VEC_PROD    false
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end
```

12.6.4 Further Decomposition of the Frozen Energy

The frozen interaction energy in ALMO-EDA is defined as the energy difference between the unrelaxed frozen (Heitler-London) wave function and the isolated fragments. In other literature (*e.g.* Ref. 37), this interaction is often decomposed in a classical fashion:

$$\Delta E_{\mathrm{frz}} = \Delta E_{\mathrm{elec}}^{\mathrm{cls}} + \Delta E_{\mathrm{Pauli}}^{\mathrm{cls}}, \tag{12.8}$$

where the contribution from permanent electrostatics is defined as the Coulomb interaction between isolated fragment charge distributions:

$$\Delta E_{\mathrm{elec}}^{\mathrm{cls}} = \sum_{A<B} \int_{r_1} \int_{r_2} \rho_A^{\mathrm{tot}}(\mathbf{r}_1) \frac{1}{\mathbf{r}_{12}} \rho_B^{\mathrm{tot}}(\mathbf{r}_2) \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2 \tag{12.9}$$

and the remainder constitutes the Pauli (or exchange) term. Such a decomposition (referred to as the classical decomposition below) is associated with two issues: (i) the evaluation of permanent electrostatics makes use of the "promolecule" state (whose density is the simple sum of monomer densities) rather than a properly anti-symmetrized wave function; (ii) when dispersion-corrected density functionals are used, the Pauli term contains dispersion interaction and thus loses its original meaning.

Horn *et al.*[41] proposed a new scheme to further decompose the frozen term into contributions from permanent electrostatics (ELEC), Pauli repulsion (PAULI) and dispersion (DISP):

$$\Delta E_{\mathrm{frz}} = \Delta E_{\mathrm{elec}} + \Delta E_{\mathrm{Pauli}} + \Delta E_{\mathrm{disp}} \tag{12.10}$$

This approach is compatible with the use of all kinds of density functionals except double hybrids, and all three components of the FRZ term are computed with the antisymmetrized frozen wave function. The key step of this method is the orthogonal decomposition of the 1PDM associated with the frozen wave function into contributions from individual fragments: $\mathbf{P}_{\mathrm{frz}} = \sum_A \tilde{\mathbf{P}}_A$. This is achieved by minimizing an objective function as follows:

$$\Omega = \sum_A E_A[\tilde{\mathbf{P}}_A] - E_A[\mathbf{P}_A] \tag{12.11}$$

while interfragment orthogonality is enforced between $\tilde{\mathbf{P}}_A$s. The readers are referred to Ref. 41 for more details about the orthogonal decomposition.

The ELEC term is then defined as the Coulomb interaction between distorted fragment densities ($\tilde{\rho}_A(\mathbf{r})$):

$$\Delta E_{\mathrm{elec}} = \sum_{A<B} \int_{r_1} \int_{r_2} \tilde{\rho}_A^{\mathrm{tot}}(\mathbf{r}_1) \frac{1}{\mathbf{r}_{12}} \tilde{\rho}_B^{\mathrm{tot}}(\mathbf{r}_2) \mathrm{d}\mathbf{r}_1 \mathrm{d}\mathbf{r}_2. \tag{12.12}$$

The DISP term is evaluated by subtracting the dispersion-free part of the total exchange-correlation (XC) interaction, where an auxiliary "dispersion-free" XC (DFXC) functional is used in company with the primary (target) XC functional:

$$\Delta E_{\mathrm{disp}} = \left(E_{\mathrm{xc}}[\mathbf{P}_{\mathrm{frz}}] - \sum_A E_{\mathrm{xc}}[\tilde{\mathbf{P}}_A] \right) - \left(E_{\mathrm{xc}}^{\mathrm{DF}}[\mathbf{P}_{\mathrm{frz}}] - \sum_A E_{\mathrm{xc}}^{\mathrm{DF}}[\tilde{\mathbf{P}}_A] \right). \tag{12.13}$$

It has been suggested[83] that HF is an appropriate DFXC for use with dispersion-inclusive or dispersion-corrected hybrid functionals such as ωB97M-V or B3LYP-D3, while revPBE is appropriate for semi-local functionals such as B97M-V. However, more recent work suggests that for hybrid GGA functionals (*e.g.*, PBE0-D3), the correlation energy should be included in the DFXC, which would then be HF-PBE rather than HF alone.[28] This avoids putting the correlation contribution to electrostatics and Pauli repulsion into the dispersion term, and affords better agreement with dispersion energies from symmetry-adapted perturbation theory.

The remainder of the frozen interaction goes into the PAULI term, which includes the net repulsive interaction given by Eq. (12.11) and the "dispersion-free" part of the XC interaction:

$$\Delta E_{\mathrm{Pauli}} = \sum_A (E_A[\tilde{\mathbf{P}}_A] - E_A[\mathbf{P}_A]) + \left(E_{\mathrm{xc}}^{\mathrm{DF}}[\mathbf{P}_{\mathrm{frz}}] - \sum_A E_{\mathrm{xc}}^{\mathrm{DF}}[\tilde{\mathbf{P}}_A] \right). \tag{12.14}$$

The PAULI term and the ELEC term can also be combined together and reported as the dispersion-free frozen (DFFRZ) term if desired.

In Q-CHEM's implementation of "EDA2", the classical frozen decomposition and the new scheme defined by Eqs. (12.12)–(12.14) are both computed by default. The classical ELEC term only depends on monomer properties and the distances between fragments, therefore, it can be particularly useful for scenarios such as force field development (as the reference for permanent electrostatics). When the DISP term calculated by the new scheme is available, a modified classical Pauli term[79] is also reported, which is simply defined as

$$\Delta E_{\mathrm{Pauli}}^{\mathrm{mod}} = \Delta E_{\mathrm{Pauli}}^{\mathrm{cls}} - \Delta E_{\mathrm{disp}}, \tag{12.15}$$

i.e., the dispersion contribution is removed from the classical Pauli term computed using its original definition. The overall decomposition of the frozen energy with the classical scheme is given by

$$\Delta E_{\mathrm{frz}} = \Delta E_{\mathrm{elec}}^{\mathrm{cls}} + \Delta E_{\mathrm{Pauli}}^{\mathrm{mod}} + \Delta E_{\mathrm{disp}} \tag{12.16}$$

Alternatively, this can also be achieved without performing the orthogonal decomposition, by setting EDA_CLS_DISP to TRUE. This also evaluates the DISP term via Eq. (12.13) except that undistorted monomer densities ($\{\mathbf{P}_A\}$) are used instead of their distorted counterparts ($\{\tilde{\mathbf{P}}_A\}$):

$$\Delta E_{\mathrm{disp}} = \left(E_{\mathrm{xc}}[\mathbf{P}_{\mathrm{frz}}] - \sum_A E_{\mathrm{xc}}[\mathbf{P}_A] \right) - \left(E_{\mathrm{xc}}^{\mathrm{DF}}[\mathbf{P}_{\mathrm{frz}}] - \sum_A E_{\mathrm{xc}}^{\mathrm{DF}}[\mathbf{P}_A] \right). \tag{12.17}$$

FRZ_ORTHO_DECOMP

Perform the decomposition of frozen interaction energy based on the orthogonal decomposition of the 1PDM associated with the frozen wave function.

TYPE:

BOOLEAN

DEFAULT:

FALSE (automatically set to TRUE by EDA2 options 1–5)

OPTIONS:

FALSE Do not perform the orthogonal decomposition.

TRUE Perform the frozen energy decomposition using orthogonal fragment densities.

RECOMMENDATION:

Use default value automatically set by "EDA2". Note that users are allowed to turn off the orthogonal decomposition by setting FRZ_ORTHO_DECOMP to -1. Also, for calculations that involve ECPs, it is automatically set to FALSE since unreasonable results will be produced otherwise.

FRZ_ORTHO_DECOMP_CONV

Convergence criterion for the minimization problem that gives the orthogonal fragment densities.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n 10^{-n}

RECOMMENDATION:

Use the default unless tighter convergence is preferred.

EDA_CLS_ELEC

Perform the classical decomposition of the frozen term.

TYPE:

BOOLEAN

DEFAULT:

FALSE (automatically set to TRUE by EDA2 options 1–5)

OPTIONS:

FALSE Do not compute the classical ELEC and PAULI terms.

TRUE Perform the classical decomposition.

RECOMMENDATION:

TRUE

EDA_CLS_DISP

Compute the DISP contribution without performing the orthogonal decomposition, which will then be subtracted from the classical PAULI term.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Use the DISP term computed with orthogonal decomposition (if available).

TRUE Use the DISP term computed using undistorted monomer densities.

RECOMMENDATION:

Set it to TRUE when orthogonal decomposition is not performed.

DISP_FREE_X

Specify the employed “dispersion-free" exchange functional.

TYPE:

STRING

DEFAULT:

HF

OPTIONS:

Exchange functionals (*e.g.* revPBE) or exchange-correlation functionals (*e.g.* B3LYP) supported by Q-CHEM.

RECOMMENDATION:

HF is recommended for hybrid (primary) functionals (*e.g.*ωB97X-V) and revPBE for semi-local ones (*e.g.*B97M-V). Other reasonable options (*e.g.* B3LYP for B3LYP-D3) can also be applied.

DISP_FREE_C

Specify the employed “dispersion-free" correlation functional.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Correlation functionals supported by Q-CHEM.

RECOMMENDATION:

Put the appropriate correlation functional paired with the chosen exchange functional (*e.g.* put PBE if DISP_FREE_X is revPBE); put NONE if DISP_FREE_X is set to an exchange-correlation functional.

12.6.5 Job Control for EDA2

The use of the FERF model for the evaluation of polarization energy and the further decomposition of the frozen term define the second generation of the ALMO-EDA method. Meanwhile, under the same code structure, the original AO-block based ALMO model and other related methods (such as the constrained relaxation of the frozen wave function[39] which renders the frozen energy variationally computed, and the polMO model[6] that arguably gives a lower limit to the polarization contribution) are also available. This entire set of methods implemented in Q-CHEM based on GEN_SCFMAN (see Section 4.3) is referred to as “EDA2". In Q-CHEM 5.2 and after, “EDA2" is used as the default ALMO-EDA driver when “JOBTYPE = EDA" is requested.

The job control for EDA2 is largely simplified by a series of preset options provided by the developers. The option number is set through the EDA2 *$rem* variable (introduced below). Besides that, for the sake of flexibility, users are allowed to overwrite the values of part of the preset *$rem* variables:

- Related to the isolated fragment calculations:
  - EDA_CHILD_SUPER_BASIS: use the super-system basis for fragment calculations (default: FALSE).
  - FRAGMO_GUESS_MODE: as introduced in Section 12.3 (default: 0).
- Related to the decomposition of the FRZ term:
  - FRZ_ORTHO_DECOMP: it can be turned off by setting its value to -1 in the *$rem* section (default: TRUE).
  - FRZ_ORTHO_DECOMP_CONV: as introduced in Section 12.6.4 (default: 6).

  - EDA_CLS_DISP: as introduced in Section 12.6.4 (default: FALSE).
  - DISP_FREE_X: as introduced in Section 12.6.4 (default: HF).
  - DISP_FREE_C: as introduced in Section 12.6.4 (default: NONE).

- Related to the evaluation and analysis of POL:
  - CHILD_MP_ORDERS: as introduced in Section 12.6.3 (default: 232 (DQ)).
  - SCFMI_FREEZE_SS: as introduced in Section 12.6.2 (default: 0).
  - EDA_POL_A: perform non-perturbative polarization analysis (default: FALSE; see Section 12.6.8).

- Related to the evaluation of CT and BSSE:
  - EDA_NO_CT: skip the evaluation of the CT term in the EDA procedure (default: FALSE (automatically turned on when SCFMI_FREEZE_SS = TRUE)).
  - EDA_BSSE: use counterpoise-corrected monomer calculations to evaluate the BSSE (default: FALSE).
  - EDA_PCT_A: turn on perturbative (Roothaan step based) charge transfer analysis (default: FALSE).
  - EDA_VCT_A: turn on non-perturbative charge transfer analysis (default: FALSE; see Section 12.6.8).
  - EDA_COVP: perform COVP analysis for charge transfer (default: FALSE; see Section 12.5.3).
  - EDA_PRINT_COVP: dump COVPs to the MO coefficient file (default: FALSE; see Section 12.5.3). Note: EDA2 can automatically generate the cubes for the dominant complementary occupied-virtual orbitals for each pair of donor and acceptor fragments when EDA_PRINT_COVP is greater than 1.

- Miscellaneous:
  - EDA2_PRINT_LEVEL: print additional details for some of the EDA terms (*e.g.* solvation-corrected ELEC/PAULI) when set to > 1 (default: 1)

EDA2

Switch on EDA2 and specify the option set number.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 Do not run through EDA2.

1 Frozen energy decomposition + nMDQ-uFERF polarization
(the standard EDA2 option)

2 Frozen energy decomposition + (AO-block-based) ALMO polarization
(old scheme with the addition of frozen decomposition)

3 Frozen energy decomposition + oDQ-FERF polarization
(NOT commonly used)

4 Frozen wave function relaxation + Frozen energy decomposition + nDQ-FERF polarization
(NOT commonly used)

5 Frozen energy decomposition + polMO polarization
(NOT commonly used).

10 No preset. Completely controlled by user's *$rem* input
(for developers only)

RECOMMENDATION:

Turn on EDA2 for Q-CHEM's ALMO-EDA jobs unless CTA with the old scheme is desired. Option 1 is recommended in general, especially when substantially large basis sets are employed. The original ALMO scheme (option 2) can be used when the employed basis set is of small or medium size (arguably no larger than augmented triple-ζ). The other options are rarely used for routine applications.

Note that starting with Q-CHEM v. 5.2, if JOBTYPE = EDA is requested while but the *$rem* variable EDA2 is not specified by the user, the latter defaults to EDA2 = 2 with EDA_PCT_A = TRUE.

For calculations based on EDA2, the default SCF convergence criterion is set to 10^{-8} (rather than 10^{-5} as in Q-CHEM's normal SCF calculations). The fragment calculations involved are forced to use the same SCF convergence criterion as the parent job.

In cases where the radical is a single atom (*e.g.* •Cl) or of a highly symmetric geometry (*e.g.* •OH), there can be multiple degenerate electronic configurations with the unpaired electron residing in different orbitals, resulting in arbitrariness in the definition of the frozen state. For such systems, it is desirable to obtain the orientation of fragment spin that leads to the lowest-energy frozen state. This can be achieved by employing a "polarize-then-depolarize" (PtD) approach,[81] using interfragment polarization to resolve the degeneracy of radical's electronic states: one first converges the polarization (SCF-MI) calculation for the full system, and then recalculates the SCF solutions for each isolated fragment using their corresponding blocks in the ALMO coefficient matrix as the initial guess. To ensure that the "depolarized" fragments are of the same electronic configuration as in the fully polarized wavefunction, the initial maximum overlap method[8] (IMOM) is used in these fragment calculations. The fragment orbitals obtained therefrom then uniquely determine the frozen wavefunction.

In Q-CHEM 5.2 and after, the procedure described above is performed for unrestricted ALMO-EDA calculations by default. It can also be manually requested by setting EDA_ALIGN_FRGM_SPIN to > 0 values. Note that this setting further ensures that one obtains a stable SCF-MI solution in the initial polarization step (see below), which is crucial for the success of this approach. Occasionally, one may find that the frozen state constructed from "spin-aligned" fragments is of a higher energy than the initial one. This indicates that the fragment spin alignment procedure is not

functioning well, and in such cases we recommend the user to run EDA2 calculation without this procedure by setting FRZ_RELAX = -1.

With Q-CHEM 6.2.1 or later versions, the user is also allowed to specify fragments to which the PtD scheme is not applied. These fragments can be specified through the *$skip_frgm_spin_align* section (fragment index starts from 1). For instance,

```
$skip_frgm_spin_align
 2
$end
```

indicates that the 2nd fragment will skip the spin alignment procedure. Typically, for interactions between radicals and closed-shell molecules, spin alignment only needs to be applied to the radicals.

EDA_ALIGN_FRGM_SPIN

Turn on the fragment spin alignment procedure

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not performed the spin alignment procedure (turned on by default in unrestricted cases)

1 Perform fragment spin alignment; use GDM for the polarization step preceding the MOM calculations

2 Perform fragment spin alignment; use GDM and perform stability analysis for the polarization step

RECOMMENDATION:

Use 1 or 2 when the radical is of highly symmetric structure

Another feature that can be useful for systems involving open-shell species is the capability of performing stability analysis on user-specified fragments, since it is important to ensure the stability of each fragment's SCF solution. This can be done through the *$frgm_stability* input section:

```
$frgm_stability
   [frgm_idx1]  [frgm_idx2]  ...
$end
```

where one simply puts the indices of fragments that require stability analysis.

Example 12.14 Energy decomposition analysis for the ammonia-borane complex. The FERF-nDQ model is used for the POL term (as very large basis set is employed here), and decomposition of the frozen interaction energy is performed (Hartree-Fock is employed as the DFXC functional by default).

```
$molecule
   0 1
   --
   0 1
   N       0.000000     0.000000    -0.727325
   H       0.947371     0.000000    -1.091577
   H      -0.473685    -0.820448    -1.091577
   H      -0.473685     0.820448    -1.091577
   --
   0 1
   B       0.000000     0.000000     0.930725
   H      -1.165774     0.000000     1.243063
   H       0.582887    -1.009590     1.243063
   H       0.582887     1.009590     1.243063
$end

$rem
   JOBTYPE           eda
   EDA2              1
   METHOD            wB97M-V
   BASIS             def2-TZVPPD
   MEM_TOTAL         4000
   MEM_STATIC        1000
   THRESH            14
   SCF_CONVERGENCE   8
   XC_GRID           000099000590
   NL_GRID           1
   FD_MAT_VEC_PROD   false
   INTEGRAL_SYMMETRY false
$end
```

Example 12.15 Energy decomposition analysis of the water dimer with a low-cost model chemistry. The original ALMO model is used for the evaluation of polarization energy, and revPBE is chosen as the DFXC functional. Counterpoise correction for the BSSE is applied.

```
$molecule
0 1
--
0 1
H1
O1 H1 0.95641
H2 O1 0.96500  H1 104.77306
--
0 1
O2 H2 dist     O1 171.85474 H1 180.000
H3 O2 0.95822  H2 111.79807 O1 -58.587
H4 O2 0.95822  H2 111.79807 O1 58.587

dist = 2.0
$end

$rem
   JOBTYPE          eda
   EDA2             2
   METHOD           b97m-v
   BASIS            def2-svpd
   SCF_CONVERGENCE  8
   THRESH           14
   DISP_FREE_X      revPBE
   DISP_FREE_C      PBE
   EDA_BSSE         true
   INTEGRAL_SYMMETRY false
$end
```

Example 12.16 Unrestricted EDA calculation for the $H_2O\cdots Cl^{\bullet}$ complex. The fragment spin alignment procedure is performed to ensure the uniqueness of the frozen wavefunction (EDA_ALIGN_FRGM_SPIN = 2).

```
$molecule
   0 2
   --
   0 2
   Cl          0.00127        0.00000       -0.88139
   --
   0 1
   O          -0.06700        0.00000        1.72173
   H           0.50943       -0.76061        1.83598
   H           0.50943        0.76061        1.83598
$end

$rem
   JOBTYPE               eda
   METHOD                m06-2x
   BASIS                 6-31+g(d)
   EDA2                  2
   UNRESTRICTED          true
   SCF_ALGORITHM         diis
   SCF_CONVERGENCE       8
   MAX_SCF_CYCLES        200
   THRESH                14
   EDA_BSSE              true
   EDA_ALIGN_FRGM_SPIN   2
   INTEGRAL_SYMMETRY     false
   POINT_GROUP_SYMMETRY  false
$end
```

12.6.6 ALMO-EDA with Implicit Solvent Models

Since the majority of chemical processes occur in the condensed phase, it is often desirable to investigate intermolecular interactions in the presence of solvents. The solvation environment can affect intermolecular interactions in a variety of ways such that the gas-phase ALMO-EDA may not be capable of revealing the physical picture of these interactions correctly. To address this gap, Mao *et al.* have proposed the ALMO-EDA(solv) approach,[84] which, unlike many other EDA schemes, incorporates the solvation effect in the evaluation of all the energy components. Currently, ALMO-EDA(solv) supports two widely used implicit solvent models: PCM and SMD (see Sec. 11.2). More solvation models will be made compatible in future releases of Q-CHEM.

Within the ALMO-EDA(solv) scheme, the interaction energy to be decomposed is given by the energy difference between the solvated, fully relaxed complex and the sum of the energies of individually solvated, non-interacting fragments.[84] As in gas-phase ALMO-EDA, the total interaction energy (INT) can be partitioned into frozen (FRZ), polarization (POL), and charge transfer (CT) contributions:

$$\begin{aligned}\Delta E_{\mathrm{INT}}^{(\mathrm{s})} &= E_{\mathrm{Full}}^{(\mathrm{s})} - \sum_A E_A^{(\mathrm{s})} \\ &= \Delta E_{\mathrm{FRZ}}^{(\mathrm{s})} + \Delta E_{\mathrm{POL}}^{(\mathrm{s})} + \Delta E_{\mathrm{CT}}^{(\mathrm{s})}\end{aligned} \tag{12.18}$$

Here the superscript "(s)" indicates that the energetic terms are evaluated with the solvent taken into account.

The frozen interaction energy ($\Delta E_{\mathrm{FRZ}}^{(\mathrm{s})}$) is defined as the energy change upon the formation of a solvated complex from several individually solvated non-interacting fragments *without* relaxing their orbitals, which can be further decom-

posed into permanent electrostatics (ELEC), Pauli repulsion (PAULI), and dispersion (DISP) contributions:

$$\begin{aligned}\Delta E^{(\mathrm{s})}_{\mathrm{FRZ}} &= E^{(\mathrm{s})}_{\mathrm{FRZ}} - \sum_A E^{(\mathrm{s})}_A \\ &= (E^{(\mathrm{s})}_{\mathrm{FRZ}} - E^{(0)}_{\mathrm{FRZ}}) - \sum_A (E^{(\mathrm{s})}_A - E^{(0)}_A) + E^{(0)}_{\mathrm{FRZ}} - \sum_A E^{(0)}_A \\ &= \Delta E_{\mathrm{SOL}} + \Delta E^{(0)}_{\mathrm{ELEC}} + \Delta E^{(0)}_{\mathrm{PAULI}} + \Delta E^{(0)}_{\mathrm{DISP}}\end{aligned} \tag{12.19}$$

Here we have introduced a new term,

$$\Delta E_{\mathrm{SOL}} = (E^{(\mathrm{s})}_{\mathrm{FRZ}} - E^{(0)}_{\mathrm{FRZ}}) - \sum_A (E^{(\mathrm{s})}_A - E^{(0)}_A) \tag{12.20}$$

to quantify the loss/gain of solvation energy upon the formation of the frozen complex. The other three terms in Eq. 12.19, $\Delta E^{(0)}_{\mathrm{ELEC}}$), $\Delta E^{(0)}_{\mathrm{PAULI}}$, and $\Delta E^{(0)}_{\mathrm{DISP}}$, are evaluated in the same way as in vacuum[41] (as indicated by the superscripts "(0)") but using MOs of solvated fragments.

In the most general cases, the solvent contribution to the frozen interaction (ΔE_{SOL}) includes both electrostatic ($\Delta E^{\mathrm{el}}_{\mathrm{SOL}}$) and non-electrostatic ($\Delta E^{\text{non-el}}_{\mathrm{SOL}}$) components, which can be combined with the "gas-phase" ELEC and PAULI terms, respectively. In addition, we ignore the solvent contribution to dispersion, an effect that cannot be captured by dispersion-corrected DFT that ALMO-EDA(solv) is based upon, which leads to $\Delta E^{(0)}_{\mathrm{DISP}} \approx \Delta E^{(\mathrm{s})}_{\mathrm{DISP}}$. The decomposition of the frozen energy in the solvation environment (Eq. 12.19) can thus be rewritten as

$$\begin{aligned}\Delta E^{(\mathrm{s})}_{\mathrm{FRZ}} &= (\Delta E^{(0)}_{\mathrm{ELEC}} + \Delta E^{\mathrm{el}}_{\mathrm{SOL}}) + (\Delta E^{(0)}_{\mathrm{PAULI}} + \Delta E^{\text{non-el}}_{\mathrm{SOL}}) + \Delta E^{(0)}_{\mathrm{DISP}} \\ &= \Delta E^{(\mathrm{s})}_{\mathrm{ELEC}} + \Delta E^{(\mathrm{s})}_{\mathrm{PAULI}} + \Delta E^{(\mathrm{s})}_{\mathrm{DISP}}\end{aligned} \tag{12.21}$$

Starting from the solvated frozen complex, one can relax the fragment orbitals using the SCF-MI technique in presence of solvent. The associated energy lowering is defined as the polarization energy in ALMO-EDA(solv) ($\Delta E^{(\mathrm{s})}_{\mathrm{POL}}$):

$$\Delta E^{(\mathrm{s})}_{\mathrm{POL}} = E^{(\mathrm{s})}_{\mathrm{POL}} - E^{(\mathrm{s})}_{\mathrm{FRZ}} \tag{12.22}$$

where $E^{(\mathrm{s})}_{\mathrm{POL}}$ is the converged SCF-MI energy with solvent. Similarly, the charge-transfer term is given by

$$\Delta E^{(\mathrm{s})}_{\mathrm{CT}} = E^{(\mathrm{s})}_{\mathrm{Full}} - E^{(\mathrm{s})}_{\mathrm{POL}} \tag{12.23}$$

where $E^{(\mathrm{s})}_{\mathrm{Full}}$ is the full SCF energy evaluated with the presence of solvent. With that, the solvation effects are implicitly incorporated in the POL and CT terms produced by the ALMO-EDA(solv) scheme.

Example 12.17 EDA calculation for the water-Mg^{2+} complex in PCM water.

```
$molecule
2 1
--
0 1
H1
H2 H1 1.55618
O1 H2 0.97619  H1 37.14891
--
2 1
Mg1 O1 scan    H2 127.14892  H1 180.0

scan = 1.91035
$end

$rem
   JOBTYPE           eda
   EDA2              2
   METHOD            wb97m-v
   BASIS             6-31+g(d)
   UNRESTRICTED      false
   SCF_ALGORITHM     diis
   SCF_CONVERGENCE   8
   MAX_SCF_CYCLES    200
   THRESH            14
   SOLVENT_METHOD    pcm
   EDA_CLS_DISP      true
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

$PCM
   THEORY                    CPCM
   METHOD                    SWIG
   SOLVER                    INVERSION
   HPOINTS                   302
   HEAVYPOINTS               302
$END

$SOLVENT
   DIELECTRIC                78.39
$END
```

Example 12.18 EDA calculation for the water-Mg^{2+} complex in SMD water.

```
$molecule
2 1
--
0 1
H1
H2 H1 1.55618
O1 H2 0.97619  H1 37.14891
--
2 1
Mg1 O1 scan    H2 127.14892  H1 180.0

scan = 1.91035
$end

$rem
   JOBTYPE           eda
   EDA2              2
   METHOD            wb97m-v
   BASIS             6-31+g(d)
   UNRESTRICTED      false
   SCF_ALGORITHM     diis
   SCF_CONVERGENCE   8
   MAX_SCF_CYCLES    200
   THRESH            14
   SOLVENT_METHOD    smd
   EDA_CLS_DISP      true
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

$smx
   solvent water
$end
```

12.6.7 ALMO-EDA with Non-*Aufbau* Electronic Configurations

Q-CHEM also supports ALMO-EDA calculations with one or multiple fragments in non-*aufbau* electronic configurations.[82] This method is particularly useful for cases where the energetically most stable electronic state of a given fragment changes upon the formation of intermolecular complex, which is common for systems involving open-shell species such as carbenes, transition metal cations, etc. For example, the lowest-energy electronic configuration of an isolated Ba^+ radical cation is $6s^1$; however, when it is ligated with CO, the complex is more stable when the unpaired electron of Ba^+ is promoted from $6s^1$ to $5d^1$, which is thus in a non-*aufbau* state relative to the isolated Ba^+ cation. A sensible strategy to study the formation of such a complex is to perform an ALMO-EDA calculation with Ba^+ in the $5d^1$ electronic configuration, and then evaluate the energy difference between $Ba^+(5d^1)$ and $Ba^+(6s^1)$ and interpret that as the monomer "electronic preparation" energy. To ensure that the system stays in a given non-*aufbau* electronic configuration throughout, the Maximum Overlap Method[25] (MOM, see Sec. 4.5.14 for details) is applied to almost all stages in ALMO-EDA, including (i) isolated fragment calculations, (ii) calculation for the polarized wavefunction, and (iii) full SCF calculation for the whole system. Among them, (i) and (iii) are standard SCF calculations using MOM, while (ii) involves the application of MOM to an SCF-MI calculation,[82] which is compatible with the use of DIIS algorithm to solve locally projected SCF equations (with FRGM_METHOD = STOLL or GIA, see Section 12.5.1).

Specifically, an ALMO-EDA calculation for a complex within a non-*aufbau* electronic configuration (assuming only one of the fragments is excited) comprises the following steps:

- Perform ground-state SCF calculations for all fragments
- Calculate the non-*aufbau* electronic configuration for the fragment as specified in the input using MOM: the energy consumed to excite this fragment is reported in the output as the *preparation* energy (E_{prp}); construct the frozen state for the system with that one fragment in the non-*aufbau* state and the rest in ground state
- Starting from the frozen state, perform an SCF-MI calculation with MOM to obtain the polarized state within the non-*aufbau* electronic configuration
- Perform a full SCF calculation with MOM, starting from the polarized wavefunction in the non-*aufbau* state

Such a calculation requires EDA2_MOM = TRUE in addition to the setup of standard EDA2 jobs. The non-*aufbau* electronic state is specified through the *$scfmi_mom* input section, which has the following format:

```
$scfmi_mom
frag_idx1   X_1   Y_1   spin_1
frag_idx2   X_2   Y_2   spin_2
...
$end
```

In each row, the first entry specifies the index of the non-*aufbau* fragment (starting from 1). The second and third entries specify the non-*aufbau* electronic configuration, which is prepared by promoting an electron from a given fragment's HOMO$-$X to LUMO$+$Y once the ground-state SCF calculation of that fragment is finished. The last entry specifies the spin of the occupied and virtual orbitals that are being swapped ("a" for α electrons and "b" for β electrons, and α orbitals will be used by default if the fourth entry is left blank). For instance, for a HOMO$\rightarrow$LUMO excitation on the first fragment, this section would simply look like

```
$scfmi_mom
1  0  0  a
$end
```

Q-CHEM 5 also allows one to perform SCF-MI calculations for non-*aufbau* electronic configurations,[82] without going through the entire ALMO-EDA procedure. To do that, one can simply add SCFMI_MOM = TRUE to the *$rem* setup of an SCF-MI calculation (see Section 12.6.2). The non-*aufbau* electronic configuration can be specified through the *$scfmi_mom* section in the same way.

EDA2_MOM

Perform ALMO-EDA calculation with non-*aufbau* electronic configurations using MOM

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard ALMO-EDA calculation

TRUE ALMO-EDA for non-*aufbau* states

RECOMMENDATION:

None

SCFMI_MOM

Perform an SCF-MI calculation with non-*aufbau* electronic configurations using MOM

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard SCF-MI calculation

TRUE SCF-MI calculation with MOM

RECOMMENDATION:

None

Note that EDA2_MOM and SCFMI_MOM can be used without explicitly setting the *$scfmi_mom* section, where the electronic configuration of the frozen state constructed from two ground-state fragments will be preserved in the SCF-MI or ALMO-EDA calculation.

Example 12.19 ALMO-EDA calculation for the $[Ba(CO)]^+$ complex with Ba^+ in the $5d^1$ electronic configuration at the B3LYP/def2-TZVP level. The *$scfmi_mom* section specifies that the unpaired α electron in the 6s orbital is promoted to one of the 5d orbitals.

```
$molecule
1 2
--
1 2
Ba   0.0  0.0  0.0
--
0 1
C    0.0  0.0  2.734
O    0.0  0.0  3.876
$end

$rem
   JOBTYPE           EDA    !by default EDA2 = 2 with CT analysis
   METHOD            B3LYP
   BASIS             DEF2-TZVP
   ECP               DEF2-ECP
   UNRESTRICTED      TRUE
   SCF_CONVERGENCE   8
   SCF_ALGORITHM     DIIS
   EDA2_MOM          TRUE
$end

$scfmi_mom
1 0 2
$end
```

12.6.8 ALMO-EDA with Non-Perturbative Polarization and Charge Transfer Analysis

Q-CHEM also supports the non-perturbative[112] (variational) energy and charge decomposition analysis in the second-generation ALMO-EDA framework. The advantage of this method over the perturbative CT analysis method is that the energy and charge decompositions are both exact and there are no higher-order terms left. Currently, this method is implemented for both restricted and unrestricted SCF calculations, and is eligible to analyze both polarization[105] (POL) and charge transfer (CT) processes of intermolecular interactions.

12.6.8.1 Non-Perturbative Polarization and Charge Transfer Analysis with the GEN_SCFMAN_EDA2 Driver

The non-perturbative polarization and charge transfer analysis can be performed using the GEN_SCFMAN_EDA2 driver (GEN_SCFMAN_EDA2 = TRUE). Perturbative CT analysis can be invoked by setting EDA_PCT_A = TRUE, while non-perturbative POL and CT analysis can be invoked by setting EDA_POL_A = TRUE and EDA_VCT_A = TRUE, respectively. The POL analysis[105] uses the recently developed fragment density matrix connecting method and can break the energy lowering and charge shift in the POL process into exactly fragment-wise additive sums. The CT analysis will print out the pairwise energy and charge changes between fragments as matrices, with fragment labels starting from 0, and the interpretation is that the energy lowering and charge transfer happens due to electron donation from the fragments labeled in the rows to the fragments labeled in the columns.

The default orbital analysis method is the complementary occupied-virtual pairs (COVP) method. To select the significant COVPs, set EDA_COVP_THRESH = N to print out the COVPs that contributes more than $0.001 \times N$ kJ/mol to the energy decrease, and the default value is set to 500. To visualize the COVPs, set EDA_SAVE_COVP = TRUE, MAKE_CUBE_FILES = TRUE and PLOTS = TRUE to save the orbitals as cube files, which can be visualized using IQmol, VMD, or other visualization software. The natural orbitals for chemical valence (NOCV) analysis can also be performed by setting EDA_NOCV = TRUE. To select the significant NOCVs, set EDA_NOCV_THRESH = N to print out NOCVs that contributes more than $0.001 \times N$ kJ/mol to energy decrease, and the default value is also set to 500. The NOCV analysis includes the POL process, CT process, and the traditional NOCV results which combine the POL and CT processes. As pointed out in this paper[104], the ETS-NOCV method approximates the effective Fock matrix integration with only one quadrature, which is set by EDA_NOCV_QUADRATURE =1, and Q-CHEM also supports quadrature number 3 and 5.

Unrestricted analysis will be used if at least one of the fragments has an open-shell structure. It can also be forced by setting UNRESTRICTED=TRUE. One caveat is that the current implementation does not support the OCC_RI_K algorithm, and the calculation will crash if this REM is set true.

GEN_SCFMAN_EDA2

Perform ALMO-EDA calculations using the GEN_SCFMAN_EDA2 driver (differing from jobs with EDA2 > 0)

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use the new ALMO-EDA framework

TRUE Use the new ALMO-EDA framework

RECOMMENDATION:

Set to TRUE to perform non-perturbative CT analysis using this driver

EDA_PCT_A

Perform perturbative CT analysis

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform perturbative CT analysis

TRUE Perform perturbative CT analysis

RECOMMENDATION:

Set to TRUE to perform perturbative CT analysis

EDA_VCT_A

Perform non-perturbative CT analysis

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform non-perturbative CT analysis

TRUE Perform non-perturbative CT analysis.

RECOMMENDATION:

Set to TRUE to perform non-perturbative CT analysis

EDA_POL_A

Perform EDA for polarization process

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform EDA for polarization process

TRUE Perform EDA for polarization process

RECOMMENDATION:

Set to TRUE to perform EDA for polarization process

EDA_COVP_THRESH

Specifies the significance above which the COVPs will be saved

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

N COVPs that contributes more than $0.001 \times N$ kJ/mol in energy decrease will be saved

RECOMMENDATION:

None

EDA_SAVE_COVP

Save significant COVPs or not

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not save significant COVPs

TRUE Save significant COVPs

RECOMMENDATION:

Set to TRUE to save COVPs. Note that REMs for plotting cube files need also be set

EDA_NOCV

Perform NOCV analysis

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do NOCV analysis

TRUE Do NOCV analysis

RECOMMENDATION:

None

EDA_NOCV_THRESH

Specifies the significance above which the NOCVs will be saved

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

N NOCVs that contributes more than $0.001 \times N$ kJ/mol in energy decrease will be saved

RECOMMENDATION:

None

EDA_SAVE_NOCV

Save significant COVPs or not

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not save significant NOCVs

1 Save significant NOCVs

RECOMMENDATION:

Set to 1 to save NOCVs. Note REMs for plotting cube files need also be set

EDA_NOCV_QUADRATURE

Number of quadratures used to integrate effective fock matrix

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use 1 quadrature

3 Use 3 quadratures

5 Use 5 quadratures

RECOMMENDATION:

Most of the time, one quadrature is enough. However, in cases where the NOCV energy decreases are very different from the corresponding COVP results, it is recommended to increase the quadrature numbers.

Example 12.20 Restricted EDA calculation for the H_2O-Na^+ system with non-perturbative POL and CT analysis.

```
$molecule
1 1
--
0 1
H          -0.73946         0.94887         0.78379
O          -1.16910         0.63297        -0.02844
H          -2.12156         0.70793         0.14730
--
1 1
Na         -0.17266        -0.04338        -1.86190
$end

$comment
EDA2 (R) calculation with variational POL and CT analyses
based on the gen_scfman_eda2 driver
$end

$rem
method                B3LYP
scf_print_frgm        1
gen_scfman_final      1
scf_print_frgm        1
basis                 6-31G*
scf_algorithm         diis
thresh                12
incfock               false
mem_total             16000
scf_convergence       7
scf_final_print       2
child_mp              1
child_mp_orders       232 ! nDQ
gen_scfman_eda2       1
eda_pol_a             1
eda_pct_a             0
eda_vct_a             1
eda_covp_thresh       500
eda_save_covp         0
make_cube_files       true
plots                 true
point_group_symmetry  false
integral_symmetry     false
$end

$plots
 grid_points 50 50 50
$end
```

Example 12.21 Unrestricted EDA calculation for the CH_3-Na^+ system with non-perturbative POL and CT analysis.

```
$molecule
1 2
--
0 2
C -1.447596 -0.000023  0.000019
H -1.562749  0.330361 -1.023835
H -1.561982  0.721445  0.798205
H -1.561187 -1.052067  0.225866
--
1 1
Na  1.215591  0.000036 -0.000032
$end

$comment
EDA2 (U) calculation with variational POL and CT analyses
based on the gen_scfman_eda2 driver
$end

$rem
method               B3LYP
scf_print_frgm       true
basis                aug-cc-PVTZ
scf_algorithm        diis
thresh               12
incfock              false
mem_total            16000
scf_convergence      7
child_mp             1
child_mp_orders      232 ! nDQ
gen_scfman_eda2      1
eda_pol_a            1
eda_pct_a            0
eda_vct_a            1
eda_covp_thresh      500
eda_save_covp        0
make_cube_files      true
plots                true
point_group_symmetry false
integral_symmetry    false
$end

$plots
 grid_points 50 50 50
$end
```

Example 12.22 Unrestricted EDA calculation for the Rn^{+}-CH_4 system with non-perturbative POL and CT analysis and NOCV analysis

```
$molecule
1 2
--
0 1
C          0.0551597051    0.0364080371   -0.0375528310
H         -0.0563410694   -1.0144740143   -0.3153331540
H         -0.0537558345    0.7868509829   -0.8318467516
H         -0.8718690962    0.2438110339    0.5677588397
H          1.0408273042    0.1810559688    0.4109829953
--
1 2
Rn        -0.2646060092    0.6373989916    3.0026329017
$end

$rem
method              b3lyp
gen_scfman_final    1
scf_print_frgm      1
basis               def2-tzvp
ecp                 def2-ecp
scf_algorithm       diis
thresh              12
incfock             0
mem_total           16000
scf_final_print     2
iprint              20000000
scf_convergence     7
child_mp            1
child_mp_orders     232 ! nDQ
gen_scfman_eda2     1
eda_pol_a           1
eda_pct_a           0
eda_vct_a           1
eda_save_covp       0
eda_nocv            1
eda_save_nocv       0
eda_covp_thresh     500
eda_nocv_thresh     500
eda_nocv_quadrature 1
make_cube_files     true
plots               true
point_group_symmetry false
integral_symmetry    false
$end

$plots
 grid_points 100 100 100
$end
```

12.6.8.2 Non-Perturbative Polarization and Charge Transfer Analysis with the EDA2 Driver

In Q-CHEM 6.1 and after, the non-perturbative POL and CT analysis methods are enabled in EDA2 driver so that features available in EDA2 can be used, such as frozen energy decomposition, inclusion of solvation models, etc. A

sample input file is shown below.

Example 12.23 Unrestricted EDA calculation for the CH_3-Na^+ system with non-perturbative POL and CT analysis using the EDA2 driver. The significant COVPs will be plotted.

```
$molecule
1 2
--
0 2
C   -1.447596 -0.000023  0.000019
H   -1.562749  0.330361 -1.023835
H   -1.561982  0.721445  0.798205
H   -1.561187 -1.052067  0.225866
--
1 1
Na   1.215591  0.000036 -0.000032
$end

$comment
EDA2 (U) calculation with variational POL and CT analyses
based on the eda2 driver
$end

$rem
jobtype              eda
eda2                 1
method               B3LYP
point_group_symmetry false
basis                6-31g*
scf_algorithm        gdm
thresh               12
eda_pol_a            true
eda_vct_a            true
eda_covp_thresh      500
eda_save_covp        true
eda_nocv             true
eda_save_nocv        false
eda_nocv_thresh      500
eda_nocv_quadrature  1
make_cube_files      true
plots                true
$end

$plots
 grid_points 50 50 50
$end
```

12.6.9 Visualization Tools Associated with ALMO-EDA

The following visualization tools are enabled in EDA2:

- Automated generation of complementary occupied-virtual pairs (COVP)
- Electron density difference (EDD) maps between intermediate states (FRZ→POL, POL→Full)
- Plots for Natural Orbitals for Chemical Valence (NOCV)
- Plots of frozen and polarized ALMOs

As introduced in Sec. 12.5.4, the COVPs can help elucidate the details of a charge-transfer process by showing the chemically most relevant donor-acceptor orbitals. In its implementation in EDA2, we enabled an automated selection of the most significant occupied-virtual pairs (based on a threshold on singular values). The MO cube files of these selected COVPs are then generated, and thus there is no need to specify which orbitals to plot. This new feature can be turned on by setting EDA_COVP_PRINT = AUTOMATED. Also, both the old and new formats of the *$plots* section are supported for automated COVP cube generation in EDA2. The old format requires MAKE_CUBE_FILES = TRUE and the new format requires PLOTS = TRUE. The plotted COVPs are indexed as `covp_a.N.cube` and `covp_b.N.cube` and the energetic significance of each of them can be looked up from the output file.

EDA2 also enabled electron density difference (EDD) plots to show the redistribution of electron density upon polarization ($\Delta\mathbf{P}_{\mathrm{pol}} = \mathbf{P}_{\mathrm{pol}} - \mathbf{P}_{\mathrm{frz}}$) and charge transfer ($\Delta\mathbf{P}_{\mathrm{ct}} = \mathbf{P}_{\mathrm{full}} - \mathbf{P}_{\mathrm{pol}}$). For unrestricted ALMO-EDA calculations, the spin density at FRZ, POL, and fully relaxed states are plotted together. Another related quantity that can be visualized is the so-called natural orbitals for chemical valence (NOCV),[89] which are defined as the eigenvectors of $\Delta\mathbf{P} = \mathbf{P}_{\mathrm{full}} - \mathbf{P}_{\mathrm{frz}}$. The NOCVs appear in pairs ψ_k and ψ_{-k}, whose associated eigenvalues are n_k and $-n_k$, respectively. The energy lowering associated with each pair of NOCVs can be calculated using the extended transition state (ETS) approach (see Ref. 89 for details). The NOCVs are useful tools for illustrating the underlying orbital interactions, including both polarization and charge transfer, in chemical bonding.

In EDA2, the EDD maps are plotted when EDA_PLOT_DIFF_DEN = TRUE. The calculation of NOCVs are performed when EDA_NOCV > 0. The most significant NOCVs are automatically selected based on a threshold on the eigenvalues $\{n_k\}$. When EDA_NOCV = 1, Q-CHEM will only plot the contribution from each significant NOCV pair (ψ_{-k}, ψ_k) to the density deformation ($\Delta\rho_k$):

$$\Delta\rho_k(\mathbf{r}) = -n_k|\psi_{-k}(\mathbf{r})|^2 + n_k|\psi_k(\mathbf{r})|^2 \tag{12.24}$$

When EDA_NOCV = 2, Q-CHEM will plot not only the NOCV pair contributions to density deformation but also the NOCVs themselves. Note that the new format of the *$plots* section is required for these visualizations (see Sec. 10.5.4.1 for details).

Finally, Q-CHEM 5.2.2 enables the visualization of frozen and polarized ALMOs, which is controlled by *$rem* variables PLOT_ALMO_FRZ and PLOT_ALMO_POL. The user needs to specify which orbitals to plot for each fragment through the *$almo_print* section:

```
$almo_print
   frgm_idx1   orb1  orb2 ... (spin)
   frgm_idx2   orb1  orb2 ... (spin)
   . . .
$end
```

One can use the format "orb1:orb2" to specify a range of orbitals to plot for each fragment. For unrestricted cases, at the end of each line one can write "a" or "b" to specify whether alpha or beta orbitals are plotted (alpha orbitals will be plotted by default if there is no specification). As above, a *$plots* section with its new format is required for the visualization of ALMOs.

In the following table, we summarize the names of the cube files generated by each type of plots. Note that for the EDD plots, "0" refers to the EDD between POL and FRZ states, while "1" refers to the EDD between full SCF and POL states; for the spin density plots, "0", "1", and "2" correspond to the FRZ, POL, and fully relaxed states, respectively.

Table 12.1: Cube file names generated by visualization tools in EDA2

Plot type	Orbital type	Cube file name
COVP	R	`covp_a.N.cube`
	U	`covp_a.N.cube`, `covp_b.N.cube`
EDD	R	`dens.0.cube`, `dens.1.cube`
	U	`dens_alpha.0.cube`, `dens_alpha.1.cube`
		`dens_beta.0.cube`, `dens_beta.1.cube`
		`dens_spin.0.cube`, `dens_spin.1.cube`, `dens_spin.2.cube`
NOCV	R	`nocv_diffden_a.N.cube`, `nocv_a.N.cube`
	U	`nocv_diffden_a.N.cube`, `nocv_a.N.cube`
		`nocv_diffden_b.N.cube`, `nocv_b.N.cube`
ALMO	R	`almo_frz_a.N.cube`, `almo_pol_a.N.cube`
	U	`almo_frz_a.N.cube`, `almo_frz_b.N.cube`
		`almo_pol_a.N.cube`, `almo_pol_b.N.cube`

EDA_PLOT_DIFF_DEN

Plot changes in electron density due to POL and CT

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not make EDD plots

TRUE Make EDD plots

RECOMMENDATION:

None

EDA_NOCV

Perform the NOCV analysis and plot the significant NOCVs

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform NOCV analysis

1 Plot NOCV pair contributions to density deformation

2 Plot both NOCV pair contribution to density deformation and NOCV orbitals

RECOMMENDATION:

None

PLOT_ALMO_FRZ

Plot ALMOs at the frozen stage of EDA2 calculations

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not plot frozen ALMOs

TRUE Plot frozen ALMOs

RECOMMENDATION:

None

PLOT_ALMO_POL

Plot ALMOs after the polarization calculation

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not plot polarized ALMOs

TRUE Plot polarized ALMOs

RECOMMENDATION:

None

Example 12.24 Performing perturbative CTA through EDA2 (using the default settings). The automatic COVP generation is enabled by setting EDA_PRINT_COVP = AUTOMATED, which plots the most significant COVPs automatically. The new format of the *$plots* section is used by setting PLOTS = TRUE, and there is no need to specify how many orbitals to plot.

```
$molecule
   0 1
   --
   0 1
   O      -1.521720     0.129941     0.000000
   H      -1.924536    -0.737533     0.000000
   H      -0.571766    -0.039961     0.000000
   --
   0 1
   O       1.362840    -0.099704     0.000000
   H       1.727645     0.357101    -0.759281
   H       1.727645     0.357101     0.759281
$end

$rem
   JOBTYPE               EDA
   METHOD                B3LYP
   BASIS                 6-31+G(d)
   MEM_TOTAL             8000
   MEM_STATIC            2000
   BASIS_LIN_DEP_THRESH  6
   THRESH                14
   SCF_CONVERGENCE       8
   MAXSCF                200
   EDA_COVP              TRUE
   EDA_PRINT_COVP        AUTOMATED !auto-generation of covp cube files
   MAKE_CUBE_FILES       TRUE
   PLOTS                 TRUE      !new format for the plot section
   INTEGRAL_SYMMETRY     FALSE
   POINT_GROUP_SYMMETRY  TRUE
$end

$plots
   GRID_POINTS           100 100 100
$end
```

Example 12.25 Electron density difference (EDD) plots and NOCV analysis for the $NH_3 \cdots BH_3$ complex. Both the significant NOCV pairs and each pair's contribution to density deformation due to orbital interaction (polarization and charge transfer) are plotted.

```
$molecule
0 1
--
0 1
N          0.0000001517    0.7279666667    0.0000000000
H          0.9488005016    1.0881357449    0.0000000000
H         -0.4743994984    1.0881371276   -0.8216800000
H         -0.4743994984    1.0881371276    0.8216800000
--
0 1
B         -0.0000014567   -0.9275533333    0.0000000000
H         -1.1719117610   -1.2408021948    0.0000000000
H          0.5859582390   -1.2408039026   -1.0149100000
H          0.5859582390   -1.2408039026    1.0149100000
$end

$rem
   JOBTYPE                 eda
   EDA2                    2   !ALMO-POL
   METHOD                  b3lyp
   BASIS                   6-31g(d)
   SCF_ALGORITHM           diis
   XC_GRID                 1
   SCF_CONVERGENCE         8
   MAX_SCF_CYCLES          200
   THRESH                  14
   EDA_PLOT_DIFF_DEN TRUE  !plot EDD maps
   EDA_NOCV  2             !NOCV analysis
   INTEGRAL_SYMMETRY       false
   POINT_GROUP_SYMMETRY    false
$end

$plots
grid_points 100 100 100
$end
```

Example 12.26 Plot HOMO and LUMO for each fragment (frozen and polarized ALMOs) in an EDA2 calculation

```
$molecule
0 1
--
0 1
H1
O1 H1 0.95641
H2 O1 0.96500  H1 104.77306
--
0 1
O2 H2 dist     O1 171.85474 H1 180.000
H3 O2 0.95822  H2 111.79807 O1 -58.587
H4 O2 0.95822  H2 111.79807 O1 58.587

dist = 2.0
$end

$rem
   JOBTYPE          eda
   METHOD           b3lyp
   BASIS            6-31g
   EDA2             2   !ALMO-POL
   UNRESTRICTED     false
   SCF_ALGORITHM    diis
   SCF_CONVERGENCE  8
   MAX_SCF_CYCLES   200
   THRESH           14
   PLOT_ALMO_FRZ    true
   PLOT_ALMO_POL    true
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY false
$end

$almo_print
1 5:6
2 5:6
$end

$plots
grid_points 60 60 60
$end
```

12.7 Additional ALMO-EDA Capabilities

12.7.1 ALMO-EDA for the MP2 Method

The previously described EDA methods are limited to SCF methods such as HF and DFT. However, for many systems, it is preferable to use a wave function based correlation method. For this reason, the ALMO-EDA has been extended to MP2.[110,111] The MP2 ALMO-EDA is based on the first-generation ALMO-EDA. It provides an MP2 correction to the FRZ, POL, and CT terms defined by the ALMO-EDA and also adds in a term corresponding to the London dispersion force. This is done by defining a constrained intermediate MP2 wave function corresponding to each HF intermediate.

The current implementation is limited to only RI-MP2 rather than full MP2, and only works in the closed shell, spin restricted case. Frozen core and spin scaling are also not yet supported. Attempting to use the EDA with a correlation

method other then RI-MP2 or with unrestricted orbitals will result in a crash. Frozen core and spin scaling settings will be ignored with a warning by the EDA, but not by the final energy, leading to inconsistent results.

Though the MP2 EDA is based on the first generation ALMO-EDA, the code path and REM settings are shared with the second generation ALMO-EDA. The MP2 ALMO-EDA does not define any new REM variables of its own. Rather, running an EDA job with EDA2 and GEN_SCFMAN will trigger an MP2 ALMO-EDA when the correlation method is RI-MP2. The correlation setting causes the SCF portion of the EDA to be switched back to the original scheme and will also decompose the correlation energy. Most settings intended for the second generation ALMO-EDA are not supported, but EDA_NO_CT and EDA_BSSE are. An example appears below.

Example 12.27 MP2 energy decomposition analysis of the water dimer.

```
$molecule
0 1
--
0 1
O  -0.031783  -0.057754   0.000000
H  -0.415035   0.819269   0.000000
H   0.919546   0.097478   0.000000
--
0 1
O   2.960796   0.171800   0.000000
H   3.290569  -0.313410  -0.758561
H   3.290569  -0.313410   0.758561
$end

$rem
   JOBTYPE          EDA
   GEN_SCFMAN       TRUE
   EDA2             TRUE
   FRGM_METHOD      STOLL
   EXCHANGE         HF
   CORRELATION      RIMP2
   BASIS            aug-cc-pVTZ
   AUX_BASIS        rimp2-aug-cc-pVTZ
   THRESH           14
   SCF_CONVERGENCE  10
   N_FROZEN_CORE    0
   EDA_BSSE         TRUE
   USE_LIBQINTS     0
   INTEGRAL_SYMMETRY FALSE
$end
```

12.7.2 ALMO-EDA for Bonded Interactions

EDA schemes have been very successful at elucidating the nature of noncovalent interactions. On the other hand, these methods are often inadequate for the analysis of covalent bonds. In fragment-based approaches, the key difficulty arises from the need to correctly spin-couple two open-shell radical fragments into a closed-shell bond in a spin-pure way. The ALMO-EDA methodology was extended by Levine to accomplish this within HF and KS DFT[67–69]. If HF is used, the final wave function whose interaction energy is being decomposed is the CAS(2,2)/1-pair perfect-pairing/TCSCF wave function. At present, only a single bond may be analyzed by these schemes.

The method begins with two doublet radical fragments, each of which is described by a restricted open-shell (RO) Hartree-Fock (HF) or Kohn-Sham DFT single determinant. In the bonded EDA scheme, because orbital re-hybridization can play such a large role in the energy, ΔE_{PREP} includes the energy required to distort each radical fragment to the geometry that it adopts in the bonded state ΔE_{GEOM}, as well as an electronic preparation due to orbital re-hybridization

ΔE_{HYBRID}. For example, an F atom has an unpaired electron in a p orbital, while an F atom in a bond will be sp-hybridized. The amine radical, NH_2, is sp^2-hybridized with an unpaired electron in a p orbital, while an amine group is often sp^3- or sp^2-hybridized with a lone pair in the p orbital in a molecule. Then,

$$\Delta E_{\text{PREP}} = \Delta E_{\text{GEOM}} + \Delta E_{\text{HYBRID}} \tag{12.25}$$

This re-hybridized state is obtained by relaxing the ALMO supersystem obtained from the fragments, permitting only on-fragment doubly-occupied–singly-occupied orbital rotations (which are well-defined due to the RO nature of the fragments). The so-optimized fragments are then re-separated and the energy difference between the electronically distorted fragments and the ground state fragments is ΔE_{HYBRID}. This corresponds to fixing the α-density and allowing the β-hole to re-optimize in the span of that α-density. This is a kind of polarization, although we draw a distinction from the electronic polarization step which appears later in both the bonded and non-bonded ALMO schemes. Another reason why it makes sense to place the re-hybridization energy here is that it is already partially accounted for by the fact that the geometry of the radical fragment is fixed to be that of the interacting fragment. For instance, free methyl radical is an sp^2-hybridized planar molecule, while a methyl group in a bond is a pyramidalized sp^3 fragment; the re-hybridzation cost was already paid by the geometric distortion.

The FRZ energy in the bonded scheme corresponds to the ALMO supersystem formed by combining the RO fragments to form a spin-pure triplet single-determinant wave function without allowing the orbitals to relax further. This term is entirely a non-bonded interaction and will typically be repulsive for a chemical bond because of Pauli repulsion. It includes contributions from inter-fragment electrostatics, Pauli repulsion, exchange-correlation, and dispersion. The EDA2 frozen decomposition scheme may be applied to this term in the spin-projected formalism (*i.e.*, BONDED_EDA = 2).

A new term is introduced for the bonded EDA scheme: ΔE_{SC} of the spin-coupling energy. This energy difference is caused by electron pairing and loosely corresponds to the idea of covalency. Like FRZ, SC will be evaluated with frozen orbitals, but while FRZ is typically strongly repulsive (dominated by Pauli repulsion), SC is typically strongly attractive in the overlapping regime associated with covalent bond formation. For this reason and because we are primarily interested in the singlet surface (as opposed to the triplet surface of the initial supersystem), FRZ and SC may be grouped together into a total frozen orbital term (FRZ + SC).

In the KS DFT scheme,[68] this spin-coupled wave function is formed by forming the broken-symmetry DFT determinant and spin-projecting out the triplet contaminant. Since the wave function is constructed from RO fragment, spin-contamination can only occur within the half-occupied space and hence the triplet contaminant is the only possible contaminant. We therefore obtain and exact singlet wave function. For HF, this is exactly equivalent to the scheme based on nonorthogonal CI,[69] so long as there are only two unpaired spins among the fragments (*i.e.*, the supersystem is closed shell). This is usually the case and so, as the schemes are equivalent, we advocate only using the spin-projected formalism as it is much more efficient.

The POL and CT terms are similarly defined as in the non-bonded ALMO scheme. FERFs may be used to define polarization but monopole FERFs, which describe the expansion or contraction of orbitals (which occurs in some cases on bond formation should be included).[67] The CT term gives an indication of the level of ionic character in the bond. Taken together the various terms describe a fingerprint for the bond being studied. Further details for how to analyze the results may be found in the referenced literature.

Considerations for using the bonded ALMO-EDA:

- SCFMI_MODE = 1 is required.
- ROSCF = TRUE must be set.
- There are no presets of the bonded ALMO-EDA. Therefore, set EDA2 = 10.

- DIIS may not be used for SCF_ALGORITHM. Use GDM_LS for BONDED_EDA = 2 and L_BFGS for BONDED_EDA = 1.

BONDED_EDA

Use the bonded ALMO-EDA.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform bonded ALMO-EDA.

1 Perform ALMO-EDA with non-orthogonal CI.

2 Perform ALMO-EDA with spin-projected formalism.

RECOMMENDATION:

Set to 2 for all cases where the supersystem is closed shell, only use 1 for cases where the fragments have more than one unpaired spin each.

EDA_CONTRACTION_ANAL

Perform analysis separating orbital contraction from the rest of POL.

TYPE:

BOOLEAN

DEFAULT:

0

OPTIONS:

FALSE Do not perform contraction analysis.

TRUE Perform contraction analysis.

RECOMMENDATION:

No recommendation

Example 12.28 Bonded EDA of F_2 with MDQ FERFs, frozen analysis in the spin-projected formalism

```
$molecule
   0 3
   --
   0 2
   F 0.0 0.0 0.0
   --
   0 2
   F 0.0 0.0 1.382
$end

$rem
   JOBTYPE              eda
   EXCHANGE             wb97x-d
   BASIS                aug-cc-pvdz
   EDA2                 10
   BONDED_EDA           2
   SCF_CONVERGENCE      6
   MAX_SCF_CYCLES       200
   ROSCF                true
   SCF_GUESS            fragmo
   SCF_ALGORITHM        gdm_ls
   SCFMI_MODE           1
   SCF_PRINT_FRGM       true
   CHILD_MP             true
   CHILD_MP_ORDERS      1233
   FRZ_RELAX            true
   FRZ_RELAX_METHOD     2
   FRZ_ORTHO_DECOMP     false
   INTEGRAL_SYMMETRY    false
   POINT_GROUP_SYMMETRY false
   EDA_CLS_ELEC         true
$end

$rem_frgm
   scf_convergence   7
   scf_algorithm     gdm_ls
   scf_guess         sad
$end
```

Example 12.29 Bonded EDA of CH with MDQ FERFS, contraction analysis in the non-orthogonal CI formalism.

```
$molecule
   0 4
   --
   0 3
   C          0.0000000000    0.0000000000   -0.0525358999
   --
   0 2
   H          0.0000000000    0.0000000000    1.0525358999
$end

$rem
   JOBTYPE                 eda
   EXCHANGE                hf
   BASIS                   aug-cc-pvdz
   EDA2                    10
   BONDED_EDA              1
   SCF_CONVERGENCE         6
   MAX_SCF_CYCLES          2000
   ROSCF                   true
   SCF_GUESS               fragmo
   SCF_ALGORITHM           l_bfgs
   SCFMI_MODE              1
   SCF_PRINT_FRGM          true
   CHILD_MP                true
   CHILD_MP_ORDERS         1233
   FRZ_RELAX               true
   FRZ_RELAX_METHOD        2
   EDA_CONTRACTION_ANAL    true
   INTEGRAL_SYMMETRY       false
   POINT_GROUP_SYMMETRY    false
$end

$rem_frgm
   scf_convergence   7
   scf_algorithm     gdm_ls
   scf_guess         sad
$end
```

12.7.3 Adiabatic ALMO-EDA and VFB Analysis

Despite the huge success and usefulness of today's most popular EDA methods, they still face some limitations in their capabilities. For instance, EDAs are usually performed at complex geometries that are obtained from unconstrained electronic structure calculations (*e.g.*, optimized equilibrium geometries). For strongly interacting systems, close intermolecular contacts driven by POL and particularly CT often result in largely unfavorable FRZ interaction, which offers little physical insights besides indicating obviously substantial intermolecular overlap. Another limitation is that the conventional EDA methods often partitions a "single-point" interaction energy evaluated at a given geometry. Therefore, the influence of FRZ, POL and CT on the structural and vibrational properties of an intermolecular complex cannot be directly characterized.

Recently Mao *et al.* reformulated the original ALMO-EDA method in an adiabatic picture,[80] where the term "adiabatic" is borrowed from spectroscopy and indicates that energy differences are evaluated at relaxed geometry on each potential energy surface. In this scheme, the total binding energy (including monomer geometry distortions) is repartitioned into

adiabatic FRZ, POL and CT terms:

$$\Delta E_{\text{bind}} = \Delta E_{\text{frz}}^{(\text{ad})} + \Delta E_{\text{pol}}^{(\text{ad})} + \Delta E_{\text{ct}}^{(\text{ad})}. \tag{12.26}$$

The adiabatic frozen interaction energy is given by the difference between the energy minimum of the frozen potential surface (on which the energy of each point is computed using the corresponding frozen wave function) and the sum of fully relaxed, non-interacting fragment energies:

$$\Delta E_{\text{frz}}^{(\text{ad})} = E[\mathbf{P}_{\text{frz}}^{(\text{frz})}] - \sum_A E_A^{(0)}. \tag{12.27}$$

Similarly, the adiabatic POL and CT terms can be obtained by performing geometry optimizations on the polarized (SCF-MI) and fully relaxed (unconstrained SCF) potential surfaces:

$$\Delta E_{\text{pol}}^{(\text{ad})} = E[\mathbf{P}_{\text{pol}}^{(\text{pol})}] - E[\mathbf{P}_{\text{frz}}^{(\text{frz})}] \tag{12.28}$$

$$\Delta E_{\text{ct}}^{(\text{ad})} = E[\mathbf{P}_{\text{full}}^{(\text{full})}] - E[\mathbf{P}_{\text{pol}}^{(\text{pol})}]\,. \tag{12.29}$$

With this method, the changes in monomer structures and intermolecular coordinates due to FRZ, POL and CT and the accompanied energetics are provided. Moreover, at the energy minimum (or other stationary points) on each potential surface, the other properties such as multipole points, vibrational frequencies and intensities can also be computed, therefore the effect of different intermolecular interaction components on them can also be characterized.

The geometry optimization on the frozen potential surface is facilitated by the analytical gradient of the frozen wave function energy implemented in Q-CHEM. As for the geometry optimization on the polarized potential surface, the nuclear gradient of the SCF-MI energy has the same form as that of the full SCF energy if the original ALMO model is used. These analytical gradients can also be used for finite difference calculations of harmonic frequencies by setting IDERIV = 1. We note that the analytical gradients of SCF-MI calculations that use FERFs are not available yet, and SCFMI_MODE = 0 is required for computing the forces on the frozen and polarized potential surfaces. Also, the current implementation of this method requires users to perform geometry optimization separately on the three potential surfaces (see the example below) and then evaluate the energy components by taking several Q-CHEM outputs (including geometry optimizations for the monomers) together, which is probably not so convenient. We look forward to extending the functionality of this method and improving its implementation in the future.

As mentioned in Section 12.6.5, for systems containing radicals of highly symmetric geometries, the frozen wavefunction obtained from concatenating the fragment MOs might be non-unique. In those cases, we recommend the user to set EDA_ALIGN_FRGM_SPIN = 1 or 2 when performing geometry optimization on the frozen potential surface. The job will then go through the fragment spin alignment procedure in each optimization cycle.

FRZ_GEOM

Compute forces on the frozen potential surface.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute forces on the frozen potential surface.

TRUE Compute forces on the frozen potential surface.

RECOMMENDATION:

Set it to TRUE when optimized geometry or vibrational frequencies on the frozen potential surface are desired.

POL_GEOM

Compute forces on the polarized (converged SCF-MI) potential surface.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute forces on the polarized potential surface.

TRUE Compute forces on the polarized potential surface.

RECOMMENDATION:

Set it to TRUE when optimized geometry or vibrational frequencies on the polarized potential surface are desired.

Example 12.12.30 Geometry optimization of the ammonia-borane complex on the fully relaxed, polarized, and frozen potential energy surfaces successively.

View input online

Example 12.31 Geometry optimization of the [Cu(CO)]$^+$ complex on the frozen potential surface, followed by a frequency calculation which is computed via finite differences.

```
$molecule
1 1
--
0 1
C          0.0000000000    0.0000000000    1.3792049588
O          0.0000000000    0.0000000000    2.4988670685
--
1 1
Cu         0.0000000000    0.0000000000   -0.9778656750
$end

$rem
   JOBTYPE            opt
   FRZ_GEOM           true
   METHOD             b3lyp
   BASIS              def2-svp
   UNRESTRICTED       false
   IDERIV             1
   FD_MAT_VEC_PROD    false
   INTEGRAL_SYMMETRY  false
   POINT_GROUP_SYMMETRY true
$end


@@@


$molecule
   read
$end

$rem
   JOBTYPE           freq
   FRZ_GEOM          true
   METHOD            b3lyp
   BASIS             def2-svp
   UNRESTRICTED      false
   IDERIV            1
   FD_MAT_VEC_PROD   false
   INTEGRAL_SYMMETRY false
   POINT_GROUP_SYMMETRY true
$end
```

To further understand the charge-transfer effects in dative complexes, in Q-CHEM 5.2.2 and after, one is allowed to separate the overall CT into contributions from forward and backward donations using the variational forward-backward (VFB) approach.[78] Such a decomposition is achieved by introducing two additional constrained intermediate states in which only one direction of CT is permitted. These two "one-way" CT states are variationally relaxed such that the associated nuclear forces can be readily obtained. This allows for a facile integration into the adiabatic ALMO-EDA scheme introduced above:

$$\Delta E_{\mathrm{ctf}}^{(\mathrm{ad})} = E[\mathbf{P}_{\mathrm{ctf}}^{(\mathrm{ctf})}] - E[\mathbf{P}_{\mathrm{pol}}^{(\mathrm{pol})}], \tag{12.30}$$

$$\Delta E_{\mathrm{ctb}}^{(\mathrm{ad})} = E[\mathbf{P}_{\mathrm{ctb}}^{(\mathrm{ctb})}] - E[\mathbf{P}_{\mathrm{pol}}^{(\mathrm{pol})}], \tag{12.31}$$

and thus the molecular property changes arising from forward and backward donations can be separately assigned. Note that in its Q-CHEM implementation, the evaluation of a VFB state always follows a polarization (standard SCF-MI)

calculation. Also, since the definition of VFB states is based on the generalized SCF-MI technique (Section 12.6.2), SCFMI_MODE = 1 is required.

VFB_CTA

Use the Variational Forward-Backward (VFB) approach to obtain "one-way" CT potential surfaces.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

FORWARD Allow 1→2 CT only (1 and 2 are two fragments).

BACKWARD Allow 2→1 CT only.

RECOMMENDATION:

None

Example 12.32 Geometry optimization on one-side CT surface (2 → 1) using the variational forward-backward (VFB) approach

```
$molecule
   0 1
   --
   0 1
   O  -1.551007  -0.114520   0.000000
   H  -1.934259   0.762503   0.000000
   H  -0.599677   0.040712   0.000000
   --
   0 1
   O   1.350625   0.111469   0.000000
   H   1.680398  -0.373741  -0.758561
   H   1.680398  -0.373741   0.758561
$end

$rem
   JOBTYPE           opt
   METHOD            wb97x-d
   BASIS             6-31g
   VFB_CTA           backward
   THRESH            14
   SCF_CONVERGENCE   9
   SCF_ALGORITHM     DIIS
   IDERIV            1
   SCFMI_MODE        1
   INTEGRAL_SYMMETRY FALSE
$end
```

In Q-CHEM v. 5.4 or later, analytical gradients for the polarized and two VFB "one-way" CT states with implicit solvent models PCM and SMD are supported so that one can perform part of the adiabatic ALMO-EDA steps (POL → CTf/CTb → Full) in solvation environments. To do this, one only needs to set the *$rem* variable SOLVENT_METHOD to PCM or SMD, which is similar to the usage of ALMO-EDA(solv) (see Section 12.6.6). The calculation of analytical forces on the frozen surface with implicit solvents is currently unavailable, and we look forward to enabling that in

future releases of Q-CHEM.

Example 12.33 Geometry optimization on the polarized surface with SMD solvent model

```
$molecule
   0 1
   --
   0 1
   H1
   O1 H1 0.95641
   H2 O1 0.96500  H1 104.77306
   --
   0 1
   O2 H2 dist     O1 171.85474 H1 180.000
   H3 O2 0.95822  H2 111.79807 O1 -58.587
   H4 O2 0.95822  H2 111.79807 O1 58.587

   dist = 2.0
$end

$rem
   JOBTYPE            OPT
   METHOD             wB97X-D
   BASIS              cc-pVDZ
   POL_GEOM           TRUE
   THRESH             14
   SCF_CONVERGENCE    9
   MEM_TOTAL          2000
   MEM_STATIC         500
   SCF_GUESS          FRAGMO
   IDERIV             1
   SCFMI_MODE         0
   SOLVENT_METHOD     SMD
   INTEGRAL_SYMMETRY FALSE
   POINT_GROUP_SYMMETRY FALSE
$end

$smx
   solvent water
$end
```

12.7.4 ALMO Force Decomposition Analysis

Building on the success of the adiabatic EDA (Section 12.7.3), a similar extension can be introduced which decomposes the forces at any geometry. Instead of following the forces to generate a geometry and evaluate the energy in that geometry as in the adiabatic EDA, the force decomposition analysis decomposes and analyzes the forces on their own at a given geometry. One important feature emerges from this approach which is the ability to decompose the frozen force, uncovering the frozen component relevant to a geometric change. This decomposition is not meaningful in the the adiabatic EDA since following purely attractive forces, for example, would collapse the molecules into each other.

Work by Aldossary *et al.* implements the analytical derivative of the classical electrostatics energy[3], allowing the frozen force to be broken down into classical electrostatics and van der Waals (Pauli and Dispersion) forces:

$$\Delta\mathbf{F}_{\text{frz}} = \Delta\mathbf{F}_{\text{cls-elec}} + \Delta\mathbf{F}_{\text{vdw}}. \tag{12.32}$$

Additionally, the frozen forces are now defined to subtract the geometric distortion forces (forces arising from geometric distortion energy). The frozen force term is now:

$$\Delta\mathbf{F}_{\text{frz}} = \mathbf{F}_{\text{frz}}[\mathbf{P}_{\text{frz}}] - \sum_{A} \mathbf{F}_{A}[\mathbf{P}_{A}], \tag{12.33}$$

where the second term on the right hand side the geometric distortion forces. Similar to the geometric distortion energy, the geometric distortion forces are subtracted from those of the isolated geometry which evaluate to zero for a geometric minimum. In other words, the geometetric distortion forces are nothing but the isolated fragments' forces in the complex geometry. The geometric distortion forces are printed separately for analysis.

By decomposing the intermolecular forces into geometric distortion, electrostatics, van der Waals, polarization, and charge transfer, we have now generated 5 times more forces than a regular force job. The $5(3N)$ forces can be difficult to analyze manually, and we use internal coordinates to make chemical sense of these forces. The Wilson's B matrix is generated using the new opt3 library and we apply the usual transformation. The user can then hone in to the bond, angle, or dihedral of interest to analyze the which force components are relevant to the stretch or compression. Despite the usefulness of the redundant internal coordinates, they are not unique, and the order of the atoms in the input file can affect the results.

FDA

Decompose intermolecular forces

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Does a regular force job.

TRUE Invokes the force decomposition analysis method

RECOMMENDATION:

Set it to TRUE with jobtype=force to decompose the force.

Example 12.34 Force Decomposition Analysis job for the $H_2O\cdots F^-$ job using the internal coordinates

```
$molecule
-1 1
--
0 1
O       0.0144306160     0.0000000000     0.0991268524
H      -0.0433242925     0.0000000000     1.2143531865
H       0.9756708294     0.0000000000    -0.0327279220
--
-1 1
F      -0.0237828221    -0.0000000000     2.5253092017
$end

$rem
JOBTYPE      FORCE
FDA          TRUE
GEN_SCFMAN   TRUE
METHOD       PBE
BASIS        def2-svp
MEM_TOTAL    2000
MEM_STATIC   500
SCF_GUESS    FRAGMO
BASIS_LIN_DEP_THRESH  6
SCFMI_MODE            0
SCF_PRINT_FRGM        TRUE
SCF_CONVERGENCE       10
THRESH                14
INTEGRAL_SYMMETRY     FALSE
POINT_GROUP_SYMMETRY  FALSE
$end
```

12.7.5 ALMO-EDA for Excited States

12.7.5.1 Theory

So far we have only covered EDA methods for intermolecular interactions between molecules in their ground states. Since electronic excited states are associated with less strongly bound electrons, modified electrostatic multipole moments (due to electron transition), and often larger polarizabilities, effects imposed by other molecules can be even larger as well as less chemically intuitive than those on ground states. Furthermore, there exist systems that are weakly bound in the ground state but much more strongly bound in the electronic excited state (*e.g.* He_2 versus He_2^*). Therefore, it is very desirable to develop an interpretation tool that can be utilized to study these important phenomena that are related to intermolecular interactions involving excited-state molecules.

Ge *et al.* recently extended the ALMO-EDA to treat *exciplexes* (where the excitation can be assigned to a single molecule within a complex)[22] and *excimers* (where multiple fragments contribute to the excitation)[20] computed at the CIS or TDDFT/TDA level of theory. Here we briefly overview the decomposition schemes. In the EDA for exciplexes, one first defines the interaction energy in the excited state ($\Delta E^*_{\mathrm{INT}}$) as

$$\Delta E^*_{\mathrm{INT}} = E^* - E^*_{\mathrm{frag}} \tag{12.34}$$

where $E^* = E + \omega$ is the energy of the excited supersystem, and E^*_{frag} can be expressed as the sum of ground-state fragment energies and the excitation energy of one of the fragments (without losing generality, this excited fragment is

denoted as fragment "1"):

$$E^*_{\text{frag}} = \sum_F E_F + \omega_1 \tag{12.35}$$

Therefore, we can rewrite the excited-state interaction as

$$\Delta E^*_{\text{INT}} = \Delta E_{\text{INT}} + \Delta\omega_{\text{INT}} \tag{12.36}$$

which contains contributions from the ground-state interaction energy ($\Delta E = E - \sum_F E_F$) and the excitation energy ($\Delta\omega_{\text{INT}} = \omega - \omega_1$). Then, as in the first-generation ALMO-EDA for ground states[53], the excited-state interaction energy can be separated into contributions from frozen interaction (FRZ), polarization (POL), and charge transfer (CT):

$$\Delta E^*_{\text{INT}} = \Delta E^*_{\text{FRZ}} + \Delta E^*_{\text{POL}} + \Delta E^*_{\text{CT}} \tag{12.37}$$

Each term on the RHS of Eq. (12.37) can be written in a similar form as Eq. (12.36):

$$\begin{aligned}
\Delta E^*_{\text{FRZ}} &= \Delta E_{\text{FRZ}} + \omega_{\text{FRZ}} - \omega_1 \\
&= \Delta E_{\text{FRZ}} + \Delta\omega_{\text{FRZ}} \\
\Delta E^*_{\text{POL}} &= \Delta E_{\text{POL}} + \omega_{\text{POL}} - \omega_{\text{FRZ}} \\
&= \Delta E_{\text{POL}} + \Delta\omega_{\text{POL}} \\
\Delta E^*_{\text{CT}} &= \Delta E_{\text{CT}} + \omega - \omega_{\text{POL}} \\
&= \Delta E_{\text{CT}} + \Delta\omega_{\text{CT}}
\end{aligned} \tag{12.38}$$

The terms ΔE_{FRZ}, ΔE_{POL}, and ΔE_{CT} can be obtained by performing a ground-state ALMO-EDA for the supersystem. To compute $\Delta\omega_{\text{FRZ}}$, $\Delta\omega_{\text{POL}}$, and $\Delta\omega_{\text{CT}}$, one needs to define ω_{FRZ} and ω_{POL}, *i.e.*, excitation energies associated with the frozen and polarized supersystem, respectively. The frozen intermediate state can be viewed as one excited fragment embedded in the environment formed by other ground-state fragments, whose effects on the excited fragment are only through the supersystem Fock matrix. The definition of the polarized intermediate state utilizes the ALMO-CIS model (see Section 12.16), where both MOs and excitation amplitudes are fragment-localized. We also note that the frozen contribution to the excited-state interaction energy, ΔE^*_{FRZ}, can be further partitioned into a classical electrostatics term (Coulomb interactions between isolated fragment charge distributions) and a non-electrostatic term (mostly Pauli repulsion if a non-dispersion-corrected model such as CIS is used):

$$\Delta E^*_{\text{FRZ}} = \Delta E^*_{\text{CLS-ELEC}} + \Delta E^*_{\text{NON-ELEC}} \tag{12.39}$$

Modifications are needed in order to extend this method to excimers, where different fragments are of degenerate or near-degenerate excited states. In such cases, we choose M reference fragment states as the initial basis. Denote the sth excited state on fragment I as the κth reference state ($\kappa = 1, 2, \ldots, M$). Similar to Eq. (12.35), we have

$$E^\kappa_{\text{frag}} = \sum_F E_F + \omega^s_I \tag{12.40}$$

The corresponding frozen excited-state wavefunction is then constructed by embedding this excited fragment into the environment formed by other fragments in their ground states:

$$|\Phi^\kappa_{\text{FRZ}}\rangle = |\Psi_1\Psi_2\cdots\Psi^s_I\cdots\Psi_N\rangle \tag{12.41}$$

and the excited-state frozen interaction energy

$$\Delta E^\kappa_{\text{FRZ}} = \Delta E_{\text{FRZ}} + \Delta\omega^\kappa_{\text{FRZ}} = \Delta E_{\text{FRZ}} + (\omega^\kappa_{\text{FRZ}} - \omega^s_I) \tag{12.42}$$

With M degenerate or near-degenerate frozen excited states, a new intermediate state is then introduced to capture the pure excitonic-splitting (EXSP) effect in the formation of excimers, which can be expressed as a linear combination of the frozen states:

$$|\Phi^\kappa_{\text{EXSP}}\rangle = \sum_{\kappa'}^{M} c^{\kappa\kappa'} |\Psi_1\Psi_2\cdots\Psi^s_I\cdots\Psi_N\rangle \tag{12.43}$$

The associated excitation energy $\omega^{\kappa}_{\mathrm{EXSP}}$ and the corresponding linear combination coefficients can be obtained by solving a secular equation in the basis of frozen states. As excitonic splitting is purely an excited-state phenomenon, we have

$$\Delta E^{\kappa}_{\mathrm{EXSP}} = \Delta\omega^{\kappa}_{\mathrm{EXSP}} = \omega^{\kappa}_{\mathrm{EXSP}} - \omega^{\kappa}_{\mathrm{FRZ}} \tag{12.44}$$

Subsequently, polarization and charge transfer are handled in a similar way as in the excimer case:

$$\begin{aligned} \Delta E^{\kappa}_{\mathrm{POL}} &= \Delta E_{\mathrm{POL}} + \Delta\omega^{\kappa}_{\mathrm{POL}} = \Delta E_{\mathrm{POL}} + (\omega^{\kappa}_{\mathrm{POL}} - \omega^{\kappa}_{\mathrm{EXSP}}) \\ \Delta E^{\kappa}_{\mathrm{CT}} &= \Delta E_{\mathrm{CT}} + \Delta\omega^{\kappa}_{\mathrm{CT}} = \Delta E_{\mathrm{CT}} + (\omega^{\kappa} - \omega^{\kappa}_{\mathrm{POL}}) \end{aligned} \tag{12.45}$$

One more complication compared to the EDA scheme for exciplexes is that since multiple (M) states are considered, extra caution needs be paid to the state-ordering at different stages (EXSP, POL and CT). In order to locate the states of interest (which can be most unambiguously identified at the EXSP stage) correctly during the entire EDA procedure, a state-tracking algorithm based on a maximum-overlap criterion is employed. The reader is referred to Ref. 20 for more details.

12.7.5.2 Job Control

The ALMO-EDA for intermolecular interactions involving excited-state molecules implemented in Q-CHEM 5.1 supports CIS and TDDFT within the Tamm-Dancoff approximation (TDA) for closed-shell systems, *i.e.*, excited states calculated by TDDFT and unrestricted systems are currently *not* supported. The EDA procedure is triggered by setting EX_EDA = TRUE. The code first performs a customized ground-state calculation (using AO-based ALMOs) through the "EDA2" driver. During the isolated fragment calculations in this EDA, the fragment excited states are also computed after its ground-state SCF is converged, which is controlled by a new input section *$frgm_cis_n_roots*. The format of this input section is as follows:

```
$frgm_cis_n_roots
frgm_idx1 nstates_to_calc nstates_as_exciton_basis
frgm_idx2 nstates_to_calc nstates_as_exciton_basis
.   .   .
$end
```

Here "nstates_to_calc" specifies the number of states to calculate for each fragment (the value of CIS_N_ROOTS for each fragment calculation), and "nstates_as_exciton_basis" refers to the number of calculated fragment states that are used to construct the EXSP state (whose sum gives M in Eq. (12.43)). When the supersystem is considered as an exciplex where the excitation is assigned to a specific fragment, only one row is needed in this section, and there is no need to specify the number of states used as the basis for the EXSP state.

The other relevant rem variables includes CIS_N_ROOTS, which specifies the number of roots to calculate in the ALMO-CIS/TDA and full CIS/TDA calculations, and EIGSLV_METH (see Section 12.16) that is set to 1 (using the Davidson iterative solver) by default. Note that the number of states that the EDA is concerned with is controlled by the number of isolated fragment states (the exciplex case) or the total number of states that are excitonically coupled (the excimer case). In the latter case, CIS_N_ROOTS is usually set to a value that is larger than M to ensure that all states of interest are captured in the ALMO-CIS/TDA and full CIS/TDA calculations, as changes in state-ordering might occur.

EX_EDA

Perform an ALMO-EDA calculation with one or more fragments excited.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform EDA with excited-state molecule(s) taken into account.

FALSE

RECOMMENDATION:

None

Example 12.35 EDA for the lowest two excited states of the formamide-water complex at the CIS/6-31+G(d) level of theory. Both excited states are assigned to the formamide molecule and the system is regarded as an exciplex.

```
$molecule
0 1
--
0 1
C          1.1508059365    0.2982718924    0.0240277739
O          0.3545181649    1.2334803420   -0.0015882208
N          0.8104369587   -1.0072797234    0.0043506838
H          2.2327270535    0.4686363261    0.0666232655
H         -0.1675092286   -1.2596328526   -0.0352400180
H          1.5210524537   -1.7122494331    0.0139809901
--
0 1
O         -1.9693273428   -0.2999882700   -0.2293071572
H         -1.3827632725    0.4697313642   -0.1375254289
H         -2.7470364523   -0.0962178118    0.2907490329
$end

$rem
   JOBTYPE           eda
   METHOD            hf
   BASIS             6-31+g(d)
   EX_EDA            true
   SCF_CONVERGENCE   8
   CIS_N_ROOTS       2
   CIS_TRIPLETS      false
   THRESH            12
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

$frgm_cis_n_roots
1  2
$end
```

Example 12.36 EDA for the lowest two states (1s->2s) of the He_2^* excimer computed at the CIS/6-311(2+,2+)G (customized) level of theory. For each He, eight excited states are calculated and only the lowest one is used to construct the EXSP state, giving rise to two supersystem states.

```
$molecule
0 1
--
0 1
He    0.0   0.0  0.0
--
0 1
He    3.0   0.0  0.0
$end

$rem
   JOBTYPE         eda
   EX_EDA          true
   METHOD          hf
   BASIS           gen   !6-311(2+,2+)G
   CIS_N_ROOTS     8
   CIS_TRIPLETS    false
   THRESH          12
   EIGSLV_METH     0   !direct
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

$frgm_cis_n_roots
1  8  1
2  8  1
$end

$basis
He   0
S    3    1.000000
9.81243000E+01    2.87452000E-02
1.47689000E+01    2.08061000E-01
3.31883000E+00    8.37635000E-01
S    1    1.000000
8.74047000E-01    1.00000000E+00
S    1    1.000000
2.44564000E-01    1.00000000E+00
SP   1    1.000000
4.80000000E-02    1.00000000E+00   1.00000000E+00
SP   1    1.000000
1.44578313E-02    1.00000000E+00   1.00000000E+00
****
$end
```

12.8 The Explicit Polarization (XPol) Method

12.8.1 Theory

XPol is an approximate, fragment-based molecular orbital method that was developed as a "next-generation" force field.[19,117–119] The basic idea of the method is to treat a molecular liquid, solid, or cluster as a collection of fragments,

where each fragment is a molecule. Intra-molecular interactions are treated with a self-consistent field method (Hartree-Fock or DFT), but each fragment is embedded in a field of point charges that represent electrostatic interactions with the other fragments. These charges are updated self-consistently by collapsing each fragment's electron density onto a set of atom-centered point charges, using charge analysis procedures (Mulliken, Löwdin, or ChElPG, for example; see Section 10.2.2). This approach incorporates many-body polarization, at a cost that scales linearly with the number of fragments, but neglects the anti-symmetry requirement of the total electronic wave function. As a result, intermolecular exchange-repulsion is neglected, as is dispersion since the latter is an electron correlation effect. As such, the XPol treatment of polarization must be augmented with empirical, Lennard–Jones-type intermolecular potentials in order to obtain meaningful optimized geometries, vibrational frequencies or dynamics.

The XPol method is based upon an *ansatz* in which the super-system wave function is written as a direct product of fragment wave functions,

$$|\Psi\rangle = \prod_{A}^{N_{\text{frag}}} |\Psi_A\rangle, \tag{12.46}$$

where N_{frag} is the number of fragments. We assume here that the fragments are molecules and that covalent bonds remain intact. The fragment wave functions are anti-symmetric with respect to exchange of electrons within a fragment, but not to exchange between fragments. For closed-shell fragments described by Hartree-Fock theory, the XPol total energy is[45,118]

$$E_{\text{XPol}} = \sum_{A} \left[2 \sum_{a} \mathbf{c}_a^{\dagger} \left(\mathbf{h}^A + \mathbf{J}^A - \tfrac{1}{2}\mathbf{K}^A \right) \mathbf{c}_a + E_{\text{nuc}}^A \right] + E_{\text{embed}}. \tag{12.47}$$

The term in square brackets is the ordinary Hartree-Fock energy expression for fragment A. Thus, $\mathbf{c}_a$ is a vector of occupied MO expansion coefficients (in the AO basis) for the occupied MO $a \in A$; $\mathbf{h}^A$ consists of the one-electron integrals; and $\mathbf{J}^A$ and $\mathbf{K}^A$ are the Coulomb and exchange matrices, respectively, constructed from the density matrix for fragment A. The additional terms in Eq. (12.47),

$$E_{\text{embed}} = \tfrac{1}{2} \sum_{A} \sum_{B \neq A} \sum_{J \in B} \left(-2 \sum_{a} \mathbf{c}_a^{\dagger} \mathbf{I}_J \mathbf{c}_a + \sum_{I \in A} L_{IJ} \right) q_J, \tag{12.48}$$

arise from the electrostatic embedding. The matrix $\mathbf{I}_J$ is defined by its AO matrix elements,

$$(\mathbf{I}_J)_{\mu\nu} = \left\langle \mu \left| \frac{1}{|\vec{r} - \vec{R}_J|} \right| \nu \right\rangle, \tag{12.49}$$

and L_{IJ} is given by

$$L_{IJ} = \frac{Z_I}{|\vec{R}_I - \vec{R}_J|}. \tag{12.50}$$

According to Eqs. (12.47) and (12.48), each fragment is embedded in the electrostatic potential arising from a set of point charges, $\{q_J\}$, on all of the other fragments; the factor of $1/2$ in Eq. (12.48) avoids double-counting. Exchange interactions between fragments are ignored, and the electrostatic interactions between fragments are approximated by interactions between the charge density of one fragment and point charges on the other fragments.

Crucially, the vectors $\mathbf{c}_a$ are constructed within the ALMO *ansatz*,[52] so that MOs for each fragment are represented in terms of only those AOs that are centered on atoms in the same fragment. This choice affords a method whose cost grows linearly with respect to N_{frag}, and where basis set superposition error is excluded by construction. In compact basis sets, the ALMO *ansatz* excludes inter- fragment charge transfer as well.

The original XPol method of Xie *et al.*[117–119] uses Mulliken charges for the embedding charges q_J in Eq. (12.48), though other charge schemes could be envisaged. In non-minimal basis sets, the use of Mulliken charges is beset by severe convergence problems,[45] and Q-CHEM's implementation of XPol offers the alternative of using either Löwdin charges, Charge Model 5 (CM5) charges,[85] or ChElPG charges,[10] the latter being derived from the electrostatic potential as discussed in Section 10.2.2. The ChElPG charges are found to be stable and robust, albeit with a somewhat

larger computational cost as compared to Mulliken or Löwdin charges.[31,45] An algorithm to compute ChElPG charges using atom-centered Lebedev grids rather than traditional Cartesian grids is available (see Section 10.2.2),[36] which uses far fewer grid points and thus can significantly improve the performance for the XPol/ChElPG method, where these charges must be iteratively updated. A cost-effective and slightly more accurate alternative to the ChElPG charges are the CM5 charges.[75] The CM5 charge derivatives are significantly cheaper to compute than those for ChElPG, and because XPol must iteratively update the charges the CM5 charges are considerably less expensive.

Researchers who use Q-CHEM's XPol code are asked to cite Refs. 31,45.

12.8.2 Supplementing XPol with Empirical Potentials

In order to obtain physical results, one must either supplement the XPol energy expression with either empirical intermolecular potentials or else with an *ab initio* treatment of intermolecular interactions. The latter approach is described in Section 12.10. Here, we describe how to add Lennard-Jones or Buckingham potentials to the XPol energy, using the *$xpol_mm* and *$xpol_params* sections described below.

The Lennard-Jones potential is

$$V_{\mathrm{LJ}}(R_{ij}) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{R_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{R_{ij}}\right)^{6}\right], \tag{12.51}$$

where R_{ij} represents the distance between atoms i and j. This potential is characterized by two parameters, a well depth ϵ_{ij} and a length scale σ_{ij}. Although quite common, the R^{-12} repulsion is unrealistically steep. The Buckingham potential replaces this with an exponential function,

$$V_{\mathrm{Buck}}(R_{ij}) = \epsilon_{ij}\left[Ae^{-B\frac{R_{ij}}{\sigma_{ij}}} - C\left(\frac{\sigma_{ij}}{R_{ij}}\right)^{6}\right], \tag{12.52}$$

Here, A, B, and C are additional (dimensionless) constants, independent of atom type. In both Eq. (12.51) and Eq. (12.52), the parameters ϵ_{ij} and σ_{ij} are determined using the geometric mean of atomic well-depth and length-scale parameters. For example,

$$\sigma_{ij} = \sqrt{\sigma_i \sigma_j}\,. \tag{12.53}$$

The atomic parameters σ_i and ϵ_i must be specified using a *$xpol_mm* section in the Q-CHEM input file. The format is a molecular mechanics-like specification of atom types and connectivities. All atoms specified in the *$molecule* section must also be specified in the *$xpol_mm* section. Each line must contain an atom number, atomic symbol, Cartesian coordinates, integer atom type, and any connectivity data. The *$xpol_params* section specifies, for each atom type, a value for ϵ in kcal/mol and a value for σ in Ångstroms. A Lennard-Jones potential is used by default; if a Buckingham potential is desired, then the first line of the *$xpol_params* section should contain the string BUCKINGHAM followed by values for the A, B, and C parameters.

12.8.3 Job Control Variables for XPol

XPol calculations are enabled by setting the *$rem* variable XPOL to TRUE. The XPol method can be used in combination with Hartree-Fock theory and with most density functionals, a notable exception being that XPol is not yet implemented for meta-GGA functionals (Section 5.3). Combination of XPol with solvation models (Section 11.2) or external charges (*$external_charges*) is also not supported. Analytic gradients are available when Mulliken, Löwdin, or CM5 embedding charges are used, but not yet available for ChElPG embedding charges.

XPOL

Perform a self-consistent XPol calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform an XPol calculation.

FALSE Do not perform an XPol calculation.

RECOMMENDATION:

NONE

Other XPol options are specified via keywords contained in the *$xpol* section. These keywords are given below.

embed

Specifies which type of electrostatic embedding will be used.

INPUT SECTION: *$xpol*

TYPE:

STRING

DEFAULT:

Charges

OPTIONS:

None No embedding charges.

Charges Atomic point charges (standard XPol method).

Density Fragment densities (as in the FMO method; see Sec. 12.13)

RECOMMENDATION:

The standard XPol method uses atomic point charges.

charges

Specifies which type of atomic point charges to use.

INPUT SECTION: *$xpol*

TYPE:

STRING

DEFAULT:

CM5

OPTIONS:

Mulliken Mulliken charges

Lowdin Löwdin charges

CHELPG ChElPG charges

CM5 CM5 charges

RECOMMENDATION:

Problems with Mulliken charges in extended basis sets can lead to XPol convergence failure. Löwdin charges tend to be somewhat more stable, while ChElPG charges are quite robust and provide an accurate electrostatic embedding. However, ChElPG charges are more expensive to compute and they perform slightly worse than CM5 charges in systems with charged monomers. The CM5 charges are a cost-effective and slightly more accurate choice. For XSAPT calculations (Section 12.10), CM5 charges are recommended. However, CM5 charges are not yet available for AIFDEM jobs (Section 12.14) and in that case ChElPG charges are recommended.

print

Specifies the level of output for the XPol code.

INPUT SECTION: *$xpol*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Desired print level

RECOMMENDATION:

Higher values print additional information

Especially in the context of SAPT(KS) calculations (see Section 12.9) and XSAPT(KS) calculations (Section 12.10), in which a Kohn-Sham description of the monomers is combined with symmetry-adapted perturbation theory (SAPT), it is essentially that the Kohn-Sham density functional exhibit correct asymptotic behavior.[61,62] Most standard density functionals do *not* satisfy this criterion, but it can be imposed by using a range-separated hybrid functional in which the range-separation parameter ω is tuned so as to satisfy the ionization potential theorem of DFT, namely, $\mathrm{IE}(\omega) = -\epsilon_{\mathrm{HOMO}}(\omega)$. This condition should be enforced separately on each monomer within an XPol calculation, which requires that a different value of ω be used for each monomer. This functionality is requested by setting the **DFT-LRC** option in the *$xpol* section. (Note that no value needs to be set with this keyword; if it is present in the *$xpol* section then this option is enabled.)

DFT-LRC

Specifies whether monomer-specific range-separated hybrid functionals are to be used

INPUT SECTION: *$xpol*

TYPE:

None

DEFAULT:

Not specified

OPTIONS:

If the keyword is present, this option is enabled.

RECOMMENDATION:

Placing this keyword into the *$xpol* section indicates that monomer-specific values of ω (the range-separation parameter) are to be used, which then requires a *$lrc_omega* input section.

If **DFT-LRC** is specified, then a *$lrc_omega* input section is also required. This input section simply consists of the values $\omega_1, \omega_2, \ldots$ for each monomer, listed one per line in the order that the monomers appear in the *$molecule* section. These values have the same units as the *$rem* variable OMEGA that is used in range-separated hybrid functional calculations, namely, $\omega =$ OMEGA/1000 in atomic units. See Section 12.10.3 for an example of how the **DFT-LRC** option and the *$lrc_omega* input section are used in the context of the XSAPT(KS) method.

12.8.4 Examples

XPol on its own is not a useful method because it neglects all intermolecular interactions except for polarization, so the two examples below demonstrate the use of XPol in conjunction with a Lennard-Jones and a Buckingham potential,

respectively.

Example 12.37 An XPol single point calculation on the water dimer using a Lennard-Jones potential. CM5 atomic embedding charges (which is the default) are used.

```
$molecule
0 1
-- water 1
0 1
O          -1.364553     .041159     .045709
H          -1.822645     .429753    -.713256
H          -1.841519    -.786474     .202107
-- water 2
0 1
O           1.540999     .024567     .107209
H            .566343     .040845     .096235
H           1.761811    -.542709    -.641786
$end

$rem
   METHOD           HF
   BASIS            3-21G
   XPOL             TRUE
$end

$xpol_mm
   1 O          -1.364553     .041159     .045709 1 2 3
   2 H          -1.822645     .429753    -.713256 2 1
   3 H          -1.841519    -.786474     .202107 2 1
   4 O           1.540999     .024567     .107209 1 5 6
   5 H            .566343     .040845     .096235 2 4
   6 H           1.761811    -.542709    -.641786 2 4
$end

$xpol_params
   1            0.16     3.16
   2            0.00     0.00
$end
```

Example 12.38 An XPol single point calculation on the water dimer using a Buckingham potential.

```
$molecule
0 1
-- water 1
0 1
O          -1.364553     .041159     .045709
H          -1.822645     .429753    -.713256
H          -1.841519    -.786474     .202107
-- water 2
0 1
O           1.540999     .024567     .107209
H            .566343     .040845     .096235
H           1.761811    -.542709    -.641786
$end

$rem
   METHOD           HF
   BASIS            3-21G
   XPOL             TRUE
$end

$xpol_mm
   1 O          -1.364553     .041159     .045709 1 2 3
   2 H          -1.822645     .429753    -.713256 2 1
   3 H          -1.841519    -.786474     .202107 2 1
   4 O           1.540999     .024567     .107209 1 5 6
   5 H            .566343     .040845     .096235 2 4
   6 H           1.761811    -.542709    -.641786 2 4
$end

$xpol_params
   BUCKINGHAM   500000.0 12.5 2.25
   1            0.16     3.16
   2            0.00     0.00
$end
```

12.9 Symmetry-Adapted Perturbation Theory (SAPT)

12.9.1 Theory

Symmetry-adapted perturbation theory (SAPT) is a theory of intermolecular interactions. When computing intermolecular interaction energies one typically computes the energy of two molecules infinitely separated and in contact, then computes the interaction energy by subtraction. SAPT, in contrast, is a perturbative expression for the interaction energy itself. The various terms in the perturbation series are physically meaningful, and this decomposition of the interaction energy can aid in the interpretation of the results. A brief overview of the theory is given below; for additional technical details, the reader is referred to Jeziorski *et al.*.[47,48] Additional context can be found in a pair of more recent review articles.[35,109]

In SAPT, the Hamiltonian for the $\mathrm{A}\cdots\mathrm{B}$ dimer is written as

$$\hat{H} = \hat{F}_A + \hat{F}_B + \xi\hat{W}_A + \eta\hat{W}_B + \zeta\hat{V} \;, \tag{12.54}$$

where $\hat{W}_A$ and $\hat{W}_B$ are Møller-Plesset fluctuation operators for fragments A and B, representing intramolecular electron correlation whereas $\hat{V}$ consists of the intermolecular Coulomb operators. This part of the perturbation is conve-

niently expressed as

$$\hat{V} = \sum_{i \in A} \sum_{j \in B} \hat{v}(ij) \tag{12.55}$$

with

$$\hat{v}(ij) = \frac{1}{|\vec{r}_i - \vec{r}_j|} + \frac{\hat{v}_A(j)}{N_A} + \frac{\hat{v}_B(i)}{N_B} + \frac{V_0}{N_A N_B} \; . \tag{12.56}$$

The quantity V_0 is the nuclear interaction energy between the two fragments and

$$\hat{v}_A(j) = -\sum_{I \in A} \frac{Z_I}{|\vec{r}_j - \vec{R}_I|} \tag{12.57}$$

describes the interaction of electron $j \in B$ with nucleus $I \in A$.

Starting from a zeroth-order Hamiltonian $\hat{H}_0 = \hat{F}_A + \hat{F}_B$ and zeroth-order wave functions that are direct products of monomer wave functions, $|\Psi_0\rangle = |\Psi_A\rangle|\Psi_B\rangle$, the SAPT approach is based on a symmetrized Rayleigh-Schrödinger perturbation expansion[47,48] with respect to the perturbation parameters ξ, η, and ζ in Eq. (12.54). The resulting interaction energy can be expressed as[47,48]

$$E_{\text{int}} = \sum_{i=1}^{\infty} \sum_{j=0}^{\infty} \left(E_{\text{pol}}^{(ij)} + E_{\text{exch}}^{(ij)} \right) \; . \tag{12.58}$$

Because it makes no sense to treat $\hat{W}_A$ and $\hat{W}_B$ at different orders of perturbation theory, there are only two indices in this expansion: j for the monomer fluctuations potentials and i for the intermolecular perturbation. The terms $E_{\text{pol}}^{(ij)}$ are known collectively as the *polarization expansion*, and these are precisely the same terms that would appear in ordinary Rayleigh-Schrödinger perturbation theory, which is valid when the monomers are well-separated. The polarization expansion contains electrostatic, induction and dispersion interactions, but in the *symmetrized* Rayleigh-Schrödinger expansion, each term $E_{\text{pol}}^{(ij)}$ has a corresponding exchange term, $E_{\text{exch}}^{(ij)}$, that arises from an anti-symmetrizer $\hat{\mathcal{A}}_{AB}$ that is introduced in order to project away the Pauli-forbidden components of the interaction energy that would otherwise appear.[48]

The version of SAPT that is implemented in Q-CHEM assumes that $\xi = \eta = 0$, an approach that is usually called SAPT0.[35] Within the SAPT0 formalism, the interaction energy is formally expressed by the following symmetrized Rayleigh-Schrödinger expansion:[47,48]

$$E_{\text{int}}(\zeta) = \frac{\langle \Psi_0 | \zeta \hat{V} \hat{\mathcal{A}}_{AB} | \Psi(\zeta) \rangle}{\langle \Psi_0 | \hat{\mathcal{A}}_{AB} | \Psi(\zeta) \rangle} \; , \tag{12.59}$$

The antisymmetrizer $\hat{\mathcal{A}}_{AB}$ in this expression can be written as

$$\hat{\mathcal{A}}_{AB} = \frac{N_A! N_B!}{(N_A + N_B)!} \, \hat{\mathcal{A}}_A \hat{\mathcal{A}}_B \left(\hat{1} + \hat{\mathcal{P}}_{AB} + \hat{\mathcal{P}}' \right) \; , \tag{12.60}$$

where $\hat{\mathcal{A}}_A$ and $\hat{\mathcal{A}}_B$ are antisymmetrizers for the two monomers and $\hat{\mathcal{P}}_{AB}$ is a sum of all one-electron exchange operators between the two monomers. The operator $\hat{\mathcal{P}}'$ in Eq. (12.60) denotes all of the three-electron and higher-order exchanges. This operator is neglected in what is known as the "single-exchange" approximation,[47,48] which is expected to be quite accurate at typical van der Waals and larger intermolecular separations, but sometimes breaks down at smaller intermolecular separations.[58]

Only terms up to $\zeta = 2$ in Eq. (12.59)—that is, second order in the intermolecular interaction—have been implemented in Q-CHEM. It is common to relabel these low-order terms in the following way [*cf.* Eq. (12.58)]:

$$E_{\text{int}}^{\text{SAPT0}} = E_{\text{elst}}^{(1)} + E_{\text{exch}}^{(1)} + E_{\text{pol}}^{(2)} + E_{\text{disp}}^{(2)} \; . \tag{12.61}$$

The electrostatic part of the first-order energy correction is denoted $E_{\text{elst}}^{(1)}$ and represents the Coulomb interaction between the two monomer electron densities.[48] The quantity $E_{\text{exch}}^{(1)}$ is the corresponding first-order (*i.e.*, Hartree-Fock)

exchange correction. Explicit formulas for these corrections can be found in Ref. 47. The second-order term from the polarization expansion, denoted $E_{\rm pol}^{(2)}$ in Eq. (12.61), consists of a dispersion contribution (which arises for the first time at second order) as well as a second-order correction for induction. The latter can be written

$$E_{\rm ind}^{(2)} = E_{\rm ind}^{(2)}(A \leftarrow B) + E_{\rm ind}^{(2)}(B \leftarrow A) \ , \tag{12.62}$$

where the notation $A \leftarrow B$, for example, indicates that the frozen charge density of B polarizes the density of A. In detail,

$$E_{\rm ind}^{(2)}(A \leftarrow B) = 2\sum_{ar} t_{ar}(w_B)_{ra} \tag{12.63}$$

where

$$(w_B)_{ar} = (\hat{v}_B)_{ar} + \sum_b (ar|bb) \tag{12.64}$$

and $t_{ar} = (w_B)_{ar}/(\epsilon_a - \epsilon_r)$. The second term in Eq. (12.62), in which A polarizes B, is obtained by interchanging labels.[45] Finally, the second-order dispersion correction has a form reminiscent of the MP2 correlation energy:

$$E_{\rm disp}^{(2)} = 4\sum_{abrs} \frac{(ar|bs)(ra|sb)}{\epsilon_a + \epsilon_b - \epsilon_r - \epsilon_s} \ . \tag{12.65}$$

The induction and dispersion corrections both have accompanying exchange corrections (exchange-induction and exchange-dispersion).[47,48]

The similarity between Eq. (12.65) and the MP2 correlation energy means that SAPT jobs, like MP2 calculations, can be greatly accelerated using resolution-of-identity (RI) techniques, and an RI version of SAPT is available in Q-CHEM. To use it, one must specify an auxiliary basis set. The same ones used for RI-MP2 work equally well for RI-SAPT, but one should always select the auxiliary basis set that is tailored for use with the primary basis of interest, as in the RI-MP2 examples in Section 6.6.2.

It is common to replace $E_{\rm ind}^{(2)}$ and $E_{\text{exch-ind}}^{(2)}$ in Eq. (12.61) with their "response" ("resp") analogues, which are the infinite-order correction for polarization arising from a frozen partner density.[47,48] Operationally, this substitution involves replacing the second-order induction amplitudes, t_{ar} in Eq. (12.63), with amplitudes obtained from solution of the coupled-perturbed Hartree-Fock equations.[96] (The perturbation is simply the electrostatic potential of the other monomer.) In addition, it is common to correct the SAPT0 binding energy for higher-order polarization effects by adding a correction term of the form[35,48]

$$\delta E_{\rm int}^{\rm HF} = E_{\rm int}^{\rm HF} - \left(E_{\rm elst}^{(1)} + E_{\rm exch}^{(1)} + E_{\rm ind,resp}^{(2)} + E_{\text{exch-ind,resp}}^{(2)}\right) \tag{12.66}$$

to the interaction energy. Here, $E_{\rm int}^{\rm HF}$ is the counterpoise-corrected Hartree-Fock binding energy for A$\cdots$B. Both the response corrections and the $\delta E_{\rm int}^{\rm HF}$ correction are optionally available in Q-CHEM's implementation of SAPT.

It is tempting to replace Hartree-Fock MOs and eigenvalues in the SAPT0 formulas with their Kohn-Sham counterparts, as a low-cost means of introducing monomer electron correlation. The resulting procedure is known as SAPT(KS),[115] and does offer an improvement on SAPT0 for some strongly hydrogen-bonded systems.[31] Unfortunately, SAPT(KS) results are generally in poor agreement with benchmark dispersion energies,[31] owing to incorrect asymptotic behavior of approximate exchange-correlation potentials.[88] The dispersion energies can be greatly improved through the use of long-range corrected (LRC) functionals in which the range-separation parameter, ω, is "tuned" so as to satisfy the condition $\epsilon_{\rm HOMO} = -$IE, where $\epsilon_{\rm HOMO}$ is the HOMO energy and "IE" represents the ionization energy.[61] Monomer-specific values of ω, tuned using the individual monomer IEs, substantially improve SAPT(KS) dispersion energies, though the results are still not of benchmark quality.[61] Other components of the interaction energy, however, can be described quite accurately SAPT(KS) in conjunction with a tuned version of LRC-ωPBE.[61] Use of monomer-specific ω values is controlled by the variable **DFT-LRC** in the *$xpol* section, as described in Section 12.8.3, with monomer-specific values of ω that must be specified in a *$lrc_omega* input section (with appropriate keywords for LRC-DFT in the *$rem* section). As an example, some values of ω for various monomers obtained using the IE-tuning condition and

S22 monomers		S66 monomers		ions	
Monomer	ω / a_0^{-1}	Monomer	ω / a_0^{-1}	Monomer	ω / a_0^{-1}
adenine	0.271	$MeNH_2$	0.397	F^-	0.480
2-aminopyridine	0.293	MeOH	0.438	Cl^-	0.372
benzene	0.280	$AcNH_2$	0.453	SO_4^{2-}	0.344
ethyne	0.397	AcOH	0.381	Li^+	2.006
ethene	0.359	cyclopentane	0.420	Na^+	1.049
methane	0.454	neopentane	0.287	K^+	0.755
formamide	0.460	pentane	0.365		
formic acid	0.412	peptide	0.341		
water	0.502	pyridine	0.316		
HCN	0.452				
indole	0.267				
ammonia	0.440				
phenol	0.292				
pyrazine	0.367				
2-pyridoxine	0.294				
thymine	0.284				
uracil	0.295				

Table 12.2: Tuned values of the range separation parameter (ω) for monomers in the S22[49] and S66[114] data sets, along with a few ions.

the LRC-ωPBE functional are listed in Table 12.2. Clearly there is a non-trivial variation amongst the optimally-tuned values of ω.

Finally, some discussion of basis sets is warranted. Typically, SAPT calculations are performed in the so-called dimer-centered basis set (DCBS),[116] which means that the combined $A+B$ basis set is used to calculate the zeroth-order wave functions for both A and B. This leads to the unusual situation that there are more MOs than basis functions: one set of occupied and virtual MOs for each monomer, both expanded in the same (dimer) AO basis. As an alternative to the DCBS, one might calculate $|\Psi_A\rangle$ using only A's basis functions (similarly for B), in which case the SAPT calculation is said to employ the monomer-centered basis set (MCBS).[116] However, MCBS results are generally of poorer quality. As an efficient alternative to the DCBS, Jacobson and Herbert[45] introduced a projected ("proj") or "pseudocanonicalized" basis set, borrowing an idea from dual-basis MP2 calculations.[107] In this approach, the SCF iterations are performed in the MCBS but then Fock matrices for fragments A and B are constructed in the dimer ($A+B$) basis set and then pseudocanonicalized, meaning that the occupied-occupied and virtual-virtual blocks of these matrices are diagonalized. This procedure does not mix occupied and virtual orbitals, and thus leaves the fragment densities and zeroth-order fragment energies unchanged. However, it does provide a larger set of virtual orbitals that extend over the partner fragment. This larger virtual space is then used to evaluate the perturbative corrections. All three of these basis options (MCBS, DCBS, and projected basis) are available in Q-CHEM.

12.9.2 Job Control for SAPT Calculations

Q-CHEM's implementation of SAPT0 was designed from the start as a correction for XPol calculations, affording the "XSAPT" method that is described in Section 12.10. As such, even a traditional SAPT0 calculation is requested by setting JOBTYPE = XSAPT. However, whereas XSAPT calculations are based on XPol wave functions for the monomers, which can capture many-body polarization effects in systems composed of more than two monomers (see

Section 12.10), traditional SAPT calculations are based instead on gas-phase monomer wave functions. This can be realized by turning off the XPol charge embedding, *i.e.*, by setting **embed = none** in the *$xpol* section that was introduced in Section 12.8.

Energy components are printed separately at the end of a SAPT job. If EXCHANGE = HF, then an XSAPT calculation with XPol embedding turned off corresponds to a SAPT0 calculation. Alternatively, if a density functional level of theory is requested in the *$rem* section, then JOBTYPE = XSAPT will perform a SAPT(KS) calculation, *i.e.*, one that is based on a Kohn-Sham description of the monomers.

Note: (1) Meta-GGAs are not yet available for SAPT(KS) calculations when the projected (pseudocanonicalized) basis set is used. SAPT(KS) calculations can be performed with meta-GGAs using the monomer or dimer basis sets.
(2) Both closed- and open-shell (unrestricted) SAPT(KS) calculations are available.
(3) Frozen orbitals are *not* available for use with SAPT(KS).
(4) SAPT with PCM boundary conditions is available.[26]

The remaining job control options for SAPT calculations are specified using various keywords contained in a *$sapt* input section, as described below. Researchers who use Q-CHEM's SAPT code are asked to cite Refs. 33,45.

Algorithm

Specifies which SAPT algorithm will be used

INPUT SECTION: *$sapt*

TYPE:

STRING

DEFAULT:

MO

OPTIONS:

MO	Traditional molecular orbital-based algorithm
RI-MO	MO-based algorithm with resolution-of-identity approximation
AO	Atomic orbital-based algorithm.

RECOMMENDATION:

The standard MO-based algorithm corresponds to an MP2-like implementation of Eq. (12.65), where the RI-MO algorithm corresponds to an RIMP2-like implementation. The RI implementation is generally much faster and introduces negligible errors (as compared to the standard implementation), provided that the auxiliary basis set is matched to the primary basis set. (The former must be specified using (AUX_BASIS in the *$rem* section.) The AO-based algorithm does not implement Eq. (12.65) and is intended only for use with XSAPT(KS)+*ai*D calculations; see Section 12.10.3.

Exchange

Specifies how the first-order exchange interaction will be evaluated.

INPUT SECTION: *$sapt*

TYPE:

STRING

DEFAULT:

S_Squared

OPTIONS:

S_Squared	Use the single-exchange ("S^2") approximation.
S_Inverse	Compute the exact first order exchange.

RECOMMENDATION:

The single-exchange approximation is expected to be adequate except possibly at very short intermolecular distances, and is somewhat faster to compute.

Basis

Controls which basis is used to evaluate the SAPT corrections

INPUT SECTION: *$sapt*

TYPE:

STRING

DEFAULT:

monomer

OPTIONS:

monomer	Use the monomer-centered basis set (MCBS).
dimer	Use the dimer-centered basis set (DCBS).
projected	Use the projected (pseudocanonicalized) basis set.

RECOMMENDATION:

The DCBS (in which the monomer wave functions are iterated to convergence using the dimer basis set) is the preferred choice in traditional SAPT, although it is more costly than the MCBS (which uses only the monomer basis set for the monomer wave functions). The DCBS is ill-defined, and therefore unavailable, for use with XPol charge embedding. The projected basis set is an efficient approximation to the DCBS for traditional SAPT calculations,[45] and *is* available for use with XPol embedding.

CPHF

Requests that the second-order corrections $E^{(2)}_{\mathrm{ind}}$ and $E^{(2)}_{\mathrm{exch\text{-}ind}}$ be replaced by their infinite-order "response" analogues, $E^{(2)}_{\mathrm{ind,resp}}$ and $E^{(2)}_{\mathrm{exch\text{-}ind,resp}}$.

INPUT SECTION: *$sapt*

TYPE:

None

DEFAULT:

Not specified

OPTIONS:

Response quantities are calculated if the keyword is present.

RECOMMENDATION:

Computing the response corrections requires solving CPHF equations for each pair of monomers, which is somewhat expensive but may improve the accuracy, especially when the monomers are polar and induction contributions are large.

DSCF

Requests the $\delta E^{\mathrm{HF}}_{\mathrm{int}}$ correction

INPUT SECTION: *$sapt*

TYPE:

None

DEFAULT:

Not specified

OPTIONS:

The $\delta E^{\mathrm{HF}}_{\mathrm{int}}$ correction is computed if this keyword is present.

RECOMMENDATION:

Evaluating $\delta E^{\mathrm{HF}}_{\mathrm{int}}$ requires an SCF calculation on the entire (super)system. In the context of SAPT0 calculations, this correction essentially results in a "Hartree-Fock plus dispersion" estimate of the interaction energy.

Print

Specifies the level of output for the XPol code.

INPUT SECTION: *$sapt*

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Desired print level

RECOMMENDATION:

Higher values print additional information

Example 12.39 Example showing a SAPT0 calculation using the RI approximation in a dimer-centered basis.

```
$molecule
0 1
-- formamide
   0 1
   C -2.018649  0.052883  0.000000
   O -1.452200  1.143634  0.000000
   N -1.407770 -1.142484  0.000000
   H -1.964596 -1.977036  0.000000
   H -0.387244 -1.207782  0.000000
   H -3.117061 -0.013701  0.000000
-- formamide
   0 1
   C  2.018649 -0.052883  0.000000
   O  1.452200 -1.143634  0.000000
   N  1.407770  1.142484  0.000000
   H  1.964596  1.977036  0.000000
   H  0.387244  1.207782  0.000000
   H  3.117061  0.013701  0.000000
$end

$rem
   JOBTYPE           XSAPT
   BASIS             AUG-CC-PVDZ
   AUX_BASIS         RIMP2-AUG-CC-PVDZ
   METHOD            HF
$end

$sapt
  algorithm    ri-mo
  basis        dimer
$end
```

12.10 The XPol+SAPT (XSAPT) Method

12.10.1 Introduction

The "XSAPT" method, which may be regarded either as an acronym for "XPol+SAPT" or for "extended" symmetry adapted perturbation theory (SAPT), was originally introduced by Jacobson and Herbert,[31,45] and later by Lao and Herbert,[59,60,62,64,65] as a low-scaling, systematically-improvable method for intermolecular interactions that could be

applicable to large systems. The idea was to replace the need for empirical parameters in the XPol method with on-the-fly evaluation of exchange-repulsion and dispersion interactions via pairwise-additive SAPT. Stated differently, XSAPT uses XPol to evaluate many-body (non-pairwise-additive) polarization effects, but then assumes that dispersion and exchange-repulsion interactions *are* pairwise additive, and evaluates them via pairwise SAPT0 or SAPT0(KS) calculations. The method was significantly extended by Lao, Herbert, and co-workers,[13,27,59,60,62,64,65,75] with various approximations applied in place of the SAPT0 or SAPT0(KS) dispersion terms,[14] which are both the least accurate and most expensive contributions to second-order SAPT. A concise overview of XSAPT-based methods can be found in Ref. 14 and a comprehensive review in Ref. 33. In particular, the XSAPT+MBD method[13] stands out as a way to obtain qualitative insight about noncovalent interactions in large systems, backed by *quantitative* energetics calculations.[14] In many cases, this type of analysis has upended textbook "conventional wisdom".[30]

12.10.2 Theory

The zeroth-order Hamiltonian for XSAPT is taken by the sum of fragment Fock operators defined by the XPol procedure, and the perturbation is the usual SAPT intermolecular perturbation [Eq. (12.56)] less the intermolecular interactions contained in the XPol fragment Fock operators. A standard SAPT0 correction (see Section 12.9) is then computed for each pair of monomers, using Eq. (12.61) in conjunction with the modified perturbation. This affords the dimer interaction energy, $E_{\rm int}^{AB}$. The total XSAPT energy is[31]

$$E_{\rm XSAPT} = \sum_A \left(\sum_a \Big[2\,\epsilon_a^A - \mathbf{c}_a^\dagger (\mathbf{J}^A - \tfrac{1}{2}\mathbf{K}^A)\mathbf{c}_a \Big] + E_{\rm nuc}^A + \sum_{B>A} E_{\rm int}^{AB} \right) , \tag{12.67}$$

which is equal to the sum of the XPol monomer energies plus the pairwise SAPT corrections. In this expression, we have removed the over-counting of two-electron interactions present in Hartree-Fock theory, effectively taking the *intra*fragment perturbation to first order. The generalization to a Kohn-Sham description of the monomers is straightforward, which extends the SAPT0(KS) approach to clusters larger than dimers.

The inclusion of many-body polarization within the zeroth-order Hamiltonian makes the subsequent SAPT corrections less meaningful in terms of energy decomposition analysis. For instance, the first-order electrostatic correction in XSAPT is *not* the total electrostatic energy, since the former corrects for errors in the approximate electrostatic treatment at zeroth order (*i.e.*, the electrostatic embedding). In order to replenish some of the significance of the XSAPT electrostatics, a "corrected" electrostatic energy is obtained by subtracting the XPol embedding potential from the first-order electrostatic energy obtained in SAPT, effectively removing the zeroth-order corrections from the first-order electrostatics. The dispersion correction may be less contaminated, since all of the XSAPT modifications to the traditional SAPT perturbation are one-electron operators and therefore the pairwise dispersion correction differs from its traditional SAPT analogue only insofar as the MOs are perturbed by the electrostatic embedding. This should be kept in mind when interpreting the output of an XSAPT calculation, although Lao and Herbert[60,62] later proposed a many-body energy decomposition scheme for XSAPT that extends traditional SAPT energy decomposition to systems containing more than two monomers. (The aforementioned contamination problems are avoid through pairwise $\delta_{\rm int}^{\rm HF}$ corrections, comparing XSAPT results to traditional SAPT based on gas-phase monomers.)

An XSAPT calculation is requested by setting JOBTYPE = XSAPT in the *$rem* section. The choice of XPol charge embedding is controlled by the **embed** and **charges** keywords in the *$xpol* input section; see Section 12.8 and the example provided below. Additional job control options for the SAPT part of the calculation are specified in the *$sapt* input section as described in Section 12.9. Researchers who use Q-CHEM's XSAPT code are asked to cite Refs. 31,33.

Additional discussions of the theory can be found elsewhere.[45,46]

Example 12.40 Example of an XPol + SAPT0 calculation using ChElPG charges for the XPol calculation and computing $E^{(2)}_{\mathrm{ind,resp}}$ and $E^{(2)}_{\mathrm{exch\text{-}ind,resp}}$ by solving CPHF equations as discussed in Section 12.9.

```
$molecule
0 1
-- formic acid
   0 1
   C  -1.888896 -0.179692  0.000000
   O  -1.493280  1.073689  0.000000
   O  -1.170435 -1.166590  0.000000
   H  -2.979488 -0.258829  0.000000
   H  -0.498833  1.107195  0.000000
-- formic acid
   0  1
   C  1.888896  0.179692  0.000000
   O  1.493280 -1.073689  0.000000
   O  1.170435  1.166590  0.000000
   H  2.979488  0.258829  0.000000
   H  0.498833 -1.107195  0.000000
$end

$rem
   JOBTYPE          XSAPT
   BASIS            CC-PVDZ
   METHOD           HF
$end

$xpol
   embed   charges
   charges CHELPG    ! charges derived from electrostatic potential
$end

$sapt
   basis projected   ! use the pseudocanonicalized dimer basis
   CPHF              ! solve CPHF equations for induction response
$end
```

The latter example is simply a traditional SAPT0 (dimer) calculation but based on zeroth-order monomer wave functions computed from a charge-embedded XPol calculation. The following example corresponds to a truly "extended"

SAPT calculation, *i.e.*, one with more than two monomers.

Example 12.41 XSAPT calculation on water tetramer using the LRC-ωPBEh functional. Includes the three-body induction couplings that arise at second order in perturbation theory when the number of monomers is greater than 2 (see Ref. 31).

```
$molecule
0 1
-- water
0 1
        O        -0.459965    1.488925    0.391165
        H         0.442885    1.099622    0.558106
        H        -0.551255    2.236567    0.999244
-- water
0 1
        O        -1.111823   -1.126854    0.565807
        H        -1.153929   -0.145562    0.663733
        H        -2.016599   -1.451826    0.678719
-- water
0 1
        O         1.661160   -0.139676    0.530681
        H         1.455561   -0.313184   -0.421143
        H         1.146044   -0.835459    0.974417
-- water
0 1
        O         0.201725   -0.384036   -1.774045
        H        -0.394336   -0.876966   -1.168916
        H        -0.094680    0.533258   -1.645074
$end

$rem
   JOBTYPE     xsapt
   EXCHANGE    gen
   BASIS       6-31G*
$end

$xpol
  embed   charges
  charges chelpg
$end

$sapt
  algorithm  mo        ! could be ri-mo for RI approximation
  basis      projected ! default choice; recommended
  3b-ind     ! include the 3-body induction couplings (optional)
$end

$xc_functional
x   wPBE  0.8
x   HF    0.2
c   PBE   1.0
$end
```

12.10.3 Dispersion Models

SAPT(KS) calculations and their many-body extension, XSAPT(KS), uses a Kohn-Sham DFT description of the monomers in order to introduce *intra*molecular electron correlation in a low-cost way, then described the *inter*molecular interactions using second-order SAPT. As mentioned in The resulting interaction energies, however, are not of benchmark quality even when tuned LRC functionals are employed,[61] because although the use of DFT for the monomers

often improves the description of hydrogen bonding (relative to Hartree-Fock-based SAPT0 calculations), the description of dispersion often deteriorates.[31] In any case, SAPT0 dispersion is not of benchmark quality anyway, as it suffers from the usual MP2 overestimation of dispersion. At the same time the dispersion and exchange-dispersion terms are the most expensive parts of a SAPT0 or SAPT(KS) calculation, with a formal scaling of $\mathcal{O}(N^4)$ and $\mathcal{O}(N^5)$, respectively, with respect to system size. Other terms in SAPT0 scale no worse than $\mathcal{O}(N^3)$ and can be computed efficiently for large monomers using an atomic orbital (AO)-based implementation of the non-dispersion terms in SAPT.[64]

In view of this, both the efficiency and the accuracy of XSAPT(KS) calculations is improved if second-order dispersion, *i.e.*, $E_{\text{disp}}^{(2)} + E_{\text{exch-disp}}^{(2)}$ in Eqs (12.61) and (12.65), is replaced by an *ad hoc* atom–atom dispersion potential of the $-C_6/R^6 - C_8/R^8 - \cdots$ variety. This is reminiscent of dispersion-corrected DFT or DFT-D, as described in Section 5.7.3. Unlike the situation in DFT, however, the dispersion energy is well-defined and separable within the SAPT formalism, so it can be replaced by atom–atom potentials without any fear of double counting of correlation effects, as there inevitably is in DFT-D. Moreover, in the present case the dispersion potentials can be fit directly to *ab initio* dispersion energies from high-level SAPT calculations [SAPT(DFT) and SAPT2+(3)], since the dispersion contribution is separable. As such, while the dispersion potentials that are described here are classical in form and do contain fitting parameters, they can nevertheless reasonably be described as *ab initio* dispersion potentials. We therefore describe this method as "+*ai*D",[64] to distinguish it from the "+D" dispersion corrections of DFT-D, although we simply called it "+D" in earlier work.[59,60,62] The composite method is called XSAPT(KS)+*ai*D; see Ref. 60 for an overview and Ref. 64 for an efficient implementation in the AO basis. The latter version exhibits $\mathcal{O}(N^3)$ scaling without significant memory bottlenecks, and is applicable to supramolecular complexes whose monomers contain $\gtrsim 100$ atoms.[64]

To request an XSAPT(KS)+*ai*D calculation, set JOBTYPE = XSAPT in the *$rem* section to perform XSAPT, with an appropriate choice of SCF method (Hartree-Fock or DFT). The +*ai*D part of the algorithm is invoked by two keywords in the *$sapt* input section: first, set **Algorithm** to **AO** to select the $\mathcal{O}(N^3)$ AO-based version of XSAPT; and second, set **Dispersion** equal to **aiD**, **aiD2**, **aiD3**, or **MBD**. The latter choices correspond, respectively, to the "first generation" (+*ai*D1) potential,[59] the second-generation (+*ai*D2) potential,[60], the third-generation (+*ai*D3) dispersion potential,[62] or the many-body dispersion (+MBD) potential.[13] All four versions exhibit similar performance for total interaction energies in small molecules,[13,62] but unlike its successors, the +*ai*D1 potential was fit to reproduce total interaction energies rather than being fit directly to *ab initio* dispersion data, and as a consequence does a much poorer job of reproducing individual energy components. (It was later discovered that the performance of +*ai*D1 benefits from some error cancellation amongst energy components,[60,62] and as such its use is not recommended.) The difference between +*ai*D2 and +*ai*D3 is a larger training set for the latter, which was designed to afford better coverage of π-stacked systems. As such, the +*ai*D3 correction is the superior choice out of the pairwise potentials in the +*ai*D suite of methods.

The first three generations of +*ai*D potentials make the pairwise approximation, where the interaction potential is assumed to be additive across all pairs of atoms. The pairwise dispersion approximation employs sums over atom pairs of the form,

$$E_{\text{disp}} = -\sum_{i\in A}\sum_{\substack{j\in B\\ A\neq B}} \left[f_6(R_{ij})\frac{C_6^{ij}}{R_{ij}^6} + f_8(R_{ij})\frac{C_8^{ij}}{R_{ij}^8} \right], \tag{12.68}$$

where i and j are nuclei in molecules A and B, respectively. The pairwise approximation breaks down in the limit of very large systems because the interactions between atom pairs are modulated by the local electrodynamic environment in the molecule. It was discovered that even the +*ai*D3 potential (the best of the pairwise +*ai*D potentials) suffers from this approximation in large systems,[64] and a correction based on the difference between XSAPT and SAPT dispersion energies was proposed.[65] While this correction performs well, all of the pairwise dispersion potentials (+*ai*D1, +*ai*D2, and +*ai*D3) are rather *ad hoc* and their corrections do not depend on the applied level of theory. The most recent +MBD potential uses a modified version of the many-body dispersion potential of Ambrosetti *et al.*[5] (see Section 5.7.6 in order to naturally account for nonadditive dispersion effects.[13] Because the +MBD method is based on the electron density it is much more connected to the *ab initio* method being used, and this is presently the more accurate version

of XSAPT.[14] When combined with the XSAPT procedure, the XSAPT+MBD energy decomposition accounts for nonadditive polarization and dispersion effects. Due to its excellent performance regardless of system size, the +MBD potential (**Dispersion MBD**) is recommended, but the +*ai*D3 potential (**Dispersion aiD3**) remains quite good for smaller systems.

As with XPol, the XSAPT and XSAPT(KS)+*ai*D methods do not function with a solvation model or with external changes. Only single-point energies are available, and frozen orbitals orbitals are not allowed. Both restricted and unrestricted versions are available. Researchers who use XSAPT(KS)+*ai*D are asked to cite Ref. 59 for +*ai*D1, Ref. 60 for +*ai*D2, Ref. 62 for +*ai*D3, or Ref. 13 for +MBD, along with Ref. 64 for the AO-based version of XSAPT.

Dispersion

Requests a +*ai*D dispersion potential.

INPUT SECTION: *$sapt*

TYPE:

STRING

DEFAULT:

aiD3

OPTIONS:

aiD	First-generation pairwise dispersion potential
aiD2	Second-generation pairwise dispersion potential
aiD3	Third-generation pairwise dispersion potential
MBD	Many-body dispersion potential

RECOMMENDATION:

Use MBD. The *ai*D2, *ai*D3, and MBD potentials were parameterized using *ab initio* dispersion data and afford accurate energy components, in addition to accurate total interaction energies. The *ai*D3 potential was parameterized using an expanded data set designed to reduce some large errors observed for π-stacked complexes using *ai*D2. The MBD potential accounts for many-body dispersion effects that are very important even in moderately large systems.

Example 12.42 XSAPT+MBD with PCM boundary conditions.[26]

```
$molecule
0 1
--
0 1
C    0.780147171  -0.609914733  -1.207556891
H    0.896191595  -1.137639594  -2.144144625
C    0.477942753   0.750993631  -1.207895407
H    0.356964231   1.278167803  -2.144054074
C    0.327289279   1.431867868  -0.000000000
H    0.091465028   2.487139215   0.000000000
C    0.477942754   0.750993631   1.207895407
H    0.356964231   1.278167803   2.144054074
C    0.780147171  -0.609914733   1.207556891
H    0.896191595  -1.137639594   2.144144625
C    0.931648311  -1.289981342   0.000000000
H    1.168485730  -2.345213690  -0.000000000
--
0 1
O   -2.743831210  -0.269262567   0.000000000
H   -2.579027215  -1.213984095   0.000000000
H   -1.856530267   0.102327758   0.000000000
$end

$rem
jobtype  xsapt
basis  def2-svpd
lrc_dft  true
method  lrc-wpbe
scf_convergence 8
solvent_method  pcm
$end

$xpol
embed none
dft-lrc
print 3
$end

$sapt
algorithm ao
basis projected
order 2
print 3
dispersion mbd
$end

$lrc_omega
341
502
$end
```

12.10.4 Running an XSAPT+MBD Job

In order to run an XSAPT job, at least two separate jobs must be run, a third is optional and suggested in the case where the induction energy is thought to be important. The first job that must be run is a second-order SAPT job without charge embedding on the full system. This job provides the value for the electrostatic term, the exchange term, and the

dispersion term. (For an XSAPT+MBD method[13] job, the dispersion value will be the dispersion value obtained from the MBD calculation). The second job that needs to be run is a SAPT job with charge embedding on the full system. This SAPT job with charge embedding acts as a correction for the induction energy and must be run on each pair of monomers. The final induction energy for an XSAPT job is then the induction energy from the SAPT job without charge embedding plus the difference between the total SAPT energy with charge embedding minus the total SAPT energy without charge embedding. In equation form, the XSAPT contribution to the induction energy would be:

$$E_{\text{pol}} = \sum_{A} \sum_{B>A} \left[E_{AB}^{\text{XSAPT}}(AB) - E_{AB}^{\text{SAPT}} \right] \tag{12.69}$$

Here the term in brackets is the difference between the charge-embedded energy for the AB dimer and the SAPT energy computed without charge embedding. Therefore, the electrostatic, exchange, and dispersion energy would be the equivalent to the same terms calculated from a normal SAPT calculation, it is the induction energy that changes upon an XSAPT calculation. An additional calculation can be performed to further correct the induction energy, this is the δHF calculation, which accounts for higher than second-order induction energy. The resulting delta(SCF) energy term is then added to the induction energy term calculated as described above.

The above procedure is the required procedure for performing an XSAPT ($+\delta$HF) calculation on a dimer complex. In order to perform an XAPT ($+\delta$HF) calculation on an n-mer complex, several more calculations need to be performed. The first of which is to run a SAPT job with charge embedding on each pair of monomers. This is required because the polarization of the dimers differs from that of the dimers embedded in the charges of the entire system. The second set of jobs, if the δHF correction is being performed, is a δHF job on every pair of monomers. These sets of jobs will again provide a correction to the induction energy for the system.

Two scripts to help the user perform an XSAPT+MBD+δHF calculation exist in the `$QC/bin/tools` directory. The script named `xsapt_creation.py` creates three files for an arbitrary geometry (defined by the user), a SAPT job without embedding, a SAPT job with embedding, and a δHF job. See the script for instructions on how to use and modify the script for personal use. The other script named `xsapt_data_collection.py` looks through the output files of the three jobs created by `xsapt_creation.py` and performs the needed arithmetic to obtain the correct energy decomposition for an XSAPT+MBD+δHF job. See the script for instructions on how to use and modify

it for personal use.

Example 12.43 AO-XSAPT(KS)+*ai*D3 calculation of water-water interaction.

```
$molecule
0 1
--
   0 1
   O  -1.551007  -0.114520   0.000000
   H  -1.934259   0.762503   0.000000
   H  -0.599677   0.040712   0.000000
--
   0 1
   O   1.350625   0.111469   0.000000
   H   1.680398  -0.373741  -0.758561
   H   1.680398  -0.373741   0.758561
$end

$rem
   JOBTYPE            xsapt
   EXCHANGE           gen
   LRC_DFT            true ! add LR-HFX
   BASIS              aug-cc-pVTZ
   MEM_TOTAL          46000
   MEM_STATIC         4000
   AO2MO_DISK         35000
   CHELPG_DX          5
   CHELPG_HEAD        30
   CHELPG_H           110
   CHELPG_HA          590
$end

$xpol
   embed   charges
   charges CHELPG
   DFT-LRC
   print   3
$end

$sapt
   algorithm     ao  ! for use with +aiD dispersion
   order         2   ! 2nd-order SAPT
   basis         projected
   Dispersion    aiD3
   print         3
$end

$xc_functional
   x   wPBE  1.0
   c   PBE   1.0
$end

$lrc_omega
   502
   502
$end
```

12.11 Energy Decomposition Analysis Based on SAPT/cDFT

12.11.1 Overview

Many schemes for decomposing quantum chemical calculations of intermolecular interaction energies into physically meaningful components can be found in the literature, but the definition of the charge-transfer (CT) contribution has proven particularly vexing to define in a satisfactory way and typically depends strongly on the choice of basis set,[63,103,113] because as virtual orbitals on monomer A start to extend significantly over monomer B as the basis set approaches completeness, the distinction between polarization (excitations localized on A, introduced by the perturbing influence of B) and CT (excitations from A to B) becomes blurred.[63] This ambiguity renders orbital-dependent definitions of CT highly dependent on the choice of atomic orbital basis set. On the other hand, constrained density functional theory (cDFT, Section 5.11),[50] by means of which a CT-free reference state can be defined based on "promolecule" densities, affords a definition of CT that is scarcely dependent on the basis set and is in accord with chemical intuition in simple cases.[63]

For intermolecular interactions, the cDFT definition of CT can be combined with a definition of the remaining components of the interaction energy (electrostatics, induction, Pauli repulsion, and van der Waals interactions) based on symmetry-adapted perturbation theory (SAPT, Section 12.9). In traditional SAPT, the CT interaction energy resides within the induction energy (also known as the polarization energy), which is therefore itself highly dependent upon the basis set. However, using cDFT to define the CT component and subtracting this out of the SAPT induction energy, both the CT and the remaining induction energies are largely independent of basis set.[63] SAPT/cDFT therefore provides a stable and physically-motivated energy decomposition.

While the cDFT definition of CT exhibits only a very mild basis-set dependence, its quantitative details do depend upon how the charge constraints in cDFT are defined relative to fragment populations (Section 5.11). For SAPT/cDFT, both atomic Becke[9] and fragment-based Hirshfeld[113] (FBH) charge partitioning methods are available. The former involves construction of atomic cell functions that amount to smoothed Voronoi polyhedra centered about each atom. A switching function defines the atomic cell of atom a, and falls rapidly from ≈ 1 near the nucleus for atom a, to ≈ 0 near any other nucleus. Becke[9] defined atomic cell functions $P_a(\mathbf{r})$ that are products of switching functions and that can be used to define the cDFT integration weight for monomer A by summing over atoms $a \in A$:

$$w_A^{\text{Becke}}(\mathbf{r}) = \frac{\sum_{a\in A} P_a(\mathbf{r})}{\sum_b P_b(\mathbf{r})} \,. \tag{12.70}$$

The sum in the denominator runs over all atoms in both monomers, A and B. Becke populations, however, are rooted in a somewhat arbitrarily-defined topology, based in part on assumed atomic radii, whereas FBH partitioning derives physical significance from isolated monomer densities $\widetilde{\rho}_A(\mathbf{r})$ and $\widetilde{\rho}_B(\mathbf{r})$. The cDFT weight function for monomer A is[113]

$$w_A^{\text{FBH}}(\mathbf{r}) = \frac{\widetilde{\rho}_A(\mathbf{r})}{\widetilde{\rho}_A(\mathbf{r}) + \widetilde{\rho}_B(\mathbf{r})} \,, \tag{12.71}$$

which is the same "stockholder" scheme used to define atomic Hirshfeld populations (Section 10.2.2), but applied here to the entire monomer. In the language of cDFT, the denominator in this expression would be called the *promolecule* density for the dimer $A + B$. In order to set a molecular fragment constraint, simply retain the existing syntax in the *$cdft* input section (as described in Section 5.11) and specify all atoms within a given molecular fragment.

Due to the fact that Becke populations are rooted in a topological scheme based in part on assumed atomic radii, it is highly recommended that if CDFT_POP is set to BECKE, the rem variable BECKE_SHIFT should be set to use either the empirically derived Bragg-Slater radii[106] or *ab initio* derived radii based on the universal density criterion[95] (see Section 10.2.2 for more details). Using the UNSHIFTED (default) scheme can lead to highly unphysical results, including a charge-transfer vector that points in the opposite direction.

To perform SAPT/cDFT energy decomposition analysis, the user must request a normal SAPT or XSAPT calculation (JOBTYPE = XSAPT), and in addition specify the keyword **CDFT-EDA** in the *$sapt* input section. Users of this method are asked to cite Ref. 63.

CDFT-EDA

Requests a SAPT/cDFT-based energy decomposition analysis.

INPUT SECTION: *$sapt*

TYPE:

None

DEFAULT:

Not specified

OPTIONS:

The analysis is performed if the keyword is set.

RECOMMENDATION:

None

As shown in the example below, a *$cdft* input section is also required in order to specify the monomer charges and spins for the cDFT part of the calculation. The CDFT_POP variable may be set (in the *$rem* section) in order to specify the electron-counting mechanism for cDFT. The options are either to use atomic Becke populations (as in traditional cDFT calculations[50]), summed up for each monomer, or else fragment-based Hirshfeld partitioning in which promolecule densities for the monomers are used to obtain a whole-molecule version of Hirshfeld atomic charges.

CDFT_POP

Sets the charge partitioning scheme for cDFT in SAPT/cDFT

TYPE:

STRING

DEFAULT:

FBH

OPTIONS:

FBH Fragment-Based Hirshfeld partitioning

BECKE Atomic Becke partitioning

RECOMMENDATION:

None

Example 12.44 Energy decomposition analysis for the water dimer using AO-SAPT+*ai*D3/cDFT. The *$cdft* input section is required in order to specify the monomer charges and spins for the cDFT segment of the calculation.

```
$molecule
0 1
--
   0 1
   O   -0.702196054  -0.056060256   0.009942262
   H   -1.022193224   0.846775782  -0.011488714
   H    0.257521062   0.042121496   0.005218999
--
0 1
   O    2.220871067   0.026716792   0.000620476
   H    2.597492682  -0.411663274   0.766744858
   H    2.593135384  -0.449496183  -0.744782026
$end

$rem
   JOBTYPE            XSAPT
   EXCHANGE           gen
   BASIS              aug-cc-pvdz
   LRC_DFT            true
   CDFT_POP           FBH ! Fragment-Based Hirshfeld charge partitioning
$end

$xpol
  embed     none
  print     3
  dft-lrc
$end

$sapt
  algorithm     AO
  Dispersion    aiD3
  order         2
  basis         dimer
  cdft-eda
  print         3
$end

$xc_functional
   x   wPBE   1.0
   c    PBE   1.0
$end

$lrc_omega
   500
   500
$end

$cdft
   0
   1 1 3
   0
   1 1 3 s
$end
```

12.12 The Many-Body Expansion Method

12.12.1 Introduction

The many-body expansion (MBE) for a system of N monomers is given by

$$E = \sum_{I=1}^{N} E_I + \sum_{I}^{N}\sum_{J>I}^{N} \Delta E_{IJ} + \sum_{I}^{N}\sum_{J>I}^{N}\sum_{K>J}^{N} \Delta E_{IJK} + \cdots , \tag{12.72}$$

in which E_I represents the energy of monomer I and

$$\Delta E_{IJ} = E_{IJ} - E_I - E_J \tag{12.73a}$$

$$\Delta E_{IJK} = E_{IJK} - \Delta E_{IJ} - \Delta E_{IK} - \Delta E_{JK} - E_I - E_J - E_K \tag{12.73b}$$

are a two-body corrections for dimer IJ and a three-body correction for trimer IJK, respectively. In a large system and/or a large basis set, truncation of this expression at the two- or three-body level may dramatically reduce the amount of computer time that is required to compute the energy. Convergence of the MBE can be accelerated by embedding the monomer (E_I), dimer (E_{IJ}), trimer (E_{IJK}), ... calculations in some representation of the electrostatic potential of the rest of the system. A simple means to do this is via atom-centered point charges that could be obtained when the E_I terms are calculated; this is the so-called electrostatically-embedded many-body expansion (EE-MBE),[29,66,101,102] which we will denote as EE-MBE(n) when the expansion is truncated at n-body terms. MBE(n) and EE-MBE(n) are available in Q-CHEM, with analytic gradients, up to five-body terms ($n = 5$). Caution should be exercised when using MBE(n) or EE-MBE(n) in conjunction with DFT, especially for systems composed of small, ionic fragments (such as ion–water interactions) because delocalization error can manifest as unphysically large many-body interactions.[11,12]

It is well known that the interaction energies of non-covalent clusters are usually overestimated—often substantially—owing to basis-set superposition error (BSSE), which disappears only very slowly as the basis sets approach completeness. The widely used Boys-Bernardi counterpoise procedure corrects for this by computing all energies, cluster and individual monomers, using the full cluster basis set. (In clusters with more than two monomers, the obvious generalization of the Boys-Bernardi counterpoise correction is sometimes called the "site–site function counterpoise" correction or SSFC.) Note, however, that basis-set extrapolation is still necessary for high-quality binding energies. In $(H_2O)_6$, for example, a counterpoise-corrected MP2/aug-cc-pVQZ calculation is still ≈ 1 kcal/mol from the MP2 basis-set limit.[99] Fortunately, the MBE allows for use of large basis sets in order to perform basis-set extrapolations in sizable clusters,[99,100] and one can employ a counterpoise correction that is consistent with an n-body expansion in order to obtain an n-body approximation to the Boys-Bernardi counterpoise-corrected supersystem energy. Two such corrections have been proposed: the many-body counterpoise correction, MBCP(n),[99,100] and the n-body Valiron-Mayer function counterpoise correction, VMFC(n).[51] The two approaches are equivalent for $n = 2$ but the MBCP(n) method requires far fewer subsystem calculations starting at $n = 3$ and is thus significantly cheaper, while affording very similar results as compared to VMFC(n).[99,100]

12.12.2 Job Control

A MBE(n) calculation is requested by setting MANY_BODY_INT = TRUE in the *$rem* section. The level of theory used for the fragments will be whatever is specified in the *$rem* section. Researchers who use Q-CHEM's MBE code are asked to cite Ref. 66,101. In addition, please cite Ref. 99 for the MBCP(n) method.

A MBE(n) calculation is requested by setting MANY_BODY_INT = TRUE in the *$rem* section. The level of theory used for the fragments will be whatever is specified in the *$rem* section. Researchers who use Q-CHEM's MBE code are asked to cite Ref. 66,101. In addition, please cite Ref. 99 for the MBCP(n) method.

MANY_BODY_INT

Perform a MBE calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a MBE calculation.

FALSE Do not perform a MBE calculation.

RECOMMENDATION:

NONE

Additional MBE-specific options, such as the order of the expansion (n), are specified in a *$mbe* input section, as described below.

Order

Specifies the order of the many-body expansion.

INPUT SECTION: *$mbe*

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n Perform an MBE(n) calculation.

RECOMMENDATION:

Orders $n \leq 5$ are available.

Embed

Specifies the embedding method for EE-MBE(n).

INPUT SECTION: *$mbe*

TYPE:

STRING

DEFAULT:

None

OPTIONS:

None Do not use embedding.

Charges Use atomic point charges.

Density Full Coulomb embedding using monomer densities.

RECOMMENDATION:

Use of atomic point charges requires a *$mbe_charges* section to specify the charges.

Q-CHEM's implementation of EE-MBE(n) with atomic point charges is designed to use with a *$mbe_charges* input section to specify fixed embedding charges. (Use of these charges is intended to accelerate convergence of the MBE by capturing many-body polarization effects and thus making the higher-order n-body terms smaller, although three- and four-body terms remain non-negligible even with embedding charges.[66,74,102]) The format of the *$mbe_charges* section is simply a list of charges in the same order as the atoms in the *$molecule* section. An example is provided below.

Many-body counterpoise corrections are requested with two keywords in the *$mbe* input section: **BSSE_Type** and

BSSE_Order. These have only been implemented up to $n = 3$, as the $n = 2$ terms make by far the most significant contribution.[74]

BSSE_Order

Perform a many-body counterpoise correction of the MBCP(n) or VMFC(n) variety.

INPUT SECTION: *$mbe*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform a counterpoise correction.

n Perform a counterpoise correction truncated at order n.

RECOMMENDATION:

Orders $n \leq 3$ are available. Use the keyword **BSSE_Type** to choose between MBCP and VMFC.

BSSE_Type

Select the type of many-body counterpoise correction, MBCP(n) or VMFC(n).

INPUT SECTION: *$mbe*

TYPE:

STRING

DEFAULT:

MBCP

OPTIONS:

MBCP Use MBCP(n).

VMFC Use VMFC(n).

RECOMMENDATION:

The two methods are equivalent for $n = 2$ but different for $n \geq 3$. MBCP(n) contains fewer terms but generally provides comparable results as compared to the formally more complete VMFC(n) approach.

Example 12.45 Example showing a EE-MBE(3) calculation using TIP3P charges.

```
$molecule
0 1
--
   0 1
   O -1.126149 -1.748387 -0.423240
   H -0.234788 -1.493897 -0.661862
   H -1.062789 -2.681331 -0.218819
--
   0 1
   O -0.254210 1.611495 -1.293845
   H -1.001520 1.163510 -1.690129
   H -0.153399 2.411746 -1.809248
--
   0 1
   O 1.694541 -0.226287 1.705739
   H 0.785920 0.073487 1.677909
   H 2.047134 0.150917 2.511706
--
   0 1
   O -0.864533 0.522472 1.218817
   H -0.694120 1.093542 0.469789
   H -1.131418 -0.310426 0.829702
$end

$rem
point_group_symmetry False
   METHOD            B3LYP
   BASIS             cc-pVDZ
   MANY_BODY_INT     true
   THRESH            14
   SCF_CONVERGENCE   7
$end

$mbe
  order 3
  embed charges
$end

$mbe_charges
  -0.834
   0.417
   0.417
  -0.834
   0.417
   0.417
  -0.834
   0.417
   0.417
  -0.834
   0.417
   0.417
$end
```

Example 12.46 Example of a MBCP(3) calculation.

```
$molecule
0 1
--
   0 1
   O -1.126149 -1.748387 -0.423240
   H -0.234788 -1.493897 -0.661862
   H -1.062789 -2.681331 -0.218819
--
   0 1
   O -0.254210  1.611495 -1.293845
   H -1.001520  1.163510 -1.690129
   H -0.153399  2.411746 -1.809248
--
   0 1
   O  1.694541 -0.226287  1.705739
   H  0.785920  0.073487  1.677909
   H  2.047134  0.150917  2.511706
--
   0 1
   O -0.864533  0.522472  1.218817
   H -0.694120  1.093542  0.469789
   H -1.131418 -0.310426  0.829702
$end

$rem
   MANY_BODY_INT     TRUE
   METHOD            B3LYP
   BASIS             cc-pVDZ
   THRESH            12
   SCF_CONVERGENCE   6
$end

$mbe
  BSSE_Order 3
  BSSE_Type MBCP ! this is the default
$end
```

12.13 The Generalized Many-Body Expansion Method

12.13.1 Introduction

The generalized many-body expansion (GMBE) method approximates the energy of a supersystem using the energies of its fragments or subsystems, determined by a distance-based threshold.[86,97,98] This threshold can be defined by the minimum atomic distance between "unit fragments" of the supersystem. Alternatively, one may use a criterion based on the heavy-atom distance, which excludes hydrogen atoms. The resulting subsystems, identified based on this threshold, are referred to as primitive fragments (monomers). These primitive fragments serve as the foundation for generating a set of fragment calculations used to approximate the supersystem's energy. To prevent redundancy, the principle of inclusion and exclusion (PIE) is enforced.

In the n-body GMBE, denoted as GMBE(n), the energy of the supersystem is expressed as

$$E = \sum_{I}^{\binom{N}{n}} \mathcal{E}_I^{(n)}, \tag{12.74}$$

where N is the number of primitive fragments.

The GMBE implementation in Q-CHEM is currently limited to first-order, GMBE(1), and employs a novel binning algorithm for efficiently generating the set of primitive fragments.[7] The supersystem's energy is approximated using GMBE(1) as:

$$E \approx \sum_{I}^{n} \mathcal{E}_I^{(1)}. \tag{12.75}$$

The intersection-corrected energy for fragment I is

$$\mathcal{E}_I^{(1)} = E_I^{(1)} - \sum_{\substack{J \\ (J>I)}} E_{I\cap J}^{(1)} + \sum_{\substack{J,K \\ (K>J>I)}} E_{I\cap J\cap K}^{(1)} - \sum_{\substack{J,K,L \\ (L>K>J>I)}} E_{I\cap J\cap K\cap L}^{(1)} + \cdots, \tag{12.76}$$

where $E_I^{(1)}$, $E_{I\cap J}^{(1)}$, $E_{I\cap J\cap K}^{(1)}$, and $E_{I\cap J\cap K\cap L}^{(1)}$ denote energies respectively of the fragment I, the intersection of fragments I and J, the intersection of fragments I, J, and K, and the intersection of fragments I, J, K, and L.

Similarly, the GMBE(1) approximation for the supersystem's density matrix is given by:

$$P \approx \sum_{I}^{n} \mathcal{P}_I^{(1)} \tag{12.77}$$

where the intersection-corrected density matrix sub-blocks for subset I are defined as:

$$\mathcal{P}_I^{(1)} = P_I^{(1)} - \sum_{\substack{J \\ (J>I)}} P_{I\cap J}^{(1)} + \sum_{\substack{J,K \\ (K>J>I)}} P_{I\cap J\cap K}^{(1)} - \sum_{\substack{J,K,L \\ (L>K>J>I)}} P_{I\cap J\cap K\cap L}^{(1)} + \cdots \tag{12.78}$$

The GMBE(1) density matrix can be used to predict the supersystem's energy through a single Fock build. Alternatively, it can serve as an initial guess in a SCF calculation for the supersystem.

12.13.2 Job Control

To perform a GMBE(1) calculation, set GMBE_INT = TRUE in the *$rem* section. The level of theory applied to the fragments will correspond to the method specified in the *$rem* section. Researchers utilizing Q-CHEM's GMBE implementation are requested to cite Refs. 7 and 97.

The following keywords control the GMBE calculation in Q-CHEM. They should be specified in the *$rem* section of the input file.

GMBE_INT

Perform a GMBE Calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform GMBE.

FALSE Do not perform GMBE.

RECOMMENDATION:

NONE

GMBE_METHOD

Specify whether to perform an energy-based GMBE (GMBE-EB) or a density matrix-based GMBE (GMBE-DM).

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Perform GMBE-EB.

2 Perform GMBE-DM.

RECOMMENDATION:

NONE

GMBE_ALGO

Specify the cutoff distance criterion used to determine the primitive fragments.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Minimum distance

2 Heavy-atoms distance

RECOMMENDATION:

NONE

GMBE_PRIMARY

Specify the primary cutoff distance used to determine the primitive fragments. The input for GMBE_PRIMARY is the desired cutoff distance in Å, multiplied by 1000. For example, to use a 2.00 Å cutoff distance, set GMBE_PRIMARY to 2000. Similarly, if GMBE_PRIMARY = 3500, the cutoff distance will be 3.5 Å.

TYPE:

INTEGER

DEFAULT:

3000

OPTIONS:

The desired cutoff is $n = 1000d$, where d is the cutoff distance in Å.

RECOMMENDATION:

NONE

GMBE_MAX_PRIM_CARD

Set the maximum cardinality for the primitive fragments.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n represents the maximum cardinality of the primitive fragments.

RECOMMENDATION:

NONE

MBEDM_PURE

Perform McWeeny purification[87] for GMBE density matrix.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform McWeeney purification on the GMBE density matrix.

FALSE Do not perform McWeeney purification on the GMBE density matrix.

RECOMMENDATION:

NONE

GMBE_DM_GUESS_ONLY

Evaluate the total energy of the system directly using GMBE-DM density matrix. Enabling this option skips SCF iterations, preventing further convergence to the supersystem result.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Do not perform SCF iterations and directly evaluate the total energy of the system using the GMBE-DM density matrix.

FALSE Perform SCF iterations with the GMBE-DM density matrix as the initial guess.

RECOMMENDATION:

NONE

GMBE_PRINT_LEVEL

Control the level of details printed in the output file

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Minimal output

2 Detailed output

RECOMMENDATION:

NONE

Example 12.47 Example of a GMBE(1)-EB calculation for a water hexamer.

```
$rem
GEN_SCFMAN          FALSE
SYM_IGNORE          TRUE
METHOD              B3LYP
BASIS               def2-TZVPPD
THRESH              14
SCF_CONVERGENCE     7
MAX_SCF_CYCLES      200
MEM_TOTAL           8000
MEM_STATIC          4000
AO2MO_DISK          5000
GMBE_INT            TRUE ! Perform GMBE
GMBE_METHOD         1    ! 1 = energy-based, 2 = density matrix-based
GMBE_ALGO           1
GMBE_PRIMARY        2000
GMBE_MAX_PRIM_CARD 5
GMBE_PRINT_LEVEL    1    !1 = minimal output,   2 = detailed output
$end

$molecule
0 1
--
0 1
O -1.126149 -1.748387 -0.423240
H -0.234788 -1.493897 -0.661862
H -1.062789 -2.681331 -0.218819
--
0 1
O -0.254210 1.611495 -1.293845
H -1.001520 1.163510 -1.690129
H -0.153399 2.411746 -1.809248
--
0 1
O 1.694541 -0.226287 1.705739
H 0.785920 0.073487 1.677909
H 2.047134 0.150917 2.511706
--
0 1
O -0.864533 0.522472 1.218817
H -0.694120 1.093542 0.469789
H -1.131418 -0.310426 0.829702
--
0 1
O -2.312727 0.054310 -2.138177
H -2.002512 -0.686744 -1.617757
H -3.266931 0.004778 -2.081349
--
0 1
O 1.292355 -0.704768 -0.977342
H 1.642300 -0.601085 -0.092386
H 0.948122 0.159656 -1.202018
$end
```

Example 12.48 Example of generating an initial guess density matrix using a GMBE(1)-DM calculation for a water hexamer.

```
$rem
GEN_SCFMAN          FALSE
SYM_IGNORE          TRUE
METHOD              B3LYP
BASIS               def2-TZVPPD
THRESH              14
SCF_CONVERGENCE     7
MAX_SCF_CYCLES      200
MEM_TOTAL           4000
MEM_STATIC          4000
ao2mo_disk          5000
GMBE_INT            TRUE  ! Perform GMBE
GMBE_METHOD         2     ! 1 = energy-based, 2 = density matrix-based
GMBE_ALGO           1
GMBE_PRIMARY        2000
GMBE_MAX_PRIM_CARD 5
GMBE_DM_GUESS_ONLY FALSE ! Perform SCF iterations with the GMBE-DM initial guess
GMBE_PRINT_LEVEL    1     ! 1 = minimal output, 2 = detailed output
MBEDM_PURE          TRUE  ! Perform purification
$end

$molecule
0 1
--
0 1
O -1.126149 -1.748387 -0.423240
H -0.234788 -1.493897 -0.661862
H -1.062789 -2.681331 -0.218819
--
0 1
O -0.254210 1.611495 -1.293845
H -1.001520 1.163510 -1.690129
H -0.153399 2.411746 -1.809248
--
0 1
O 1.694541 -0.226287 1.705739
H 0.785920 0.073487 1.677909
H 2.047134 0.150917 2.511706
--
0 1
O -0.864533 0.522472 1.218817
H -0.694120 1.093542 0.469789
H -1.131418 -0.310426 0.829702
--
0 1
O -2.312727 0.054310 -2.138177
H -2.002512 -0.686744 -1.617757
H -3.266931 0.004778 -2.081349
--
0 1
O 1.292355 -0.704768 -0.977342
H 1.642300 -0.601085 -0.092386
H 0.948122 0.159656 -1.202018
$end
```

12.14 *Ab Initio* Frenkel Davydov Exciton Model (AIFDEM)

12.14.1 Theory

The Frenkel-Davydov Exciton model is an old idea for describing the (potentially delocalized) excited states of molecular crystals, or more generally noncovalent assemblies or aggregates of weakly electronically-coupled chromophores, as linear combinations of fragment-localized excited states. The Ith collective excited state, $|\Xi_I\rangle$, is thus written

$$|\Xi_I\rangle = \sum_{n}^{\text{sites}} \sum_{i}^{\text{states}} K_n^{Ii} |\Psi_n^i\rangle \prod_{m\neq n} |\Psi_m\rangle \;, \tag{12.79}$$

where $|\Psi_n^i\rangle$ is the ith excited state of the nth molecular fragment and $|\Psi_m\rangle$ is the ground-state wave function of the mth fragment. Eigenstates and energies are found by constructing and diagonalizing the electronic Hamiltonian matrix in this direct product, "exciton-site" basis.

In the *ab initio* Frenkel-Davydov exciton model (AIFDEM) developed by Morrison and Herbert,[90–93] the ground-state wave functions in Eq. (12.79) are single Slater determinants (obtained from SCF calculations on isolated fragments), and the fragment excited-state wave functions are linear combinations of singly-excited determinants:

$$|\Psi_A^*\rangle = \sum_{ia} C^{ia} |\Phi_A^{ia}\rangle \;. \tag{12.80}$$

The AIFDEM approach computes elements of the exact Hamiltonian,

$$\langle \Psi_A^* \Psi_B \Psi_C \ldots | \hat{H} | \Psi_A \Psi_B^* \Psi_C \ldots \rangle = \sum_{ia\sigma} \sum_{kb\tau} C_\sigma^{ia} C_\tau^{kb} \langle \Phi_A^{ia} \Phi_B \Phi_C \ldots | \hat{H} | \Phi_A \Phi_B^{kb} \Phi_C \ldots \rangle \;. \tag{12.81}$$

In particular, no dipole-coupling approximation is made (as is often invoked in simple exciton models). Such an approximation may be valid for well-separated chromophores but likely less so for tightly-packed chromophores in a molecular crystal. Overlap matrices

$$\langle \Psi_A^* \Psi_B \Psi_C \ldots | \Psi_A \Psi_B^* \Psi_C \ldots \rangle = \sum_{ia\sigma} \sum_{kb\tau} C_\sigma^{ia} C_\tau^{kb} \langle \Phi_A^{ia} \Phi_B \Phi_C \ldots | \Phi_A \Phi_B^{kb} \Phi_C \ldots \rangle \tag{12.82}$$

are also required because molecular orbitals located on different fragments are not orthogonal to one another. In order to reduce the number of terms in Eqs. (12.81) and (12.82), the fragment excited states are transformed into the natural transition orbital (NTO) basis (see Section 7.14.3) and then the corresponding orbitals transformation (Section 10.14.2.2) is used to compute matrix elements between non-orthogonal Slater determinants. The size of the exciton-site basis is sufficiently small such that eigenvectors and energies of the exciton Hamiltonian can be printed and saved to scratch files. Transition dipole moments between the ground state and the first ten excited states of the exciton Hamiltonian are also computed.

The cost to compute each matrix element scales with the size of the supersystem (somewhere between quadratic and quartic with monomer size), since all fragments must be included in the direct products. To reduce this scaling, a physically-motivated charge embedding scheme was introduced[90] that only treats the excited fragments, and neighbors within a user-specified distance threshold, with full QM calculation, while the other ground state fragment interactions are approximated by atomic point charges. In general, inclusion of neighboring fragments in the QM part of the matrix element evaluation does not seem to significantly improve the accuracy and diminishes the cost savings of the charge-embedding procedure. Therefore, the minimal "0 Å" threshold, where only the excited fragments are described at a QM level, can be considered optimal. Charge embedding with the minimal threshold affords an algorithm that scales as $N_\text{F}^2 \times \mathcal{O}(n_\text{pair}^{2-4})$, where N_F is the number of fragments and n_pair is the size of a pair of fragments.

The exciton-site basis can be expanded to include higher-lying fragment excited states which affords the wave function increased variational flexibility, and can significantly improve the accuracy for polar systems and delocalized excited

states. The number of fragment excited states included in the basis is specified by the CIS_N_ROOTS keyword, which must be ≥ 1. A cost effective means of including polarization effects is to use the XPol method to compute fragment ground state, and the nature of the atomic point charges is therefore controlled by keywords specified in the *$xpol* input section, as described in Section 12.8. Fragment excited states are then computed using the XPol-polarized MOs. Two additional types of states can be included in the AIFDEM basis. These may be of the charge transfer-type, where the states are of the form $|\Phi_A^+\Phi_B^-\Phi_C\ldots\rangle$, where $\Phi_A^{\pm}$ are cationic or anionic determinants from unrestricted SCF calculations on the isolated fragments. "Multi-exciton" states, of the type $|^1(\Phi_A^{\mathrm{T}}\Phi_B^{\mathrm{T}})\Phi_C\ldots\rangle$, are also easily included in the basis. These states couple triplet wave functions on two different monomers to an overall singlet, and have been used to study the singlet fission process in tetracene and pentacene.[2,91]

The exciton-site basis states are spin-adapted to form proper $\hat{S}^2$ eigenstates.Their multiplicity determines that of the target excited state and this must be specified by setting CIS_SINGLETS or CIS_TRIPLETS to TRUE. The number of terms included in Eqs. (12.81) and (12.82) can be rationally truncated at some fraction of the norm of the fragment NTO amplitudes, in order to reduce cost at the expense of accuracy, although the approximation is controllable by means of the truncation threshold. Computation time scales approximately quadratically with the number of terms and a threshold of about 85% has been found to maintain acceptable accuracy for organic molecules with reasonable cost. The fragment orbitals and excited states may be computed with any SCF and single-excitation theory, including DFT and TDDFT, however the coupling matrix elements are always computed with a CIS-like Hamiltonian with no DFT exchange-correlation.

There are many chemical processes of interest where motion of nuclei induces electronic transitions, in a breakdown of the Born-Oppenheimer approximation. In order to investigate such processes it is useful to calculate some quantity that codifies the coupling of adiabatic electronic states dues to nuclear motion. In molecular electronic structure theory this quantity is the nonadiabatic coupling (or "derivative coupling") vector $\langle\Psi_I|(\partial/\partial x)|\Psi_J\rangle$, which describes how the nuclear position derivative $\partial/\partial x$ couples adiabatic (Born-Oppenheimer) electronic states Ψ_I and Ψ_J. In solid-state physics these ideas are typically not discussed in terms of the Born-Oppenheimer approximation but rather in terms of the so-called "Holstein" and "Peierls" exciton/phonon coupling constants (see below). Within the framework of the AIFDEM, each of these quantities can be computed from a common intermediate $\mathbf{H}^{[x]}$, which is the derivative of the AIFDEM Hamiltonian matrix with respect to some nuclear coordinate x.[91,92] Diagonal matrix elements $H_{AA}^{[x]} \equiv \partial H_{AA}/\partial x$ describe how the exciton-site energies are modulated by nuclear motion and off-diagonal matrix elements $H_{AB}^{[x]} \equiv \partial H_{AB}/\partial x$ describe how nuclear motion modifies the energy-transfer couplings.

Once $\mathbf{H}^{[x]}$ has been constructed, then the nonadiabatic coupling vector between eigenstates Ψ_I and Ψ_J of the exciton Hamiltonian is simply

$$\mathbf{h}^{IJ} = \mathbf{K}_I^{\dagger}\mathbf{H}^{[x]}\mathbf{K}_J \ . \tag{12.83}$$

The Holstein ($A = B$) and Peierls ($A \neq B$) coupling constants $g_{AB\theta}$, expressed in dimensionless normal mode coordinates, are[92]

$$g_{AB\theta} = (2\mu_\theta\omega_\theta)^{-1/2}\sum_x \widetilde{H}_{AB}^{[x]}L_{x\theta} \tag{12.84}$$

where the matrix $\mathbf{L}$ is the transformation between Cartesians coordinates and normal (phonon) modes. The columns of $\mathbf{L}$ contain the normalized Cartesian displacements of normal mode θ, whose frequency and effective mass are ω_θ and μ_θ, respectively. The tilde on, $\widetilde{H}_{AB}^{[x]}$ indicates that the matrix element derivatives have been orthogonalized (including the derivative of the orthogonalization transformation).[92]

12.14.2 Job Control Variables

A basic AIFDEM calculation is requested by setting AIFDEM = TRUE in the *$rem* section. Additional job control variables dictate which types of states are included in the exciton-site basis as well as the NTO threshold that is used to truncate the expansion in Eq. (12.80). These variables and some others are described below.

AIFDEM

Perform an AIFDEM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform an AIFDEM calculation.

TRUE Perform an AIFDEM calculation.

RECOMMENDATION:

False

AIFDEM_NTOTHRESH

Controls how many NTOs that are retained in the exciton-site basis states.

TYPE:

INTEGER

DEFAULT:

99

OPTIONS:

n Retain enough NTOs to recover $n\%$ of the norm of the original CIS or TDDFT vectors in Eq. (12.80).

RECOMMENDATION:

A threshold of 85% gives a good trade-off of computational time and accuracy for organic molecules.

AIFDEM_EMBED_RANGE

Specifies the size of the QM region for charge embedding

TYPE:

INTEGER

DEFAULT:

FULL_QM

OPTIONS:

FULL_QM No charge embedding.

0 Treat only excited fragments with QM.

n Range (in Å) from excited fragments within which to treat other fragments with QM.

RECOMMENDATION:

The minimal threshold of zero typically maintains accuracy while significantly reducing computational time.

AIFDEM_CTSTATES

Include charge-transfer-like cation/anion pair states in the AIFDEM basis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Include CT states.

FALSE Do not include CT states.

RECOMMENDATION:

Use if CT states are desired in the basis.

AIFDEM_SINGFIS

Include multi-exciton states in the AIFDEM basis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Include multi-exciton states.

FALSE Do not include multi-exciton states.

RECOMMENDATION:

Use if multi-exciton states are desired in the basis. This option requires the use of AIFDEM_SEGSTART and AIFDEM_SEGEND in the *$rem* section.

For calculations on large systems, it may be desirable to break up the calculation of the AIFDEM matrix elements into batches. The user can specify can specify the first and last matrix element for a calculation with the following variables.

AIFDEM_SEGSTART

Indicates the index of the first matrix element to be computed.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n First matrix element of the chunk to be computed.

RECOMMENDATION:

Needs to be used with AIFDEM_SEGEND

AIFDEM_SEGEND

Indicates the index of the last matrix element to be computed.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Last matrix element of thhe chunk to be computed.

RECOMMENDATION:

Needs to be used with AIFDEM_SEGSTART

When computing AIFDEM matrix elements in batches, one can avoid running fragment SCF calculations for each job, by first running a job that runs only the fragment SCF calculations. These are saved in the scratch directory. These may then be read into subsequent calculations. The following variables control this utility.

AIFDEM_FRGM_WRITE

Fragment SCF calculations only.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Only fragment SCF calculations are carried out, no computation of matrix elements.

FALSE Regular AIFDEM calculation as specified by other *$rem* variables.

RECOMMENDATION:

None

AIFDEM_FRGM_READ

Skips fragment SCF calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Skips fragment SCF calculations, only computation of matrix elements.

FALSE Regular AIFDEM calculation as specified by other *$rem* variables.

RECOMMENDATION:

Requires a prior calculation that computes fragment SCF data.

12.14.3 Examples

For charge-embedded AIFDEM calculations, set XPOL = TRUE in the *$rem* section and then select the type of embedding charges via the *$xpol* input section, as described in Section 12.8 and illustrated in the following example.

Example 12.49 Example showing singlet excited state calculation, on $(H_2O)_4$. XPol is used to generate monomer wave functions with ChElPG charges. Minimal QM charge embedding is used for the exciton model with three excited states per fragment.

```
$molecule
0 1
--H2O 0
   0 1
   O          1.74078        1.59716       -1.49814
   H          2.22908        2.18316       -2.08914
   H          0.88038        2.04726       -1.32684
--H2O 1
   0 1
   O          1.31998       -1.18934       -1.91734
   H          1.49988       -0.22974       -1.89044
   H          1.69058       -1.52594       -1.07704
--H2O 2
   0 1
   O         -0.68982        2.59476       -0.72224
   H         -1.14372        3.37086       -1.07364
   H         -1.35592        1.84986       -0.78334
--H2O 3
   0 1
   O         -1.27512       -1.77394       -1.69524
   H         -0.32252       -1.52884       -1.85604
   H         -1.53992       -2.30454       -2.45644
$end

$rem
   BASIS                aug-cc-pvdz
   EXCHANGE             HF
   CIS_N_ROOTS          3
   CIS_TRIPLETS         FALSE
   XPOL                 TRUE
   AIFDEM               TRUE
   AIFDEM_EMBED_RANGE   0
   AIFDEM_NTOTHRESH     90
   NTO_PAIRS            1
$end

$xpol
  embed   charges
  charges CHELPG
$end
```

The multi-exciton keyword AIFDEM_SINGFIS invokes the use of multi-exciton states, $|^1(\Phi_A^{\mathrm{T}}\Phi_B^{\mathrm{T}})\Phi_C \ldots\rangle$. The following example illustrates this, along with the use of AIFDEM_SEGSTART and AIFDEM_SEGEND to split up the calculation of

matrix elements into segments. This is required for multi-exciton calculations.

Example 12.50 AIFDEM example with multi-exciton states in the basis.

```
$rem
BASIS          aug-cc-pvdz
EXCHANGE       HF
CIS_N_ROOTS    3
CIS_SINGLETS   TRUE
CIS_TRIPLETS   TRUE
XPOL           TRUE
AIFDEM         TRUE
AIFDEM_EMBED_RANGE 0
AIFDEM_NTOTHRESH   90
NTO_PAIRS          1
AIFDEM_SINGFIS     TRUE
AIFDEM_CTSTATES    TRUE
SCF_PRINT_FRGM     true
AIFDEM_SEGSTART    1
AIFDEM_SEGEND      2
$end

$molecule
0 1
-- H2O 0
0 1
O 1.74078 1.59716 -1.49814
H 2.22908 2.18316 -2.08914
H 0.88038 2.04726 -1.32684
-- H2O 1
0 1
O 1.31998 -1.18934 -1.91734
H 1.49988 -0.22974 -1.89044
H 1.69058 -1.52594 -1.07704
$end

$xpol
embed   charges
charges chelpg
$end
```

To compute AIFDEM derivatives $\mathbf{H}^{[x]}$ and $\mathbf{S}^{[x]}$ of the Hamiltonian and overlap matrices, the user should request a standard AIFDEM job and in addition set CIS_STATE_DERIV = 1. Currently, the AIFDEM derivatives do not support charge embedding so the keyword AIFDEM_EMBED_RANGE must be omitted from these jobs, which precludes the use of XPol wavefunctions for the fragments. Furthermore, only one excited state per fragment is supported; therefore, CIS_N_ROOTS = 1 is required.

The derivatives of the AIFDEM Hamiltonian matrix and overlap matrix are printed in the output file in sets of the three Cartesian coordinates that belong to a single atom. For convenience, the orthogonalized AIFDEM Hamiltonian matrix elements are saved in the scratch directory, `$QCSCRATCH/aifdem_deriv`. These are organized such that the derivatives for each unique matrix element are stored in individual files in the order of the atomic Cartesian coordinates.

These files can facilitate external calculation of exciton/phonon coupling constants.

Example 12.51 A basic AIFDEM derivative calculation on a chain of helium atoms.

```
$molecule
0 1
--frgm 0
0 1
He    0.000   0.000   0.000
He    0.000   0.000   1.400
--frgm 1
0 1
He    0.000   0.000   2.800
He    0.000   0.000   4.200
$end

$rem
   BASIS             =  cc-pvdz
   EXCHANGE          =  hf
   AIFDEM            =  true
   CIS_N_ROOTS       =  1
   CIS_SINGLETS      =  true
   CIS_TRIPLETS      =  false
   CIS_STATE_DERIV   =  1
   NTO_PAIRS         =  1
   MEM_TOTAL         =  1000
   MEM_STATIC        =  1000
   MAX_CIS_CYCLES    =  200
   MAX_SCF_CYCLES    =  200
   THRESH            =  10
   AIFDEM_NTOTHRESH  =  100
$end
```

12.15 TDDFT for Molecular Interactions

12.15.1 Introduction

There exist a broad class of weakly interacting molecular complexes which give rise to interesting excited-state properties that are potentially very different from those of a single chromophore. The “TDDFT for molecular interactions" or TDDFT(MI) method is designed for efficient excited-state calculations in such cases in (potentially large) systems composed of weakly-interacting but electronically-coupled monomers.[32,72] Such systems include molecular aggregates, chromophores in explicit solvent, and even proteins, for which the traditional TDDFT method become prohibitively expensive. TDDFT(MI) starts from a ground-state SCF MI calculation, and the use of ALMOs is central to its efficiency. In addition, the excitations are confined within monomer units and the explicit charge-transfer excitations are ignored, significantly reduced the two-electron integrals cost. The method works by coupling together excitations computed individually on different molecular fragments, and the number of excited states per fragment can be increased (at very low cost) in order to increase the variational flexibility of this exciton-type basis. Thus, despite the localized nature of the basis states, TDDFT(MI) is capable of describing collective excitations that are delocalized over multiple monomer units, as for example in the case of organic semiconductors. In general, TDDFT(MI) reproduces full super-system TDDFT excitation energies to within $\sim$0.2 eV, but with an order or magnitude reduction in total CPU time.[72] Formally, the cost of the method scales as $\mathcal{O}(N_{\text{fragment}}^2 N_{\text{roots}}^2 N_{\text{sub-AO}}^x)$ where N_{fragment} is the number of monomers, N_{roots} is the number of excited states per monomer, and $N_{\text{sub-AO}}$ is the number of AOs on a dimer subsystem. The exponent x (with $2 \leq x \leq 4$) reflects the cost of forming the Fock-like matrices of a traditional TDDFT calculation.

An especially promising application of the TDDFT(MI) method is to study excitation energies of a single chromophore in solution using a large number of explicit, quantum-mechanical solvent molecules. In such cases, the excitations are localized on the single chromophore and we can introduce a local excitation approximation (LEA) to TDDFT(MI) in which all of the Coulomb and exchange couplings between the solvent molecules and the chromophore are neglected.[73] Following the ground-state SCF(MI) calculation, the cost of the TDDFT part of the calculation becomes essentially the same as the cost of a TDDFT calculation on the gas-phase chromophore. In addition, this approach avoids the appearance of, and mixing with, spurious charge-transfer-to-solvent states,[32,73] of the sort that are known to arise in TDDFT calculations with explicit solvent.[44,57] Three versions of LEA-TDDFT(MI), named LEA0, LEA-Q and LEAc, have been implemented in Q-CHEM.[73] In the LEA0 method, ALMOs from the ground state SCF(MI) calculation are used to perform the TDDFT calculation. In LEAc, a sub-block of the TDDFT(MI) working equation localized on chromophore is extracted to calculate the excitation energies. Finally, LEA-Q is almost the same as LEAc except for some transformations to eliminate the overlap matrices. These approaches have been applied to converge solvatochromatic shifts for several aqueous chromophores.[73]

12.15.2 Job Control

In addition to the normal TDDFT job controls variables described in Section 7.3.5, there are several others to request TDDFT(MI). Note that only single-point energies (not gradients) are available for this method.

TDDFT_MI

Perform an TDDFT(MI) calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform an TDDFT(MI) calculation

TRUE Perform an TDDFT(MI) calculation

RECOMMENDATION:

False

MI_ACTIVE_FRAGMENT

Sets the active fragment

TYPE:

INTEGER

DEFAULT:

NO DEFAULT

OPTIONS:

n Specify the fragment on which the TDDFT calculation is to be performed, for LEA-TDDFT(MI).

RECOMMENDATION:

None

MI_LEA

Controls the LEA-TDDFT(MI) methods

TYPE:

INTEGER

DEFAULT:

NO DEFAULT

OPTIONS:

0 The LEA0 method

1 The LEA-Q method

2 The LEAc method

RECOMMENDATION:

1

12.16 ALMO-CIS/TDA and Its Charge-Transfer Correction

12.16.1 Introduction

The ALMO-CIS[15] and ALMO-CIS+CT[21] methods are local variants of Configuration Interaction Singles (CIS) for excited states, which are formulated based on the locality of Absolutely Localized Molecular Orbitals (ALMOs). The ALMO-CIS method shares same spirit with the TDDFT(MI) method[72] (Section 12.15), but it was originally designed to calculate a large number of excited states in atomic/molecular clusters, *e.g.*, the entire $n = 2$ band in helium clusters that contain hundreds of atoms.[15,21]

In ALMO-CIS and ALMO-CIS+CT, one solves a truncated non-orthogonal CIS eigenvalue problem:

$$A_{ia,jb}t^{jb} = \omega S_{ia,jb}t^{jb} \tag{12.85}$$

The use of ALMOs allows associating each MO index (i, a, j, or b) to a fragment. In ALMO-CIS, only the CIS amplitudes corresponding to intrafragment transitions are retained, *i.e.*, $t^{jb} = 0$ if the occupied orbital j and the virtual orbital b reside on two different fragments. The Hamiltonian and overlap matrix are also truncated, with i (j) and a (b) belonging to the same fragment. This approximation excludes interfragment charge transfer (CT) excitations entirely, which sometimes turns out to be insufficiently accurate. In ALMO-CIS+CT, the CT effect is reintroduced by providing a distance-based cutoff (r_{cut}) so that transitions between neighboring fragments within a range of r_{cut} are allowed, *i.e.*, i (j) and a (b) that are on a pair of neighboring fragments are also included in Eq. (12.85). In both ALMO-CIS and ALMO-CIS+CT, the dimension of the eigenvalue problem scales linearly with the system size rather than having a quadratic scaling as in standard CIS. Because of the reduction of matrix size, it is computationally feasible to explicitly build the Hamiltonian and directly diagonalize it to obtain a full band of excited states for relatively large systems. The overall scaling of the diagonalization step in ALMO-CIS/ALMO-CIS+CT is cubic, in contrast to the sixth-order scaling of standard CIS for full-spectrum calculation. To accelerate the construction of the CIS Hamiltonian (the $\mathbf{A}$ matrix in Eq. (12.85)), the resolution-of-the-identity (RI) technique is employed to evaluate some of the 2-electron terms (see Ref. 15 for details).

Besides the full-spectrum calculations described above, use of the Davidson algorithm is also available for ALMO-CIS and ALMO-CIS+CT, which targets a few lowest excited states as in standard CIS/TDDFT calculations. This iterative method, unlike the original full-spectrum version, also supports the ALMO variant of linear-response TDDFT within the Tame-Dancoff approximation (TDA),[34] which is referred to as ALMO-TDA and shares the same working equation (Eq. 12.85).

12.16.2 Job Control

In addition to the standard CIS job controls variables described in Section 7.2.4, there are several additional *$rem* variables to specify for an ALMO-CIS/ALMO-CIS+CT calculation.

LOCAL_CIS

Invoke ALMO-CIS/TDA or ALMO-CIS/TDA+CT calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Regular CIS/TDDFT calculations

1 ALMO-CIS/TDA without RI

2 ALMO-CIS with RI

RECOMMENDATION:

Use 2 when running full-spectrum ALMO-CIS calculations (EIGSLV_METH = 0)

Use 1 when running the iterative version of ALMO-CIS/TDA (EIGSLV_METH = 1)

EIGSLV_METH

Control the method for solving the ALMO-CIS eigen-equation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Explicitly build the Hamiltonian then diagonalize (full-spectrum)

1 Use the Davidson method (currently only available for restricted cases)

RECOMMENDATION:

None; use 1 for ALMO-TDA calculations (0 unavailable)

NN_THRESH

The distance cutoff for neighboring fragments (between which CT excitation occurs).

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not include interfragment transitions (ALMO-CIS/TDA)

n Include interfragment excitations between pairs of fragments the distances between whom are smaller than $n\ a_0$ (ALMO-CIS/TDA+CT)

RECOMMENDATION:

None

Example 12.52 ALMO-CIS+CT calculation ($r_{\mathrm{cut}} = 10\ a_0$) for all the $n = 2$ states of a helium dimer.

```
$molecule
0 1
--
   0 1
   He      2.8    0.     0.
--
   0 1
   He      0.     0.     0.
$end

$rem
   BASIS          gen
   AUX_BASIS      rimp2-cc-pvdz
   PURECART       1111
   METHOD         hf
   FRGM_METHOD    stoll
   CIS_N_ROOTS    8
   CIS_TRIPLETS   false
   LOCAL_CIS      2 ! use RI for ALMO-CIS
   NN_THRESH      10
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

$rem_frgm
   cis_n_roots 0
$end

$basis
****
HE   0
S    3    1.000000
3.84216340D+01    2.37660000D-02
5.77803000D+00    1.54679000D-01
1.24177400D+00    4.69630000D-01
S    1    1.000000
2.97964000D-01    1.00000000D+00
SP   1    1.000000
4.80000000D-02    1.00000000D+00   1.00000000D+00
****
$end
```

Example 12.53 Attachment-detachment density plots for the first two ALMO-CIS states of the formamide-water complex

```
$molecule
0 1
--
0 1
  C         1.1508059365    0.2982718924    0.0240277739
  O         0.3545181649    1.2334803420   -0.0015882208
  N         0.8104369587   -1.0072797234    0.0043506838
  H         2.2327270535    0.4686363261    0.0666232655
  H        -0.1675092286   -1.2596328526   -0.0352400180
  H         1.5210524537   -1.7122494331    0.0139809901
--
0 1
  O        -1.9693273428   -0.2999882700   -0.2293071572
  H        -1.3827632725    0.4697313642   -0.1375254289
  H        -2.7470364523   -0.0962178118    0.2907490329
$end

$rem
basis           6-31+g(d)
method          hf
scf_convergence 8
cis_n_roots     2
cis_triplets   false
thresh         12
frgm_method     stoll
local_cis       1 ! no RI when doing iterative ALMO-CIS
eigslv_meth     1 ! use iterative solver for ALMO-CIS
nn_thresh       0 ! ALMO-CIS without CT correction
make_cube_files true
plots           true
point_group_symmetry false
integral_symmetry    false
$end

$plots
grid_points 50 50 50
attachment_detachment_density 1-2
$end
```

12.16.3 ALMO-CIS/TDA with Selected Fragment Occupied-Virtual Pairs

Q-CHEM 6.0 and later versions support ALMO-CIS/TDA calculations with selected fragment occupied-virtual pairs, *i.e.*, only excitation amplitudes that correspond to transitions between selected occupied and virtual orbitals are considered in Eq. (12.85). To run this type of calculations one needs to set ALMOCIS_FRAGOV > 0, and currently three different modes are supported:

ALMOCIS_FRAGOV

Doing ALMO-CIS/TDA calculations with transitions from occupied orbitals on the 1st fragment and virtuals in the full system

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Doing standard ALMO-CIS/TDA calculations (if LOCAL_CIS > 0)

1 Reading user-specified active fragment O-V pairs from the *$frag_ov_pairs section*

2 Excitations on the *first* fragment only

3 Excitations from the occupied orbitals on the *first* fragment to all virtuals in the system

RECOMMENDATION:

None

The format of the *$frag_ov_pairs* section:

```
$frag_ov_pairs
   [number of frag_ov_pairs]
   [occ_frg_idx1]  [vir_frg_idx1]
   [occ_frg_idx2]  [vir_frg_idx2]
   ...
$end
```

These modified ALMO-CIS/TDA models can be used to model excited states in complex environments, such as the local excitation of a chromophore in solution or its charge-transfer-to-solvent (CTTS) excitations. Note that the iterative Davidson algorithm is required for these calculations, *i.e.*, EIGSLV_METH = 1.

Example 12.54 ALMO-TDA calculation for foramide water with user-specified occupied-virtual pairs: O(1) -> V(1) and O(1) -> V(2)

```
$molecule
0 1
--
0 1
  C          1.1508059365    0.2982718924    0.0240277739
  O          0.3545181649    1.2334803420   -0.0015882208
  N          0.8104369587   -1.0072797234    0.0043506838
  H          2.2327270535    0.4686363261    0.0666232655
  H         -0.1675092286   -1.2596328526   -0.0352400180
  H          1.5210524537   -1.7122494331    0.0139809901
--
0 1
  O         -1.9693273428   -0.2999882700   -0.2293071572
  H         -1.3827632725    0.4697313642   -0.1375254289
  H         -2.7470364523   -0.0962178118    0.2907490329
$end

$rem
jobtype     sp
basis       6-31G*
method      pbe0
frgm_method stoll
cis_n_roots     4
thresh         12
local_cis       1
almocis_fragov  1
eigslv_meth     1  ! iterative method
point_group_symmetry false
integral_symmetry    false
$end

$frag_ov_pairs
2
1 1
1 2
$end
```

References and Further Reading

[1] The VMD program may be downloaded from http://www.ks.uiuc.edu/Research/vmd.

[2] B. Alam, A. F. Morrison, and J. M. Herbert. *J. Phys. Chem. C*, 124:24653, 2020. DOI: 10.1021/acs.jpcc.0c07932.

[3] A. Aldossary, M. Gimferrer, Y. Mao, H. Hao, A. K. Das, P. Salvador, T. Head-Gordon, and M. Head-Gordon. *J. Phys. Chem. A*, 127:1760, 2023. DOI: 10.1021/acs.jpca.2c08061.

[4] A. Aldossary, H. Shen, Z. Wang, and M. Head-Gordon. *Chem. Phys. Lett.*, page 141825, 2024. DOI: 10.1016/j.cplett.2024.141825.

[5] A. Ambrosetti, A. M. Reilly, R. A. DiStasio, Jr., and A. Tkatchenko. *J. Chem. Phys.*, 140:18A508, 2014. DOI: 10.1063/1.4865104.

[6] R. J. Azar, P. R. Horn, E. J. Sundstrom, and M. Head-Gordon. *J. Chem. Phys.*, 138:084102, 2013. DOI: 10.1063/1.4792434.

[7] F. Ballesteros, J. A. Tan, and K. U. Lao. *J. Chem. Phys.*, 159:074107, 2023. DOI: 10.1063/5.0160810.

[8] G. M. J. Barca, A. T. B. Gilbert, and P. M. W. Gill. *J. Chem. Theory Comput.*, 14:1501, 2018. DOI: 10.1021/acs.jctc.7b00994.

[9] A. D. Becke. *J. Chem. Phys.*, 88:2547, 1988. DOI: 10.1063/1.454033.

[10] C. M. Breneman and K. B. Wiberg. *J. Comput. Chem.*, 11:361, 1990. DOI: 10.1002/jcc.540110311.

[11] D. R. Broderick and J. M. Herbert. *Chem. Sci.*, 15:19893, 2024. DOI: 10.1039/D4SC05955G.

[12] D. R. Broderick and J. M. Herbert. *J. Phys. Chem. Lett.*, 16:2793, 2025. DOI: 10.1021/acs.jpclett.4c03619.

[13] K. Carter-Fenk, K. U. Lao, K.-Y. Liu, and J. M. Herbert. *J. Phys. Chem. Lett.*, 10:2706, 2019. DOI: 10.1021/acs.jpclett.9b01156.

[14] K. Carter-Fenk, K. U. Lao, and J. M. Herbert. *Acc. Chem. Res.*, 54:3679, 2021. DOI: 10.1021/acs.accounts.1c00387.

[15] K. D. Closser, Q. Ge, Y. Mao, Y. Shao, and M. Head-Gordon. *J. Chem. Theory Comput.*, 11:5791, 2015. DOI: 10.1021/acs.jctc.5b00703.

[16] E. A. Cobar, R. Z. Khaliullin, R. G. Bergman, and M. Head-Gordon. *Proc. Natl. Acad. Sci. USA*, 104:6963, 2007. DOI: 10.1073/pnas.0610295104.

[17] E. E. Dahlke and D. G. Truhlar. *J. Chem. Theory Comput.*, 3:46, 2007. DOI: 10.1021/ct600253j.

[18] D. G. Fedorov and K. Kitaura. Theoretical background of the fragment molecular orbital (FMO) method and its implementation in GAMESS. In D. G. Fedorov and K. Kitaura, editors, *The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems*, chapter 2, pages 5–36. CRC Press, Boca Rotan, 2009. DOI: 10.1201/9781420078497.

[19] J. Gao. *J. Chem. Phys.*, 109:2346, 1998. DOI: 10.1063/1.476802.

[20] Q. Ge and M. Head-Gordon. *J. Chem. Theory Comput.*, 14:5156, 2018. DOI: 10.1021/acs.jctc.8b00537.

[21] Q. Ge, Y. Mao, A. F. White, E. Epifanovsky, K. D. Closser, and M. Head-Gordon. *J. Chem. Phys.*, 146:044111, 2017. DOI: 10.1063/1.4973611.

[22] Q. Ge, Y. Mao, and M. Head-Gordon. *J. Chem. Phys.*, 148:064105, 2018. DOI: 10.1063/1.5017510.

[23] D. Ghosh, D. Kosenkov, V. Vanovschi, C. F. Williams, J. M. Herbert, M. S. Gordon, M. W. Schmidt, L. V. Slipchenko, and A. I. Krylov. *J. Phys. Chem. A*, 114:12739, 2010. DOI: 10.1021/jp107557p.

[24] E. Gianinetti, M. Raimondi, and E. Tornaghi. *Int. J. Quantum Chem.*, 60:157, 1996. DOI: 10.1002/(SICI)1097-461X(1996)60:1<157::AID-QUA17>3.0.CO;2-C.

[25] A. T. B. Gilbert, N. A. Besley, and P. M. W. Gill. *J. Phys. Chem. A*, 112:13164, 2008. DOI: 10.1021/jp801738f.

[26] M. Gray and J. M. Herbert. Water vs. the cation–π interaction: The effect of solvation on intermolecular energy decomposition analysis. (in preparation).

[27] M. Gray and J. M. Herbert. *J. Chem. Phys.*, 155:034103, 2021. DOI: 10.1063/5.0059364.

[28] M. Gray and J. M. Herbert. *Annu. Rep. Comput. Chem.*, 20:1, 2024. DOI: 10.1016/bs.arcc.2024.03.001.

[29] J. M. Herbert. *J. Chem. Phys.*, 151:170901, 2019. DOI: 10.1063/1.5126216.

[30] J. M. Herbert. *J. Phys. Chem. A*, 125:7125, 2021. DOI: 10.1021/acs.jpca.1c05962.

[31] J. M. Herbert, L. D. Jacobson, K. U. Lao, and M. A. Rohrdanz. *Phys. Chem. Chem. Phys.*, 14:7679, 2012. DOI: 10.1039/c2cp24060b.

[32] J. M. Herbert, X. Zhang, A. F. Morrison, and J. Liu. *Acc. Chem. Res.*, 49:931, 2016. DOI: 10.1021/acs.accounts.6b00047.

[33] J. M. Herbert, M. Gray, K.-Y. Liu, and K. Carter-Fenk. Extended symmetry-adapted perturbation theory (XSAPT): A cubic-scaling platform for computing accurate intermolecular interaction energies and *ab initio* energy decomposition analysis. *ChemRxiv*, 2025. DOI: 10.26434/chemrxiv-2025-vctg5.

[34] S. Hirata and M. Head-Gordon. *Chem. Phys. Lett.*, 314:291, 1999. DOI: 10.1016/S0009-2614(99)01149-5.

[35] E. G. Hohenstein and C. D. Sherrill. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2:304, 2012. DOI: 10.1002/wcms.84.

[36] Z. C. Holden, R. M. Richard, and J. M. Herbert. *J. Chem. Phys.*, 139:244108, 2013. DOI: 10.1063/1.4850655.

[37] M. Hopffgarten and G. Frenking. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2:43, 2012. DOI: 10.1002/wcms.71.

[38] P. R. Horn and M. Head-Gordon. *J. Chem. Phys.*, 143:114111, 2015. DOI: 10.1063/1.4930534.

[39] P. R. Horn and M. Head-Gordon. *J. Chem. Phys.*, 144:084118, 2016. DOI: 10.1063/1.4941849.

[40] P. R. Horn, E. J. Sundstrom, T. A. Baker, and M. Head-Gordon. *J. Chem. Phys.*, 138:134119, 2013. DOI: 10.1063/1.4798224.

[41] P. R. Horn, Y. Mao, and M. Head-Gordon. *J. Chem. Phys.*, 144:114107, 2016. DOI: 10.1063/1.4942921.

[42] P. R. Horn, Y. Mao, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 18:23067, 2016. DOI: 10.1039/C6CP03784D.

[43] W. Humphrey, A. Dalke, and K. Schulten. *J. Molec. Graphics*, 14:33, 1996. DOI: 10.1016/0263-7855(96)00018-5.

[44] C. M. Isborn, B. D. Mar, B. F. E. Curchod, I. Tavernelli, and T. J. Martínez. *J. Phys. Chem. B*, 117:12189, 2013. DOI: 10.1021/jp4058274.

[45] L. D. Jacobson and J. M. Herbert. *J. Chem. Phys.*, 134:094118, 2011. DOI: 10.1063/1.3560026.

[46] L. D. Jacobson, R. M. Richard, K. U. Lao, and J. M. Herbert. *Annu. Rep. Comput. Chem.*, 9:25, 2013. DOI: 10.1016/B978-0-444-62672-1.00002-9.

[47] B. Jeziorski, R. Moszynski, A. Ratkiewicz, S. Rybak, K. Szalewicz, and H. L. Williams. SAPT: A program for many-body symmetry-adapted perturbation theory calculations of intermolecular interaction energies. In E. Clementi, editor, *Methods and Techniques in Computational Chemistry: METECC-94*, volume B, chapter 3, page 79. STEF, Cagliari, 1993.

[48] B. Jeziorski, R. Moszynski, and K. Szalewicz. *Chem. Rev.*, 94:1887, 1994. DOI: 10.1021/cr00031a008.

[49] P. Jurecka, J. Sponer, J. Cerny, and P. Hobza. *Phys. Chem. Chem. Phys.*, 8:1985, 2006. DOI: 10.1039/B600027D.

[50] B. Kaduk, T. Kowalczyk, and T. Van Voorhis. *Chem. Rev.*, 112:321, 2012. DOI: 10.1021/cr200148b.

[51] M. Kamiya, S. Hirata, and M. Valiev. *J. Chem. Phys.*, 128:074103, 2008. DOI: 10.1063/1.2828517.

[52] R. Z. Khaliullin, M. Head-Gordon, and A. T. Bell. *J. Chem. Phys.*, 124:204105, 2006. DOI: 10.1063/1.2191500.

[53] R. Z. Khaliullin, E. A. Cobar, R. C. Lochan, A. T. Bell, and M. Head-Gordon. *J. Phys. Chem. A*, 111:8753, 2007. DOI: 10.1021/jp073685z.

[54] R. Z. Khaliullin, A. T. Bell, and M. Head-Gordon. *J. Chem. Phys.*, 128:184112, 2008. DOI: 10.1063/1.2912041.

[55] R. Z. Khaliullin, A. T. Bell, and M. Head-Gordon. *Chem. Eur. J*, 15:851, 2009. DOI: 10.1002/chem.200802107.

[56] K. Kitaura, E. Ikeo, T. Asada, T. Nakano, and M. Uebayasi. *Chem. Phys. Lett.*, 313:701, 1999. DOI: 10.1016/S0009-2614(99)00874-X.

[57] A. Lange and J. M. Herbert. *J. Chem. Theory Comput.*, 3:1680, 2007. DOI: 10.1021/ct700125v.

[58] K. U. Lao and J. M. Herbert. *J. Phys. Chem. A*, 116:3042, 2012. DOI: 10.1021/jp300109y.

[59] K. U. Lao and J. M. Herbert. *J. Phys. Chem. Lett.*, 3:3241, 2012. DOI: 10.1021/jz301015p.

[60] K. U. Lao and J. M. Herbert. *J. Chem. Phys.*, 139:034107, 2013. DOI: 10.1063/1.4813523.

[61] K. U. Lao and J. M. Herbert. *J. Chem. Phys.*, 140:044108, 2014. DOI: 10.1063/1.4862644.

[62] K. U. Lao and J. M. Herbert. *J. Phys. Chem. A*, 119:235, 2015. DOI: 10.1021/jp5098603.

[63] K. U. Lao and J. M. Herbert. *J. Chem. Theory Comput.*, 12:2569, 2016. DOI: 10.1021/acs.jctc.6b00155.

[64] K. U. Lao and J. M. Herbert. *J. Chem. Theory Comput.*, 14:2955, 2018. DOI: 10.1021/acs.jctc.8b00058.

[65] K. U. Lao and J. M. Herbert. *J. Chem. Theory Comput.*, 14:5128, 2018. DOI: 10.1021/acs.jctc.8b00527.

[66] K. U. Lao, K.-Y. Liu, R. M. Richard, and J. M. Herbert. *J. Chem. Phys.*, 144:164105, 2016. DOI: 10.1063/1.4947087.

[67] D. S. Levine and M. Head-Gordon. *J. Phys. Chem. Lett.*, 8:1967, 2017. DOI: 10.1021/acs.jpclett.7b00766.

[68] D. S. Levine and M. Head-Gordon. *Proc. Natl. Acad. Sci. USA*, 114:12649, 2017. DOI: 10.1073/pnas.1715763114.

[69] D. S. Levine, P. R. Horn, Y. Mao, and M. Head-Gordon. *J. Chem. Theory Comput.*, 12:4812, 2016. DOI: 10.1021/acs.jctc.6b00571.

[70] W. Z. Liang and M. Head-Gordon. *J. Phys. Chem. A*, 108:3206, 2004. DOI: 10.1021/jp0374713.

[71] W. Z. Liang and M. Head-Gordon. *J. Chem. Phys.*, 120:10379, 2004. DOI: 10.1063/1.1729870.

[72] J. Liu and J. M. Herbert. *J. Chem. Phys.*, 143:034106, 2015. DOI: 10.1063/1.4926837.

[73] J. Liu and J. M. Herbert. *J. Chem. Theory Comput.*, 12:157, 2016. DOI: 10.1021/acs.jctc.5b00828.

[74] K.-Y. Liu and J. M. Herbert. *J. Chem. Phys.*, 147:161729, 2017. DOI: 10.1063/1.4986110.

[75] K.-Y. Liu, K. Carter-Fenk, and J. M. Herbert. *J. Chem. Phys.*, 151:031102, 2019. DOI: 10.1063/1.5111869.

[76] R. C. Lochan, R. Z. Khaliullin, and M. Head-Gordon. *Inorg. Chem.*, 47:4032, 2008. DOI: 10.1021/ic701625g.

[77] M. Loipersberger, J. Lee, Y. Mao, A. K. Das, K. Ikeda, J. Thirman, T. Head-Gordon, and M. Head-Gordon. *J. Phys. Chem. A*, 123:9621, 2019. DOI: 10.1021/acs.jpca.9b08586.

[78] M. Loipersberger, Y. Mao, and M. Head-Gordon. *J. Chem. Theory Comput.*, 16:1073, 2020. DOI: 10.1021/acs.jctc.9b01168.

[79] Y. Mao, O. Demerdash, M. Head-Gordon, and T. Head-Gordon. *J. Chem. Theory Comput.*, 12:5422, 2016. DOI: 10.1021/acs.jctc.6b00764.

[80] Y. Mao, P. R. Horn, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 19:5944, 2017. DOI: 10.1039/C6CP08039A.

[81] Y. Mao, D. S. Levine, M. Loipersberger, P. R. Horn, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 22:12867, 2020. DOI: 10.1039/D0CP01933J.

[82] Y. Mao, A. Montoya-Castillo, and T. E. Markland. *J. Chem. Phys.*, 153:244111, 2020. DOI: 10.1063/5.0035593.

[83] Y. Mao, M. Loipersberger, P. R. Horn, A. K. Das, O. N. Demerdash, D. S. Levine, S. P. Veccham, T. Head-Gordon, and M. Head-Gordon. *Annu. Rev. Phys. Chem.*, 72:641, 2021. DOI: 10.1146/annurev-physchem-090419-115149.

[84] Y. Mao, M. Loipersberger, K. J. Kron, J. S. Derrick, C. J. Chang, S. M. Sharada, and M. Head-Gordon. *Chem. Sci.*, 12:1398, 2021. DOI: 10.1039/D0SC05327A.

[85] A. V. Marenich, S. V. Jerome, C. J. Cramer, and D. G. Truhlar. *J. Chem. Theory Comput.*, 8:527, 2012. DOI: 10.1021/ct200866d.

[86] N. J. Mayhall and K. Raghavachari. *J. Chem. Theory Comput.*, 8:2669, 2012. DOI: 10.1021/ct300366e.

[87] R. McWeeny. *Rev. Mod. Phys.*, 32:335, 1960. DOI: 10.1103/RevModPhys.32.335.

[88] A. J. Misquitta and K. Szalewicz. *Chem. Phys. Lett.*, 357:301, 2002. DOI: 10.1016/S0009-2614(02)00533-X.

[89] M. P. Mitoraj, A. Michalak, and T. Ziegler. *J. Chem. Theory Comput.*, 5:962, 2008. DOI: 10.1021/ct800503d.

[90] A. F. Morrison and J. M. Herbert. *J. Phys. Chem. Lett.*, 6:4390, 2015. DOI: 10.1021/acs.jpclett.5b02109.

[91] A. F. Morrison and J. M. Herbert. *J. Phys. Chem. Lett.*, 8:1442, 2017. DOI: 10.1021/acs.jpclett.7b00230.

[92] A. F. Morrison and J. M. Herbert. *J. Chem. Phys.*, 146:224110, 2017. DOI: 10.1063/1.4985607.

[93] A. F. Morrison, Z.-Q. You, and J. M. Herbert. *J. Chem. Theory Comput.*, 10:5366, 2014. DOI: 10.1021/ct500765m.

[94] T. Nagata, O. Takahashi, K. Saito, and S. Iwata. *J. Chem. Phys.*, 115:3553, 2001. DOI: 10.1063/1.1388039.

[95] L. F. Pacios. *J. Comput. Chem.*, 16:133, 1995. DOI: 10.1002/jcc.540160202.

[96] J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley. *Int. J. Quantum Chem. Symp.*, 13:225, 1979. DOI: 10.1002/qua.560160825.

[97] R. M. Richard and J. M. Herbert. *J. Chem. Phys.*, 137:064113, 2012. DOI: 10.1063/1.4742816.

[98] R. M. Richard and J. M. Herbert. *J. Chem. Theory Comput.*, 9:1408, 2013. DOI: 10.1021/ct300985h.

[99] R. M. Richard, K. U. Lao, and J. M. Herbert. *J. Phys. Chem. Lett.*, 4:2674, 2013. DOI: 10.1021/jz401368u.

[100] R. M. Richard, K. U. Lao, and J. M. Herbert. *J. Chem. Phys.*, 139:224102, 2013. DOI: 10.1063/1.4836637.

[101] R. M. Richard, K. U. Lao, and J. M. Herbert. *J. Chem. Phys.*, 141:014108, 2014. DOI: 10.1063/1.4885846.

[102] R. M. Richard, K. U. Lao, and J. M. Herbert. *Acc. Chem. Res.*, 47:2828, 2014. DOI: 10.1021/ar500119q.

[103] E. Ronca, L. Belpassi, and F. Tarantelli. *ChemPhysChem*, 15:2682, 2014. DOI: 10.1002/cphc.201402321.

[104] H. Shen, Z. Wang, and M. Head-Gordon. *J. Chem. Theory Comput.*, 18:7428, 2022. DOI: 10.1021/acs.jctc.2c00901.

[105] H. Shen, S. P. Veccham, and M. Head-Gordon. *J. Chem. Theory Comput.*, 19:8624, 2023. DOI: 10.1021/acs.jctc.3c00872.

[106] J. C. Slater. *J. Chem. Phys.*, 41:3199, 1964. DOI: 10.1063/1.1725697.

[107] R. P. Steele, R. A. DiStasio, Jr., Y. Shao, J. Kong, and M. Head-Gordon. *J. Chem. Phys.*, 125:074108, 2006. DOI: 10.1063/1.2234371.

[108] H. Stoll, G. Wagenblast, and H. Preuss. *Theor. Chem. Acc.*, 57:169, 1980. DOI: 10.1007/BF00574903.

[109] K. Szalewicz. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 2:254, 2012. DOI: 10.1002/wcms.86.

[110] J. Thirman and M. Head-Gordon. *J. Chem. Phys.*, 143:084124, 2015. DOI: 10.1063/1.4929479.

[111] J. Thirman and M. Head-Gordon. *J. Phys. Chem. A*, 121:717, 2017. DOI: 10.1021/acs.jpca.6b11516.

[112] S. P. Veccham, J. Lee, Y. Mao, P. R. Horn, and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 23:928, 2021. DOI: 10.1039/d0cp05852a.

[113] J. Řezáč and A. de la Lande. *J. Chem. Theory Comput.*, 11:528, 2015. DOI: 10.1021/ct501115m.

[114] J. Řezáč, K. E. Riley, and P. Hobza. *J. Chem. Theory Comput.*, 7:2427, 2011. DOI: 10.1021/ct2002946.

[115] H. L. Williams and C. F. Chabalowski. *J. Phys. Chem. A*, page 646, 2001. DOI: 10.1021/jp003883p.

[116] H. L. Williams, E. M. Mas, and K. Szalewicz. *J. Chem. Phys.*, 103:7374, 1995. DOI: 10.1063/1.470309.

[117] W. Xie and J. Gao. *J. Chem. Theory Comput.*, 3:1890, 2007. DOI: 10.1021/ct700167b.

[118] W. Xie, L. Song, D. G. Truhlar, and J. Gao. *J. Chem. Phys.*, 128:234108, 2008. DOI: 10.1063/1.2936122.

[119] W. Xie, M. Orozco, D. G. Truhlar, and J. Gao. *J. Chem. Theory Comput.*, 5:459, 2009. DOI: 10.1021/ct800239q.

Chapter 13

Specialized Topics

13.1 Geminal Models

13.1.1 Introduction

Computational models that use single reference wave function describe molecules in terms of independent electrons interacting via mean Coulomb and exchange fields. It is natural to improve this description by using correlated electron pairs, or *geminals*, as building blocks for molecular wave functions. Requirements of computational efficiency and size consistency constrain geminals to have $S_z = 0$,[73] with each geminal spanning its own subspace of molecular orbitals.[3] Geminal wave functions were introduced into computational chemistry by Hurley, Lennard-Jones, and Pople.[42] An excellent review of the history and properties of geminal wave functions is given by Surjan.[81]

We implemented a size consistent model chemistry based on Singlet type Strongly orthogonal Geminals (SSG). In SSG, the number of molecular orbitals in each singlet electron pair is an adjustable parameter chosen to minimize total energy. Open-shell orbitals remain uncorrelated. The SSG wave function is computed by setting SSG *$rem* variable to 1. Both spin-restricted (RSSG) and spin-unrestricted (USSG) versions are available, chosen by the UNRESTRICTED *$rem* variable.

The wave function has the form

$$\begin{aligned} \Psi_{\mathrm{SSG}} &= \hat{A}\left[\psi_1(\mathbf{r}_1,\mathbf{r}_2)\ldots\psi_{n_\beta}(\mathbf{r}_{2n_\beta-1},\mathbf{r}_{2n_\beta})\phi_i(\mathbf{r}_{2n_\beta+1})\ldots\phi_j(\mathbf{r}_{n_\beta+n_\alpha})\right] \\ \psi_a(\mathbf{r}_1,\mathbf{r}_2) &= \sum_{k\in A}\frac{D_i^A}{\sqrt{2}}[\phi_k(\mathbf{r}_1)\bar{\phi}_k(\mathbf{r}_2)-\phi_k(\mathbf{r}_2)\bar{\phi}_k(\mathbf{r}_1)] \\ \phi_k(\mathbf{r}_1) &= \sum_\lambda C_\lambda^k\chi_\lambda(\mathbf{r}_1) \\ \bar{\phi}_k(\mathbf{r}_1) &= \sum_\lambda \bar{C}_\lambda^k\chi_\lambda(\mathbf{r}_1) \end{aligned} \tag{13.1}$$

with the coefficients C, D, and subspaces A chosen to minimize the energy

$$E_{\mathrm{SSG}} = \frac{\langle\Psi_{\mathrm{SSG}}|\hat{H}|\Psi_{\mathrm{SSG}}\rangle}{\langle\Psi_{\mathrm{SSG}}|\Psi_{\mathrm{SSG}}\rangle} \tag{13.2}$$

evaluated with the exact Hamiltonian $\hat{H}$. A constraint $\bar{C}_\lambda^k = C_\lambda^k$ for all MO coefficients yields a spin-restricted version of SSG.

SSG model can use any orbital-based initial guess. It is often advantageous to compute Hartree-Fock orbitals and then read them as initial guess for SSG. The program distinguishes Hartree-Fock and SSG initial guess wave functions,

and in former case makes preliminary assignment of individual orbital pairs into geminals. The verification of orbital assignments is performed every ten wave function optimization steps, and the orbital pair is reassigned if total energy is lowered.

The convergence algorithm consists of combination of three types of minimization steps. Direct minimization steps[78] seek a minimum along the gradient direction, rescaled by the quantity analogous to the orbital energy differences in SCF theory.[73] If the orbitals are nearly degenerate or inverted, a perturbative re-optimization of single geminal is performed. Finally, new set of the coefficients C and D is formed from a linear combination of previous iterations, in a manner similar to DIIS algorithm.[70,71] The size of iterative subspace is controlled by the DIIS_SUBSPACE_SIZE keyword.

After convergence is achieved, SSG reorders geminals based on geminal energy. The energy, along with geminal expansion coefficients, is printed for each geminal. Presence of any but the leading coefficient with large absolute value (value of 0.1 is often used for the definition of "large") indicates the importance of electron correlation in the system. The Mulliken population analysis is also performed for each geminal, which enables easy assignment of geminals into such chemical objects as core electron pairs, chemical bonds, and lone electron pairs.

As an example, consider the sample calculation of ScH molecule with 6-31G basis set at the experimental bond distance of 1.776 Å. In its singlet ground state the molecule has 11 geminals. Nine of them form core electrons on Sc. Two remaining geminals are:

Geminal 10 E = -1.342609
0.99128 -0.12578 -0.03563 -0.01149 -0.01133 -0.00398

Geminal 11 E = -0.757086
0.96142 -0.17446 -0.16872 -0.12414 -0.03187 -0.01227 -0.01204 -0.00435 -0.00416 -0.00098

Mulliken population analysis shows that geminal 10 is delocalized between Sc and H, indicating a bond. It is moderately correlated, with second expansion coefficient of a magnitude 0.126. The geminal of highest energy is localized on Sc. It represents $4s^2$ electrons and describes their excitation into $3d$ orbitals. Presence of three large expansion coefficients show that this effect cannot be described within GVB framework.[8]

13.1.2 Perturbative Corrections

The SSG description of molecular electronic structure can be improved by perturbative description of missing inter-geminal correlation effects. We have implemented Epstein-Nesbet form of perturbation theory[28,61] that permits a balanced description of one- and two-electron contributions to excited states' energies in SSG model. This form of perturbation theory is especially accurate for calculation of weak intermolecular forces. Also, two-electron $[i\bar{j}, j\bar{i}]$ integrals are included in the reference Hamiltonian in addition to intra-geminal $[i\bar{j}, i\bar{j}]$ integrals that are needed for reference wave function to be an eigenfunction of the reference Hamiltonian.[74]

All perturbative contributions to the SSG(EN2) energy (second-order Epstein-Nesbet perturbation theory of SSG wave function) are analyzed in terms of largest numerators, smallest denominators, and total energy contributions by the type of excitation. All excited states are subdivided into dispersion-like with correlated excitation within one geminal coupled to the excitation within another geminal, single, and double electron charge transfer. This analysis permits careful assessment of the quality of SSG reference wave function. Formally, the SSG(EN2) correction can be applied both to RSSG and USSG wave functions. Experience shows that molecules with broken or nearly broken bonds may have divergent RSSG(EN2) corrections. USSG(EN2) theory is balanced, with largest perturbative corrections to the wave function rarely exceeding 0.1 in magnitude.

SSG

Controls the calculation of the SSG wave function.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute the SSG wave function

1 Do compute the SSG wave function

RECOMMENDATION:

See also the UNRESTRICTED and DIIS_SUBSPACE_SIZE *$rem* variables.

13.2 Intracules

13.2.1 Introduction

The many dimensions of electronic wave functions makes them difficult to analyze and interpret. It is often convenient to reduce this large number of dimensions, yielding simpler functions that can more readily provide chemical insight. The most familiar of these is the one-electron density $\rho(\mathbf{r})$, which gives the probability of an electron being found at the point $\mathbf{r}$. Analogously, the one-electron momentum density $\pi(\mathbf{p})$ gives the probability that an electron will have a momentum of $\mathbf{p}$. However, the wave function is reduced to the one-electron density much information is lost. In particular, it is often desirable to retain explicit two-electron information. Intracules are two-electron distribution functions and provide information about the *relative* position and momentum of electrons. A detailed account of the different type of intracules can be found in Ref. 32. Q-CHEM's intracule package was developed by Aaron Lee and Nick Besley, and can compute the following intracules for or HF wave functions:

- Position intracules, $P(u)$: describes the probability of finding two electrons separated by a distance u.
- Momentum intracules, $M(v)$: describes the probability of finding two electrons with relative momentum v.
- Wigner intracule, $W(u,v)$: describes the combined probability of finding two electrons separated by u *and* with relative momentum v.

13.2.2 Position Intracules

The intracule density, $I(\mathbf{u})$, represents the probability for the inter-electronic vector $\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_2$:

$$I(\mathbf{u}) = \int \rho(\mathbf{r}_1\mathbf{r}_2)\ \delta(\mathbf{r}_{12} - \mathbf{u})\ d\mathbf{r}_1\, \mathrm{d}\mathbf{r}_2 \tag{13.3}$$

where $\rho(\mathbf{r}_1, \mathbf{r}_2)$ is the two-electron density. A simpler quantity is the spherically averaged intracule density,

$$P(u) = \int I(\mathbf{u}) \mathrm{d}\Omega_{\mathbf{u}}\ , \tag{13.4}$$

where $\Omega_{\mathbf{u}}$ is the angular part of $\mathbf{v}$, measures the probability that two electrons are separated by a scalar distance $u = |\mathbf{u}|$. This intracule is called a position intracule.[32] If the molecular orbitals are expanded within a basis set

$$\psi_a(\mathbf{r}) = \sum_{\mu} c_{\mu a}\phi_{\mu}(\mathbf{r}) \tag{13.5}$$

The quantity $P(u)$ can be expressed as

$$P(u) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma} (\mu\nu\lambda\sigma)_{\rm P} \tag{13.6}$$

where $\Gamma_{\mu\nu\lambda\sigma}$ is the two-particle density matrix and $(\mu\nu\lambda\sigma)_{\rm P}$ is the position integral

$$(\mu\nu\lambda\sigma)_{\rm P} = \int \phi^*_\mu(\mathbf{r})\ \phi_\nu(\mathbf{r})\ \phi^*_\lambda(\mathbf{r}+\mathbf{u})\phi_\sigma(\mathbf{r}+\mathbf{u})\ d\mathbf{r}\ d\Omega \tag{13.7}$$

and $\phi_\mu(\mathbf{r})$, $\phi_\nu(\mathbf{r})$, $\phi_\lambda(\mathbf{r})$ and $\phi_\sigma(\mathbf{r})$ are basis functions. For HF wave functions, the position intracule can be decomposed into a Coulomb component,

$$P_{\rm J}(u) = \frac{1}{2} \sum_{\mu\nu\lambda\sigma} D_{\mu\nu} D_{\lambda\sigma} (\mu\nu\lambda\sigma)_{\rm P} \tag{13.8}$$

and an exchange component,

$$P_{\rm K}(u) = -\frac{1}{2} \sum_{\mu\nu\lambda\sigma} \left[D^\alpha_{\mu\lambda} D^\alpha_{\nu\sigma} + D^\beta_{\mu\lambda} D^\beta_{\nu\sigma} \right] (\mu\nu\lambda\sigma)_{\rm P} \tag{13.9}$$

where $D_{\mu\nu}$ *etc.* are density matrix elements. The evaluation of $P(u)$, $P_{\rm J}(u)$ and $P_{\rm K}(u)$ within Q-CHEM has been described in detail in Ref. 52.

Some of the moments of $P(u)$ are physically significant,[31] for example

$$\int_0^\infty u^0 P(u) du = \frac{n(n-1)}{2} \tag{13.10}$$

$$\int_0^\infty u^0 P_{\rm J}(u) du = \frac{n^2}{2} \tag{13.11}$$

$$\int_0^\infty u^2 P_{\rm J}(u) du = nQ - \mu^2 \tag{13.12}$$

$$\int_0^\infty u^0 P_{\rm K}(u) du = -\frac{n}{2} \tag{13.13}$$

where n is the number of electrons and, μ is the electronic dipole moment and Q is the trace of the electronic quadrupole moment tensor. Q-CHEM can compute both moments and derivatives of position intracules.

13.2.3 Momentum Intracules

Analogous quantities can be defined in momentum space; $\bar{I}(\mathbf{v})$, for example, represents the probability density for the relative momentum $\mathbf{v} = \mathbf{p}_1 - \mathbf{p}_2$:

$$\bar{I}(\mathbf{v}) = \int \pi(\mathbf{p}_1, \mathbf{p}_2)\ \delta(\mathbf{p}_{12} - \mathbf{v}) d\mathbf{p}_1 d\mathbf{p}_2 \tag{13.14}$$

where $\pi(\mathbf{p}_1, \mathbf{p}_2)$ momentum two-electron density. Similarly, the spherically averaged intracule

$$M(v) = \int \bar{I}(\mathbf{v}) \mathrm{d}\Omega_{\mathbf{v}} \tag{13.15}$$

where $\Omega_{\mathbf{v}}$ is the angular part of $\mathbf{v}$, is a measure of relative momentum $v = |\mathbf{v}|$ and is called the momentum intracule. The quantity $M(v)$ can be written as

$$M(v) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma}\ (\mu\nu\lambda\sigma)_{\rm M} \tag{13.16}$$

where $\Gamma_{\mu\nu\lambda\sigma}$ is the two-particle density matrix and $(\mu\nu\lambda\sigma)_{\rm M}$ is the momentum integral[6]

$$(\mu\nu\lambda\sigma)_{\rm M} = \frac{v^2}{2\pi^2}\int \phi^*_\mu(\mathbf{r})\phi_\nu(\mathbf{r}+\mathbf{q})\phi^*_\lambda(\mathbf{u}+\mathbf{q})\phi_\sigma(\mathbf{u})j_0(qv)\;d\mathbf{r}\;d\mathbf{q}\;d\mathbf{u} \tag{13.17}$$

The momentum integrals only possess four-fold permutational symmetry, *i.e.*,

$$(\mu\nu\lambda\sigma)_{\rm M} = (\nu\mu\lambda\sigma)_{\rm M} = (\sigma\lambda\nu\mu)_{\rm M} = (\lambda\sigma\mu\nu)_{\rm M} \tag{13.18}$$

$$(\nu\mu\lambda\sigma)_{\rm M} = (\mu\nu\sigma\lambda)_{\rm M} = (\lambda\sigma\nu\mu)_{\rm M} = (\sigma\lambda\mu\nu)_{\rm M} \tag{13.19}$$

and therefore generation of $M(v)$ is roughly twice as expensive as $P(u)$. Momentum intracules can also be decomposed into Coulomb $M_{\rm J}(v)$ and exchange $M_{\rm K}(v)$ components:

$$M_J(v) = \frac{1}{2}\sum_{\mu\nu\lambda\sigma} D_{\mu\nu}D_{\lambda\sigma}(\mu\nu\lambda\sigma)_{\rm M} \tag{13.20}$$

$$M_{\rm K}(v) = -\frac{1}{2}\sum_{\mu\nu\lambda\sigma}\left[D^\alpha_{\mu\lambda}D^\alpha_{\nu\sigma} + D^\beta_{\mu\lambda}D^\beta_{\nu\sigma}\right](\mu\nu\lambda\sigma)_{\rm M} \tag{13.21}$$

Again, the even-order moments are physically significant:[6]

$$\int_0^\infty v^0 M(v)dv = \frac{n(n-1)}{2} \tag{13.22}$$

$$\int_0^\infty u^0 M_{\rm J}(v)dv = \frac{n^2}{2} \tag{13.23}$$

$$\int_0^\infty v^2 P_{\rm J}(v)dv = 2nE_{\rm T} \tag{13.24}$$

$$\int_0^\infty v^0 M_{\rm K}(v)dv = -\frac{n}{2} \tag{13.25}$$

where n is the number of electrons and $E_{\rm T}$ is the total electronic kinetic energy. Currently, Q-CHEM can compute $M(v)$, $M_{\rm J}(v)$ and $M_{\rm K}(v)$ using s and p basis functions only. Moments are generated using quadrature and consequently for accurate results $M(v)$ must be computed over a large and closely spaced v range.

13.2.4 Wigner Intracules

The intracules $P(u)$ and $M(v)$ provide a representation of an electron distribution in either position *or* momentum space but neither alone can provide a complete description. For a combined position *and* momentum description an intracule in phase space is required. Defining such an intracule is more difficult since there is no phase space second-order reduced density. However, the second-order Wigner distribution,[7]

$$W_2(\mathbf{r}_1,\mathbf{p}_1,\mathbf{r}_2,\mathbf{p}_2) = \frac{1}{\pi^6}\int \rho_2(\mathbf{r}_1+\mathbf{q}_1,\mathbf{r}_1-\mathbf{q}_1,\mathbf{r}_2+\mathbf{q}_2,\mathbf{r}_2-\mathbf{q}_2)e^{-2i(\mathbf{p}_1\cdot\mathbf{q}_1+\mathbf{p}_2\cdot\mathbf{q}_2)}d\mathbf{q}_1 d\mathbf{q}_2 \tag{13.26}$$

can be interpreted as the probability of finding an electron at $\mathbf{r}_1$ with momentum $\mathbf{p}_1$ and another electron at $\mathbf{r}_2$ with momentum $\mathbf{p}_2$. [The quantity $W_2(\mathbf{r}_1,\mathbf{r}_2,\mathbf{p}_1,\mathbf{p}_2$ is often referred to as "quasi-probability distribution" since it is not positive everywhere.]

The Wigner distribution can be used in an analogous way to the second order reduced densities to define a combined position and momentum intracule. This intracule is called a Wigner intracule, and is formally defined as

$$W(u,v) = \int W_2(\mathbf{r}_1,\mathbf{p}_1,\mathbf{r}_2,\mathbf{p}_2)\delta(\mathbf{r}_{12}-\mathbf{u})\delta(\mathbf{p}_{12}-\mathbf{v})d\mathbf{r}_1\, d\mathbf{r}_2\, d\mathbf{p}_1\, d\mathbf{p}_2\, d\Omega_{\mathbf{u}}\, d\Omega_{\mathbf{v}} \tag{13.27}$$

If the orbitals are expanded in a basis set, then $W(u,v)$ can be written as

$$W(u,v) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma}\,(\mu\nu\lambda\sigma)_{\mathrm{W}} \tag{13.28}$$

where $(\mu\nu\lambda\sigma)_{\mathrm{W}}$ is the Wigner integral

$$(\mu\nu\lambda\sigma)_{\mathrm{W}} = \frac{v^2}{2\pi^2}\int\int \phi^*_\mu(\mathbf{r})\phi_\nu(\mathbf{r}+\mathbf{q})\phi^*_\lambda(\mathbf{r}+\mathbf{q}+\mathbf{u})\phi_\sigma(\mathbf{r}+\mathbf{u})j_0(q\,v)\; d\mathbf{r}\; dq\; d\Omega_{\mathbf{u}} \tag{13.29}$$

Wigner integrals are similar to momentum integrals and only have four-fold permutational symmetry. Evaluating Wigner integrals is considerably more difficult that their position or momentum counterparts. The fundamental $[ssss]_{\mathrm{w}}$ integral,

$$\begin{aligned}[ssss]_{\mathrm{W}} \;=\; & \frac{u^2v^2}{2\pi^2}\int\int \exp\left[-\alpha|\mathbf{r}-\mathbf{A}|^2-\beta|\mathbf{r}+\mathbf{q}-\mathbf{B}|^2-\gamma|\mathbf{r}+\mathbf{q}+\mathbf{u}-\mathbf{C}|^2-\delta|\mathbf{r}+\mathbf{u}-\mathbf{D}|^2\right]\times \\ & j_0(qv)\; d\mathbf{r}\; d\mathbf{q}\; d\Omega_{\mathbf{u}}\end{aligned} \tag{13.30}$$

can be expressed as

$$[ssss]_{\mathrm{W}} = \frac{\pi u^2v^2\; e^{-(R+\lambda^2u^2+\mu^2v^2)}}{2(\alpha+\delta)^{3/2}(\beta+\gamma)^{3/2}}\int e^{-\mathbf{P}\cdot\mathbf{u}} j_0\left(|\mathbf{Q}+\eta\mathbf{u}|v\right)\; d\Omega_u \tag{13.31}$$

or alternatively

$$[ssss]_{\mathrm{W}} = \frac{2\pi^2u^2v^2e^{-(R+\lambda^2u^2+\mu^2v^2)}}{(\alpha+\delta)^{3/2}(\beta+\gamma)^{3/2}}\sum_{n=0}^{\infty}(2n+1)i_n(P\,u)j_n(\eta uv)j_n(Qv)P_n\left(\frac{\mathbf{P}\cdot\mathbf{Q}}{P\,Q}\right) \tag{13.32}$$

Two approaches for evaluating $(\mu\nu\lambda\sigma)_{\mathrm{W}}$ have been implemented in Q-CHEM, full details can be found in Ref. 90. The first approach uses the first form of $[ssss]_{\mathrm{W}}$ and used Lebedev quadrature to perform the remaining integrations over $\Omega_{\mathbf{u}}$. For high accuracy large Lebedev grids[49–51] should be used, grids of up to 5294 points are available in Q-CHEM. Alternatively, the second form can be adopted and the integrals evaluated by summation of a series. Currently, both methods have been implemented within Q-CHEM for s and p basis functions only.

When computing intracules it is most efficient to locate the loop over u and/or v points within the loop over shell-quartets.[19] However, for $W(u,v)$ this requires a large amount of memory to store all the integrals arising from each (u,v) point. Consequently, an additional scheme, in which the u and v points loop is outside the shell-quartet loop, is available. This scheme is less efficient, but substantially reduces the memory requirements.

13.2.5 Intracule Job Control

The following *$rem* variables can be used to control the calculation of intracules.

INTRACULE

Controls whether intracule properties are calculated (see also the *$intracule* section).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE No intracule properties.

TRUE Evaluate intracule properties.

RECOMMENDATION:

None

WIG_MEM

Reduce memory required in the evaluation of $W(u, v)$.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use low memory option.

TRUE Use low memory option.

RECOMMENDATION:

The low memory option is slower, so use the default unless memory is limited.

WIG_LEB

Use Lebedev quadrature to evaluate Wigner integrals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Evaluate Wigner integrals through series summation.

TRUE Use quadrature for Wigner integrals.

RECOMMENDATION:

None

WIG_GRID

Specify angular Lebedev grid for Wigner intracule calculations.

TYPE:

INTEGER

DEFAULT:

194

OPTIONS:

Lebedev grids up to 5810 points.

RECOMMENDATION:

Larger grids if high accuracy required.

N_WIG_SERIES

Sets summation limit for Wigner integrals.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

$n < 100$

RECOMMENDATION:

Increase n for greater accuracy.

N_I_SERIES

Sets summation limit for series expansion evaluation of $i_n(x)$.

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

N_J_SERIES

Sets summation limit for series expansion evaluation of $j_n(x)$.

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

13.2.6 Format for the *$intracule* Section

int_type	0	Compute $P(u)$ only
	1	Compute $M(v)$ only
	2	Compute $W(u,v)$ only
	3	Compute $P(u)$, $M(v)$ and $W(u,v)$
	4	Compute $P(u)$ and $M(v)$
	5	Compute $P(u)$ and $W(u,v)$
	6	Compute $M(v)$ and $W(u,v)$
u_points		Number of points, start, end.
v_points		Number of points, start, end.
moments	0–4	Order of moments to be computed ($P(u)$ only).
derivs	0–4	order of derivatives to be computed ($P(u)$ only).
accuracy	n	(10^{-n}) specify accuracy of intracule interpolation table ($P(u)$ only).

Example 13.1 Compute HF/STO-3G $P(u)$, $M(v)$ and $W(u,v)$ for Ne, using Lebedev quadrature with 974 point grid.

```
$molecule
   0  1
   Ne
$end

$rem
   METHOD      hf
   BASIS       sto-3g
   INTRACULE   true
   WIG_LEB     true
   WIG_GRID    974
$end

$intracule
   int_type   3
   u_points  10   0.0  10.0
   v_points   8   0.0   8.0
   moments    4
   derivs     4
   accuracy   8
$end
```

Example 13.2 Compute HF/6-31G $W(u,v)$ intracules for H_2O using series summation up to n=25 and 30 terms in the series evaluations of $j_n(x)$ and $i_n(x)$.

```
$comment
  Note only a few points are calculated in this sample
$end

$molecule
   0  1
   H1
   O   H1  r
   H2  O   r  H1  theta

   r = 1.1
   theta = 106
$end

$rem
   METHOD          hf
   BASIS           6-31G
   INTRACULE       true
   WIG_MEM         true
   N_WIG_SERIES    25
   N_I_SERIES      40
   N_J_SERIES      50
$end

$intracule
   int_type    2
   u_points    2   0.0   15.0
   v_points    2   0.0   10.0
$end
```

13.3 CASE Approximation

The Coulomb Attenuated Schrödinger Equation (CASE) approximation[2] follows from the KWIK algorithm[27] in which the Coulomb operator is separated into two pieces using the error function, Eq. (5.12). Whereas in Section 5.6 this partition of the Coulomb operator was used to incorporate long-range Hartree-Fock exchange into DFT, within the CASE approximation it is used to attenuate all occurrences of the Coulomb operator in Eq. (4.2), by neglecting the long-range portion of the identity in Eq. (5.12). The parameter ω in Eq. (5.12) is used to tune the level of attenuation. Although the total energies from Coulomb attenuated calculations are significantly different from non-attenuated energies, it is found that relative energies, correlation energies and, in particular, wave functions, are not, provided a reasonable value of ω is chosen.

By virtue of the exponential decay of the attenuated operator, ERIs can be neglected on a proximity basis yielding a rigorous $\mathcal{O}(N)$ algorithm for single point energies. CASE may also be applied in geometry optimizations and frequency calculations.

OMEGA

Controls the degree of attenuation of the Coulomb operator.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

INTEGRAL_2E_OPR

Determines the two-electron operator.

TYPE:

INTEGER

DEFAULT:

-2 Coulomb Operator.

OPTIONS:

-1 Apply the CASE approximation.

-2 Coulomb Operator.

RECOMMENDATION:

Use the default unless the CASE operator is desired.

13.4 Molecular Junctions

The conductance and current-voltage relationships of molecular junctions can be calculated using either Landauer or Non-Equilibrium Green's Function (NEGF) levels of theory. In both cases, the Green's function formulation is employed using a chosen electronic structure level. See Refs. 24,25 for further introduction.

In molecular junctions the current-voltage curve depends on the electron transmission function, which can be calculated using the quantum transport code developed by the Dunietz group (Kent State). The scattering-free approach, (Landauer), provides a zero-bias limit, the non-equilibrium approach, (NEGF), provides a voltage-dependent transmission by solving iteratveily for the bias-affected electronic density.

This quantum transport utility is invoked by setting the *$rem* variable TRANS_ENABLE.

TRANS_ENABLE

To invoke the molecular transport code.

INPUT SECTION: *$REM*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform electron transport calculation.

1 Landauer; zero bias limit.

3 A self-consistent Green's function calculation at zero bias voltage (NEGF with zero bias, typically used for preparation of bias dependent NEGF).

4 Full NEGF.

-1 Print matrices needed for generating bulk model data files.

RECOMMENDATION:

Values 3 or 4 must be set with SCF_ALGORITHM = NEGF in the *$rem* section and should set POINT_GROUP_SYMMETRY = FALSE to fix the atomic coordinates. For NEGF calculations, the transport axis is assumed to be along X axis, along which the bias potential is dropped.

In addition to information included in the Q-CHEM output file the following files are generated:

- `transmission.txt` (Transmission function in the requested energy window)
- `TDOS.txt` (Total density of states)
- `current.txt` (I-V plot only for the Landauer level)
- `IV-NEGF-all.txt` (I-V plot obtained by NEGF method)

The transport flag can be used to print out data matrices of the junction or the bulk models that can be used in subsequent transport calculations:

- `FAmat.dat` (Hamiltonian matrix for follow up calculations and analysis)
- `Smat.dat` (Overlap matrix for follow up calculations and analysis)
- `DAmat.dat` (Density matrix for follow up calculations and analysis [only for NEGF])

(In case of printing bulk information the files need to be renamed as explained below.)

In the case of unrestricted spin the output file names are appended by A[B] to indicate the spin state. (*e.g.* `transmissionA.txt` and `transmissionB.txt`). We note that in the closed-shell spin-restricted case, the `transmission.txt` corresponds to the α spin, where the total transmission due to the spin symmetry is twice the values included in the file.

The Landauer level generates a single set of output files. NEGF calculation provides directories for the different biases where the directories `Vbias1`, `Vbias2`, $\cdots$ (the numbers in the directory names are used to index the bias voltage) contain the output files at the corresponding bias.

The T-Chem program allows for large flexibility in setting up the transport calculation and therefore requires setting several parameters. Accordingly, the actual molecular junction model is partitioned into domains representing the electrode self-energy (SEs) and the central bridge region, and the procedure by which the SEs are calculated.

This setup is addressed in two transport-specific sections in the input file:

- *$trans_model* (molecular junction regions)
- *$trans_method* (sets various transport flags include details for the SE evaluation, NEGF integration, grid parameters, and various input parameters as the Fermi energy and the bias.)

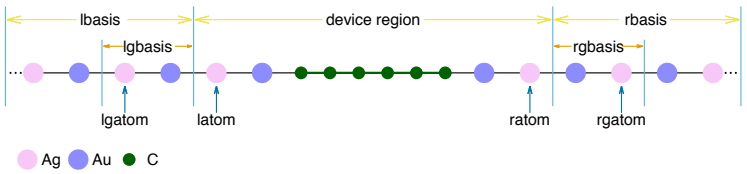


Figure 13.1: Illustration for the different regions of the molecular junction for Landauer calculation.

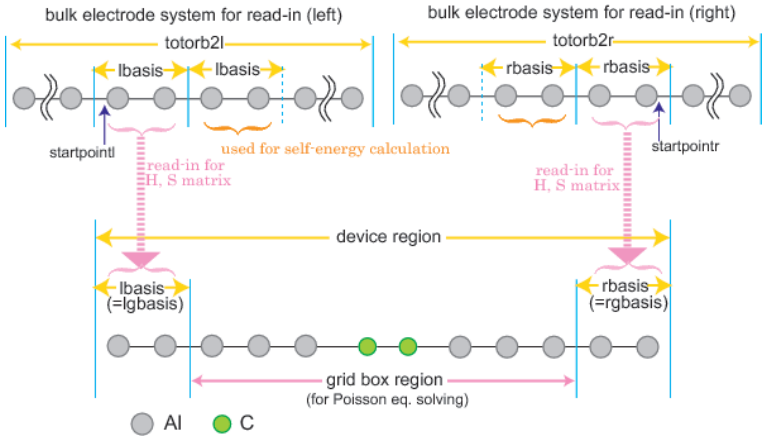


Figure 13.2: Illustration for the different regions of the molecular junction for NEGF calculation.

The *$trans_model* section partitions the molecular model, assigning the atomic basis functions to different regions. There are no default values. The different regions are set as illustated in Fig. 13.1 and Fig. 13.2 for the Landauer and NEGF levels, respectively.

The parameters in the *$trans_model* section can be provided either in terms of the atom numbers or basis functions. **Note:** Only one set of these numbers is required. The assignment based on **trans_latom** etc. has priority.

- **trans_latom**: INTEGER, atom index, where the central/bridge region starts (the first atom in junction area).
- **trans_ratom**: INTEGER, atom index, where the central/bridge region ends (the last atom in junction area).
- **trans_lgatom**: INTEGER, atom index where the repeat unit of the left electrode starts.
- **trans_rgatom**: INTEGER, atom index where the repeat unit of the right electrode ends.

Atoms within numbers **trans_lgatom**, **trans_lgatom**+1, .., **trans_latom**-1 define the repeat unit of the left electrode (similarly for right electrode).

- **trans_lbasis**: INTEGER, the number of basis functions appearing to the left of the device region (the index of the first AO within the central region).
- **trans_rbasis**: INTEGER, the number of basis functions appearing to the right of the device region right electrode (total number of basis functions minus this number equals last AO that is within the device).
- **trans_lgbasis**: INTEGER, the number of basis functions of the repeat unit of the left electrode.

- **trans_rgbasis**: INTEGER, the number of basis functions of the repeat unit of the right electrode.

For the NEGF calculation, **trans_lbasis** and **trans_lgbasis** must be the same number as shown in Fig. 13.2 (also for **trans_rbasis** and **trans_rgbasis**). The NEGF calculations require precalculated bulk data, while the Landauer can use such precalculated data as an option.

The example in Fig. 13.1 includes a total of 18 atoms, where the Au and Ag basis sets each contain 22 basis functions per atom. The repeat unit includes a pair of Au Ag atoms. Therefore, the parameters should be given as follows (only the first two columns are required, the rest are included for explanation):

trans_lbasis	88	No. of functions representing left electrode region (2×22 for Ag + 2×22 for Au)
trans_rbasis	88	No. of functions representing right electrode (2×22 for Ag + 2×22 for Au)
trans_lgbasis	44	Size of the repeating unit of the left electrode (22 for Ag + 22 for Au)
trans_rgbasis	44	Size of the repeating unit of the right electrode (22 for Ag + 22 for Au)

Alternatively this section can be based on the atom numbers corresponding to their position in the *$molecule* section:

trans_lgatom	3	Third atom is used to define the repeat unit of the left electrode
trans_latom	5	Fifth atom is the first atom of the junction
trans_ratom	14	Fourteenth atom is the last atom of the junction
trans_rgatom	16	Sixteenth atom is used to define the repeat unit of the right electrode

In this example, we have used the same repeat unit for the left and right electrodes; this symmetry is *not* required.

WARNING: In both illustrated examples atomic wires are used: Au/Ag wire in the Landauer case and Al wire for the NEGF example. Here The Au/Ag pair or a single Al atom represent a bulk repeat unit, in more detailed models it is essential that the atoms within the unit are introduced at the same order in all units. All bulk layers have to appear in order (”left to right”) including those units that are designated to be the first and last bulk unit included within the central region.

Note: The order of atoms in the *$molecule* section is important and assumes the following:
Repeating units (left) - Molecular Junction - Repeating units (right)
The atoms are provided first by the leftmost repeat unit in the left electrode then proceeds to the next repeat unit up to the surface unit. Next the bridge atoms are provided followed by the right surface unit and the right electrode region. The right electrode region starts with the surface layer and ends with the most distant layer within the bulk. The atoms order within each of the included electrode repeat units must be consistent. The atoms order within the bridge region (excluding the electrode repeat unit atoms) is arbitrary.

That is, the order of atoms in the molecule section has to adhere to the following:

1. atoms of the leftmost repeat unit
2. atoms of the next repeat unit(s) (if avaialble)
3. atoms of the left surface unit device
4. bridge atoms
5. atoms of the right surface unit
6. atoms of the next right electrode unit(s) (if avaialble)
7. atoms of the rightmost repeat unit

T-Chem allows for complete flexibility in determining the different regions of the electrode models. As a consequence, incorrect setting of the regions cannot be caught by the program and may result in transmission functions that are

unphysical (*e.g.* large or even negative values). Such errors can occur where the cluster model is partitioned (by mistake) within the orbital space of an atom. Regions should be defined between the repeat units. The atoms within each repeat unit must be always provided in the same internal order, and must be consistent with the order (and orientation!) used in the precalculation of the bulk model (if employed).

Note: At least a single repeat unit of the electrodes should be included in the bridge region. With the Landauer model, if **trans_readhs** == 0, then at least one unit beyond the bridge region *has* to be included, an additional unit (total of two at least) is required, when also **trans_method** != 0.

The necessary parameters in *$trans_method* section are listed as follows:

trans_spin

Spin coordinates.

INPUT SECTION: *$trans_method*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Restricted spin calculations or closed-shell singlet states.

3 Unrestricted spin calculations or open-shell systems (both spins are calculated).

RECOMMENDATION:

None

trans_method

Electrode surface GF model.

INPUT SECTION: *$trans_method*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 A wide band limit (WBL) with a constant parameter trans_greens (default).

1 WBL using the Ke-Baranger-Yang TB at the Fermi energy.

2 WBL using the Lopez-Sancho TB at the Fermi energy.

3 Tight-binding (TB) following the Ke-Baranger-Yang algorithm.

4 TB following the Lopez-Sancho algorithm (decimation).

RECOMMENDATION:

Note: Only option 0 is currently available for NEGF calculations.

trans_readhs

Flag to read the Hamiltonian and overlap matrices of the bulk model.

INPUT SECTION: *$trans_method*

TYPE:

INTEGER

DEFAULT:

0 Use the current Hamiltonian and overlap matrices to parse out the electrode matrices.

OPTIONS:

1 Use pre-calculated electrode Hamiltonian and overlap matrices. Required for NEGF.

RECOMMENDATION:

If set to 1, the following files are requred: `FAmat2l.dat` and `Smat2l.dat` (for left electrode model), `FAmat2r.dat` and `Smat2r.dat` (for right electrode model). If both electrodes are of the same type, may use symbolic links to the same files. (For unrestricted spin model, `FBmat2l.dat` and `FBmat2r.dat` are also necessary)

For **trans_readhs** = 1, the paramters for parsing the precalculated bulk matrices (`FAmat2l.dat`, etc) have to be provided. These indices define the regions within the bulk full model space to set the onsite and coupling matrcies for the SE calculations.

- **trans_totorb2** (Integer):
  Total number of basis functions in the electrode models (if set, then same size is assumed for both electrodes) (no default value).
- **trans_totorb2l**: Integer number, total number of basis functions in left electrode model (no default value).
- **trans_totorb2r**: Integer number, total number of basis functions in right electrode model (no default value).
- **trans_startpoint**: Integer number, start point (basis number) for reading the TB integrals. Note, the basis number, that is, index number of basis function starts from 0. (if set, then same size is assumed for both electrodes) (default value 0).
- **trans_startpointl**: given as integer number, left start point (basis number) for reading the TB integrals. Note, the basis number, that is, index number of basis function starts from 0. (default value 0).
- **trans_startpointr**: given as integer number, right start point (basis number) for reading the TB integrals. Note, the basis number, that is, index number of basis function starts from 0. (no default value).

Note: NEGF requires **trans_readhs** to be set where SEs are precalculated. The Landauer level can obtain bulk parameters from the current job of a molecular junction model.

Options to control the output:

trans_npoint (Integer):

Number of grid points within the energy window of the transmission spectra calculation.

300 Default value.

trans_printdos (Integer):

Controls the printout of TDOS.

0 Default, no total DOS printing.

1 A TDOS (of the junction region) will be printed to TDOS.txt (closed shell).

trans_printiv (Integer):
Calculated current (at the zero bias case).
0 Default, no current calculated and printed
1 Current will be printed to `current.txt` (closed shell) or `currentA/B.txt` (unrestricted or open shell).

trans_ipoints (Integer):
Number of points for current calculation (at the zero bias case).
300 Default value.

The following parameters control the imaginary smearing in calculating GFs:

trans_greens (Double):
Imaginary smearing/Broadening (in eV) added to the surface Green's function (only relevant for trans_method==0).
0.07 Default

trans_devsmear (Double):
Imaginary smearing (in eV) added to the real Hamiltonian in central region retarded GF evaluation.
0.01 Default

trans_bulksmear (Double):
Imaginary smearing (in eV) added to the real Hamiltonian in electrodes GF evaluation (relevant for trans_method>0).
0.01 Default

trans_efermi (Double):
Fermi energy of the electrode (for α spin) (in eV) used for defining the energy window of the T(E).
-5.0 Default

trans_efermib (Double):
Fermi energy for β spin (in eV). If this is not given, the same value of α spin is used.

trans_adjustefermi (Integer):
Resetting of the Fermi energy (FE) in NEGF using **trans_enable** = 3 or 4
0 Default. Fixed FE specified by **trans_efermi** (and **trans_efermib**) is used.
1 FE is chosen as midpoint of HOMO and LUMO levels
2 FE is adjusted so that charge neutrality is satisfied (recommended for zero bias calculation to establish the Fermi energy of the model)

Options 1, 2 use the same FE for α and β spins. Negative options allow for different FEs for α and β spins.

trans_vmax (Double):
Maximum voltage bias (V).
1.0 Default

Note: The default energy window for transmission and current calculations is defined as:
trans_emin = **trans_efermi** - **trans_vmax**/2 and **trans_emax** = **trans_efermi** + **trans_vmax**/2
if **trans_emin** and **trans_emax** are not given.

The **trans_emin** and **trans_emax** values can be set to determine the energy window for calculating the transmission function (both must be specified). If specified, these values will override the window defined by **trans_efermi** and **trans_vmax** values.

The following paramter establish the voltages for the NEGF level (trans_enable==4):

trans_vstart (Double):
(Only for **trans_enable** = 4). Starting voltage bias (V). The bias voltage increases from **trans_vstart** to **trans_vmax**

in the NEGF calculation.
0.0 Default

trans_nvbias (Integer):
(Only for **trans_enable** = 4), number of points of bias voltage.
1 Default The bias voltage values to be calculated are defined by dividing the range between **trans_vstart** and **trans_vmax** with this number. For example, when **trans_vstart** = 0.0, **trans_vmax** = 1.0, and **trans_nvbias** = 5, the voltages are 0.0, 0.25, 0.50, 0.75, and 1.0. For the case of **trans_nvbias** = 1, the voltage to be calculated is **trans_vmax**.

trans_gridoffset (Double):
(Only for **trans_enable** = 4), Offset distance (in A) to define extended grid box size used for obtaining matrix elements that involve the electrostatic potential in the Fock matrix. See below for details.
5.0 Default

The bias voltage of $+V/2$ is added on the left electrode, and $-V/2$ on the right electrode, where the bias potential slope is along x-axis. The offset should be set to the location of the left surface layer from the leftmost atom in the atomic coordinates (the first atom in the molecule section). The box size is defined by adding/substracting the offset distance to maximum/minimum of each of the x, y and z atomic coordinates. The bias voltage $V(\mathbf{r})$ on the grid points in the box is used for calculating correction term for the Fock matrix (*i.e.* $\langle i|V|j\rangle$).

Note that this grid is *not* for obtaining electrostatic potential by solving the Poisson equation. The setup of the Poisson equation solver is described below. The grid for the Poisson equation is given by *$plots* block keyword (see also example below). The same grid size is used for both boxes.

The following are parameters for addressing the NEGF iterations, **trans_enable** = 3, 4:

trans_restart (Integer):
Flag to restart reading the density matrix files, `DAmat.dat` (and `DBmat.dat`) from the "TransRestart/" directory.
0 No restart (default).
1 Read file of density matrix.

trans_mixing (Double):
Mixing ratio of DIIS mixing method for updating the central block of density matrix in the NEGF iteration.
1.0 Default

trans_mixhistory (Integer):
The number of NEGF iteration steps in which the history of density matrix is stored for the DIIS method.
40 Default

The following parameters are used to set the NEGF density integration:

trans_dehcir (Double):
Grid size, dE (in eV), for integrating the Greens function on the half circle path on imaginary plane.
1.0 Default

trans_delpart (Double):
Grid size, dE (in eV), for integrating the Greens function on the path of the linear part on imaginary plane.
0.01 Default

trans_debwin (Double):
(For **trans_enable** = 4). Grid size, dE (in eV), for integration on the non-equilibrium term.
0.01 Default

trans_numres (Integer):

The number of poles at Fermi energy enclosed by the contour into the imaginary plane.
100 Default

Parameters for controlling the electrostatic potential calculation, the Poisson solver:

trans_peconv (Integer):
The convergence criteria of the iteration of the Poisson equation. The threshold is 10^{-n} hartree of maximum energy difference over the all grid points.
9 Default

trans_pemaxite (Integer):
Maximum iterations of the Poisson solver.
1000 Default

trans_readesp (Integer):
Storing or reading in the electrostatic potential. The data is stored in "ReadInESP/" directory.
-1 Write the ESP data to the "ReadInESP/" directory at the first step and stop calculation.
0 Write the ESP data to the "ReadInESP/" at the first step and continue calculation using it (default).
1 Read the pre-calculated read-in ESP data from "ReadInESP/" and continue calculation.

As an example of the Landauer calculation, the sample Q-CHEM input is given below.

Example 13.3 Quantum transport Landauer calculation applied to C_6 between two gold electrodes.

```
$molecule
   0 1
   Ag   -11.000    0.000  0.000
   Au    -8.300    0.000  0.000
   Ag    -5.600    0.000  0.000
   Au    -2.900    0.000  0.000
   Ag    -0.200    0.000  0.000
   Au     2.500    0.000  0.000
   C      4.800    0.000  0.000
   C      6.500    0.000  0.000
   C      8.200    0.000  0.000
   C      9.900    0.000  0.000
   C     11.600    0.000  0.000
   C     13.300    0.000  0.000
   Au    15.600    0.000  0.000
   Ag    18.300    0.000  0.000
   Au    21.000    0.000  0.000
   Ag    23.700    0.000  0.000
   Au    26.400    0.000  0.000
   Ag    29.100    0.000  0.000
$end

$rem
   METHOD            B3LYP
   BASIS             lanl2dz
   ECP               lanl2dz
   ECP_FIT           true
   INCDFT            FALSE
   MOLDEN_FORMAT     TRUE
   SCF_CONVERGENCE   6
   TRANS_ENABLE      1
$end

$trans-method
   trans_spin        0
   trans_npoints     300
   trans_method      0
   trans_readhs      0
   trans_printdos    1
   trans_efermi      -6.50
   trans_vmax        4.00
$end

$trans-model
   trans_lgatom      3
   trans_latom       5
   trans_ratom       14
   trans_rgatom      16
$end
```

A sample for unrestricted spin calculation can be found in the `$QC/samples/tchem` directory.

For NEGF calculations, note the following comments:

- Only WBL method for evaluating self-energy is available for the NEGF in the current version (**trans_method** = 0)

- In the *$plots* section (when defining a grid box for the Poisson equation solver), the grid box region must cover all atoms except for the left and right electrode parts as defined in *$trans_model*. All integer index flags in *$plots* can be 0.
- For calculations on bulk repeat unit cluster, the structure and orientation of the repeat unit must be consistent with the unit included in the junction region.
- Fermi energy adjustment to satisfy charge neutrality (**trans_adjustefermi** = 2 or 3) are recommended at zero bias (**trans_enable** = 3) before performing a full NEGF calculation.
- The criterion for convergence in the NEGF iterations is the maximum difference in density matrix elements, and is not based on energy.

A NEGF calculation involves the following steps:

Step 1: Pre-calculations:

A: Hamiltonian and overlap matrices of the repeat units in the electrodes (required; obtained using Trans_enable == -1 on the bulk model)

B: A converged junction electronic density by standard DFT (recommended; a conventional Q-CHEM restart file obtained at the DFT level for the junctional model)

C: Electrostatic potential of large electrode region (optional)

Step 2: Self-consistent Greens function calculation at zero bias:

The resulting convered zero-bias density matrices should be used for the biased NEGF calculations.

Step 3: NEGF calculations up to the desired bias.

Example of these steps applied to C_2 between two aluminum electrodes:

Example 13.4 Step 1-A of the NEGF calculation, the pre-calculation of the bulk electrode. Flag keyword of printing matrices must be set (*i.e.* **trans_enable** ==-1).

```
$molecule
   0 1
   Al -15.04250 0.0 0.0
   Al -12.30750 0.0 0.0
   Al -9.572500 0.0 0.0
   Al -6.837500 0.0 0.0
   Al -4.102500 0.0 0.0
   Al -1.367500 0.0 0.0
   Al 1.367500 0.0 0.0
   Al 4.102500 0.0 0.0
   Al 6.837500 0.0 0.0
   Al 9.572500 0.0 0.0
   Al 12.30750 0.0 0.0
   Al 15.04250 0.0 0.0
$end

$rem
   METHOD            hf
   BASIS             hwmb
   ECP               hwmb
   UNRESTRICTED      true
point_group_symmetry False
   ECP_FIT           TRUE
   SCF_CONVERGENCE   4
$end

@@@

$molecule
   read
$end

$rem
   METHOD            b3lyp
   BASIS             hwmb
   ECP               hwmb
   UNRESTRICTED      true
point_group_symmetry False
   SCF_GUESS         read
   SCF_GUESS_MIX     3
   ECP_FIT           TRUE
   SCF_CONVERGENCE   4
$end

@@@

$molecule
   read
$end

$rem
   METHOD            b3lyp
   ECP               hwmb
   BASIS             hwmb
   UNRESTRICTED      true
point_group_symmetry False
   SCF_GUESS         read
   SCF_GUESS_MIX     3
   ECP_FIT           TRUE
   SCF_CONVERGENCE   4
   TRANS_ENABLE      -1
$end

$trans-method
```

As preparation for the next step, the following setups are necessary:

1. The files `FAmat2l.dat`, `FAmat2r.dat`, `Smat2l.dat`, and `Smat2r.dat` (also `FBmat2l.dat` and `FBmat2r.dat` if the calculation is spin-unrestricted) that link to output dat file of the bulk clculation must be placed in the same directory of the following Q-CHEM input file. Coping or linking the output files of the step 1-A.
2. Restart directory of the standard DFT obtained in the step 1-B is used by the SCF_GUESS.
3. (Optional) Read-in electrostatic potential data in "ReadInESP/" directory must be placed if this option is used (for **trans_readesp** = 1). This option can provide more bulk electrode electrostatic environment as the boundary condition of Poisson equation solving (see the sample files in `$QC/samples/tchem`) for more details).

. **Example 13.13.5** Step 2 of the NEGF calculation (device region).

```
{\em Find file for step 1C in the samples directory.}

$molecule
   read
$end
$rem
   JOBTYPE               sp
   UNRESTRICTED          true
   POINT_GROUP_SYMMETRY  false
   MAXSCF                500
   EXCHANGE              b3lyp
   CORRELATION           none
   BASIS                 hwmb
   ECP                   hwmb
   ECP_FIT               TRUE
   SCF_CONVERGENCE       4
   SCF_ALGORITHM         negf
   SCF_GUESS             read
   MEM_TOTAL             16000
   MEM_STATIC            4000
   TRANS_ENABLE          3
$end
$plots
For NEGF (for Poisson equation)
   190    -9.5     9.5
    80    -4.0     4.0
    80    -4.0     4.0
     0  0  0  0
     0
$end
$trans-method
   trans_spin          2
   trans_npoints     500
   trans_method        0
   trans_printdos      1
   trans_printiv       1
   trans_adjustefermi  1
   trans_vmax        1.0
   trans_emin       -6.5
   trans_emax       -2.5
   trans_mixing        0.1
   trans_mixhistory   50
   trans_dehcir        1.0
   trans_delpart       0.01
   trans_numres      100
   trans_peconv        8
   trans_pemaxite   1000
   trans_readesp       0
   trans_readhs        1
   trans_totorb2      48
   trans_startpointl  16
   trans_startpointr  32
$end
$trans-model
   trans_lbasis   8
   trans_rbasis   8
   trans_lgbasis  8
   trans_rgbasis  8
$end
```

As preparation for the next step, the following files and directories are necessary:

1. In the same way as step 2, `FAmat2l.dat`, `FAmat2r.dat`, `Smat2l.dat`, and `Smat2r.dat` (also `FBmat2l.dat` and `FBmat2r.dat` for spin-unrestricted calculations) must be placed.
2. Restart directory for Q-CHEM generated in the step 2 must be copied to here (only coordinates are used).
3. T-Chem Restart directory "TransRestart/" for density matrix must be created and `DAmat.dat` (and `DBmat.dat` for spin-unrestricted) generated in the step 2 must be copied or linked to the directory.
4. Electrostatic potential data in "ReadInESP/" directory used in the step 2 must be copied to here.
5. Set Fermi energy obtained from Step 2 using the **trans_fermi** (recommended).

Example 13.13.6 Step 3 of the NEGF calculation.

```
$molecule
   read
$end

$rem
   UNRESTRICTED      true
   MAXSCF            500
   POINT_GROUP_SYMMETRY false
   EXCHANGE          b3lyp
   ECP               hwmb
   ECP_FIT               TRUE
   SCF_CONVERGENCE   4
   SCF_ALGORITHM     negf
   MEM_TOTAL         16000
   MEM_STATIC        4000
   TRANS_ENABLE      4
$end

$trans-method
   trans_spin         2
   trans_npoints    500
   trans_method       0
   trans_printdos     1
   trans_printiv      1
   trans_adjustefermi 0
   trans_efermi      -4.421836
   trans_vmax         0.5
   trans_emin        -6.5
   trans_emax        -2.5
   trans_mixing       0.2
   trans_mixhistory   50
   trans_dehcir       1.0
   trans_delpart      0.01
   trans_debwin       0.01
   trans_numres       100
   trans_peconv       8
   trans_pemaxite     1000
   trans_gridoffset   4.0
   trans_nvbias       6
   trans_restart      1
   trans_readesp      1
   trans_readhs       1
   trans_totorb2      48
   trans_startpointl  16
   trans_startpointr  32
$end
```

```
$plots
For NEGF calculation
   190    -9.5     9.5
    80    -4.0     4.0
    80    -4.0     4.0
     0  0  0  0
     0
$end

$trans-model
   trans_lbasis   8
   trans_rbasis   8
   trans_lgbasis  8
   trans_rgbasis  8
$end
```

13.5 Nuclear–Electronic Orbital Method

13.5.1 Introduction

The nuclear-electronic orbital (NEO) method[65,89] provides the framework for the accurate and efficient description of multicomponent systems in which more than one type of particle is treated quantum mechanically. Typically, the NEO method treats all electrons and specified protons quantum mechanically on the same level with molecular orbital techniques. An advantage of the NEO method is that anharmonicity, proton delocalization, and zero point energy contributions of the quantum protons are included directly in the energy calculations, geometry optimizations, reaction paths, and dynamics. For a broad overview of the NEO framework and to help guide your selection, see the JCP perspective Ref. 39 and the Chemical Reviews article Ref. 65 both by Hammes-Schiffer.

13.5.2 Overview of Available NEO Models

13.5.2.1 NEO-Hartree-Fock

The simplest method within the NEO framework is the Hartree-Fock (NEO-HF) method, where the total nuclear-electronic wavefunction is approximated as a product of electronic ($\mathbf{\Phi}_0^{\mathrm{e}}(\mathbf{x}_{\mathrm{e}})$) and nuclear ($\mathbf{\Phi}_0^{\mathrm{p}}(\mathbf{x}_{\mathrm{p}})$) Slater determinants composed of electronic and protonic spin orbitals, respectively:

$$\mathbf{\Psi}_{\text{NEO-HF}}(\mathbf{x}_{\mathrm{e}},\mathbf{x}_{\mathrm{p}}) = \mathbf{\Phi}_0^{\mathrm{e}}(\mathbf{x}_{\mathrm{e}})\mathbf{\Phi}_0^{\mathrm{p}}(\mathbf{x}_{\mathrm{p}}) = |0^{\mathrm{e}}0^{\mathrm{p}}\rangle \ . \tag{13.33}$$

Here, $\mathbf{x}_{\mathrm{e}}$ and $\mathbf{x}_{\mathrm{p}}$ are collective spatial and spin coordinates of the quantum electrons and protons. The NEO-HF energy for a restricted Hartree-Fock (RHF) treatment of the electrons and a high-spin open-shell treatment of the quantum protons is

$$\begin{aligned} E_{\text{NEO-HF}} = 2\sum_{i}^{N_{\mathrm{e}}/2} h_{ii}^{\mathrm{e}} + \sum_{i}^{N_{\mathrm{e}}/2}\sum_{j}^{N_{\mathrm{e}}/2}\Big(2(ii|jj) - (ij|ij)\Big) \\ + \sum_{I}^{N_{\mathrm{p}}} h_{II}^{\mathrm{p}} + \frac{1}{2}\sum_{I}^{N_{\mathrm{p}}}\sum_{J}^{N_{\mathrm{p}}}\Big((II|JJ) - (IJ|IJ)\Big) - 2\sum_{i}^{N_{\mathrm{e}}/2}\sum_{I}^{N_{\mathrm{p}}}(ii|II). \end{aligned} \tag{13.34}$$

The $i, j, \cdots$, indices denote occupied spatial electronic orbitals, and the $I, J, \cdots$, indices correspond to occupied spatial protonic orbitals. In Eq. (13.34), h_{ij}^{e} and $(ij|kl)$ are conventional electronic core Hamiltonian and two-electron integrals, respectively, and the corresponding terms for quantum protons are defined analogously. The last term in Eq. (13.34) is the Coulomb interaction between the electrons and the quantum protons. The spatial electronic and

protonic orbitals [$\psi_i^{\mathrm{e}}(\mathbf{r}_{\mathrm{e}})$ and $\psi_I^{\mathrm{p}}(\mathbf{r}_{\mathrm{p}})$] are expanded as linear combinations of electronic or protonic Gaussian basis functions [$\phi_\mu^{\mathrm{e}}(\mathbf{r}_{\mathrm{e}})$ or $\phi_{\mu'}^{\mathrm{p}}(\mathbf{r}_{\mathrm{p}})$]:

$$\psi_i^{\mathrm{e}}(\mathbf{r}_{\mathrm{e}}) = \sum_{\mu}^{N_{\mathrm{e}}^{bf}} C_{\mu i}^{\mathrm{e}} \phi_\mu^{\mathrm{e}}(\mathbf{r}_{\mathrm{e}}) \tag{13.35a}$$

$$\psi_I^{\mathrm{p}}(\mathbf{r}_{\mathrm{p}}) = \sum_{\mu'}^{N_{\mathrm{p}}^{bf}} C_{\mu' I}^{\mathrm{p}} \phi_{\mu'}^{\mathrm{p}}(\mathbf{r}_{\mathrm{p}}) \ . \tag{13.35b}$$

The lower-case Greek letters without and with primes denote basis functions for electrons and protons, respectively, and $C_{\mu i}^{\mathrm{e}}$ and $C_{\mu' I}^{\mathrm{p}}$ are electronic and protonic MO expansion coefficients, respectively.

Analogous to the conventional electronic Hartree-Fock method, the electronic and protonic coefficients are determined by variationally minimizing the energy in Eq. (13.34) via the self-consistent field (SCF) procedure. This procedure leads to a set of coupled electronic and protonic Roothaan equations:

$$\mathbf{F}^{\mathrm{e}}\mathbf{C}^{\mathrm{e}} = \mathbf{S}^{\mathrm{e}}\mathbf{C}^{\mathrm{e}}\mathbf{E}^{\mathrm{e}} \tag{13.36a}$$

$$\mathbf{F}^{\mathrm{p}}\mathbf{C}^{\mathrm{p}} = \mathbf{S}^{\mathrm{p}}\mathbf{C}^{\mathrm{p}}\mathbf{E}^{\mathrm{p}} \ , \tag{13.36b}$$

where $\mathbf{S}^{\mathrm{e}}$ and $\mathbf{S}^{\mathrm{p}}$ are electronic and protonic overlap matrices, respectively. The electronic and protonic Fock elements in Eqs. (13.36a) and (13.36b) are given by

$$F_{\mu\nu}^{\mathrm{e}} = h_{\mu\nu}^{\mathrm{e}} + \sum_{\rho\lambda} P_{\lambda\rho}^{\mathrm{e}} \Big((\mu\nu|\rho\lambda) - \frac{1}{2}(\mu\lambda|\rho\nu) \Big) - \sum_{\mu'\nu'} P_{\nu'\mu'}^{\mathrm{p}} (\mu\nu|\mu'\nu') \tag{13.37a}$$

$$F_{\mu'\nu'}^{\mathrm{p}} = h_{\mu'\nu'}^{\mathrm{p}} + \sum_{\rho'\lambda'} P_{\lambda'\rho'}^{\mathrm{p}} \Big((\mu'\nu'|\rho'\lambda') - (\mu'\lambda'|\rho'\nu') \Big) - \sum_{\mu\nu} P_{\nu\mu}^{\mathrm{e}} (\mu'\nu'|\mu\nu) \ . \tag{13.37b}$$

The electronic and protonic density matrix elements in Eqs. (13.37a) and (13.37b) are defined as

$$P_{\nu\mu}^{\mathrm{e}} = 2 \sum_{i}^{N_{\mathrm{e}}/2} C_{\nu i}^{\mathrm{e}} C_{\mu i}^{\mathrm{e}*} \tag{13.38a}$$

$$P_{\nu'\mu'}^{\mathrm{p}} = \sum_{I}^{N_{\mathrm{p}}} C_{\nu' I}^{\mathrm{p}} C_{\mu' I}^{\mathrm{p}*} \ . \tag{13.38b}$$

The generalization to the unrestricted Hartree-Fock (NEO-UHF) treatment of electrons is accomplished by introducing separate spatial orbitals for α and β electron spins.

The analytical gradients of the NEO-HF energy[89] with respect to the classical nuclear coordinates (or coordinates of the centers of the quantum proton basis functions) are available. These gradients allow geometry optimizations within the NEO framework. The analytical Hessians of the NEO-HF energy with respect to the classical nuclear coordinates are also available.[76] The Hessians can identify whether the optimized geometries are minima or transition states on the ground state vibronic potential energy surface.

The essential *$rem* variables for toggling a NEO-HF calculation are NEO = TRUE and METHOD = HF. Supported JOBTYPE's include SP, FORCE, OPT, TS, and FREQ. Consult Section 13.5.4 for the full list of variables as well as see the examples in Section 13.5.6.

13.5.2.2 NEO-DFT

NEO density functional theory (NEO-DFT)[16,17,62] is an extension of DFT to multicomponent systems within the NEO framework. The Hohenberg-Kohn theorems have been extended to multicomponent systems, where the reference is

expressed as the product of electronic and nuclear Slater determinants composed of Kohn-Sham orbitals. The NEO-DFT total energy is

$$E[\rho^{\mathrm{e}},\rho^{\mathrm{p}}] = E_{\mathrm{ext}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}}] + E_{\mathrm{ref}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}}] + E_{\mathrm{exc}}[\rho^{\mathrm{e}}] + E_{\mathrm{pxc}}[\rho^{\mathrm{p}}] + E_{\mathrm{epc}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}}] \; . \tag{13.39}$$

In this equation, $E_{\mathrm{ext}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}}]$ is the interaction of the electronic and protonic densities with the external potential created by the classical nuclei, and $E_{\mathrm{ref}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}}]$ contains the electron–electron, proton–proton, and electron-proton classical Coulomb energies, as well as the noninteracting kinetic energies of the quantum particles. The terms $E_{\mathrm{exc}}[\rho^{\mathrm{e}}]$, $E_{\mathrm{pxc}}[\rho^{\mathrm{p}}]$, and $E_{\mathrm{epc}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}}]$ are the electron-electron exchange-correlation functional, the proton-proton exchange-correlation functional, and the electron-proton correlation functional, respectively. The quantities

$$\rho^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}}) = 2\sum_{i=1}^{N_{\mathrm{e}}/2} |\psi_i^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})|^2 \tag{13.40a}$$

$$\rho^{\mathrm{p}}(\mathbf{r}_1^{\mathrm{p}}) = \sum_{I=1}^{N_{\mathrm{p}}} |\psi_I^{\mathrm{p}}(\mathbf{r}_1^{\mathrm{p}})|^2 \tag{13.40b}$$

are the electron and proton densities, respectively, and $\psi_i^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})$ and $\psi_I^{\mathrm{p}}(\mathbf{r}_1^{\mathrm{p}})$ are the electronic and protonic Kohn-Sham spatial orbitals, respectively. These orbitals are obtained by solving two sets of coupled Kohn-Sham equations for the electrons and quantum protons:

$$\Big(-\frac{1}{2}\nabla^2 + v_{\mathrm{eff}}^{\mathrm{e}}(\mathbf{r}_1^{\mathrm{e}})\Big)\psi_i^{\mathrm{e}} = \epsilon_i^{\mathrm{e}}\,\psi_i^{\mathrm{e}} \tag{13.41a}$$

$$\Big(-\frac{1}{2m_{\mathrm{p}}}\nabla^2 + v_{\mathrm{eff}}^{\mathrm{p}}(\mathbf{r}_1^{\mathrm{p}})\Big)\psi_I^{\mathrm{p}} = \epsilon_I^{\mathrm{p}}\,\psi_I^{\mathrm{p}} \; . \tag{13.41b}$$

The effective potentials v_{eff} and v_{eff} are obtained by taking the derivative of the total energy expression in Eq. (13.39) with respect to electron density and proton density, respectively. Analogous to NEO-HF, these electronic and protonic Kohn-Sham orbitals are expanded as linear combinations of electronic or protonic Gaussian basis functions ($\phi_\mu^{\mathrm{e}}(\mathbf{r}_{\mathrm{e}})$ and $\phi_{\mu'}^{\mathrm{p}}(\mathbf{r}_{\mathrm{p}})$). The extension to open-shell electron systems is analogous to the NEO-UHF method.

The practical implementation of the NEO-DFT method requires an electron-electron exchange-correlation functional, a proton-proton exchange-correlation functional, and an electron-proton correlation functional. Any conventional electron-electron exchange-correlation functional can be used within the NEO-DFT framework.[10] Because the proton-proton exchange and correlation are negligible in molecular systems, only the exchange at the NEO-Hartree-Fock level is included to eliminate self-interaction error in the NEO-DFT method. A suitable electron-proton correlation functional is essential for obtaining accurate proton densities and energies, and the epc17-2[9,96] and epc19[83] functionals are designed to achieve this goal. These two functionals are based on the multicomponent extension of the Colle-Salvetti formalism. The epc17-2 functional is of the local density approximation (LDA) type with the functional form:

$$E_{\mathrm{epc}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}}] = -\int d\mathbf{r}\frac{\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})}{a - b[\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})]^{1/2} + c\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})} \; . \tag{13.42}$$

The epc19 functional is its multicomponent generalized gradient approximation (GGA) extension that depends on the electron and proton density gradients and is of the form:

$$\begin{aligned} E_{\mathrm{epc}}[\rho^{\mathrm{e}},\rho^{\mathrm{p}},\hat{\boldsymbol{\nabla}}\rho^{\mathrm{e}},\hat{\boldsymbol{\nabla}}\rho^{\mathrm{p}}] = &-\int d\mathbf{r}\frac{\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})}{a - b[\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})]^{1/2} + c\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})}\times \\ &\left\{1 - d\left(\frac{[\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})]^{-1/3}}{(1+m_{\mathrm{p}})^2}\left[m_{\mathrm{p}}^2\frac{\hat{\nabla}^2\rho^{\mathrm{e}}(\mathbf{r})}{\rho^{\mathrm{e}}(\mathbf{r})} - 2m_{\mathrm{p}}\frac{\hat{\boldsymbol{\nabla}}\rho^{\mathrm{e}}(\mathbf{r})\cdot\hat{\boldsymbol{\nabla}}\rho^{\mathrm{p}}(\mathbf{r})}{\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})} + \frac{\hat{\nabla}^2\rho^{\mathrm{p}}(\mathbf{r})}{\rho^{\mathrm{p}}(\mathbf{r})}\right]\right)\exp\left[\frac{-k}{[\rho^{\mathrm{e}}(\mathbf{r})\rho^{\mathrm{p}}(\mathbf{r})]^{1/6}}\right]\right\} \; . \end{aligned} \tag{13.43}$$

In addition to the parameters a, b, and c in the epc17-2 functional,[9] the epc19 functional[83] has the d and k parameters and also depends on the proton mass m_{p}. Analogous to the NEO-HF analytical energy gradients, the NEO-DFT analytical gradients[84] are available for these two functionals, allowing geometry optimizations on the ground state vibronic potential energy surface. The NEO-DFT analytical Hessians[84] are available for the epc17-2 functional or when no electron-proton correlation functional is used and allow characterization of the stationary points.

The essential *$rem* variables for toggling a NEO-DFT calculation are NEO = TRUE and one of the two options: (1) specify METHOD or (2) specify EXCHANGE and CORRELATION. We inherit Q-CHEM's conventional, electronic DFT library, and therefore METHOD, EXCHANGE, and CORRELATION should be set to a functional listed in Section 5.3. Although it has not been tested across all possible combinations and permutations, NEO-DFT should be able to support the vast majority of the electronic XC functionals listed. Supported JOBTYPE's include SP, FORCE, OPT, TS, and FREQ. Consult Section 13.5.4 for the full list of variables as well as see the examples in Section 13.5.6. Lastly, the electron-proton correlation functional should be set to one of the supported types:

NEO_EPC
Specifies the electron-proton correlation functional.
TYPE:
STRING
DEFAULT:
No default
OPTIONS:
NAME Use NEO_EPC = *NAME*, where *NAME* can be either epc172 or epc19.
RECOMMENDATION:
Consult the NEO literature to guide your selection.

13.5.2.3 NEO-MSDFT

The NEO multistate DFT (NEO-MSDFT) method was developed to describe hydrogen transfer and hydrogen tunneling systems within the NEO framework.[26,99] Analogous to the conventional electronic MSDFT method,[30,38,57,60] the NEO-MSDFT method linearly combines localized NEO-DFT states in a nonorthogonal configuration interaction (NOCI) scheme.[79] This approach captures the delocalized, bilobal proton densities needed to describe hydrogen tunneling and avoids complications of local minima in orbital space.

Consider a system with N transferring protons, where each proton moves in a double-well potential in the Born-Oppenheimer picture. In NEO-DFT calculations, the protonic density tends to localize in one well of a symmetric or nearly symmetric double-well potential instead of delocalizing over both wells. Quantizing each transferring proton with the NEO approach leads to 2^N diabatic NEO-DFT states. Each diabatic state has the protonic density of each transferring proton localized in one of the two wells of its corresponding double-well potential. In practice, higher-energy diabatic states can be excluded from the NOCI expansion. The set of all diabatic NEO-DFT states is $\{|\tilde{\Psi}_0\rangle, |\tilde{\Psi}_1\rangle, \ldots |\tilde{\Psi}_n\rangle\}$, where $n = 2^N - 1$. Each diabatic state is the product of a Kohn-Sham electronic and protonic determinant, as discussed in Section 13.5.2.2.

The adiabatic NEO-MSDFT states $\{|\Psi_0\rangle, |\Psi_1\rangle, \ldots |\Psi_n\rangle\}$ are linear combinations of all diabatic NEO-DFT states:

$$\begin{aligned} |\Psi_0\rangle &= D_0^0|\tilde{\Psi}_0\rangle + D_1^0|\tilde{\Psi}_1\rangle + \cdots + D_n^0|\tilde{\Psi}_n\rangle \\ |\Psi_1\rangle &= D_0^1|\tilde{\Psi}_0\rangle + D_1^1|\tilde{\Psi}_1\rangle + \cdots + D_n^1|\tilde{\Psi}_n\rangle \\ &\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ |\Psi_n\rangle &= D_0^n|\tilde{\Psi}_0\rangle + D_1^n|\tilde{\Psi}_1\rangle + \cdots + D_n^n|\tilde{\Psi}_n\rangle \end{aligned} \tag{13.44}$$

The coefficients in Eq. (13.44) are determined by solving the $2^N \times 2^N$ matrix eigenvalue problem

$$\mathbf{HD} = \mathbf{SDE}. \tag{13.45}$$

The overlap matrix $\mathbf{S}$ and effective Hamiltonian matrix $\mathbf{H}$ contain the overlap and couplings, respectively, between pairs of localized diabatic states. Note that the diagonal elements of $\mathbf{S}$ are unity, and the diagonal elements of $\mathbf{H}$ are

the NEO-DFT energies of the diabatic states. For the sake of brevity, the analytical forms of the off-diagonal terms of these matrices are excluded here but are given in previous work.[26]

The limitations of the epc functionals and the resulting inaccuracies of the overlap between two localized NEO-DFT states, as well as the approximate form of the off-diagonal Hamiltonian matrix elements, can be accounted for by applying a simple correction function to the off-diagonal elements of the **S** matrix:

$$S'_{ij} = \alpha(S_{ij})^{\beta}. \tag{13.46}$$

Users can control the values of α and β used in NEO-MSDFT calculations, and their default values were determined though a fitting process discussed in our previous work.[99] Setting both these parameters to unity produces the original overlap matrix. However, in general the alpha and beta parameters should be kept at their default values unless another parameterization procedure is performed to determine alternative values.

A NEO-MSDFT calculation can be enabled by setting the *$rem* variable NEO_MSDFT = 1 in addition to the variables in the NEO-DFT subsection (see 13.5.2.2):

NEO_MSDFT

Enable a NEO-MSDFT calculation

TYPE:

INTEGER

DEFAULT:

0 No NEO-MSDFT calculation.

OPTIONS:

1 Enable a NEO-MSDFT calculation.

0 Disable a NEO-MSDFT calculation.

RECOMMENDATION:

See Section 13.5.2.3 for details on customizing a NEO-MSDFT calculation.

Analytical gradients have also been implemented,[102] allowing users to optimize geometries on specified NEO-MSDFT surfaces by setting JOBTYPE = OPT. By default, analytical derivative couplings between NEO-MSDFT states are computed directly following an analytical gradient calculation,[101] but this calculation can be disabled. Note that the only gradients and derivative couplings printed out are those pertaining to the ground and first-excited NEO-MSDFT states, as these are the only states of physical relevance (as explained elsewhere[26]). Users can also request a semi-numerical Hessian calculation by setting IDERIV = 2. Further control of a NEO-MSDFT calculation is given by the *$neo_msdft* section, whose job-control options are given below:

OPT_STATE

Controls which NEO-MSDFT state is geometry-optimized when JOBTYPE = OPT or which NEO-MSDFT surface the semi-numerical Hessian will be calculated on when IDERIV = 2.

INPUT SECTION: *$neo_msdft*

TYPE:

INTEGER

DEFAULT:

0 Ground-State

OPTIONS:

n Indicates optimization/semi-numerical Hessian calculation will occur on the nth NEO-MSDFT surface.

RECOMMENDATION:

Ensure that n is strictly less than the number of diabatic states included in the adiabatic state expansion of Eq. (13.44).

ALPHA

Sets the α parameter used for the overlap correction of Eq. (13.46).

INPUT SECTION: *$neo_msdft*

TYPE:

FLOAT

DEFAULT:

0.0604

OPTIONS:

User-defined.

RECOMMENDATION:

Keep default value unless another parameterization procedure is performed.

BETA

Sets the β parameter used for the overlap correction of Eq. (13.46).

INPUT SECTION: *$neo_msdft*

TYPE:

FLOAT

DEFAULT:

0.492

OPTIONS:

User-defined.

RECOMMENDATION:

Keep default value unless another parameterization procedure is performed.

NACV

Controls if analytical derivative couplings are calculated following an analytical gradient calculation.

INPUT SECTION: *$neo_msdft*

TYPE:

INTEGER

DEFAULT:

1 Enables derivative coupling calculation.

OPTIONS:

1 Enables derivative coupling calculation.

0 Disables derivative coupling calculation.

RECOMMENDATION:

None.

DENPLT

Controls the generation of proton density cube files for NEO-MSDFT states.

INPUT SECTION: *$neo_msdft*

TYPE:

INTEGER

DEFAULT:

-1 Disables generation of proton density cube files.

OPTIONS:

n Enables generation of proton density cube files for NEO-MSDFT states $\{0, \ldots, n\}$.

RECOMMENDATION:

Users can also generate electron density cube files for each of the specified NEO-MSDFT states via the *$plots* section (refer to Section 10.5.4.1 for details). Also note that if $n > 0$, a directory `neo_msdft_denplt` will be created in the current working directory where all the generated cube files will be written.

CPSCF_THRESH

Controls the convergence criteria for the NEO-CPSCF routine. Solving the NEO-CPSCF equations is required for calculating the analytical gradients of the NEO-MSDFT energies.[102] The NEO-CPSCF routine is considered converged when the error is less than $10^{-\text{CPSCF\_THRESH}}$.

INPUT SECTION: *$neo_msdft*

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

User-defined.

RECOMMENDATION:

Tightening the NEO-CPSCF convergence will improve the reliability of the analytical gradients.

CPSCF_NSTEPS

Controls the maximum number of NEO-CPSCF iterations permitted. Solving the NEO-CPSCF equations is required for calculating the analytical gradients of the NEO-MSDFT energies.[102] The NEO-CPSCF routine will fail if it does not converge within a CPSCF_NSTEPS number of steps.

INPUT SECTION: *$neo_msdft*

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

User-defined.

RECOMMENDATION:

If CPSCF_THRESH is higher than the default, increasing the number of iterations permitted in the NEO-CPSCF routine may be needed.

FDIFF_STEP
Sets the distance each center is perturbed if a semi-numerical Hessian calculation is requested.
INPUT SECTION: *$neo_msdft*
TYPE:
FLOAT
DEFAULT:
10^{-4} Å
OPTIONS:
User-defined.
RECOMMENDATION:
If INPUT_BOHR = FALSE, then the step size input by the user will be measured in Å. If INPUT_BOHR = TRUE, then the step size input by the user will be measured in bohr.

As discussed in previous work,[26,99] each quantum transferring proton in NEO-MSDFT is given two basis function centers: one center localized near the donor and another center localized near the acceptor. When requesting a NEO-MSDFT calculation, in the *$molecule* section, each quantum transferring proton must have one center input as a standard H atom center and its other center input as a ghost H atom center. These centers must be listed consecutively in the *$molecule* section. Additionally, it is assumed that all ghost H atom centers in the *$molecule* section of a NEO-MSDFT calculation belong to a quantum transferring proton.

By default, all possible 2^N NEO-DFT diabatic states are included in the adiabatic state expansion of Eq. (13.44). If this is not desired, users can provide an input to the *$neo_msdft_diabat_control* section to specify which diabatic states to include. This input section takes in an array of Boolean values (input as ones or zeros), where each row is a unique string corresponding to a particular diabatic state. These Booleans refer to whether or not a quantum transferring proton is localized on each proton basis function center in the order in which they appear in the *$molecule* section. Thus, if the number of diabatic states the user wants to include is m and the number of quantum transferring protons is N, the *$neo_msdft_diabat_control* section should contain an array with m rows and $2N$ columns, where the i-th row specifies which proton basis function centers have a localized quantum proton in the i-th diabatic state. Note that other protons can be quantized with NEO that do not correspond to a transferring proton and therefore are represented by a single proton basis function center; these protons should not be included in the *$neo_msdft_diabat_control* section. Examples showing how to set up NEO-MSDFT calculations/input files with both one and multiple transferring protons can be found in Section 13.5.6.

The overlap correction of Eq. (13.46) is parameterized to produce accurate proton densities and tunneling splittings when the proton basis function centers are optimized for separate NEO-DFT diabatic states (see the procedures explained in our previous work).[26,99] However, users may wish to perform geometry optimizations on NEO-MSDFT surfaces instead. One can either optimize the positions of just the proton basis function centers for fixed geometries of classical nuclei or optimize the positions of both the classical nuclei and proton basis function centers simultaneously. Both cases require the user to set JOBTYPE = OPT, but in the former case, the *$opt* section must be used to freeze the classical nuclei during the optimization (see Section 9.4.3 for details). In either case, the input of the *$molecule* section serves as an initial guess for the optimization, so users must make an informed guess as to where to initially place each of the two proton basis function centers for each quantum transferring proton.

13.5.2.4 CNEO

In NEO-SCF calculations, the nuclear solution corresponds to the variational nuclear wavefunction (density) for the specific method and basis sets. The nuclear orbitals in molecular systems are localized, and therefore the quantized nuclei can be treated as distinguishable when analyzing their position expectation values. These expectation values

are generally not equal to the input coordinates of the NEO nuclear basis function centers. Control over the position expectation values of the nuclei can be achieved with the constrained (C)NEO approach,[93] based on the original NEO method[89] and in analogy to the conventional constrained DFT (Section 5.11). This is accomplished by imposing the following constraint:

$$\langle \psi_I^{\mathrm{p}} | \mathbf{r}_1^{\mathrm{p}} | \psi_I^{\mathrm{p}} \rangle = \mathbf{R}_I \tag{13.47}$$

where $\mathbf{R}_I$ is the chosen value for the position expectation value for nucleus I. For geometry optimizations and vibrational frequency calculations, typically $\mathbf{R}_I$ is the position of the NEO nuclear basis function center, as defined by the input coordinates. The constraint is imposed by using Lagrange multipliers. In this case, the protonic part of the coupled Kohn-Sham equations of NEO-DFT (Eq. (13.41b)) becomes:

$$\Big(-\frac{1}{2m_{\mathrm{p}}}\nabla^2 + v_{\mathrm{eff}}^{\mathrm{p}}(\mathbf{r}_1^{\mathrm{p}}) + \mathbf{f}_I \cdot \mathbf{r}_1^{\mathrm{p}} \Big) \psi_I^{\mathrm{p}} = \epsilon_I^{\mathrm{p}}\, \psi_I^{\mathrm{p}} \tag{13.48}$$

where $\mathbf{f}_I$ is the Lagrange multiplier. In practice, the value of $\mathbf{f}_I$ is optimized numerically to satisfy the constraint in Eq. (13.47).

CNEO calculations can be invoked by setting CNEO = TRUE in addition to the variables for NEO-HF 13.5.2.1 (or NEO-DFT 13.5.2.2):

CNEO
Enable a CNEO calculation. Using CNEO with multiple quantum protons automatically activates the nuclear Hartree product approximation.

TYPE:
LOGICAL/INTEGER

DEFAULT:
FALSE No CNEO calculation.

OPTIONS:
TRUE (or 1) Enable a CNEO calculation.
FALSE (or 0) Disable a CNEO calculation.

RECOMMENDATION:
None.

Analytical CNEO gradients[94] and Hessians[95] are available, allowing geometry optimizations with JOBTYPE = OPT and vibrational frequency calculations with JOBTYPE = FREQ. CNEO can be used with Hartree-Fock or with DFT in conjunction with the epc17-2 or epc19 electron-proton correlation functional and the electron exchange-correlation functionals available in Q-CHEM. Note that using CNEO with multiple quantum protons invokes the nuclear Hartree product approximation (Section 13.5.2.5).

13.5.2.5 Nuclear Hartree Product Approximation

In NEO-HF, the total wavefunction is a product of electronic and nuclear Slater determinants (Section 13.5.2.1); relatedly, in NEO-DFT, the reference is a product of Slater determinants (Section 13.5.2.2). Because the nuclear orbitals in molecular systems are highly localized and the overlap between them is effectively zero, the nuclear exchange interactions are numerically insignificant,[65] and we can treat the quantum nuclei as distinguishable. This means that we can approximate the nuclear wavefunction as a Hartree product instead of a Slater determinant:[4]

$$\Phi_{\mathrm{Hartree}}^{\mathrm{p}}(\mathbf{x}_{\mathrm{p}}) = \prod_{I=1}^{N_{\mathrm{p}}} \chi_I^{\mathrm{p}}(\mathbf{x}_I^{\mathrm{p}}) \,. \tag{13.49}$$

Here, the product is over N_p quantum protons, $\mathbf{x}_\mathrm{p}$ are the collective spatial and spin coordinates for the quantum protons, and $\chi_I^\mathrm{p}(\mathbf{x}_I^\mathrm{p})$ is the spin orbital for the Ith quantum proton. The total wavefunction is then (see also Eq. (13.33)):

$$\Psi'_{\text{NEO-HF}}(\mathbf{x}_\mathrm{e},\mathbf{x}_\mathrm{p}) = \Phi_0^\mathrm{e}(\mathbf{x}_\mathrm{e}) \prod_{I=1}^{N_\mathrm{p}} \chi_I^\mathrm{p}(\mathbf{x}_I^\mathrm{p}) \; . \tag{13.50}$$

Note that a high-spin configuration is assumed for the protonic wavefunction, and therefore the protonic spin orbitals can be replaced with spatial orbitals multiplied by the same spin function. For NEO-HF, Eq. (13.50) leads to N_p coupled one-nucleus Roothaan equations instead of one N_p-nucleus Roothaan equation (Eq. (13.36b)). For NEO-DFT, it leads to N_p coupled one-nucleus Kohn-Sham equations (Eq. (13.41b)). Making use of the localized nature of the quantum nuclei, the spatial nuclear orbitals for each nucleus (Eq. (13.35)) are expanded in the local basis set for that nucleus (i.e., different nuclei do not share basis functions).

The nuclear Hartree product approximation can be activated by setting NEO_N_HARTREE_PROD = TRUE in addition to the variables for NEO-HF 13.5.2.1 (or NEO-DFT 13.5.2.2):

NEO_N_HARTREE_PROD

Enable the nuclear Hartree product approximation.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

TRUE (or 1) Enable a nuclear Hartree product calculation.

FALSE (or 0) Disable a nuclear Hartree product calculation.

RECOMMENDATION:

Use simultaneous NEO-SCF for faster convergence.

It is recommended to use the simultaneous NEO-SCF algorithms for faster convergence (Section 13.5.4.3). By default, the Coulomb and exchange self-interaction terms for each nucleus are computed. Similarly to NEO-SCF calculations with one quantum proton, the evaluation of these terms can be turned off with NEO_VPP = FALSE to further accelerate the calculations. Note that this does not change the NEO-SCF energy and the occupied orbitals, but it does affect the nuclear virtual orbitals.[82] Note that the supported JOBTYPE's are the same as for CNEO 13.5.2.4 and include SP, FORCE, OPT, TS, and FREQ.

13.5.2.6 LR-NEO-TDDFT

The LR-NEO-TDDFT method[97] is a multicomponent extension of the linear response TDDFT method within the NEO framework. It allows the simultaneous calculation of the electronic and proton vibrational excitation energies. In the (LR)-NEO-TDDFT method, the linear response of the NEO Kohn-Sham system to perturbative external fields is computed. The NEO-TDDFT working equation is

$$\begin{bmatrix} \mathbf{A}^\mathrm{e} & \mathbf{B}^\mathrm{e} & \mathbf{C} & \mathbf{C} \\ \mathbf{B}^\mathrm{e} & \mathbf{A}^\mathrm{e} & \mathbf{C} & \mathbf{C} \\ \mathbf{C}^\mathrm{T} & \mathbf{C}^\mathrm{T} & \mathbf{A}^\mathrm{p} & \mathbf{B}^\mathrm{p} \\ \mathbf{C}^\mathrm{T} & \mathbf{C}^\mathrm{T} & \mathbf{B}^\mathrm{p} & \mathbf{A}^\mathrm{p} \end{bmatrix} \begin{bmatrix} \mathbf{X}^\mathrm{e} \\ \mathbf{Y}^\mathrm{e} \\ \mathbf{X}^\mathrm{p} \\ \mathbf{Y}^\mathrm{p} \end{bmatrix} = \omega \begin{bmatrix} \mathbf{I} & 0 & 0 & 0 \\ 0 & -\mathbf{I} & 0 & 0 \\ 0 & 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}^\mathrm{e} \\ \mathbf{Y}^\mathrm{e} \\ \mathbf{X}^\mathrm{p} \\ \mathbf{Y}^\mathrm{p} \end{bmatrix} \tag{13.51}$$

where

$$A^{\mathrm{e}}_{ia,jb} = (\epsilon_a - \epsilon_i)\delta_{ab}\delta_{ij} + \langle aj|ib\rangle + \frac{\delta^2 E_{\mathrm{exc}}}{\delta P^{\mathrm{e}}_{jb}\delta P^{\mathrm{e}}_{ai}} + \frac{\delta^2 E_{\mathrm{epc}}}{\delta P^{\mathrm{e}}_{jb}\delta P^{\mathrm{e}}_{ai}} \tag{13.52}$$

$$B^{\mathrm{e}}_{ia,jb} = \langle ab|ij\rangle + \frac{\delta^2 E_{\mathrm{exc}}}{\delta P^{\mathrm{e}}_{jb}\delta P^{\mathrm{e}}_{ia}} + \frac{\delta^2 E_{\mathrm{epc}}}{\delta P^{\mathrm{e}}_{jb}\delta P^{\mathrm{e}}_{ia}} \tag{13.53}$$

$$A^{\mathrm{p}}_{IA,JB} = (\epsilon_A - \epsilon_I)\delta_{AB}\delta_{IJ} + \langle AJ|IB\rangle + \frac{\delta^2 E_{\mathrm{pxc}}}{\delta P^{\mathrm{p}}_{JB}\delta P^{\mathrm{p}}_{AI}} + \frac{\delta^2 E_{\mathrm{epc}}}{\delta P^{\mathrm{p}}_{JB}\delta P^{\mathrm{p}}_{AI}} \tag{13.54}$$

$$B^{\mathrm{p}}_{IA,JB} = \langle AB|IJ\rangle + \frac{\delta^2 E_{\mathrm{pxc}}}{\delta P^{\mathrm{p}}_{JB}\delta P^{\mathrm{p}}_{IA}} + \frac{\delta^2 E_{\mathrm{epc}}}{\delta P^{\mathrm{p}}_{JB}\delta P^{\mathrm{p}}_{IA}} \tag{13.55}$$

$$C_{ia,JB} = -\langle aB|iJ\rangle + \frac{\delta^2 E_{\mathrm{epc}}}{\delta P^{\mathrm{p}}_{JB}\delta P^{\mathrm{e}}_{ai}} \tag{13.56}$$

Here, the occupied electronic orbitals are denoted with indices i and j, whereas the unoccupied electronic orbitals are denoted with indices a and b. The analogous upper case indices denote protonic orbitals. The solution of Eq. (13.51) provides the electronic and proton vibrational excitation energies ω, as well as the transition excitation and de-excitation amplitudes, $\mathbf{X}$ and $\mathbf{Y}$, respectively. Analogous to the TDDFT method, the Tamm-Dancoff approximation (TDA)[41] can be imposed within the NEO framework, defining the NEO-TDDFT-TDA method that is represented by

$$\begin{bmatrix} \mathbf{A}^{\mathrm{e}} & \mathbf{C} \\ \mathbf{C}^{\mathrm{T}} & \mathbf{A}^{\mathrm{p}} \end{bmatrix} \begin{bmatrix} \mathbf{X}^{\mathrm{e}} \\ \mathbf{X}^{\mathrm{p}} \end{bmatrix} = \omega \begin{bmatrix} \mathbf{X}^{\mathrm{e}} \\ \mathbf{X}^{\mathrm{p}} \end{bmatrix} . \tag{13.57}$$

The extension of the NEO-TDDFT and NEO-TDDFT-TDA approaches to open-shell electron systems is straightforward.[22] NEO-TDHF and NEO-CIS have similar forms as NEO-TDDFT and NEO-TDA without electron-proton, electron-electron, or proton-proton correlation. The analytical gradients for NEO-CIS/NEO-TDA/NEO-TDHF/NEO-TDDFT are available,[84] enabling geometry optimizations on the excited state vibronic potential energy surfaces. For NEO-TDA and NEO-TDDFT, analytical gradients are available for the epc17-2 functional or when no electron-proton correlation functional is used. The transition densities can be analyzed to determine the percentages of electronic and protonic character for each vibronic excited state.

The LR-NEO-TDHF (or TDDFT) method is activated by the following keyword:

RPA

Do a NEO-TDDFT or NEO-TDHF calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do a NEO-TDA or NEO-CIS calculation.

TRUE Do a NEO-TDDFT or NEO-TDHF calculation.

RECOMMENDATION:

Consult the NEO literature to guide your selection.

Furthermore, the variables for NEO-HF 13.5.2.1 (or NEO-DFT 13.5.2.2) should be included. Supported JOBTYPE's include SP and OPT. Additional customization can be controlled by the following *$rem* variables:

CIS_N_ROOTS

Sets the number of NEO excited state roots to find by Davidson or display the number of roots obtained by direct diagonalization.

TYPE:

INTEGER

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

n $n > 0$ Looks for n NEO excited states.

RECOMMENDATION:

None

RPA

Do a NEO-TDDFT or NEO-TDHF calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do a NEO-TDA or NEO-CIS calculation.

TRUE Do a NEO-TDDFT or NEO-TDHF calculation.

RECOMMENDATION:

Consult the NEO literature to guide your selection.

DIRECT_DIAG

Perform direct diagonalization to obtain all the NEO excitation energies.

TYPE:

INTEGER

DEFAULT:

0 Use Davidson algorithm.

OPTIONS:

1 Do the direct diagonalization.

0 Use Davidson algorithm.

RECOMMENDATION:

Only use this option when Davidson solutions are not stable.

CIS_STATE_DERIV

This keyword is used to specify for which NEO excited state the gradient or geometry optimization is needed.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

n $n > 0$ Looks to calculate gradient or conduct geometry optimization for the nth NEO excited state.

RECOMMENDATION:

Consult the keyword NEO_SET_ESTATE if gradient is desired for a vibronic excited state with dominant electronic character.

NEO_SET_ESTATE

This keyword is used to specify for which vibronic excited state with dominant electronic character the gradient or geometry optimization is needed.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

n $n > 0$ Looks to calculate gradient or conduct geometry optimization for the nth NEO vibronic excited state with dominant electronic character.

RECOMMENDATION:

Make sure enough roots are requested by the CIS_N_ROOTS keyword because the vibronic excited states with dominant protonic character usually come before.

NEO_SET_OPT

Enable a NEO excited state geometry optimization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Enable a NEO excited state geometry optimization.

0 Disable a NEO excited state geometry optimization.

RECOMMENDATION:

Need to use with CIS_STATE_DERIV. Consult the keyword NEO_SET_ESTATE if geometry optimization is desired for a vibronic excited state with dominant electronic character.

NEO_ZVEC_LINEAR

Use linear solver for Z-vector equations for NEO excited state gradient.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Use linear solver

0 Use iterative conjugate gradient solver

RECOMMENDATION:

Use the default iterative conjugate gradient solver because it is more memory efficient.

NEO_ZVEC_CG_MAXITER

Controls the maximum number of iterative gradient solver iterations permitted.

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

n Use $n > 0$ iterations.

RECOMMENDATION:

None.

NEO_ZVEC_CG_CONV

The convergence threshold ($10^{-\text{NEO\_ZVEC\_CG\_CONV}}$) for the iterative gradient solver for NEO Z-vector equations.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n Use $n > 0$ iterations.

RECOMMENDATION:

None.

SET_SUBSPACE

Specify the number of protonic guess vectors for NEO-TDDFT

TYPE:

INTEGER

DEFAULT:

Number of states desired (as set by CIS_N_ROOTS) if the number is smaller than the size of the protonic subspace (number of protonic occupied orbitals $\times$ number of protonic virtual orbitals) or the size of the protonic subspace

OPTIONS:

n Use $n > 0$ vectors.

RECOMMENDATION:

None.

13.5.2.7 RT-NEO-TDDFT

Real-time NEO time-dependent density functional theory (RT-NEO-TDDFT or RT-NEO for brevity) is a multicomponent extension of conventional electronic RT-TDDFT within the NEO framework [103] and is an approach that enables the modeling of nonequilibrium, non-Born-Oppenheimer nuclear-electronic quantum dynamics. This approach assumes the form of the single Slater determinant product *ansatz* of the NEO-DFT reference system, where $\mathbf{x}^{\mathrm{e}}$ and $\mathbf{x}^{\mathrm{p}}$ are collective spatial and spin coordinates of the quantum electrons and protons, $\mathbf{P}^{\mathrm{e}}$ (or $\mathbf{P}^{\mathrm{p}}$) is the single particle density matrix and $\mathbf{F}^{\mathrm{e}}$ (or $\mathbf{F}^{\mathrm{p}}$) is the Kohn-Sham matrix for the electrons (or protons), and t is the time. The time-dependent Schrödinger equation

$$i\hbar\frac{\partial}{\partial t}\Psi_{\mathrm{NEO}}\left(\mathbf{x}^{\mathrm{e}}\,,\mathbf{x}^{\mathrm{p}}\,;t\right)=H\left(\mathbf{x}^{\mathrm{e}}\,,\mathbf{x}^{\mathrm{p}}\,;t\right)\Psi_{\mathrm{NEO}}\left(\mathbf{x}^{\mathrm{e}}\,,\mathbf{x}^{\mathrm{p}}\,;t\right) \tag{13.58}$$

can be propagated according to the set of multicomponent von Neumann equations:

$$i\hbar\frac{\partial}{\partial t}\mathbf{P}^{\mathrm{e}}\left(t\right)=\left[\mathbf{F}^{\mathrm{e}}\left(t,\mathbf{P}^{\mathrm{e}}\left(t\right),\mathbf{P}^{\mathrm{p}}\left(t\right)\right),\mathbf{P}^{\mathrm{e}}\left(t\right)\right] \tag{13.59a}$$

$$i\hbar\frac{\partial}{\partial t}\mathbf{P}^{\mathrm{p}}\left(t\right)=\left[\mathbf{F}^{\mathrm{p}}\left(t,\mathbf{P}^{\mathrm{p}}\left(t\right),\mathbf{P}^{\mathrm{e}}\left(t\right)\right),\mathbf{P}^{\mathrm{p}}\left(t\right)\right] \tag{13.59b}$$

Note that these equations are not propagated independently, but are actually coupled through Fock terms that depend on both particle densities. The electron and proton densities are evolved together in time. The resulting electronic and protonic dipole moments can be obtained at every time step and analyzed via post-processing to provide relevant spectral features. Fourier transformation of these signals from the corresponding time-domain to the frequency-domain can yield both electronic and vibrational spectra that have been shown to agree well with LR-NEO-TDDFT. In addition, this approach provides the nonequilbrium, real-time dynamics of the electronic and protonic densities.

Up to this point, the equations of motion presented above [Eq. (13.59a) and Eq. (13.59b) assume that the classical nuclei are held at a fixed geometry throughout the entire simulation. To combine the real-time dynamics of the electrons and protons together with the mean-field motion of the classical nuclei, the RT-NEO-TDDFT-Ehrenfest approach has been formulated.[104] In this approach, the classical nuclei are evolved according to Newtonian dynamics, while the nonequilibrium electrons and protons are still being propagated according to Eq. (13.59a) and Eq. (13.59b), respectively:

$$M_A \ddot{\mathbf{R}}_A = -\nabla_A \left\langle \Psi_{\text{NEO}} \left| \hat{H}_{\text{NEO}} \right| \Psi_{\text{NEO}} \right\rangle \tag{13.60}$$

Here we denote M_A and $\mathbf{R}_A$ to be the mass and the position, respectively, of the Ath classical nucleus, and the abbreviated form of the total nuclear gradient expression is $\nabla_A \left\langle \Psi_{\text{NEO}} \left| \hat{H}_{\text{NEO}} \right| \Psi_{\text{NEO}} \right\rangle$. We support both of these approaches, *i.e.*, RT-NEO with fixed classical nuclei and RT-NEO with moving classical nuclei, and we have designated those methods as METHOD = REALTIME and METHOD = EHRENFEST, respectively.

In addition to electrons, it is important that the proton dynamics is adequately described throughout the course of a given trajectory. This is especially important in cases where proton delocalization may start to occur, such as during a proton transfer reaction or when the geometry of the classical nuclei is no longer held fixed. To provide the flexibility for describing proton delocalization, one may use a larger protonic basis set and/or utilize multiple fixed proton basis (FPB) function centers placed at pre-designated positions. Alternatively, a traveling proton basis (TPB) approach has been formulated and will be made available in a future release.[104] Examples of using either one or multiple FPB function centers can be found in Section 13.5.6) of this manual .

In certain applications, the requirement of using a small time step due to the fast electronic dynamics can prohibit long simulation times. For systems that are electronically adiabatic, the electronic Born-Oppenheimer (BO) approximation can be invoked by quenching the electrons to an instantaneous electronic ground state via the standard self-consistent-field (SCF) procedure instead of propagating Eq. (13.59a) at every time step:[53]

$$\mathbf{P}^{\text{e}}(t) = \text{SCF}\left[\mathbf{P}^{\text{p}}(t), \{\mathbf{R}_A(t)\}\right] \tag{13.61}$$

This approach enables the use of a time step that is several orders of magnitude larger than that required for integrating Eq. (13.59a), therefore rendering longer simulation times more tractable. The electronic BO approximation can be invoked for RT-NEO with fixed classical nuclei or for RT-NEO with moving classical nuclei by toggling the methods METHOD = BO-REALTIME or METHOD = BO-EHRENFEST, respectively.

A RT-NEO-HF or RT-NEO-TDDFT calculation can be invoked by setting NEO_TDKS = TRUE in the *$rem* section. In addition, the variables for NEO-HF 13.5.2.1 (or NEO-DFT 13.5.2.2) should be included, and note that the JOBTYPE *$rem* variable is not used as it's not applicable:

NEO_TDKS

Enable a real-time NEO-TDHF or NEO-TDDFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE No RT-NEO calculation.

OPTIONS:

TRUE Enable a RT-NEO calculation.

FALSE Disable a RT-NEO calculation.

RECOMMENDATION:

None.

Additional job control is provided in its own input section. The input file for a RT-NEO propagation is illustrated in several examples in Section 13.5.6, however, the available controls are also discussed below. Note that the keywords

and field types defined for *$tdks* (Section 7.4) are not the same parameters as those defined for the *$neo_tdks*, and *vice versa*. Only parameters/keywords documented below are supported. The parameters of the jobs are set using options specified in the *$neo_tdks* input section. The format of the *$neo_tdks* section is analogous to the *$rem* section:

```
$neo_tdks
  <Keyword>  <parameter/option>
$end
```

Note: The following job control variables belong *only* in the *$neo_tdks* section. Do not place them in the *$rem* section.

METHOD

Specifies which RT-NEO dynamics approach will be used to propagate the system.

INPUT SECTION: *$neo_tdks*

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Realtime	Fixed classical nuclei[103]
Ehrenfest	Moving classical nuclei[104]
BO-Realtime	Fixed classical nuclei with electronic BO approximation[53]
BO-Ehrenfest	Moving classical nuclei with electronic BO approximation[53]

RECOMMENDATION:

None.

DT

Specifies the time step Δt, in atomic units.

INPUT SECTION: *$neo_tdks*

TYPE:

DOUBLE

DEFAULT:

0.04

OPTIONS:

Δt $\Delta t > 0$ User-selected.

RECOMMENDATION:

None.

MAXITER

Specifies the maximum number of time steps to simulate.

INPUT SECTION: *$neo_tdks*

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

The total propagation length is $\Delta t \times$ **MAXITER**.

ELECTRONIC_HOMO_TO_LUMO

Performs a HOMO → LUMO excitation for a single electron prior to time propagation.

An unrestricted calculation is performed when this keyword is invoked.

INPUT SECTION: *$neo_tdks*

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Should not use in conjunction with the electronically adiabatic approximation.

To perturb the system, a time-dependent electric field pulse $\boldsymbol{\mathcal{E}}(t)$ can be applied via a dipole coupling term added to the electronic and/or protonic Fock matrices,

$$\mathbf{F}^{\mathrm{e}} = \mathbf{F}^{\mathrm{e},0} - \boldsymbol{\mu}^{\mathrm{e}} \cdot \boldsymbol{\mathcal{E}}(t) \tag{13.62a}$$

$$\mathbf{F}^{\mathrm{p}} = \mathbf{F}^{\mathrm{p},0} - \boldsymbol{\mu}^{\mathrm{p}} \cdot \boldsymbol{\mathcal{E}}(t) \ , \tag{13.62b}$$

where $\mathbf{F}^{\mathrm{e},0}$ (or $\mathbf{F}^{\mathrm{p},0}$) is the unperturbed Kohn-Sham matrix and $\boldsymbol{\mu}^{\mathrm{e}}$ (or $\boldsymbol{\mu}^{\mathrm{p}}$) is the dipole moment matrix for the electrons (or protons). Note that the dipole moment (*i.e.*, $\boldsymbol{\mu}^{\mathrm{e}}$ and $\boldsymbol{\mu}^{\mathrm{p}}$) have the opposite sign for electrons and protons. Tunable parameters for **FIELD_TYPE** are provided in the additional keywords found below. For the supported field types below, note that the electric field vector $\boldsymbol{\mathcal{E}} = (\mathcal{E}_x, \mathcal{E}_y, \mathcal{E}_z)$ is a quantity whose magnitude is controlled by **FIELD_AMP** and whose direction is controlled by **FIELD_DIRECTION**.

1. **Delta** simulates a Dirac δ-function kick with a field that is turned on only at time $t_0 = 0$:

$$\boldsymbol{\mathcal{E}}(t) = \mathbf{A}_0 \delta\left(t - t_0\right) \tag{13.63}$$

   - The amplitude of $\mathbf{A}_0$ and its direction are set using **FIELD_AMP** and **FIELD_DIRECTION**, respectively.

2. **Gaussian** simulates the following impulse field that is turned on at all times $t \geq 0$:

$$\boldsymbol{\mathcal{E}}(t) = \mathbf{A}_0 \exp\left(\frac{(t - t_{\mathrm{peak}})^2}{\tau^2}\right) \sin(\omega t) \tag{13.64}$$

   - The amplitude of $\mathbf{A}_0$ and its direction are set using **FIELD_AMP** and **FIELD_DIRECTION**, respectively.
   - The center of the pulse t_{peak} is set using the **FIELD_PEAK** keyword.
   - The pulse half-width τ is set using the **FIELD_TAU** keyword.
   - The pulse frequency ω is set using the **FIELD_FREQUENCY** keyword.

FIELD_TYPE

The external applied field

INPUT SECTION: *$tdks*

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

DELTA	δ-function kick
GAUSSIAN	Impulse field (Gaussian envelope)
NONE	No field

RECOMMENDATION:

No recommendation.

FIELD_AMP

The amplitude of the external field, in atomic units.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0.01

OPTIONS:

NONE

RECOMMENDATION:

No recommendation.

FIELD_PEAK

The peak position t_{peak} (in atomic units of time) for the Gaussian impulse field.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

0.0

OPTIONS:

NONE

RECOMMENDATION:

No recommendation.

FIELD_TAU

The value of τ (in atomic units of time) for the Gaussian impulse field.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

800.0

OPTIONS:

NONE

RECOMMENDATION:

No recommendation.

FIELD_FREQUENCY

The frequency ω of the external field, in eV units.

INPUT SECTION: *$tdks*

TYPE:

DOUBLE

DEFAULT:

6.0

OPTIONS:

NONE

RECOMMENDATION:

No recommendation.

FIELD_DIRECTION

The direction of the external applied field vector.

INPUT SECTION: *$tdks*

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

XYZ	Pulse along the direction {1, 1, 1}
X	Pulse along the direction {1, 0, 0}
Y	Pulse along the direction {0, 1, 0}
Z	Pulse along the direction {0, 0, 1}

NONE

RECOMMENDATION:

No recommendation.

FIELD_PARTICLE

The subsystem on which to apply the external field perturbation.

INPUT SECTION: *$tdks*

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

BOTH Electrons and quantized protons

ELECTRONS

PROTONS

RECOMMENDATION:

No recommendation.

13.5.2.8 NEO-DFT(V)

By quantizing select nuclei, NEO gives rise to a potential energy surface with fewer nuclear degrees of freedom, which prevents a direct calculation of the vibrational frequencies of the entire molecule. Consequently, diagonalization of a coordinate Hessian in the NEO framework yields vibrational frequencies and accompanying normal modes of only the classical nuclei, with the quantum nuclei responding instantaneously to the motion of the classical nuclei.[76] Although the fundamental anharmonic vibrational frequencies of the quantum nuclei can be accurately obtained through LR-NEO-TDDFT,[22] the couplings between the vibrations of the classical and quantum nuclei are missing. To obtain the fully coupled molecular vibrations, an effective strategy denoted NEO-DFT(V) was developed.[23,98] The NEO-DFT(V) method has been shown to incorporate key anharmonic effects in full molecular vibrational analyses and to produce accurate molecular vibrational frequencies compared to experiments.[23,98]

The NEO-DFT(V) method involves diagonalization of an extended NEO Hessian composed of partial second derivatives of the coordinates of the classical nuclei ($\mathbf{r}_c$) and the expectation values of the quantum nuclei ($\mathbf{r}_q$). This extended Hessian matrix is composed of three sub-matrices: $\mathbf{H}_0 = (\partial^2 E/\partial \mathbf{r}_c^2)|_{\mathbf{r}_c=\mathbf{r}_q}$, $\mathbf{H}_1 = \partial^2 E/\partial \mathbf{r}_q \partial \mathbf{r}_c$, and $\mathbf{H}_2 = \partial^2 E/\partial \mathbf{r}_q^2$, where in each case, all other coordinates of the classical nuclei and expectation values of the quantum nuclei are fixed. The extended Hessian has the following structure:

$$\mathbf{H}_{\text{DFT(V)}} = \begin{bmatrix} \mathbf{H}_0 & \mathbf{H}_1^\top \\ \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix} \tag{13.65}$$

where

$$\mathbf{H}_2 = \mathbf{U}^\dagger \mathbf{\Omega} \mathbf{M} \mathbf{U} \tag{13.66a}$$

$$\mathbf{H}_1 = -\mathbf{H}_2 \mathbf{R} \tag{13.66b}$$

$$\mathbf{H}_0 = \mathbf{H}_{\text{NEO}} + \mathbf{R}^\top \mathbf{H}_2 \mathbf{R} \,. \tag{13.66c}$$

The quantity $\mathbf{R} = d\mathbf{r}_q/d\mathbf{r}_c$ and the NEO Hessian matrix is $\mathbf{H}_{\text{NEO}} = d^2E/d\mathbf{r}_c^2$ (without the constraint that the expectation values of the quantum nuclei are fixed). In the expression for the $\mathbf{H}_2$ matrix, $\mathbf{M}$ is the diagonal mass matrix, and $\mathbf{\Omega}$ is the diagonal matrix with elements corresponding to the squares of the LR-NEO-TDDFT fundamental vibrational frequencies.[98] $\mathbf{U}$ is a unitary matrix that transforms $\mathbf{\Omega}$ to the target coordinate system and is approximated with the transition dipole moment vectors afforded by a LR-NEO-TDDFT calculation.[23]

Diagonalization of $\mathbf{H}_{\text{DFT(V)}}$ produces the fully coupled molecular vibrational frequencies including anharmonic effects associated with the quantum protons. NEO-SCF(V) calculations can be invoked by setting NEO_SCFV = TRUE:

NEO_SCFV

Enable a NEO-SCFV calculation

TYPE:

INTEGER

DEFAULT:

0 No NEO-SCFV calculation.

OPTIONS:

1 Enable a NEO-SCFV calculation.

0 Disable a NEO-SCFV calculation.

RECOMMENDATION:

None.

In addition, the variables for NEO-HF 13.5.2.1 (or NEO-DFT 13.5.2.2) should be included, and JOBTYPE should be set to FREQ. Furthermore, the NEO-HF(V) method involves building the extended NEO-Hessian based on the NEO-HF Hessian and inputs from NEO-TDHF, and therefore the variables specific to LR-NEO-TDHF (TDDFT) (13.5.2.6) can be also be used to tune the calculation.

13.5.2.9 NEO-CC

An alternative route for inclusion of correlation effects between quantum particles (*i.e.*, electrons and protons) is with wave functions methods that are systematically improvable and parameter-free.[65,89] Among the various developed multicomponent wave function methods, the NEO coupled cluster (NEO-CC) methods have been particularly successful.[63,64,66,67] The NEO-CC wave function is given by

$$|\Psi_{\text{NEO-CC}}\rangle = e^{\hat{T}}|0^{\text{e}}0^{\text{p}}\rangle, \tag{13.67}$$

where $\hat{T}$ is the cluster operator that incorporates the correlation effects between quantum particles, and $|0^{\text{e}}0^{\text{p}}\rangle$ is the NEO-HF reference wave function. In the NEO-CCSD method,[64] the cluster operator is given by

$$\hat{T} = \hat{T}_1 + \hat{T}_2 = t_a^i a_i^a + t_A^I a_I^A + \frac{1}{4} t_{ab}^{ij} a_{ij}^{ab} + \frac{1}{4} t_{AB}^{IJ} a_{IJ}^{AB} + t_{aA}^{iI} a_{iI}^{aA} = \sum_\alpha t_\alpha a^\alpha, \tag{13.68}$$

where $a^\alpha = a_\alpha^\dagger = \{a_i^a, a_I^A, a_{ij}^{ab}, a_{IJ}^{AB}, a_{iI}^{aA}\}$ are the excitation operators expressed in terms of creation/annihilation ($a_p^\dagger/a_p$) fermionic operators, and α is the excitation rank. Here, the $i, j, \ldots$ indices denote occupied electronic orbitals, the $a, b, \ldots$ indices denote unoccupied electronic orbitals, and the $p, q, \ldots$ indices denote general electronic orbitals. The protonic orbitals are denoted analogously using the capitalized indices. The unknown t_α wave function parameters (amplitudes) are determined by solving the set of nonlinear equations for each α:[64]

$$\langle 0^{\text{e}}0^{\text{p}}|a_\alpha e^{-\hat{T}_1-\hat{T}_2}\hat{H}_{\text{NEO}}e^{\hat{T}_1+\hat{T}_2}|0^{\text{e}}0^{\text{p}}\rangle = 0\ . \tag{13.69}$$

In this equation, $H_{\text{NEO}} = h_q^p a_p^q + \frac{1}{2} g_{rs}^{pq} a_{pq}^{rs} + h_Q^P a_p^q + \frac{1}{2} g_{RS}^{PQ} a_{PQ}^{RS} - g_{qQ}^{pP} a_{pP}^{qQ}$ is the second-quantized NEO Hamiltonian, where $h_q^p = \langle q|\hat{h}^{\text{e}}|p\rangle$ and $g_{rs}^{pq} = \langle rs|pq\rangle$ are conventional electronic core Hamiltonian and two-electron integrals, respectively. The remaining protonic (h_Q^P and g_{RS}^{PQ}) and electron-proton (g_{qQ}^{pP}) integrals are defined analogously. Lastly, the NEO-CCSD energy is calculated from

$$E_{\text{NEO-CCSD}} = \left\langle 0^{\text{e}}0^{\text{p}}\middle|e^{-\hat{T}_1-\hat{T}_2}\hat{H}_{\text{NEO}}e^{\hat{T}_1+\hat{T}_2}\middle|0^{\text{e}}0^{\text{p}}\right\rangle . \tag{13.70}$$

To increase the computational efficiency and reduce the memory requirements for the NEO-CCSD method, the two-particle integrals can be approximated with the density fitting (DF) approximation,[67] in which the two-particle four-center integrals are factorized into a sum of products of three-center and two-center two-particle integrals. In particular,

the four-center two-electron integrals are approximated by

$$(\mu\nu|\rho\sigma) = \langle\mu\rho|\nu\sigma\rangle \approx \sum_{XY}(\mu\nu|X)(X|Y)^{-1}(Y|\rho\sigma) \;, \tag{13.71}$$

where $(\mu\nu|X)$ and $(X|Y)$ are three-center and two-center two-electron integrals, respectively. In this equation, $\mu, \nu, \ldots$ and $X, Y, \ldots$ indices denote electronic and auxiliary electronic basis functions, respectively. The four-center two-proton integrals are approximated analogously by

$$(\mu'\nu'|\rho'\sigma') = \langle\mu'\rho'|\nu'\sigma'\rangle \approx \sum_{X'Y'}(\mu'\nu'|X')(X'|Y')^{-1}(Y'|\rho'\sigma') \;, \tag{13.72}$$

where primed indices denote protonic basis functions and $(\mu'\nu'|X)$ and $(X'|Y')$ are three-center and two-center two-proton integrals, respectively. Finally, the four-center electron-proton integrals are approximated as

$$(\mu\nu|\mu'\nu') = \langle\mu\mu'|\nu\nu'\rangle \approx \sum_{X'Y'}(\mu\nu|X')(X'|Y')^{-1}(Y'|\mu'\nu'). \tag{13.73}$$

By employing the DF approximation, the memory requirements for storing four-center two-particle integrals are reduced from N_{bf}^4 to $N_{\text{bf}}^2 \times N_{\text{aux}}$, where N_{bf} and N_{aux} are the number of electronic or protonic basis functions and auxiliary basis functions, respectively.[67]

Currently only JOBTYPE = SP is supported. NEO-CC is activated by the following keyword in addition to the variables for NEO-HF (NEO = TRUE and METHOD = HF):

NEO_RICCSD

Enable a NEO-RICCSD calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Enable this option.

0 Disable this option.

RECOMMENDATION:

Both electronic and protonic auxiliary basis sets must be specified.

The following additional *$rem* variables can also be used to customize the NEO-RICCSD calculation (also see the examples in Section 13.5.6):

NEO_CCSD_MAX_CYCLES

Controls the maximum number of CC iterations permitted.

TYPE:

INTEGER

DEFAULT:

5000

OPTIONS:

n Set the maximum number of iterations to $n > 0$.

RECOMMENDATION:

None

NEO_CCSD_CONVERGENCE

NEO-RICCSD is considered converged when the energy error is less than $10^{-\text{NEO\_CCSD\_CONVERGENCE}}$.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

User-defined

RECOMMENDATION:

None

13.5.2.10 NEO-MP2

Second-order Møller-Plesset perturbation theory provides a useful framework for wave function based correlation effects with lower cost than NEO-CC[82]. The NEO-MP2 correlation energy is composed of the electron-electron, proton-proton and electron-proton contributions:

$$E^{(2)}_{\text{NEO}} = E^{(2)}_{\text{ee}} + E^{(2)}_{\text{pp}} + E^{(2)}_{\text{ep}}, \tag{13.74}$$

where

$$E^{(2)}_{\text{ee}} = \frac{1}{4}\sum_{ijab}\frac{|\langle ij|ab\rangle - \langle ij|ba\rangle|^2}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \tag{13.75a}$$

$$E^{(2)}_{\text{pp}} = \frac{1}{4}\sum_{IJAB}\frac{|\langle IJ|AB\rangle - \langle IJ|BA\rangle|^2}{\varepsilon_I + \varepsilon_J - \varepsilon_A - \varepsilon_B} \tag{13.75b}$$

$$E^{(2)}_{\text{ep}} = \sum_{iIaA}\frac{|\langle iI|aA\rangle|^2}{\varepsilon_i + \varepsilon_I - \varepsilon_a - \varepsilon_A}\,. \tag{13.75c}$$

Here, i and j are occupied electronic spin orbitals with energies ε_i and ε_j, a and b are virtual (unoccupied) electronic spin orbitals with energies ε_a and ε_b, and uppercase indices indicate the analogous protonic spin orbitals and energies.

To provide accuracy that is competitive with NEO-CCSD, NEO-MP2 has been extended to include orbital optimization and empirical spin-component and electron-proton correlation scaling.[66,67] This leads to the NEO equivalents of SOS-OOMP2 (O2)[56] and SOS-MP2[34,43] (Sections 6.6.5 and 6.6.6). The NEO equivalents are referred to as NEO-SOS′-OOMP2 and NEO-SOS′-MP2 to indicate the inclusion of an electron-proton correlation scaling factor:

$$E^{(2)}_{\text{NEO}} = c_{\text{ss}}E^{\text{ss}(2)}_{\text{ee}} + c_{\text{os}}E^{\text{os}(2)}_{\text{ee}} + c_{\text{ep}}E^{(2)}_{\text{ep}} + E^{(2)}_{\text{pp}}, \tag{13.76}$$

where $E^{\text{ss}(2)}_{\text{ee}}$ and $E^{\text{os}(2)}_{\text{ee}}$ are the same- and opposite-spin parts of the electron-electron correlation energy, which are scaled by c_{ss} and c_{os}, respectively, and c_{ep} is the electron-proton scaling factor.

Orbital optimization is performed by minimizing the NEO-MP2 energy with respect to the unitary operator $e^{\hat{X}-\hat{X}^\dagger}$ formed from the rotation operator $\hat{X} = \hat{X}^{\text{e}} + \hat{X}^{\text{p}} = x^i_a a^a_i + x^I_A a^A_I$, with $\mathbf{x} = \{x^i_a, x^I_A\}$ being the set of unknown electronic and protonic orbital rotation parameters. Optimal orbital rotation parameters are solved for by finding the stationary points of electronic and protonic orbital gradients $\mathbf{w}_{\text{e}}$ and $\mathbf{w}_{\text{p}}$, which have elements

$$(\mathbf{w}_{\text{e}})^i_a = \frac{\partial E_{\text{NEO-MP2}}(\mathbf{x})}{\partial x^a_i}\bigg|_{\mathbf{x}_{\text{e}}=0} \tag{13.77a}$$

and

$$(\mathbf{w}_{\text{p}})^I_A = \frac{\partial E_{\text{NEO-MP2}}(\mathbf{x})}{\partial x^A_I}\bigg|_{\mathbf{x}_{\text{p}}=0}, \tag{13.77b}$$

and are solved in an alternating fashion until self-consistency is achieved. For computational efficiency, all NEO-MP2 methods are implemented with the density fitting approximation for two-particle integrals,[29] as described in more detail in Section 13.5.2.9.

Currently only JOBTYPE = SP is supported. NEO-MP2 methods are invoked with NEO_RIMP2 = 1 to run without orbital optimization and NEO_RIMP2 = 2 to run with orbital optimization. Additionally, the variables for NEO-HF (NEO = TRUE and METHOD = HF) should be included:

NEO_RIMP2

Enable a NEO-MP2 or NEO-OOMP2 calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

2 Perform a NEO-OOMP2 calclulation.

1 Perform a NEO-MP2 calculation.

0 Disable this option.

RECOMMENDATION:

Both electronic and protonic auxiliary basis sets must be specified.

Spin-component scaling settings are controlled with the SCS variable. Custom c_{ss} and c_{os} can be set with SSS_FACTOR and SOS_FACTOR, respectively, when SCS = 3 is set (arbitrary scaling). By default, c_{ep} is set to the same value as c_{os}, but a custom electron-proton scaling factor can be input with the EP_FACTOR variable, which will always override the default set by SCS. These settings are as follows (also see the examples in Section 13.5.6):

SCS

Set the type of spin-component scaling.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Turns on spin-component scaling with SCS ($c_{\mathrm{ss}} = 0.33$, $c_{\mathrm{os}} = 1.2$).

2 Turns on spin-component scaling with SOS ($c_{\mathrm{ss}} = 0.0$, $c_{\mathrm{os}} = 1.2$ for MP2, $c_{\mathrm{os}} = 1.3$ for OOMP2).

3 arbitrary SCS (set with SSS_FACTOR and SOS_FACTOR).

0 no spin-component scaling.

RECOMMENDATION:

NONE

SSS_FACTOR

Controls the strength of the same-spin component of the MP2 electron-electron correlation energy.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n Corresponding to $c_{\mathrm{ss}} = n/10^6$.

RECOMMENDATION:

NONE

SOS_FACTOR

Controls the strength of the opposite-spin component of the MP2 electron-electron correlation energy.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n Corresponding to $c_{\mathrm{os}} = n/10^6$.

RECOMMENDATION:

NONE

EP_FACTOR

Controls the strength of the electron-proton component of the NEO-MP2 correlation energy.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n Corresponding to $c_{\mathrm{ep}} = n/10^6$.

RECOMMENDATION:

NONE

13.5.2.11 NEO Configuration Interaction

The NEO Configuration Interaction (NEO-CI) wavefunction is a linear combination of NEO configurations:

$$\Psi^{\mathrm{NEO}}(\mathbf{R}^{\mathrm{c}}) = \sum_{iI} c_{iI} |\Phi_i^{\mathrm{e}}\rangle |\Phi_I^{\mathrm{n}}\rangle \tag{13.78}$$

where $|\Phi_i^{\mathrm{e}}\rangle$ is electronic determinant i and $|\Phi_I^{\mathrm{n}}\rangle$ is nuclear determinant I. The electronic determinant is composed of electronic orbitals and the nuclear determinant is composed of protonic orbitals.[58]

The NEO-CI wavefunction is almost always truncated to include only single, or single and double excitations into the virtual orbital space. In the single-reference case, the electronic and nuclear determinants use the NEO-HF electronic and nuclear orbitals, respectively. In the multi-reference case, the electronic and nuclear orbitals are optimized via NEO-MCSCF. The orbitals can be optimized for multiple states with state-averaging.

Available models include:

- NEO-MCSCF(CASSCF,RASSCF)
- NEO-CI
- NEO-MR-CI
- NEO-OOMRCI

In addition to setting NEO = TRUE, METHOD = HF, and JOBTYPE = SP, the job settings for these available models are controlled by the following *$rem* variables:

Note: Wavefunction Keywords

NEO_WF_N_ELECS
Total number of electrons in wavefunction.
TYPE:
INTEGER
DEFAULT:
(Read from $molecule input.)
OPTIONS:
User-defined
RECOMMENDATION:
NONE

NEO_WF_E_FROZEN_CORE
Number of electronic frozen core orbitals. Orbitals are doubly occupied in all configurations.
TYPE:
INTEGER
DEFAULT:
0
OPTIONS:
User-defined
RECOMMENDATION:
Generally, 1s orbitals can be treated as frozen core orbitals.

NEO_WF_E_FROZEN_VIRT
Number of electronic frozen virtual orbitals. These orbitals are unoccupied in all configurations.
TYPE:
INTEGER
DEFAULT:
0
OPTIONS:
User-defined
RECOMMENDATION:
Virtual orbitals contribute to the dynamic correlation recovered by CI calculations,
so it is not recommended that this value is set, except in special cases.

NEO_WF_E_DOCC

Number of electronic *reference* doubly-occupied orbitals *minus* the CI frozen core orbitals. There are **NEO_WF_E_XLEVEL** excitations out of these orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_RAS

Number of electronic *reference* restricted active space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_RAS_NHOLES

Maximum Number of electronic *reference* excitations out of the RAS orbitals and into the CAS and AUX orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_CAS

Number of electronic *reference* complete active space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_AUX

Number of electronic *reference* auxilliary space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_AUX_NELECS

Maximum number of electronic *reference* excitations into the AUX orbitals from the RAS and AUX orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_XLEVEL

Electronic excitation level of NEO-CI expansion: 1 = single, 2 = single and double, etc.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_N_FROZEN_VIRT

Number of nuclear frozen virtual orbitals. These orbitals are unoccupied in all configurations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

Virtual orbitals contribute to the dynamic correlation recovered by CI calculations, so it is not recommended that this value is set, except in special cases.

NEO_WF_N_DOCC

Number of nuclear *reference* doubly-occupied orbitals.

There are **NEO_WF_N_XLEVEL** excitations out of these orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_N_CAS

Number of nuclear *reference* complete active space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_N_XLEVEL

Nuclear excitation level of NEO-CI expansion: 1 = single, 2 = single and double, etc.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

Note: NEO-MRCI Keywords

NEO_MRCI

Perform NEO Multireference Configuration Interaction

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Run a NEO-MRCI Calculation.

FALSE Do not run a NEO-MRCI Calculation.

RECOMMENDATION:

NONE

NEO_MRCI_MAX_ITER

Maximum NEO CI iterations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_MRCI_NUM_ROOTS

Number of NEO CI states to converge.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

User-defined

RECOMMENDATION:

Due to the spectrum and structure of the NEO CI Hamiltonian, it is advisable to set NEO_MRCI_NUM_ROOTS to be greater than the number of desired roots.

NEO_MRCI_KSPACE_MIN

Minimum size of Krylov subspace.

TYPE:

INTEGER

DEFAULT:

NEO_MRCI_NUM_ROOTS + 1

OPTIONS:

User-defined

RECOMMENDATION:

Increasing the number of guess vectors improves CI convergence but has increased memory costs.

NEO_MRCI_KSPACE_MAX

Maximum size of Krylov subspace.

TYPE:

INTEGER

DEFAULT:

NEO_MRCI_KSPACE_MIN + 5

OPTIONS:

User-defined

RECOMMENDATION:

Increasing the number of guess vectors before contraction improves CI convergence but has increased memory costs.

NEO_MRCI_PREDIAG_ROUTINE

Choose how initial guess vectors are created.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Linear combination of lowest NEO_MRCI_PREDIAG_DIM NEO-CI diagonal elements.

1 Eigenvectors of NEO_MRCI_PREDIAG_DIM x NEO_MRCI_PREDIAG_DIM block of NEO-CI Hamiltonian.

2 MCSCF vectors are used for initial guess.

RECOMMENDATION:

0

NEO_MRCI_PREDIAG_DIM

Dimension for NEO_MRCI_PREDIAG_ROUTINE

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

User-defined

RECOMMENDATION:

The larger the size, the better the initial guess.

NEO_MRCI_CONV_THRESH

Convergence tolerance for roots of NEO-CI. It is defined as $10^{-\text{NEO\_MRCI\_CONV\_THRESH}}$

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

4 Sufficient for energies.

5 Gradients and couplings.

RECOMMENDATION:

4

NEO_MRCI_PRINT_CIVECS

Print the final NEO CI vectors to civec.dat.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MRCI_READ_CIVECS

Read in NEO CI vectors from civec.dat.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

User-defined

RECOMMENDATION:

This can be used to restart from a previously unconverged CI calculation.

NEO_MRCI_READ_MOS

Read in MO coefficients from mocoefs.electronic and mocoefs.protonic.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MRCI_PRINT_LEVEL

Determines print level for NEO CI calculation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 No convergence information is printed. Only iteration number and timings.

1 Convergence information is printed.

>1 Detailed printing for debugging.

RECOMMENDATION:

1

NEO_MRCI_CIVEC_ANALYSIS

Perform brief analysis for CI vectors, largest contributing configurations.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE No analysis is performed.

TRUE Analysis is performed.

RECOMMENDATION:

FALSE

NEO_MRCI_CIDEN_PRINT

Print the NEO CI protonic densities of the converged roots to .cube files

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print protonic densities.

TRUE Print protonic densities.

RECOMMENDATION:

None

Note: NEO-MCSCF Keywords

NEO_MCSCF

Run a NEO-MCSCF calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not run a NEO-MCSCF.

TRUE Run a NEO-MCSCF.

RECOMMENDATION:

None

NEO_MCSCF_DEBUG

Run NEO-MCSCF in debug mode.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not debug NEO-MCSCF.

1 Debug NEO-MCSCF

RECOMMENDATION:

0

NEO_MCSCF_MACRO_MAX_ITER

Maximum number of NEO-MCSCF macroiterations.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

RECOMMENDATION:

Most CAS calculations should require few macroiterations. More complex RASSCF wavefunctions may require more iterations as convergence can be slow.

NEO_MCSCF_MACRO_CONV_THRESH
Orbital gradient convergence tolerance for NEO-MCSCF. Defined as $10^{-\text{NEO\_MCSCF\_MACRO\_CONV\_THRESH}}$.
TYPE:
INTEGER
DEFAULT:
5
OPTIONS:
User-defined
RECOMMENDATION:
5

NEO_MCSCF_MICRO_MAX_ITER
Maximum NEO-CI iterations.
TYPE:
INTEGER
DEFAULT:
1000
OPTIONS:
User-defined
RECOMMENDATION:
Set as high as necessary to converge all roots required.

$\geq$1000

NEO_MCSCF_MICRO_OPT_MAX_ITER
Maximum number of orbital optimizations per macroiteration.
TYPE:
INTEGER
DEFAULT:
5
OPTIONS:
User-defined
RECOMMENDATION:
5

NEO_MCSCF_EMC_CONV_THRESH
Convergence threshold for NEO-MCSCF energy. Defined as $10^{-\text{NEO\_MCSCF\_EMC\_CONV\_THRESH}}$.
TYPE:
INTEGER
DEFAULT:
8
OPTIONS:
User-defined
RECOMMENDATION:
$\geq$8

NEO_MCSCF_NSTATES

Number of NEO-CI roots to compute during microiterations.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_SA

Perform state-averaged NEO-MCSCF.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Perform CASSCF.

TRUE Perform SA-MCSCF

RECOMMENDATION:

None

NEO_MCSCF_SA_NSTATES

Number of states to include in state-averaging.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_PRINT_CIVECS

Print the NEO-MCSCF CI vectors to mcvec.dat.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE No printing.

TRUE MCSCF CI vectors are printed to mcvec.dat and wavefunction is printed to mcwf.dat.

RECOMMENDATION:

None

NEO_MCSCF_READ_MOS

Read MCSCF MO coefficients from mocoefs.electronic and mocoefs.protonic.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE NEO-HF MO coefficients used for initial guess.

TRUE MO coefficients from files used as initial guess.

RECOMMENDATION:

None

NEO_MCSCF_PRINT_MOS

Print MCSCF MO coefficients to mocoefs.electronic.new and mocoefs.protonic.new.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Print MO coefficients.

FALSE MO coefficients are not printed.

RECOMMENDATION:

TRUE

NEO_MCSCF_PRINT_LEVEL

Print level for NEO-MCSCF.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Only convergence information is printed.

1 Macro and micro iteration information is printed.

≥ 2 Detailed printing.

RECOMMENDATION:

1

NEO_MCSCF_OPT_METHOD

Orbital optimization method for NEO-MCSCF.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 GDM macroiterations. No microiterations.

1 Augmented Hessian macroiterations. Augmented Hessian microiterations.

2 Augmented Hessian macroiterations. GDM microiterations.

RECOMMENDATION:

2

NEO_MCSCF_KSPACE_MIN

Minimum size of the Krylov subspace for NEO-CI microiterations.

TYPE:

INTEGER

DEFAULT:

NEO_MCSCF_NSTATES + 2

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_KSPACE_MAX

Maximum size of the Krylov subspace for NEO-CI microiterations.

TYPE:

INTEGER

DEFAULT:

NEO_MCSCF_KSPACE_MIN + 10

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_CIDEN_PRINT

Print the CI protonic density for MCSCF states to *.cube files.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print density.

TRUE Print density.

RECOMMENDATION:

None

NEO_MCSCF_REORDER_MOS

Reorder initial guess molecular orbitals. Requires reorder_mo.txt and reorder_nuc_mo.txt.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Orbitals are not reordered.

TRUE Orbitals are reordered.

RECOMMENDATION:

None

13.5.3 Additional Customization to NEO Models

Additional customization to the NEO Methods in the previous section can also be invoked. Table 13.1 shows the current customization options we have made available, and Table 13.2 shows the combination of NEO Methods upon which these features have been tested.

Feature	Single-point Energies	Analytical Gradients	Analytical Hessian
Implicit Solvation	✓	✓	✓
Electrostatic Embedding	✓	✓	–
Empirical Dispersion Correction	✓	✓	✓
Pseudopotentials	✓	✓	✓

Table 13.1: Features that have been extended to the NEO framework.

Level of Theory (Algorithm)	Implicit Solvation	Electrostatic Embedding	Empirical Dispersion Correction	Pseudopotentials
NEO-HF/DFT	✓	✓	✓	✓
NEO-MSDFT	–	–	–	–
CNEO	✓	✓	✓	✓
NEO Hartree Product	✓	✓	✓	✓
LR-NEO-TDHF/TDDFT	–	–	–	–
RT-NEO-TDHF/TDDFT	–	✓	✓	✓
NEO-SCF(V)	–	–	–	–
NEO-CC	–	–	–	–
NEO-MP2	–	–	–	–
NEO-CI	–	–	–	–

Table 13.2: NEO method compatibility. Check mark indicates current support.

13.5.3.1 Implicit Solvation

Bulk solvent effects can be directly incorporated into NEO calculations through the application of various implicit solvation models (Section 11.2) within the NEO framework.[91] The polarizable continuum model (PCM) constitutes one family of implicit solvation models and itself encompasses several different formulations:[40,47] C-PCM,[5,87] IEF-PCM,[14,18] SS(V)PE,[18] etc.[40] In the PCM approach, the solute molecule is placed in a cavity that is embedded in dielectric continuum solvent, and the cavity surface is discretized into i tesserae grid points. The solvent response is represented by a partial charge q_i centered at each tesserae grid point $\mathbf{s}_i$.[40,86]

For each SCF iteration, the current electronic and protonic densities, together with the fixed classical nuclei, define the solute's charge distribution. This charge distribution gives rise to the solute's electrostatic potential V_i at each tesserae grid point:

$$V_i = \sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{s}_i|} - \sum_{\mu\nu} P^{\mathrm{e}}_{\mu\nu} \int d\mathbf{r}_{\mathrm{e}} \frac{\phi^{\mathrm{e}}_\mu(\mathbf{r}_{\mathrm{e}})\,\phi^{\mathrm{e}}_\nu(\mathbf{r}_{\mathrm{e}})}{|\mathbf{r}_{\mathrm{e}} - \mathbf{s}_i|} + \sum_{\mu'\nu'} P^{\mathrm{p}}_{\mu'\nu'} \int d\mathbf{r}_{\mathrm{p}} \frac{\phi^{\mathrm{p}}_{\mu'}(\mathbf{r}_{\mathrm{p}})\,\phi^{\mathrm{p}}_{\nu'}(\mathbf{r}_{\mathrm{p}})}{|\mathbf{r}_{\mathrm{p}} - \mathbf{s}_i|} \,. \tag{13.79}$$

The solute electrostatic potential is then used to compute q_i using standard PCM methods. Once obtained, the set of tesserae charges is included as an additional one-electron (one-proton) contribution to the electronic (protonic) Fock or

analogous Kohn-Sham matrix:

$$F_{\mu\nu}^{\mathrm{e,solv}} = F_{\mu\nu}^{\mathrm{e},0} - \sum_i q_i \int d\mathbf{r}_\mathrm{e} \frac{\phi_\mu^\mathrm{e}(\mathbf{r}_\mathrm{e})\,\phi_\nu^\mathrm{e}(\mathbf{r}_\mathrm{e})}{|\mathbf{r}_\mathrm{e} - \mathbf{s}_i|} \tag{13.80a}$$

$$F_{\mu'\nu'}^{\mathrm{p,solv}} = F_{\mu'\nu'}^{\mathrm{p},0} + \sum_i q_i \int d\mathbf{r}_\mathrm{p} \frac{\phi_{\mu'}^\mathrm{p}(\mathbf{r}_\mathrm{p})\,\phi_{\nu'}^\mathrm{p}(\mathbf{r}_\mathrm{p})}{|\mathbf{r}_\mathrm{p} - \mathbf{s}_i|}\,, \tag{13.80b}$$

where $F_{\mu\nu}^{\mathrm{e},0}$ and $F_{\mu'\nu'}^{\mathrm{p},0}$ refer to the gas-phase, electronic [Eq. (13.37a)] and protonic [Eq. (13.37b)] Fock or analogous Kohn-Sham matrix elements, respectively.

NEO-PCM calculations involve iterative, self-consistent convergence of the nuclear-electronic wavefunction in the presence of the dielectric continuum solvent. NEO-PCM analytic gradients and analytic Hessians are also implemented. The calculation can be invoked by setting SOLVENT_METHOD = PCM in the *$rem* input section. Non-electrostatic contributions to the solvation free energy can also be included by setting SOLVENT_METHOD = SMD:

SOLVENT_METHOD

Sets the preferred solvent method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0	Do not use a solvation model.
PCM	Use an apparent surface charge, polarizable continuum model (Section 11.2.3).
SMD	Use SMD (Section 11.2.9.3).

RECOMMENDATION:

No recommendation.

The SMD approach builds upon PCM by including non-electrostatic interactions, namely the cavitation, dispersion, and solvent structure energy terms. A caveat is that SMD was originally parameterized to reproduce experimental free energies of solvation for conventional, electronic DFT calculations (i.e., solute nuclei represented as point charges).[21]

In the simplest approach, the cavity surface is discretized into point charges. However, a more sophisticated approach utilizing Gaussian-smeared charges is also supported.[40,45,46,48] Selection of these various PCM schemes and related variables can be set in the following input sections. For NEO-PCM calculations, use the *$pcm* and *$solvent* input sections. For NEO-SMD calculations, use the *$pcm* and *$smx* input sections. The capabilities from Q-CHEM's solvation library have been inherited to work with NEO methods, and the full list of variables that have been tested to work with NEO are listed in the pages that follow. See Section 11.2 for more comprehensive information on these solvation models as well as consult John Herbert's review article[40] to guide your selection. Lastly, examples on setting up a solvated NEO calculation can be found in Section 13.5.6.

The format of the *$pcm* section is analogous to the *$rem* section:

```
$pcm
  <Keyword>  <parameter/option>
$end
```

Note: The following job control variables belong *only* in the *$pcm* section. Do not place them in the *$rem* section.

Theory

Specifies the which polarizable continuum model will be used.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

CPCM

OPTIONS:

CPCM (or CPCM1)	Conductor-like PCM of Ref. 20, with $f_\varepsilon = (\varepsilon - 1)/\varepsilon$.
CPCM2	Original conductor-like screening model of Ref. 44, with $f_\varepsilon = (\varepsilon - 1)/(\varepsilon + 1/2)$.
IEFPCM	IEF-PCM with an asymmetric $\mathbf{K}$ matrix.
SSVPE	SS(V)PE model, equivalent to IEF-PCM with a symmetric $\mathbf{K}$ matrix.

RECOMMENDATION:

We recommend C-PCM with NEO methods. The IEF-PCM model is more exact and better suited for low-dielectric solvents, but analytic Hessians are not supported.

Method

Specifies which surface discretization method will be used.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

SwiG

OPTIONS:

SwiG	Switching/Gaussian method
Spherical	Use a single, fixed sphere for the cavity surface.
Fixed	Use discretization point charges instead of smooth Gaussians.

RECOMMENDATION:

Use the default.

Solver

Specifies the algorithm used to solve the PCM equations.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

INVERSION

OPTIONS:

INVERSION	Direct matrix inversion
CG	Iterative conjugate gradient

RECOMMENDATION:

Use the default.

SwitchThresh

Threshold for discarding grid points on the cavity surface.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n Discard grid points when the switching function is less than 10^{-n}.

RECOMMENDATION:

Use the default.

Radii

Specifies which set of atomic van der Waals radii will be used to define the solute cavity.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

Bondi

OPTIONS:

Bondi Use the (extended) set of Bondi radii (Ref. 75).

FF Use Lennard-Jones radii from a molecular mechanics force field.

UFF Use radii from the Universal Force Field (Ref. 72).

Read Read the atomic radii from a *$van_der_waals* input section.

RECOMMENDATION:

Do not alter this section if doing SMD calculations.

vdwScale

Scaling factor for the atomic van der Waals radii used to define the solute cavity.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

1.2

OPTIONS:

α Use a scaling factor of $\alpha > 0$.

RECOMMENDATION:

Use the default.

SASradius

Form a "solvent accessible" surface with the given solvent probe radius.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

0.0

OPTIONS:

r Use a solvent probe radius of r, in Å.

RECOMMENDATION:

Use the default.

SurfaceType

Selects the solute cavity surface construction.

INPUT SECTION: *$pcm*

TYPE:

STRING

DEFAULT:

VDW_SAS

OPTIONS:

VDW_SAS	van der Waals or solvent-accessible surface
SES	solvent-excluded surface

RECOMMENDATION:

Use the default.

Note: The acceptable values for the number of Lebedev points per sphere are $N = 6$, 26, 38, 50, 86, 110, 146, 170, 194, 302, 350, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, 5294.

HeavyPoints

The number of Lebedev grid points to be placed non-hydrogen atoms in the QM system.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

194

OPTIONS:

Acceptable values are listed above.

RECOMMENDATION:

Use the default for geometry optimizations. For absolute solvation energies, the user may want to examine convergence with respect to N.

HPoints

The number of Lebedev grid points to be placed on H atoms in the QM system.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

110

OPTIONS:

Acceptable values are listed above.

RECOMMENDATION:

Use the default for geometry optimizations. For absolute solvation energies, the user may want to examine convergence with respect to N.

PrintLevel

Controls the printing level during PCM calculations.

INPUT SECTION: *$pcm*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Prints PCM energy and basic surface grid information. Minimal additional printing.

1 Level 0 plus PCM solute-solvent interaction energy components and Gauss' Law error.

2 Level 1 plus surface grid switching parameters and a `.PQR` file for visualization of the cavity surface apparent surface charges.

3 Level 2 plus a `.PQR` file for visualization of the electrostatic potential at the surface grid created by the converged solute.

4 Level 3 plus additional surface grid information, electrostatic potential and apparent surface charges on each SCF cycle.

5 Level 4 plus extensive debugging information.

RECOMMENDATION:

Use the default unless further information is desired.

CavityRadius

Specifies the solute cavity radius.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

None

OPTIONS:

R Use a radius of R, in Ångstroms.

RECOMMENDATION:

None.

CavityCenter

Specifies the center of the spherical solute cavity.

INPUT SECTION: *$pcm*

TYPE:

FLOAT

DEFAULT:

0.0 0.0 0.0

OPTIONS:

$x\ y\ z$ Coordinates of the cavity center, in Ångstroms.

RECOMMENDATION:

The format is **CavityCenter** followed by three floating-point values, delineated by spaces.

Uses the same coordinate system as the *$molecule* section.

The format of the *$solvent* section is analogous to the *$rem* section:

```
$solvent
  <Keyword>  <parameter/option>
$end
```

Note: The following job control variables belong *only* in the *$solvent* section. Do not place them in the *$rem* section.

Dielectric

Specifies the static dielectric constant for the PCM solvent.

INPUT SECTION: *$solvent*

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

ε Use a dielectric constant $\varepsilon > 0$.

RECOMMENDATION:

The static (*i.e.*, zero-frequency) dielectric constant is what is usually called "the" dielectric constant. The default corresponds to water at 25°C.

The format of the *$smx* section is analogous to the *$rem* section:

```
$smx
  <Keyword>  <parameter/option>
$end
```

Note: The following job control variables belong *only* in the *$smx* section. Do not place them in the *$rem* section.

Solvent

Sets the SMx solvent

INPUT SECTION: *$smx*

TYPE:

STRING

DEFAULT:

water

OPTIONS:

Any name from the list of solvents given in Table 11.7.

RECOMMENDATION:

NONE

13.5.3.2 Electrostatic Embedding

A set of external charges can be incorporated into a NEO calculation by specifying the *$external_charges* input section along with setting QM_MM = TRUE in the *$rem* input section. The format is shown below and consists of Cartesian coordinates and the value of the point charge, with one charge per line. The charge is in atomic units and the coordinates are in Ångstroms, unless bohrs are selected by setting the *$rem* keyword INPUT_BOHR to TRUE. The external charges are rotated with the molecule into the standard nuclear orientation and are specified in the following format:

```
$external_charges
   x-coord1  y-coord1  z-coord1  charge1
   x-coord2  y-coord2  z-coord2  charge2
   x-coord3  y-coord3  z-coord3  charge3
$end
```

Note that the electric field and analytic gradients are automatically saved to disk when the *$external_charges* keyword is present. This therefore enables NEO-QM/MM calculations to be performed using Q-CHEM paired an external, molecular mechanics package. When using an external, molecular mechanics driver, the following *$rem* keyword should be set to FALSE to avoid issues with double counting:

SKIP_CHARGE_SELF_INTERACT

Ignores the electrostatic interactions among external charges in a QM/MM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE No electrostatic interactions among external charges.

FALSE Computes the electrostatic interactions among external charges.

RECOMMENDATION:

None

13.5.3.3 Empirical Dispersion Correction

All of the available DFT-D empirical dispersion methods for modeling non-covalent interactions introduced in Section 5.7.3 have been extended to NEO (including analytic gradients and analytic Hessians) and can be requested via the *$rem* variable DFT_D, which is discussed below.

Two caveats are worth pointing out: First, note that the functional form of the DFT-D empirical dispersion potential, $E_{\text{disp}}^{\text{D}}(R_{AB})$, is a sum over all pairwise atomic contributions. For convenience, our implementation of the DFT-D correction uses the positions of the quantum proton(s) basis function center(s) for computing the interatomic distances R_{AB}.[53] Second, the DFT-D dispersion potentials have been parameterized for classical nuclei represented as point charges.

DFT_D

Controls the empirical dispersion correction to be added.

TYPE:

LOGICAL

DEFAULT:

None

OPTIONS:

FALSE	(or 0) Do not apply the DFT-D2, DFT-CHG, or DFT-D3 scheme
EMPIRICAL_GRIMME	DFT-D2 dispersion correction from Grimme[35]
EMPIRICAL_CHG	DFT-CHG dispersion correction from Chai and Head-Gordon[15]
EMPIRICAL_GRIMME3	DFT-D3(0) dispersion correction from Grimme (deprecated as of Q-CHEM 5.0)
D3_ZERO	DFT-D3(0) dispersion correction from Grimme *et al.*[36]
D3S_ZERO	DFT-D3S(0) dispersion correction from Tkachenko and Head-Gordon[85]
D3_BJ	DFT-D3(BJ) dispersion correction from Grimme *et al.*[37]
D3S_BJ	DFT-D3S(BJ) dispersion correction from Tkachenko and Head-Gordon[85]
D3_CSO	DFT-D3(CSO) dispersion correction from Schröder *et al.*[77]
D3_ZEROM	DFT-D3M(0) dispersion correction from Smith *et al.*[80]
D3_BJM	DFT-D3M(BJ) dispersion correction from Smith *et al.*[80]
D3_OP	DFT-D3(op) dispersion correction from Witte *et al.*[92]
D3	Automatically select the "best" available D3 dispersion correction
D4	DFT-D4 dispersion correction from Caldeweyher *et al.*[11–13]

RECOMMENDATION:

Use D4 if the specified functional is avialable. Currently, only a subset of functionals in DFT-D4 is supported. It includes B3LYP, B97, B1LYP, PBE0, PW6B95, M06L, M06, WB97, WB97X, CAMB3LYP, PBE02, PBE0DH, MPW1K, MPWB1K, B1B95, B1PW91, B2GPPLYP, B2PLYP, B3P86, B3PW91, O3LYP, REVPBE, REVPBE0, REVTPSS, REVTPSSH, SCAN, TPSS0, TPSSH, X3LYP, TPSS, BP86, BLYP, BPBE, MPW1PW91, MPW1LYP, PBE, RPBE, and PW91.

Note: The following variable is applicable only to the DFT-CHG dispersion model.

DFT_D_A

Controls the strength of dispersion corrections in the Chai–Head-Gordon DFT-D scheme, Eq. (5.26).

TYPE:

INTEGER

DEFAULT:

600

OPTIONS:

n Corresponding to $a = n/100$.

RECOMMENDATION:

Use the default.

Note: The following input section is only relevant to the DFT-D2 dispersion model. This input section is optional and is only used if the user wants to change the default values recommended by Grimme.

```
$empirical_dispersion
S6 S6_value
D D_value
C6 element_1 C6_value_for_element_1 element_2 C6_value_for_element_2
VDW_RADII element_1 radii_for_element_1 element_2 radii_for_element_2
$end
```

Note: The following variables are applicable only to the DFT-D3 dispersion models.

DFT_D3_S6

The linear parameter s_6 in eq. (5.27). Used in all forms of DFT-D3.

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS6

The nonlinear parameter $s_{r,6}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,6} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_S8

The linear parameter s_8 in Eq. (5.27). Used in DFT-D3(0), DFT-D3(BJ), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_8 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS8

The nonlinear parameter $s_{r,8}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,8} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A1

The nonlinear parameter α_1 in Eqs. (5.29), (5.30), (5.31), and (5.32). Used in DFT-D3(BJ), DFT-D3(CSO), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_1 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A2

The nonlinear parameter α_2 in Eqs. (5.29) and (5.32). Used in DFT-D3(BJ), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_2 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_POWER

The nonlinear parameter β_6 in Eq. (5.32). Used in DFT-D3(op). Must be greater than or equal to 6 to avoid divergence.

TYPE:

INTEGER

DEFAULT:

600000

OPTIONS:

n Corresponding to $\beta_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_3BODY

Controls whether the three-body interaction in Grimme's DFT-D3 method should be applied (see Eq. (14) in Ref. 36).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0)	Do not apply the three-body interaction term
TRUE	Apply the three-body interaction term

RECOMMENDATION:

NONE

Note: The following variables are applicable only to the DFT-D4 dispersion model.

DFT_D4_S6

The linear parameter s_6. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n	Corresponding to $s_6 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S8

The linear parameter s_8. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n	Corresponding to $s_8 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S10

The linear parameter s_{10}. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n	Corresponding to $s_{10} = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_A1

The nonlinear parameter α_1. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_1 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_A2

The nonlinear parameter α_2. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_2 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S9

The linear parameter s_9. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_9 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_WF

Weighting factor for Gaussian weighting.

TYPE:

INTEGER

DEFAULT:

600000000

OPTIONS:

n Corresponding to $wf = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_GA

Charge scaling

TYPE:

INTEGER

DEFAULT:

300000000

OPTIONS:

n Corresponding to $ga = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_GC

Charge scaling

TYPE:

INTEGER

DEFAULT:

200000000

OPTIONS:

n Corresponding to $gc = n/100000000$.

RECOMMENDATION:

Use default

13.5.3.4 Pseudopotentials

Q-CHEM's Effective Core Potential (ECP) package has been integrated with the NEO method to enable the description of relativistic and core electronic effects for systems in which some of the atoms may bear pseudopotentials. This is done by adding an additional one-electron potential contribution, which serves to model the effects of the core electrons, into the electronic Fock or analagous Kohn-Sham matrix (also see Section 8.10). In both the electronic and protonic Fock or analogous Kohn-Sham matrices, the one-electron term corresponding to the interaction of electrons or quantum protons with the classical nucleus with an ECP utilizes the effective nuclear charge shielded by the core electrons.[54] The following *$rem* variable controls which ECP is used:

ECP

Defines the effective core potential and associated basis set to be used

TYPE:

STRING

DEFAULT:

No ECP

OPTIONS:

General, Gen User defined. (*$ecp* keyword required)

Symbol Use standard ECPs discussed above.

RECOMMENDATION:

ECPs are recommended for first row transition metals and heavier elements. Also consult Section 8.10 for more details.

13.5.4 Job Control for the NEO methods

13.5.4.1 General Keywords

The NEO method is a natural extension of the SCF method and it inherits many of its functionalities. Some keywords that are used in the SCF Job Control are used in the NEO methods with a few additional keywords. The following *$rem* variables must be specified in order to run NEO calculations:

Note: Some *$rem* variables, which we find mostly self-explanatory but not discussed below, that also influence NEO jobs in regards to geometry and symmetry considerations include INPUT_BOHR, SYM_IGNORE, and POINT_GROUP_SYMMETRY.

NEO

Enable a NEO-SCF calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Enable a NEO-SCF calculation.

FALSE Disable a NEO-SCF calculation.

RECOMMENDATION:

Set to TRUE if desired.

JOBTYPE

Specifies the type of calculation.

TYPE:

STRING

DEFAULT:

SP

OPTIONS:

SP Single point energy.

FORCE Analytical force calculation.

OPT Geometry minimization.

TS Transition structure search.

FREQ Frequency calculation.

RECOMMENDATION:

Application-dependent. Always use POINT_GROUP_SYMMETRY = FALSE with geometry optimization.

METHOD

Specifies the exchange-correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use METHOD = *NAME*, where *NAME* is one of the following: HF for Hartree-Fock theory; one of the DFT methods listed in Section 5.3.5.;

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations for DFT are indicated in bold in Section 5.3.5.

BASIS

Specifies the electronic basis sets to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen User defined (*$basis* keyword required).

Symbol Use standard basis sets as per Chapter 8.

Mixed Use a mixture of basis sets (see Chapter 8).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

The format for the user-defined basis section is as follows:

```
$basis
    X    0
    L    K    scale
    α1   C1^Lmin   C1^Lmin+1   ...   C1^Lmax
    α2   C2^Lmin   C2^Lmin+1   ...   C2^Lmax
    ⋮    ⋮         ⋮           ⋱     ⋮
    αK   CK^Lmin   CK^Lmin+1   ...   CK^Lmax
****
$end
```

where

X	Atomic symbol of the atom (atomic number not accepted)
L	Angular momentum symbol (S, P, SP, D, F, G)
K	Degree of contraction of the shell (integer)
$scale$	Scaling to be applied to exponents (default is 1.00)
α_i	Gaussian primitive exponent (positive real number)
C_i^L	Contraction coefficient for each angular momentum (non-zero real numbers).

Atoms are terminated with $\star\star\star\star$ and the complete basis set is terminated with the *$end* keyword terminator. No blank lines can be incorporated within the general basis set input. Note that more than one contraction coefficient per line is required for compound shells like SP. As with all Q-CHEM input deck information, all input is case-insensitive.

Additionally, the NEO-SCF methods require definition of the nuclear basis sets. Refer to Ref. 100 for selection of the protonic basis sets. Note that the pure (spherical harmonic) and Cartesian angular forms for the protonic basis sets are

found in the main article and in the supported information, respectively. Also, see details about the NEO_PURECART variable described in the next section.

The format for the user-defined neo_basis section is as follows:

```
$neo_basis
    X     0
    L     K      scale
    α_1   C_1^{L_min}   C_1^{L_min+1}   ...   C_1^{L_max}
    α_2   C_2^{L_min}   C_2^{L_min+1}   ...   C_2^{L_max}
    ⋮     ⋮      ⋮      ⋱     ⋮
    α_K   C_K^{L_min}   C_K^{L_min+1}   ...   C_K^{L_max}
****
$end
```

Note: Several of the post-NEO-HF methods utilizing density fitting also require the nuclear auxiliary basis set to be defined. The format for such user-defined *$neo_aux_basis* section is identical to the *$basis* and *$neo_basis* input sections. See examples in Section 13.5.6 for more details.

13.5.4.2 Additional Keywords

More *$rem* variables that can be used to customize the calculation. Some of these are NEO specific variables, while others are shared with other Q-CHEM modules:

BASIS_LIN_DEP_THRESH

Sets the threshold for determining linear dependence in the basis set

TYPE:

INTEGER

DEFAULT:

6 Corresponding to a threshold of 10^{-6}

OPTIONS:

n Sets the threshold to 10^{-n}

RECOMMENDATION:

Set to 5 or smaller if you have a poorly behaved SCF and you suspect linear dependence in you basis set. Lower values (larger thresholds) may affect the accuracy of the calculation.

NEO_BASIS_LIN_DEP_THRESH

This keyword is used to set the liner dependency threshold for nuclear basis sets. It is defined as $10^{-\text{NEO\_BASIS\_LIN\_DEP\_THRESH}}$.

TYPE:

DOUBLE

DEFAULT:

5.0

OPTIONS:

User-defined

RECOMMENDATION:

No recommendation.

PURECART

INTEGER

TYPE:

Controls the use of pure (spherical harmonic) or Cartesian angular forms

DEFAULT:

1111 Pure h, g, f, d functions

OPTIONS:

$hgfd$ Use 1 for pure and 2 for Cartesian.

RECOMMENDATION:

This is pre-defined for all standard basis sets

NEO_PURECART

This keyword is used to specify Cartesian or spherical Gaussians for nuclear basis functions.

TYPE:

INTEGER

DEFAULT:

2222

OPTIONS:

User-defined

RECOMMENDATION:

The default value corresponds to the use of Cartesian Gaussians for all angular momentum classes. The value NEO_PURECART = 1111 would use spherical Gaussians instead, similar to the use of PURECART.

NEO_ISOTOPE

Enable calculations of different types of isotopes. Only one type of isotope is allowed at present.

TYPE:

INTEGER

DEFAULT:

1 Default is the proton isotope.

OPTIONS:

1 This NEO calculation is using proton isotope.

2 This NEO calculation is using deuterium isotope.

3 This NEO calculation is using tritium isotope.

RECOMMENDATION:

Refer to the NEO literature for the best performance on the isotope effects calculations.

NEO_VPP

This keyword is used to control whether to remove $J - K$ terms from the nuclear Fock matrix and the corresponding kernel terms for NEO excited state methods. The purpose for the removal of these terms in the case of one quantum proton is to save on computational cost. Note that the ground state NEO-HF or NEO-DFT energy is the same with or without removal of these nuclear $J - K$ terms. The nuclear virtual orbitals will however be different, which may affect the results for perturbative methods (i.e., multicomponent excited-state methods and post-NEO-HF wavefunction methods).

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE

OPTIONS:

TRUE (or 1) Enable this option (include nuclear $J - K$ terms).

FALSE (or 0) Disable this option (remove nuclear $J - K$ terms).

RECOMMENDATION:

Set NEO_VPP = 0 only in the case of one quantum hydrogen.

XC_GRID

Specifies the type of grid to use for DFT calculations.

TYPE:

INTEGER

DEFAULT:

Functional-dependent; see Table 5.3.

OPTIONS:

0 Use SG-0 for H, C, N, and O; SG-1 for all other atoms.

n Use SG-n for all atoms, $n = 1, 2$, or 3

XY A string of two six-digit integers X and Y, where X is the number of radial points and Y is the number of angular points where possible numbers of Lebedev angular points, which must be an allowed value from Table 5.2 in Section 5.5.

$-XY$ Similar format for Gauss-Legendre grids, with the six-digit integer X corresponding to the number of radial points and the six-digit integer Y providing the number of Gauss-Legendre angular points, $Y = 2N^2$.

RECOMMENDATION:

We strongly recommend using an unpruned, Euler-Maclaurin-Lebedev grid type for NEO-DFT calculations. Larger grids may be required for optimization and frequency calculations.

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE Closed-shell systems.

TRUE Open-shell systems.

OPTIONS:

FALSE Constrain the spatial part of the alpha and beta orbitals to be the same.

TRUE Do not Constrain the spatial part of the alpha and beta orbitals.

RECOMMENDATION:

The ROHF method is not available. Note that for unrestricted calculations on systems with an even number of electrons it is usually necessary to break α/β symmetry in the initial guess, by using SCF_GUESS_MIX or providing *$occupied* information (see Section 4.4 on initial guesses).

S2THRESH

Cutoff for neglect of overlap integrals, defined via a two-electron shell-pair threshold of $10^{-\mathrm{S2THRESH}}$ (S2THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

Same as THRESH.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

Increase the value of S2THRESH if the program finds negative eigenvalues for the overlap matrix.

THRESH

Cutoff for neglect of two electron integrals. $10^{-\mathrm{THRESH}}$ (THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

8 For single point energies.

10 For optimizations and frequency calculations.

14 For coupled-cluster calculations.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

The value should satisfy THRESH $\geq$ 3 + SCF_CONVERGENCE, although THRESH = 4 + SCF_CONVERGENCE is the default (in most cases) since Q-CHEM v. 6.0. See Ref. 33 for recommended values of THRESH in the presence of diffuse basis functions, where tighter thresholds are often required and too-loose thresholds may lead to slower convergence or convergence failure.

MEM_STATIC

Sets the memory for AO-integral evaluations and their transformations in Q-CHEM 4.1 or older versions.

TYPE:

INTEGER

DEFAULT:

192 corresponding to 192 MB.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use the default.

MEM_TOTAL

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 2 GB

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use the default, or set to the physical memory of your machine. The minimum requirement is $3X^2$.

13.5.4.3 NEO-SCF Specific Keywords

The following *$rem* variables can be used to specify the tolerances for NEO-SCF convergence. Some of these *$rem* variables are inherited from conventional SCF calculations:

SCF_CONVERGENCE

NEO-SCF is considered converged when the electronic wave function error is less that $10^{-\text{SCF\_CONVERGENCE}}$. Adjust the value of THRESH at the same time. (Starting with Q-CHEM 3.0, the DIIS error is measured by the maximum error rather than the RMS error as in earlier versions.)

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations.

8 For geometry optimizations.

OPTIONS:

User-defined

RECOMMENDATION:

None.

NEO_N_SCF_CONVERGENCE

NEO-SCF is considered converged when the nuclear wave function error is less that $10^{-\text{NEO\_N\_SCF\_CONVERGENCE}}$.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

User-defined

RECOMMENDATION:

None.

NEO_E_CONV

Energy convergence criteria in the NEO-SCF calculations so that the difference in energy between consecutive iterations is less than $10^{-\mathrm{NEO\_E\_CONV}}$.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

User-defined

RECOMMENDATION:

Tighter criteria for geometry optimization are recommended. Note that for stepwise calculations, this convergence criteria checks the difference in energy between electronic and protonic microiterations blocks.

MAX_SCF_CYCLES

Controls the maximum number of SCF iterations permitted.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

Increase for slowly converging systems such as those containing transition metals.

NEO_E_MAX_SCF_CYCLES

Controls the maximum number of electronic SCF micoiterations permitted. This variable is only used for stepwise NEO-SCF optimizations.

TYPE:

INTEGER

DEFAULT:

150

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

None.

NEO_N_MAX_SCF_CYCLES

Controls the maximum number of nuclear SCF micoiterations permitted. This variable is only used for stepwise NEO-SCF optimizations.

TYPE:

INTEGER

DEFAULT:

150

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

None.

The NEO-SCF iteration procedure has traditionally involved solving the electronic and quantum nuclear Roothaan/ Kohn-Sham equations in alternating fashion until convergence to the variational solution is reached. This approach can become computationally expensive, especially when many SCF iterations are required to achieve self-consistency. To accelerate this procedure, those equations may instead be solved simultaneously at every step of the NEO-SCF iteration procedure. This is now the default scheme for solving the NEO-SCF equations. We support the simultaneous implementation of the Direct Inversion in the Iterative Subspace (DIIS) algorithm[55] as well as the Geometric Direct Minimization (GDM)[88] approach. The following *$rem* variables permit some customization of the NEO-SCF iterations:

NEO_SIMULTANEOUS_SCF

Enables simultaneous optimization algorithm.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Use the default unless issues with convergence are observed, in which cause set to FALSE to utilize the traditional stepwise approach.

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

DIIS Pulay DIIS.

OPTIONS:

DIIS	Pulay DIIS.
GDM	Geometric Direct Minimization.
DIIS_GDM	Use DIIS and then later switch to geometric direct minimization.

RECOMMENDATION:

In the NEO methods, the GDM procedure is recommended. Note that DIIS_GDM is only supported for simultaneous optimizations.

NEO_SCF_GUESS_N

Specifies the initial guess procedure for the nuclear orbitals in a NEO-SCF calculation.

TYPE:

STRING

DEFAULT:

TIGHT Default unless ghost atoms are present.

CORE Default when there are ghost atoms.

OPTIONS:

CORE Diagonalize the nuclear core Hamiltonian

TIGHT Occupies the least diffuse nuclear MO

AUTOSAD On-the-fly superposition of nuclear atomic densities from an atomic NEO calculation

READ Read previous nuclear MOs from disk

RECOMMENDATION:

Use the default. Note that if NEO_SCF_GUESS_N is set to READ, only the nuclear MOs from a previous NEO-SCF job are read from disk. The starting guess for the electronic MOs in a NEO calculation remains the converged result from the conventional electronic calculation of the current job. To read in both the electronic and nuclear MOs from a previous NEO-SCF job, NEO_SCF_GUESS_N must be set to READ in conjunction with SKIP_SCFMAN set to TRUE.

NEO_STEPWISE_SCF_STEPS

Specifies the number of NEO-SCF stepwise/alternating macro-iterations to perform before switching to simultaneous algorithm.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

The rate of convergence of the NEO-SCF procedure is dependent on the initial guess for the electronic and protonic orbitals. For especially difficult systems, it may be helpful to do stepwise optimization before switching to simultaneous.

Note: The following variables are specific to SCF_ALGORITHM = DIIS.

DIIS_SUBSPACE_SIZE

Controls the size of the DIIS subspace during the SCF.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

User-defined

RECOMMENDATION:

None.

DIIS_ERR_RMS

Changes the DIIS convergence metric from the maximum to the RMS error.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Use the default, the maximum error provides a more reliable criterion.

Note: The following variables are specific to SCF_ALGORITHM = DIIS_GDM.

MAX_DIIS_CYCLES

The maximum number of DIIS iterations before switching to (geometric) direct minimization.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

1 Only a single Roothaan step before switching to (G)DM

n n DIIS iterations before switching to (G)DM.

RECOMMENDATION:

None

THRESH_DIIS_SWITCH

The threshold for switching between DIIS extrapolation and direct minimization of the SCF energy is $10^{-\text{THRESH\_DIIS\_SWITCH}}$.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined.

RECOMMENDATION:

None

13.5.5 Molecular Properties and Wave Function Analysis Tools for NEO Models

Some of Q-CHEM's molecular properties and wave function analysis tools have been extended to NEO methods, which are invoked automatically and reported in the output file:

- Mulliken population analysis for ground state
- Molecular dipole moments and proton position expectation value for ground state
- MOLDEN files for protonic density plot generation from ground state NEO-SCF calculations

- MOLDEN files for protonic transition densities plot generation from excited-state LR-NEO-TDHF (TDDFT) calculations
- Vibrational analysis (including isotopic substitution) using CNEO or NEO-SCF(V) Hessians

Protonic density MOLDEN files are not automatically generated, but can be enabled by specifying the following two *$rem* variables: MAKE_CUBE_FILES = TRUE and PLOTS = TRUE. Specifying these keywords will also generate MOLDEN files for electronic density visualization. The grid parameters used for generating the electronic density can be tuned through the *$plots* input section (refer to Section 10.5.4.1 for details). Note that at this time, the grid for protonic density generation cannot be customized by the user and is instead chosen automatically based on several predefined choices.

13.5.6 Examples

Example 13.7 Input for the NEO-HF calculation on H_2O molecule with the second proton treated quantum mechanically. The electronic basis set is cc-pVDZ and the protonic is an uncontracted 2s2p2d basis set with exponents 4.0 and 8.0.

```
$molecule
   0 1
   H   -3.5008791    1.2736107    0.7596000
   O   -3.9840791    1.3301107   -0.0574000
   H   -4.9109791    1.2967107    0.1521000
$end

$rem
   METHOD   hf
   BASIS    cc-pvdz
   NEO      true
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
S    1    1.000000
   8.0 1.0
P    1    1.000000
   4.0 1.0
P    1    1.000000
   8.0 1.0
D    1    1.000000
   4.0 1.0
D    1    1.000000
   8.0 1.0
****
$end
```

Example 13.8 Input for the NEO-DFT/epc17-2 calculation of CH_2O molecule treating both protons quantum mechanically and using simultaneous DIIS algorithm to perform NEO-SCF. The electronic basis set is cc-pVDZ and the protonic basis set is an uncontracted 1s1p basis set with exponents 4.0 and 4.0. A maximum of 10 previous Fock matrices is used to form the next guess in interpolation/extrapolation, and the NEO wavefunction is considered converged when all three criteria are met (all in atomic units): (1) the largest element of the electronic error vector is below the cutoff threshold of 10^{-8}, (2) the largest element of the protonic error vector is below the cutoff threshold of 10^{-7}, and (3) the energy difference between two consecutive steps is below 10^{-8}.

```
$molecule
0 1
  C        0.000000    0.000000    0.000000
  O        0.000000    0.000000    1.220000
  H        0.935307    0.0000000  -0.5400000
  H       -0.935307    0.000000   -0.540000
$end

$rem
   JOBTYPE              SP
   SYM_IGNORE           TRUE
   INPUT_BOHR           FALSE
   BASIS                cc-pvdz
   NEO                  TRUE
   METHOD               pbe0
   xc_grid = 000099000302
   NEO_EPC              epc172
   NEO_N_SCF_CONVERGENCE = 7
   SCF_CONVERGENCE = 8
   NEO_E_CONV = 8
   SCF_ALGORITHM        DIIS
   NEO_SIMULTANEOUS_SCF TRUE
   NEO_STEPWISE_SCF_STEPS = 2
   DIIS_SUBSPACE_SIZE = 10
   DIIS_ERR_RMS         FALSE
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.9 Input for the NEO-DFT/epc17-2 geometry optimization calculation of all centers on CH_2O molecule with both protons treated quantum mechanically. The electronic exchange-correlation functional is PBE0. The electronic basis set is STO-3G and the protonic is an uncontracted 1s1p basis set with exponents 4.0. This calculation utilizes DFT grid with 99 radial and 302 spherical quadrature points along with the DIIS algorithm.

```
$molecule
   0 1
   C     0.000000    0.000000    0.000000
   O     0.000000    0.000000    1.220000
   H     0.935307    0.000000   -0.540000
   H    -0.935307    0.000000   -0.540000
$end

$rem
   JOBTYPE           OPT
   METHOD            pbe0
   BASIS             sto-3g
   NEO               true
   NEO_EPC           epc172
point_group_symmetry False
   SCF_CONVERGENCE   11
   MAX_SCF_CYCLES    100
   SCF_ALGORITHM     diis
   XC_GRID           000099000302
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.10 Input for the NEO-DFT/epc19 geometry optimization calculation of the NEO center only on open-shell OH radical molecule with a proton treated quantum mechanically. The electronic exchange-correlation functional is PBE0. The electronic basis set is 6-31G and the protonic is an uncontracted 1s1p basis set with exponents 4.0. This calculation utilizes DFT grid with 99 radial and 230 spherical quadrature points along with the DIIS algorithm.

```
$molecule
   0 2
   O    -4.511414   1.264878   0.000000
   H    -2.739325   1.866123   0.000000
$end

$rem
   JOBTYPE           OPT
   METHOD            pbe0
   BASIS             6-31g
   UNRESTRICTED      true
   INPUT_BOHR        true
   NEO               true
point_group_symmetry False
   SCF_CONVERGENCE   6
   MAX_SCF_CYCLES    100
   SCF_ALGORITHM     diis
   NEO_EPC           epc19
   XC_GRID           000099000230
$end

$opt
   FIXED
   1 XYZ
   ENDFIXED
$end

$neo_basis
H    2
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.11 Input for a NEO-MSDFT calculation on FHF^- with a fixed F–F distance of 2.70 Å. The proton in between the two F atoms is quantized and the proton basis function centers are optimized on the NEO-MSDFT ground state. The electronic basis set is STO-3G, and the protonic basis set is an uncontracted 1s1p basis set with exponents 4.0. The electronic exchange-correlation functional is B3LYP, and the electron-proton correlation functional is epc17-2. The second basis function center for the quantum proton is input as a ghost center.

```
$molecule
-1  1
F      0.00000000     0.00000000    -1.35000000
F      0.00000000     0.00000000     1.35000000
H      0.00000000     0.00000000    -0.25220000
@H     0.00000000     0.00000000     0.25220000
$end

$rem
jobtype = opt
input_bohr = false
method = b3lyp
basis = sto-3g
neo = true
neo_epc = epc172
neo_msdft = 1
sym_ignore = 1
max_scf_cycles 500
scf_convergence 6
NEO_N_SCF_CONVERGENCE 6
NEO_E_CONV = 6
SCF_ALGORITHM gdm
$end

$OPT
FIXED
1 XYZ
2 XYZ
ENDFIXED
$END

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.12 Input for a NEO-MSDFT calculation on a fixed average reactant-product structure of the formic acid dimer with a C–C distance of 3.86 Å, where both transferring protons are quantized. The electronic basis set is STO-3G, and the protonic basis set is an uncontracted 1s1p basis set with exponents 4.0. The electronic exchange-correlation functional is B3LYP, and the electron-proton correlation functional is epc17-2. Only the two *trans* states are included in the adiabatic state expansion (note the spaces between the ones and zeros of the *$neo_msdft_diabat_control* section). The second center for each quantum proton is input as a ghost center. Proton density plotting for both the ground and first excited state has been enabled.

```
$molecule
0  1
H         -3.0310572606    0.0000000000    0.0000000000
H         -0.4438453003    1.0873992328    0.0000000000
@H         0.4438452993    1.0873992291    0.0000000000
O         -1.3616953972   -1.1310228162    0.0000000000
O         -1.3616953971    1.1310228163    0.0000000000
C         -1.9296657251    0.0000000000    0.0000000000
H          3.0310572606    0.0000000000    0.0000000000
H          0.4438452993   -1.0873992291    0.0000000000
@H        -0.4438453003   -1.0873992328    0.0000000000
O          1.3616953972    1.1310228162    0.0000000000
O          1.3616953971   -1.1310228163    0.0000000000
C          1.9296657251    0.0000000000    0.0000000000
$end

$rem
jobtype = sp
input_bohr = false
method = b3lyp
basis = sto-3g
neo = true
neo_epc = epc172
neo_msdft = 1
sym_ignore = 1
scf_convergence 6
NEO_N_SCF_CONVERGENCE = 6
NEO_E_CONV = 6
max_scf_cycles 500
SCF_ALGORITHM GDM
$end

$neo_msdft
denplt = 1
$end

$neo_msdft_diabat_control
1 0 1 0
0 1 0 1
$end

$neo_basis
H    2
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    8
S    1    1.000000
   4.0 1.0
P    1    1.000000
```

Example 13.13 Input for a NEO-MSDFT calculation on a protonated water dimer with a fixed O–O distance of 2.80 Å, where all protons in the system are quantized, but only the proton between the two oxygen atoms is treated as transferring. The two centers for the transferring proton are written consecutively in the *$molecule* section, with the first one written as an H atom center and the second one is written as a ghost H atom center. The other centers do not have corresponding ghost centers and are therefore not recognized as being transferring. The electronic basis set is STO-3G, and the protonic basis set is an uncontracted 1s1p basis set with exponents 4.0. The electronic exchange-correlation functional is B3LYP, and the electron-proton correlation functional is epc17-2.

```
$molecule
1  1
O      -1.4000000000     0.0000000000     0.0000000000
O       1.4000000000     0.0000000000     0.0000000000
H      -0.3000000000     0.0000000000     0.0000000000
@H      0.3000000000     0.0000000000     0.0000000000
H       1.6600000000     0.7500000000    -0.5500000000
H       1.6600000000    -0.7500000000    -0.5500000000
H      -1.6600000000     0.7500000000     0.5500000000
H      -1.6600000000    -0.7500000000     0.5500000000
$end

$rem
jobtype = sp
input_bohr = false
method = b3lyp
basis = sto-3g
neo = true
neo_epc = epc172
neo_msdft = 1
sym_ignore = 1
scf_convergence 6
NEO_N_SCF_CONVERGENCE = 6
NEO_E_CONV = 6
max_scf_cycles 500
SCF_ALGORITHM GDM
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    5
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    6
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    7
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
```

Example 13.14 Input for the NEO-DFT/epc17-2 geometry optimization calculation of a CH_2O molecule in C-PCM water. A van der Waals surface is constructed around the fixed, classical nuclear point charge positions and the nuclear basis function center positions of the two quantum hydrogens. Here, the Bondi radii for hydrogen is used in the construction of the spherical cavity surrounding the delocalized proton densities, but the user has the option to specify a custom-defined atomic radii in a *$van_der_waals* input section. The entire cavity is scaled by a factor of $\alpha_{\text{vdW}} = 1.2$, and is then discretized via the SwiG approach.

```
$molecule
0 1
  C        0.000000    0.000000    0.000000
  O        0.000000    0.000000    1.220000
  H        0.935307    0.0000000  -0.5400000
  H       -0.935307    0.000000   -0.540000
$end

$rem
point_group_symmetry False
   JOBTYPE              OPT
   INPUT_BOHR           FALSE
   BASIS                sto-3g
   NEO                  TRUE
   METHOD               pbe0
   xc_grid = 000099000302
   NEO_EPC              epc172
   SCF_CONVERGENCE = 8
   NEO_E_CONV = 8
   SCF_ALGORITHM        GDM
   SOLVENT_METHOD       PCM
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end

$pcm
   Theory               CPCM
   Method               SWIG
   Solver               INVERSION
   HeavyPoints          194
   HPoints              194
   Radii                Bondi
   vdwScale             1.2
$end

$solvent
  Dielectric   78.39
$end
```

Example 13.15 Input for the NEO-DFT/epc17-2 geometry optimization calculation of a Au_2H, $C_{\infty v}$ molecule utilizing the fit-LANL2DZ effective core potential for the gold atoms.

```
$molecule
1 1
  Au       0.000000    0.000000    2.222000
  Au       0.000000    0.000000    5.502000
  H        0.000000    0.000000    3.842000
$end

$rem
   JOBTYPE                 OPT
   SYM_IGNORE             TRUE
   INPUT_BOHR            FALSE
   BASIS                 mixed
   NEO                    TRUE
   METHOD                b3lyp
   xc_grid      = 000099000302
   NEO_EPC              epc172
   NEO_N_SCF_CONVERGENCE   = 8
   SCF_CONVERGENCE         = 8
   NEO_E_CONV              = 8
   SCF_ALGORITHM           GDM
   ECP             fit-LANL2DZ
$end

$basis
Au 1
LANL2DZ
****
Au 2
LANL2DZ
****
H 3
def2-qzvp
****
$end

$neo_basis
H    3
S    1    1.000000
   5.973 1.0
S    1    1.000000
   10.645 1.0
S    1    1.000000
   17.943 1.0
S    1    1.000000
   28.950 1.0
P    1    1.000000
   7.604 1.0
P    1    1.000000
   14.701 1.0
P    1    1.000000
   23.308 1.0
D    1    1.000000
   9.011 1.0
D    1    1.000000
   19.787 1.0
F    1    1.000000
   10.914 1.0
F    1    1.000000
   20.985 1.0
****
$end
```

Example 13.16 Input for NEO-HF analytic Hessian calculation on HCN molecule with a proton treated quantum mechanically. The electronic basis set is STO-3G and the protonic basis is 1s1p with exponents 4.0.

```
$molecule
0 1
  C          0.0000000000    0.0000000000    0.9684140792
  N          0.0000000000    0.0000000000   -1.2085828830
  H          0.0000000000    0.0000000000    2.9046475823
$end

$rem
jobtyp = freq
input_bohr = true
point_group_symmetry = False
method = hf
basis = sto-3g
neo = true
SCF_ALGORITHM = gdm
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.17 Input for NEO-HF(V) on HCN molecule with a proton treated quantum mechanically. The electronic basis set is STO-3G and the protonic basis is 1s1p with exponents 4.0.

```
$molecule
0 1
  C          0.0000000000    0.0000000000    0.9684140792
  N          0.0000000000    0.0000000000   -1.2085828830
  H          0.0000000000    0.0000000000    2.9046475823
$end

$rem
jobtyp = freq
input_bohr = true
point_group_symmetry = False
method = hf
SCF_ALGORITHM = gdm
basis = sto-3g
neo = true
neo_scfv = 1
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.18 Input for CNEO-DFT analytic Hessian calculation on HCN molecule with a proton treated quantum mechanically. The electronic basis set is STO-3G, and the protonic basis set is an uncontracted 1s1p basis set with exponents 4.0. The electronic exchange-correlation functional is BLYP, and the electron-proton correlation functional is epc17-2.

```
$molecule
0 1
H    0.000000    0.000000   -0.013880
C    0.000000    0.000000    1.103384
N    0.000000    0.000000    2.310496
$end

$rem
jobtype     freq
sym_ignore  true
method      blyp
basis       sto-3g
neo         true
neo_epc     epc172
cneo        true
$end

$neo_basis
H    1
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.19 Input for NEO-DFT energy calculation on CH_2O molecule with the two protons treated quantum mechanically using the nuclear Hartree product approximation. The electronic basis set is STO-3G, and the protonic basis set is an uncontracted 1s1p basis set with exponents 4.0. The electronic exchange-correlation functional is BLYP, and the electron-proton correlation functional is epc17-2.

```
$molecule
0 1
C        0.000000    0.000000    0.000000
O        0.000000    0.000000    1.220000
H        0.935307    0.000000   -0.540000
H       -0.935307    0.000000   -0.540000
$end

$rem
jobtype          sp
sym_ignore       true
scf_algorithm    gdm
method           blyp
basis            sto-3g
neo              true
neo_epc          epc172
neo_n_hartree_prod     true
neo_n_scf_convergence  8
scf_convergence        8
neo_e_conv             8
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.20 Input for the LR-NEO-TDDFT/epc19 calculation on CH_2O molecule (both protons treated quantum mechanically) of the first five roots obtained with the Davidson algorithm. The electronic exchange-correlation functional is PBE0. The electronic basis set is STO-3G and the protonic is an uncontracted 1s1p basis set with exponents 4.0. This calculation utilizes DFT grid with 99 radial and 302 spherical quadrature points.

```
$molecule
   0 1
   C     0.000000    0.000000    0.000000
   O     0.000000    0.000000    1.220000
   H     0.935307    0.000000   -0.540000
   H    -0.935307    0.000000   -0.540000
$end

$rem
   METHOD            pbe0
   BASIS             sto-3g
   THRESH            14
   XC_GRID           000099000302
   S2THRESH          12
   NEO               true
   NEO_EPC           epc172
   SET_ROOTS         5
   RPA               true
   SCF_CONVERGENCE   12
   NEO_E_CONV        12
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.21 Input for the LR-NEO-TDHF calculation on the FDF^- molecule treating quantum nuclei as deuterium and employing the NO_VPP option. The electronic basis set is cc-pVDZ and the protonic is an uncontracted even-tempered 8s8p basis set.

```
$molecule
   -1 1
   F   0.000000   0.000000   -1.122987
   F   0.000000   0.000000    1.122987
   H   0.000000   0.000000    0.000000
$end

$rem
   METHOD            hf
   BASIS             cc-pvdz
   NEO               true
   SCF_ALGORITHM     GDM
   RPA               true
   CIS_N_ROOTS       100
   THRESH            14
   S2THRESH          12
   SCF_CONVERGENCE   11
   MAX_SCF_CYCLES    300
   NEO_VPP           0
   NEO_ISOTOPE       2
   NEO_E_CONV        11
$end

$neo_basis
H    3
S    1    1.000000
   2.828400 1.0
S    1    1.000000
   4.0 1.0
S    1    1.000000
   5.6569 1.0
S    1    1.000000
   8.0 1.0
S    1    1.000000
   11.3137 1.0
S    1    1.000000
   16.0 1.0
S    1    1.000000
   22.6274 1.0
S    1    1.000000
   32.0 1.0
P    1    1.000000
   2.828400 1.0
P    1    1.000000
   4.0 1.0
P    1    1.000000
   5.6569 1.0
P    1    1.000000
   8.0 1.0
P    1    1.000000
   11.3137 1.0
P    1    1.000000
   16.0 1.0
P    1    1.000000
   22.6274 1.0
P    1    1.000000
   32.0 1.0
****
$end
```

Example 13.22 Input for the analytic LR-NEO-TDDFT gradient calculation on the CH_2 molecule with both protons treated quantum mechanically. A total of four excited states are requested and the gradient is computed for the 3rd excited state. The electronic exchange-correlation functional is CAM-B3LYP, and electron-proton correlation functional epc17-2 is used. The electronic basis set is STO-3G and the protonic basis is 1s1p with exponents 4.0. This calculation utilizes DFT grid with 99 radial and 302 spherical quadrature points along with the GDM algorithm.

```
$molecule
0 3
C  0.00000000000000e+00  0.00000000000000e+00 -5.63654429543699e-02
H  1.81800983405161e+00  0.00000000000000e+00 -9.92269386019353e-01
H -1.81800983405161e+00  0.00000000000000e+00 -9.92269386019353e-01
$end

$rem
point_group_symmetry = False
input_bohr = true
method = cam-b3lyp
basis = sto-3g
thresh = 14
s2thresh = 12
neo = true
SET_ROOTS = 4
RPA = true
xc_grid = 000099000302
unrestricted = 1
neo_epc = epc172
SCF_ALGORITHM = gdm
SET_STATE_DERIV = 3
$end

$neo_basis
H    2
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.23 Input for LR-NEO-TDDFT geometry optimization on the C_2H_2 molecule with both protons treated quantum mechanically. A total of three excited states are requested and the geometry optimization is computed for the 1st excited state. The electronic exchange-correlation functional is B3LYP, and electron-proton correlation functional epc17-2 is used. The electronic basis set is STO-3G and the protonic basis is 1s1p with exponents 4.0. This calculation utilizes DFT grid with 99 radial and 302 spherical quadrature points along with the GDM algorithm.

```
$molecule
0 1
        C       0.4142076725     1.0563578037     0.0000000223
        C      -0.4142118956    -1.0563667882     0.0000000223
        H       1.1661939287     2.9673893099     0.0000000246
        H      -1.1661909474    -2.9673788285     0.0000000246
$end

$rem
point_group_symmetry = False
NEO_SET_OPT = 1
neo_epc = epc172
SET_STATE_DERIV = 1
jobtype = opt
input_bohr = true
method = b3lyp
neo = true
SCF_ALGORITHM = gdm
thresh = 14
s2thresh = 12
basis = sto-3g
rpa = true
SET_ROOTS = 3
xc_grid = 000099000302
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end
```

Example 13.24 Input for LR-NEO-TDDFT on the C_2H_2 molecule with both protons treated quantum mechanically. A total of 12 excited states are requested. Ground-state protonic and electronic densities are printed in the cube files. Protonic and electronic transition densities of the first and the second vibronic excitations with electronic dominant characters are also printed in the cube files. The electronic exchange-correlation functional is B3LYP, and electron-proton correlation functional epc17-2 is used. The electronic basis set is STO-3G and the protonic basis is 1s1p with exponents 4.0. This calculation utilizes DFT grid with 99 radial and 302 spherical quadrature points along with the GDM algorithm.

```
$molecule
0 1
       C        -0.2315710674    1.2702261467    0.0000001295
       C         0.2315702809   -1.2702255666    0.0000001295
       H         1.2946585350    2.6676952886   -0.0000000923
       H        -1.2946589903   -2.6676943717   -0.0000000923
$end

$rem
point_group_symmetry = False
input_bohr = true
method = b3lyp
neo = true
NEO_SET_ESTATE = 1
SCF_ALGORITHM = gdm
thresh = 14
s2thresh = 12
basis = sto-3g
GEOM_OPT_MAX_CYCLES = 500
rpa = true
SET_ROOTS = 12
xc_grid = 000099000302
MAKE_CUBE_FILES = true
plots = true
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end

$plots
grid information to plot protonic and electronic ground state densities and transition densiti
   100  -4.0 6.0
   100  -5.0 4.0
   100  -4.0 4.0
   0 1 2 0
   0
   0 1
$end
```

Example 13.25 Formate ion simulated with RT-NEO-TDHF and excited by a Gaussian impulse response from all three orthogonal directions, x, y, and z.

```
$molecule
-1 1
C   0.000000  0.000000  0.316833
H   0.000000  0.000000  1.430087
O   0.000000  1.135580 -0.208193
O   0.000000 -1.135580 -0.208193
$end

$rem
NEO                   = true
SYM_IGNORE            = true
INPUT_BOHR            = false
BASIS                 = cc-pvtz
METHOD                = hf
SCF_CONVERGENCE       = 9
NEO_N_SCF_CONVERGENCE = 9
NEO_E_CONV            = 9
NEO_VPP               = 0
NEO_TDKS              = true
$end

$neo_basis
H    2
S    1    1.000000
   5.973 1.0
S    1    1.000000
   10.645 1.0
S    1    1.000000
   17.943 1.0
S    1    1.000000
   28.950 1.0
P    1    1.000000
   7.604 1.0
P    1    1.000000
   14.701 1.0
P    1    1.000000
   23.308 1.0
D    1    1.000000
   9.011 1.0
D    1    1.000000
   19.787 1.0
F    1    1.000000
   10.914 1.0
F    1    1.000000
   20.985 1.0
****
$end

$neo_tdks
METHOD              = realtime
DT                  = 0.04
MAXI TER            = 20000
FIELD_TYPE          = gaussian
FIELD_AMP           = 0.02
FIELD_PEAK          = 0.0
FIELD_TAU           = 800.0
FIELD_FREQUENCY     = 6.0
FIELD_DIRECTION     = xyz
FIELD_PARTICLE_TYPE = both
$end
```

Example 13.26 Excited-state intramolecular proton transfer in *o*-hydroxylbenzaldehyde simulated with RT-NEO-TDDFT-Ehrenfest utilzing the B3LYP electronic exchange-correlation functional and epc17-2 electron-proton correlation functional. Two additional basis function centers are added along the proton transfer path, the total simulation time was specified to be 19.35 fs, and an electronic HOMO to LUMO swap was performed to prepare the initial state.

```
$molecule
0 1
C  -1.310008  1.258755  0.000000
C   0.019289  0.780580  0.000000
C   0.322586 -0.621636  0.000000
C  -0.761283 -1.465586  0.000000
C  -2.125342 -0.985468  0.000000
C  -2.398971  0.362640  0.000000
O   1.008708  1.658363  0.000000
C   1.725826 -1.062678  0.000000
O   2.682115 -0.219090  0.000000
H  -3.414260  0.733314  0.000000
H  -0.596742 -2.537818  0.000000
H  -2.926679 -1.713458  0.000000
H  -1.459388  2.331114  0.000000
H   1.924136 -2.138192  0.000000
H   1.844187  1.135044  0.000000
@H  2.042593  0.928022  0.000000
@H  2.241000  0.721000  0.000000
$end

$rem
NEO                   = true
SYM_IGNORE            = true
INPUT_BOHR            = false
BASIS                 = cc-pvtz
METHOD                = b3lyp
SCF_CONVERGENCE       = 9
NEO_N_SCF_CONVERGENCE = 9
NEO_E_CONV            = 9
NEO_EPC               = epc172
NEO_VPP               = 0
NEO_TDKS              = true
$end

$neo_basis
H    15
S    1    1.000000
4.0 1.0
P    1    1.000000
4.0 1.0
****
H    16
S    1    1.000000
4.0 1.0
P    1    1.000000
4.0 1.0
****
H    17
S    1    1.000000
4.0 1.0
P    1    1.000000
4.0 1.0
****
$end

$neo_tdks
METHOD                  = ehrenfest
DT                      = 0.04
MAXITER                 = 20000
ELECTRONIC_HOMO_TO_LUMO = true
$end
```

Example 13.27 Ground-state intramolecular proton transfer in malonaldehyde simulated with BO-RT-NEO-TDDFT-Ehrenfest. Minimal basis sets are employed as a proof of principle. Note the large time step of 4.2 atomic units is employed since the electronic dynamics are no longer being propagated.

```
$molecule
0 1
O   0.000000 -1.300873  2.045662
O   0.000000  1.290832  2.045662
C   0.000000 -1.216031  0.759259
C   0.000000  1.205990  0.759259
C   0.000000 -0.005020  0.053344
H   0.000000 -0.005020 -1.024823
H   0.000000 -2.163146  0.218967
H   0.000000  2.153105  0.218967
H   0.000000 -0.301012  2.355420
$end

$rem
NEO                   = true
SYM_IGNORE            = true
INPUT_BOHR            = false
BASIS                 = sto-3g
METHOD                = b3lyp
SCF_CONVERGENCE       = 9
NEO_N_SCF_CONVERGENCE = 9
NEO_E_CONV            = 9
NEO_EPC               = epc172
NEO_VPP               = 0
NEO_TDKS              = true
$end

$neo_basis
H    9
S    1    1.000000
4.0 1.0
P    1    1.000000
4.0 1.0
****
$end

$neo_tdks
METHOD  = bo-ehrenfest
DT      = 4.2
MAXITER = 1335
$end
```

Example 13.28 Ground-state intramolecular proton transfer in malonaldehyde simulated with BO-RT-NEO-TDDFT-Ehrenfest, and utilizing the B3LYP-D3(0) empirical dispersion correction with custom parameters.

```
$molecule
0 1
O   0.000000 -1.300873  2.045662
O   0.000000  1.290832  2.045662
C   0.000000 -1.216031  0.759259
C   0.000000  1.205990  0.759259
C   0.000000 -0.005020  0.053344
H   0.000000 -0.005020 -1.024823
H   0.000000 -2.163146  0.218967
H   0.000000  2.153105  0.218967
H   0.000000 -0.301012  2.355420
$end

$rem
NEO                   = true
SYM_IGNORE            = true
INPUT_BOHR            = false
BASIS                 = sto-3g
METHOD                = b3lyp
SCF_CONVERGENCE       = 9
NEO_N_SCF_CONVERGENCE = 9
NEO_E_CONV            = 9
NEO_EPC               = epc172
NEO_VPP               = 0
NEO_TDKS              = true
DFT_D                 = D3_ZERO
DFT_D3_S6             = 100000
DFT_D3_RS6            = 126100
DFT_D3_S8             = 170300
DFT_D3_3BODY          = FALSE
$end

$neo_basis
H    9
S    1    1.000000
4.0 1.0
P    1    1.000000
4.0 1.0
****
$end

$neo_tdks
METHOD  = bo-ehrenfest
DT      = 4.2
MAXITER = 1335
$end
```

Example 13.29 Input for the NEO-RICCSD calculation on H_2O molecule with the second proton treated quantum mechanically. The electronic basis set is STO-3G and the protonic is an uncontracted 1s1p basis set with exponents 4.0. The electronic auxiliary basis set is RIMP2-aug-cc-pVDZ and the protonic auxiliary basis set is an uncontracted even-tempered 8s8p basis set.

```
$molecule
0 1
O          0.00000       -0.07579        0.00000
H          0.86681        0.60144        0.00000
H         -0.86681        0.60144        0.00000
$end

$rem
neo = true
basis = sto-3g
aux_basis = rimp2-aug-cc-pVDZ
NEO_RICCSD 1
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end

$neo_aux_basis
H   3
S    1    1.000000
   2.8284 1.0
S    1    1.000000
   4.0 1.0
S    1    1.000000
   5.6569 1.0
S    1    1.000000
   8.0 1.0
S    1    1.000000
   11.3137 1.0
S    1    1.000000
   16.0 1.0
S    1    1.000000
   22.6274 1.0
S    1    1.000000
   32.0 1.0
P    1    1.000000
   2.8284 1.0
P    1    1.000000
   4.0 1.0
P    1    1.000000
   5.6569 1.0
P    1    1.000000
   8.0 1.0
P    1    1.000000
   11.3137 1.0
P    1    1.000000
   16.0 1.0
P    1    1.000000
   22.6274 1.0
P    1    1.000000
   32.0 1.0
****
$end
```

Example 13.30 Input for the NEO-SCS′-MP2 calculation on CH_2O molecule with the both protons treated quantum mechanically. The electronic basis set is STO-3G and the protonic is an uncontracted 1s1p basis set with exponents 4.0. The electronic auxiliary basis set is RIMP2-aug-cc-pVDZ and the protonic auxiliary basis set is an uncontracted even-tempered 8s8p basis set. SCS scaling corresponds to $c_{ss} = 0.33$, $c_{os} = 1.2$ and $c_{ep} = 1.2$.

```
$molecule
0 1
C         0.000000    0.000000    0.000000
O         0.000000    0.000000    1.220000
H         0.935307    0.0000000  -0.5400000
H        -0.935307    0.000000   -0.540000
$end

$rem
method = hf
basis = sto-3g
neo = true
sym_ignore = 1
aux_basis = rimp2-cc-pVDZ
scf_convergence 10
max_scf_cycles 100
SCF_ALGORITHM diis
NEO_E_CONV 10
NEO_RIMP2 1
SCS 1
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
H    4
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end

$neo_aux_basis
H   3
S    1    1.000000
   2.8284 1.0
S    1    1.000000
   4.0 1.0
S    1    1.000000
   5.6569 1.0
S    1    1.000000
   8.0 1.0
S    1    1.000000
   11.3137 1.0
S    1    1.000000
   16.0 1.0
S    1    1.000000
   22.6274 1.0
S    1    1.000000
   32.0 1.0
P    1    1.000000
   2.8284 1.0
P    1    1.000000
   4.0 1.0
P    1    1.000000
   5.6569 1.0
```

Example 13.31 Input for the NEO-OOMP2 calculation on H_2O molecule with the second proton treated quantum mechanically. The electronic basis set is STO-3G and the protonic is an uncontracted 1s1p basis set with exponents 4.0. The electronic auxiliary basis set is RIMP2-aug-cc-pVDZ and the protonic auxiliary basis set is an uncontracted even-tempered 8s8p basis set.

```
$molecule
0 1
O          0.00000       -0.07579        0.00000
H          0.86681        0.60144        0.00000
H         -0.86681        0.60144        0.00000
$end

$rem
input_bohr = false
method = hf
basis = sto-3g
aux_basis = rimp2-aug-cc-pVDZ
neo = true
sym_ignore = 1
scf_convergence 10
max_scf_cycles 100
SCF_ALGORITHM diis
NEO_E_CONV 10
NEO_VPP 0
NEO_RIMP2 2
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end

$neo_aux_basis
H   3
S    1    1.000000
   2.8284 1.0
S    1    1.000000
   4.0 1.0
S    1    1.000000
   5.6569 1.0
S    1    1.000000
   8.0 1.0
S    1    1.000000
   11.3137 1.0
S    1    1.000000
   16.0 1.0
S    1    1.000000
   22.6274 1.0
S    1    1.000000
   32.0 1.0
P    1    1.000000
   2.8284 1.0
P    1    1.000000
   4.0 1.0
P    1    1.000000
   5.6569 1.0
P    1    1.000000
   8.0 1.0
P    1    1.000000
   11.3137 1.0
P    1    1.000000
   16.0 1.0
```

Example 13.32 Input for the NEO-SOS′-OOMP2 calculation on H_2O molecule with the second proton treated quantum mechanically. The electronic basis set is STO-3G and the protonic is an uncontracted 1s1p basis set with exponents 4.0. The electronic auxiliary basis set is RIMP2-aug-cc-pVDZ and the protonic auxiliary basis set is an uncontracted even-tempered 8s8p basis set. SOS scaling corresponds to $c_{ss} = 0.0$ and $c_{os} = 1.2$, with a custom value of $c_{ep} = 1.5$.

```
$molecule
0 1
O          0.00000       -0.07579        0.00000
H          0.86681        0.60144        0.00000
H         -0.86681        0.60144        0.00000
$end

$rem
input_bohr = false
method = hf
basis = sto-3g
aux_basis = rimp2-aug-cc-pVDZ
neo = true
sym_ignore = 1
scf_convergence 10
max_scf_cycles 100
SCF_ALGORITHM diis
NEO_E_CONV 10
NEO_VPP 0
NEO_RIMP2 2
SCS 2
EP_FACTOR 1500000
$end

$neo_basis
H    3
S    1    1.000000
   4.0 1.0
P    1    1.000000
   4.0 1.0
****
$end

$neo_aux_basis
H   3
S    1    1.000000
   2.8284 1.0
S    1    1.000000
   4.0 1.0
S    1    1.000000
   5.6569 1.0
S    1    1.000000
   8.0 1.0
S    1    1.000000
   11.3137 1.0
S    1    1.000000
   16.0 1.0
S    1    1.000000
   22.6274 1.0
S    1    1.000000
   32.0 1.0
P    1    1.000000
   2.8284 1.0
P    1    1.000000
   4.0 1.0
P    1    1.000000
   5.6569 1.0
P    1    1.000000
   8.0 1.0
P    1    1.000000
   11.3137 1.0
```

Example 13.33 Input for NEO-SA-MCSCF on $HeHHe^+$ molecule with the proton treated quantum mechanically. 3 states are averaged. The NEO-CASSCF uses AH/GDM.

```
$molecule
1  1
He    0.0000  0.0000  0.9000
He    0.0000  0.0000 -0.9000
H     0.0000  0.0000  0.0000
$end

$rem
input_bohr = false
mem_total = 32000
method = hf
basis = mixed
use_libqints = 1
print_general_basis = 0
scf_algorithm = gdm
neo_basis_lin_dep_thresh = 3
neo = true
neo_vpp = 0
neo_purecart = 1111
neo_mcscf = true
neo_wf_e_docc = 0
neo_wf_e_cas = 4
neo_wf_n_cas = 87
neo_mcscf_sa = true
neo_mcscf_sa_nstates = 3
neo_mcscf_nstates = 6
neo_mcscf_opt_method = 2
neo_mcscf_macro_conv_thresh = 6
neo_mcscf_macro_max_iter = 100
neo_mcscf_emc_conv_thresh = 8
neo_mcscf_micro_conv_thresh = 8
neo_mcscf_micro_max_iter = 100000
neo_mcscf_micro_opt_max_iter = 5
neo_mcscf_kspace_min = 8
neo_mcscf_kspace_max = 20
neo_mcscf_reorder_mos = false
neo_mcscf_print_mos = false
neo_mcscf_read_mos = false
neo_mcscf_print_level = 0
neo_mcscf_use_prev_ci_iter = 0
$end

$neo_basis
H    3
S    1    1.0
   2.8284  1.0
S    1    1.0
   4.0000  1.0
S    1    1.0
   5.6569  1.0
S    1    1.0
   8.0000  1.0
S    1    1.0
  11.3137  1.0
S    1    1.0
  16.0000  1.0
S    1    1.0
  22.6274  1.0
S    1    1.0
  32.0000  1.0
P    1    1.0
   2.8284  1.0
P    1    1.0
   4.0000  1.0
```

Example 13.34 Input for NEO-SD$_{en}$CI on HeHHe^{+} molecule with the proton treated quantum mechanically, solving for 7 roots.

```
$molecule
1  1
He    0.0000  0.0000  0.9000
He    0.0000  0.0000 -0.9000
H     0.0000  0.0000  0.0000
$end

$rem
input_bohr = false
mem_total = 32000
method = hf
basis = mixed
use_libqints = 1
print_general_basis = 0
scf_algorithm = gdm
neo_basis_lin_dep_thresh = 3
neo = true
neo_vpp = 0
neo_purecart = 1111
neo_wf_e_docc = 2
neo_wf_e_xlevel = 1
neo_wf_n_cas = 1
neo_wf_n_xlevel = 1
neo_mrci = true
neo_mrci_max_iter = 1000
neo_mrci_num_roots = 7
neo_mrci_kspace_min = 8
neo_mrci_kspace_max = 15
neo_mrci_prediag_routine = 1
neo_mrci_prediag_dim = 1000
neo_mrci_conv_thresh = 5
$end

$neo_basis
H    3
S    1    1.0
   2.8284  1.0
S    1    1.0
   4.0000  1.0
S    1    1.0
   5.6569  1.0
S    1    1.0
   8.0000  1.0
S    1    1.0
  11.3137  1.0
S    1    1.0
  16.0000  1.0
S    1    1.0
  22.6274  1.0
S    1    1.0
  32.0000  1.0
P    1    1.0
   2.8284  1.0
P    1    1.0
   4.0000  1.0
P    1    1.0
   5.6569  1.0
P    1    1.0
   8.0000  1.0
P    1    1.0
  11.3137  1.0
P    1    1.0
  16.0000  1.0
P    1    1.0
```

13.6 Construction of Effective Hamiltonians from EOM-CC Wave Functions

Effective Hamiltonians provide a powerful framework for coarse-grained representation of complex electronic structure and for making direct comparisons with the experimentally derived parameters. It is commonly used in magnetic systems.[59] Following general theory of effective Hamiltonians,[59] Ref. 68 describes how effective Hamiltonians can be extracted from the EOM-CC wave functions by using Bloch's formalism. This approach provides an exact map of the many-body electronic structure onto a selected model space, yielding rigorous effective Hamiltonians in this model space. Currently, only EOM-SF-CCSD wave functions and Heisenberg's and Hubbard's model spaces are supported. The implemented procedure is N-to-N map, meaning that the number of model states is the same as the number of electronic states used to build the effective Hamiltonian. In the case of the Heisenberg's Hamiltonian, the number of states N should be the same as the number of open-shell electrons, since this is the number of open-shell configurations that a single spin-flipping excitation can provide. In the case of the Hubbard's Hamiltonian, N should equal to the total number of configurations that a single spin-flipping excitation can generate in the open-shell subspace (squared number of open-shell orbitals). Once CC_MAKE_EFF_HAM *$rem* variable is set to 1, the electronic states are read from the list specified in the *$eff_ham* section. Right and left amplitudes of these EOM states are used for the effective Hamiltonian construction. The Hamiltonians are constructed in two forms: the Bloch's form (non-Hermitian) and des Cloizeaux's form (Hermitian). If CC_OSFNO is invoked (recommended), the correspondence between the open-shell orbitals is established through the SVD procedure,[69] followed by Boys' localization of these orbitals. If these orbitals are desired, the orbital canonicalization should be disabled with CC_CANONIZE = FALSE. If GUI is set to 2, the orbitals used in CCMAN2 calculations are dropped to the `.fchk` file and can be visualized by IQMOL.

Note: Beware that localization procedure may fail when there are several orbitals residing on the same radical centers, since in this case Boys localization becomes ill-conditioned.

The syntax of the *$eff_ham* section is similar to the syntax of the *$trans_prop* section. First, **state_list** starts the list of electronic states that are used for the construction of the effective Hamiltonian. The EOM states are abbreviated by the type of the method (currently, only SF_STATES is supported), its irreducible representation (use 1 for OSFNO and localized orbitals), and the number in the particular irreducible representation. The list of states is terminated by **end_list**. The type of the effective Hamiltonian is specified by the **eff_ham** keyword, which can be either **Heisenberg** or **Hubbard**.

The example below is based on Molecule 1 from Ref. 68. It shows how the Heisenberg Hamiltonian can be constructed from the EOM-SF-CCSD wave-functions. It can be modified to construct the Hubbard Hamiltonian by (i) requesting

four SF states (that would include two covalent and two ionic states) and (ii) setting **eff_ham** to **Hubbard**.

Example 13.35 Input for constructing the Heisenberg Hamiltonian from EOM-SF wave-functions for a molecule with two open-shell electrons.

```
$comment
   Molecule 1 from J. Chem. Phys. 152, 094108 (2020):  propane-1,3-diyl.
   This example illustrates the construction of the Heisenberg Hamiltonian
   from the two lowest diradical states computed by EOM-SF-CCSD.
$end

$molecule
   0 3
   C    -0.0720787494   -0.2443969570    1.2525335342
   C    -0.1914499016    0.5591929000    0.0000000000
   C    -0.0720787494   -0.2443969570   -1.2525335342
   H     0.1191211842    0.2375636339    2.2059937351
   H    -0.3120216606   -1.3031986575    1.2578210464
   H     0.5519313098    1.3748537257    0.0000000000
   H    -1.1649723712    1.1083017822    0.0000000000
   H     0.1191211842    0.2375636339   -2.2059937351
   H    -0.3120216606   -1.3031986575   -1.2578210464
$end

$rem
   METHOD            eom-ccsd
   BASIS             sto-3g
   GUI               2
   SF_STATES         [2]
   CC_OSFNO          true
   CC_FNO_THRESH     9900
   CC_CANONIZE       false
   CC_MAKE_EFF_HAM   1
   CC_EOM_PROP       1
   CC_EOM_PROP_TE    1
   POINT_GROUP_SYMMETRY false
   INTEGRAL_SYMMETRY    false
$end

$eff_ham
   state_list
   sf_states 1 1
   sf_states 1 2
   end_list
   eff_ham heisenberg
$end
```

References and Further Reading

[1] Geometry Optimization (Appendix 9.1).

[2] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill. *Chem. Phys. Lett.*, 254:329, 1996. DOI: 10.1016/0009-2614(96)00280-1.

[3] T. Arai. *J. Chem. Phys.*, 33:95, 1960. DOI: 10.1063/1.1731142.

[4] B. Auer and S. Hammes-Schiffer. *J. Chem. Phys.*, 132:084110, 2010. DOI: 10.1063/1.3332769.

[5] V. Barone and M. Cossi. *J. Phys. Chem. A*, 102:1995, 1998. DOI: 10.1021/jp9716997.

[6] N. A. Besley, A. M. Lee, and P. M. W. Gill. *Mol. Phys.*, 100:1763, 2002. DOI: 10.1080/00268970110111779.

[7] N. A. Besley, D. P. O'Neill, and P. M. W. Gill. *J. Chem. Phys.*, 118:2033, 2003. DOI: 10.1063/1.1532311.

[8] F. W. Bobrowicz and W. A. Goddard III. The self-consistent field equations for generalized valence bond and open-shell Hartree–Fock wave functions. In H. F. Schaefer III, editor, *Methods of Electronic Structure Theory*, volume 3, pages 79–128. Plenum, New York, 1977. DOI: 10.1007/978-1-4757-0887-5_4.

[9] K. R. Brorsen, Y. Yang, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 8:3488, 2017. DOI: 10.1021/acs.jpclett.7b01442.

[10] K. R. Brorsen, P. E. Schneider, and S. Hammes-Schiffer. *J. Chem. Phys.*, 149:044110, 2018. DOI: 10.1063/1.5037945.

[11] E. Caldeweyher, C. Bannwarth, and S. Grimme. *J. Chem. Phys.*, 147:034112, 2017. DOI: 10.1063/1.4993215.

[12] E. Caldeweyher, S. Ehlert, A. Hansen, H. Neugebauer, S. Spicher, C. Bannwarth, and S. Grimme. *J. Chem. Phys.*, 150:154122, 2019. DOI: 10.1063/1.5090222.

[13] E. Caldeweyher, J.-M. Mewes, S. Ehlert, and S. Grimme. *Phys. Chem. Chem. Phys.*, 22:8499, 2020. DOI: 10.1039/d0cp00502a.

[14] E. Cancès and B. Mennucci. *J. Chem. Phys.*, 114:4744, 2001. DOI: 10.1063/1.1349091.

[15] J.-D. Chai and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 10:6615, 2008. DOI: 10.1039/b810189b.

[16] A. Chakraborty, M. V. Pak, and S. Hammes-Schiffer. *Phys. Rev. Lett.*, 101:153001, 2008. DOI: 10.1103/PhysRevLett.101.153001.

[17] A. Chakraborty, M. V. Pak, and S. Hammes-Schiffer. *J. Chem. Phys.*, 131:124115, 2009. DOI: 10.1063/1.3236844.

[18] D. M. Chipman. *J. Chem. Phys.*, 112:5558, 2000. DOI: 10.1063/1.481133.

[19] J. Cioslowski and G. Liu. *J. Chem. Phys.*, 105:4151, 1996. DOI: 10.1063/1.472285.

[20] M. Cossi, N. Rega, G. Scalmani, and V. Barone. *J. Comput. Chem.*, 24:669, 2003. DOI: 10.1002/jcc.10189.

[21] C. J. Cramer and D. G. Truhlar. *Acc. Chem. Res.*, 41:760, 2008. DOI: 10.1021/ar800019z.

[22] T. Culpitt, Y. Yang, F. Pavošević, Z. Tao, and S. Hammes-Schiffer. *J. Chem. Phys.*, 150:201101, 2019. DOI: 10.1063/1.5099093.

[23] T. Culpitt, Y. Yang, P. E. Schneider, F. Pavošević, and S. Hammes-Schiffer. *J. Chem. Theory Comput.*, 15:6840, 2019. DOI: 10.1021/acs.jctc.9b00665.

[24] S. Datta. *Quantum Transport: Atom to Transistor*. Cambridge University Press, Cambridge, 2005.

[25] M. Di Ventra. *Electron transport in nanoscale systems*. Cambridge University Press, Cambridge, 2008.

[26] J. A. Dickinson, Q. Yu, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 14:6170, 2023. DOI: 10.1021/acs.jpclett.3c01422.

[27] J. P. Dombroski, S. W. Taylor, and P. M. W. Gill. *J. Phys. Chem.*, 100:6272, 1996. DOI: 10.1021/jp952841b.

[28] P. S. Epstein. *Phys. Rev.*, 28:695, 1926. DOI: 10.1103/PhysRev.28.695.

[29] J. H. Fetherolf, F. Pavošević, Z. Tao, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 13:5563, 2022. DOI: 10.1021/acs.jpclett.2c01357.

[30] J. Gao, A. Grofe, H. Ren, and P. Bao. *J. Phys. Chem. Lett.*, 7:5143, 2016. DOI: 10.1021/acs.jpclett.6b02455.

[31] P. M. W. Gill. *Chem. Phys. Lett.*, 270:193, 1997. DOI: 10.1016/S0009-2614(97)00361-8.

[32] P. M. W. Gill, D. P. O'Neill, and N. A. Besley. *Theor. Chem. Acc.*, 109:241, 2003. DOI: 10.1007/s00214-002-0411-5.

[33] M. Gray, P. E. Bowling, and J. M. Herbert. *J. Phys. Chem. A*, 128:7739, 2024. DOI: 10.1021/acs.jpca.4c00283.

[34] S. Grimme. *J. Chem. Phys.*, 118:9095, 2003. DOI: 10.1063/1.1569242.

[35] S. Grimme. *J. Comput. Chem.*, 27:1787, 2006. DOI: 10.1002/jcc.20495.

[36] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg. *J. Chem. Phys.*, 132:154104, 2010. DOI: 10.1063/1.3382344.

[37] S. Grimme, S. Ehrlich, and L. Goerigk. *J. Comput. Chem.*, 32:1456, 2011. DOI: 10.1002/jcc.21759.

[38] A. Grofe, Z. Qu, D. G. Truhlar, H. Li, and J. Gao. *J. Chem. Theory Comput.*, 13:1176, 2017. DOI: 10.1021/acs.jctc.6b01176.

[39] S. Hammes-Schiffer. *J. Chem. Phys.*, 155:030901, 2021. DOI: 10.1063/5.0053576.

[40] J. M. Herbert. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 11:e1519, 2021. DOI: 10.1002/wcms.1519.

[41] J. M. Herbert. Density-functional theory for electronic excited states. In C. García-Iriepa and M. Marazzi, editors, *Theoretical and Computational Photochemistry: Fundamentals, Methods, Applications and Synergy with Experimental Approaches*, chapter 3, pages 69–118. Elsevier, 2023. DOI: 10.1016/B978-0-323-91738-4.00005-1.

[42] A. C. Hurley, J. E. Lennard-Jones, and J. A. Pople. *Proc. Roy. Soc. London A*, 220:446, 1953. DOI: 10.1098/rspa.1953.0198.

[43] Y. Jung, R. C. Lochan, A. D. Dutoi, and M. Head-Gordon. *J. Chem. Phys.*, 121:9793, 2004. DOI: 10.1063/1.1809602.

[44] A. Klamt and G. Schüürmann. *J. Chem. Soc. Perkin Trans. 2*, page 799, 1993. DOI: 10.1039/P29930000799.

[45] A. W. Lange and J. M. Herbert. *J. Phys. Chem. Lett.*, 1:556, 2010. DOI: 10.1021/jz900282c.

[46] A. W. Lange and J. M. Herbert. *J. Chem. Phys.*, 133:244111, 2010. DOI: 10.1063/1.3511297.

[47] A. W. Lange and J. M. Herbert. *Chem. Phys. Lett.*, 509:77, 2011. DOI: 10.1016/j.cplett.2011.04.092.

[48] A. W. Lange, J. M. Herbert, B. J. Albrecht, and Z.-Q. You. *Mol. Phys.*, 118:e1644384, 2020. DOI: 10.1080/00268976.2019.1644384.

[49] V. I. Lebedev. *Zh. Vychisl. Mat. Mat. Fix.*, 16:293, 1976. DOI: 10.1016/0041-5553(76)90100-2.

[50] V. I. Lebedev. *Sibirsk. Mat. Zh.*, 18:132, 1977.

[51] V. I. Lebedev and D. N. Laikov. *Dokl. Math.*, 366:741, 1999.

[52] A. M. Lee and P. M. W. Gill. *Chem. Phys. Lett.*, 313:271, 1999. DOI: 10.1016/S0009-2614(99)00935-5.

[53] T. E. Li and S. Hammes-Schiffer. *J. Chem. Phys.*, 158:114118, 2023. DOI: 10.1063/5.0142007.

[54] T. E. Li, E. Paenurk, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 15:751, 2024. DOI: 10.1021/acs.jpclett.3c03112.

[55] A. Liu, M. Chow, A. Wildman, M. Frish, S. Hammes-Schiffer, and X. Li. *J. Phys. Chem. A*, 126:7033, 2022. DOI: 10.1021/acs.jpca.2c05172.

[56] R. C. Lochan and M. Head-Gordon. *J. Chem. Phys.*, 126:164101, 2007. DOI: 10.1063/1.2718952.

[57] Y. Lu and J. Gao. *J. Phys. Chem. Lett.*, 13:7762, 2022. DOI: 10.1021/acs.jpclett.2c02088.

[58] C. Malbon and S. Hammes-Schiffer. *J. Chem. Theory Comput.*, 21:3968, 2025. DOI: 10.1021/acs.jctc.5c00042.

[59] J. P. Malrieu, R. Caballol, C. J. Calzado, C. de Graaf, and N. Guihéry. *Chem. Rev.*, 114:429, 2014. DOI: 10.1021/cr300500z.

[60] Y. Mo, P. Bao, and J. Gao. *Phys. Chem. Chem. Phys.*, 13:6760, 2011. DOI: 10.1039/C0CP02206C.

[61] R. K. Nesbet. *Proc. Roy. Soc. Ser. A*, 230:312, 1955. DOI: 10.1098/rspa.1955.0134.

[62] M. V. Pak, A. Chakraborty, and S. Hammes-Schiffer. *J. Phys. Chem. A*, 111:4522, 2007. DOI: 10.1021/jp0704463.

[63] F. Pavošević and S. Hammes-Schiffer. *J. Chem. Phys.*, 151:074104, 2019. DOI: 10.1063/1.5116113.

[64] F. Pavošević, T. Culpitt, and S. Hammes-Schiffer. *J. Chem. Theory Comput.*, 15:338, 2018. DOI: 10.1021/acs.jctc.8b01120.

[65] F. Pavošević, T. Culpitt, and S. Hammes-Schiffer. *Chem. Rev.*, 120:4222, 2020. DOI: 10.1021/acs.chemrev.9b00798.

[66] F. Pavošević, B. J. G. Rousseau, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 11:1578, 2020. DOI: 10.1021/acs.jpclett.0c00090.

[67] F. Pavošević, Z. Tao, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 12:1631, 2021. DOI: 10.1021/acs.jpclett.0c03771.

[68] P. Pokhilko and A. I. Krylov. *J. Chem. Phys.*, 152:094108, 2020. DOI: 10.1063/1.5143318.

[69] P. Pokhilko, D. Izmodenov, and A. I. Krylov. *J. Chem. Phys.*, 152:034105, 2020. DOI: 10.1063/1.5138643.

[70] P. Pulay. *Chem. Phys. Lett.*, 73:393, 1980. DOI: 10.1016/0009-2614(80)80396-4.

[71] P. Pulay. *J. Comput. Chem.*, 3:556, 1982. DOI: 10.1002/jcc.540030413.

[72] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard III, and W. M. Skiff. *J. Am. Chem. Soc.*, 114:10024, 1992. DOI: 10.1021/ja00051a040.

[73] V. A. Rassolov. *J. Chem. Phys.*, 117:5978, 2002. DOI: 10.1063/1.1503773.

[74] V. A. Rassolov, F. Xu, and S. Garaschchuk. *J. Chem. Phys.*, 120:10385, 2004. DOI: 10.1063/1.1738110.

[75] R. S. Rowland and R. Taylor. *J. Phys. Chem.*, 100:7384, 1996. DOI: 10.1021/jp953141+.

[76] P. E. Schneider, Z. Tao, F. Pavošević, E. Epifanovsky, X. Feng, and S. Hammes-Schiffer. *J. Chem. Phys.*, 154: 054108, 2021. DOI: 10.1063/5.0033540.

[77] H. Schröder, A. Creon, and T. Schwabe. *J. Chem. Theory Comput.*, 11:3163, 2015. DOI: 10.1021/acs.jctc.5b00400.

[78] R. Seeger and J. A. Pople. *J. Chem. Phys.*, 65:265, 1976. DOI: 10.1063/1.432764.

[79] J. H. Skone, M. V. Pak, and S. Hammes-Schiffer. *J. Chem. Phys.*, 123:134108, 2005. DOI: 10.1063/1.2039727.

[80] D. G. Smith, L. A. Burns, K. Patkowski, and C. D. Sherrill. *J. Phys. Chem. Lett.*, 7:2197, 2016. DOI: 10.1021/acs.jpclett.6b00780.

[81] P. R. Surján. *Topics Curr. Chem.*, 203:63, 1999. DOI: 10.1007/3-540-48972-X_4.

[82] C. Swalina, M. V. Pak, and S. Hammes-Schiffer. *Chem. Phys. Lett.*, 404:394, 2005. DOI: 10.1016/j.cplett.2005.01.115.

[83] Z. Tao, Y. Yang, and S. Hammes-Schiffer. *J. Chem. Phys.*, 151:124102, 2019. DOI: 10.1063/1.5119124.

[84] Z. Tao, S. Roy, P. E. Schneider, F. Pavošević, and S. Hammes-Schiffer. *J. Chem. Theory Comput.*, 17:5110, 2021. DOI: 10.1021/acs.jctc.1c00454.

[85] N. V. Tkatchenko and M. Head-Gordon. *J. Chem. Theory Comput.*, 20:9741, 2024. DOI: 10.1021/acs.jctc.4c01105.

[86] J. Tomasi, B. Mennucci, and R. Cammi. *Chem. Rev.*, 106:2999, 2005. DOI: 10.1021/cr9904009.

[87] T. N. Truong and E. V. Stefanovich. *Chem. Phys. Lett.*, 240:253, 1995. DOI: 10.1016/0009-2614(95)00541-B.

[88] T. Van Voorhis and M. Head-Gordon. *Mol. Phys.*, 100:1713, 2002. DOI: 10.1080/00268970110103642.

[89] S. P. Webb, T. Iordanov, and S. Hammes-Schiffer. *J. Chem. Phys.*, 117:4106, 2002. DOI: 10.1063/1.1494980.

[90] E. Wigner. *Phys. Rev.*, 40:749, 1932. DOI: 10.1103/PhysRev.40.749.

[91] A. Wildman, Z. Tao, L. Zhao, S. Hammes-Schiffer, and X. Li. *J. Chem. Theory Comput.*, 18:1340, 2022. DOI: 10.1021/acs.jctc.1c01285.

[92] J. Witte, N. Mardirossian, J. B. Neaton, and M. Head-Gordon. *J. Chem. Theory Comput.*, 13:2043, 2017. DOI: 10.1021/acs.jctc.7b00176.

[93] X. Xu and Y. Yang. *J. Chem. Phys.*, 152:084107, 2020. DOI: 10.1063/1.5143371.

[94] X. Xu and Y. Yang. *J. Chem. Phys.*, 153:074106, 2020. DOI: 10.1063/5.0014001.

[95] X. Xu and Y. Yang. *J. Chem. Phys.*, 154:244110, 2021. DOI: 10.1063/5.0055506.

[96] Y. Yang, K. R. Brorsen, T. Culpitt, M. V. Pak, and S. Hammes-Schiffer. *J. Chem. Phys.*, 147:114113, 2017. DOI: 10.1063/1.4996038.

[97] Y. Yang, T. Culpitt, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 9:1765, 2018. DOI: 10.1021/acs.jpclett.8b00547.

[98] Y. Yang, P. E. Schneider, T. Culpitt, F. Pavošević, and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 10:1167, 2019. DOI: 10.1021/acs.jpclett.9b00299.

[99] Q. Yu and S. Hammes-Schiffer. *J. Phys. Chem. Lett.*, 11:10106, 2020. DOI: 10.1021/acs.jpclett.0c02923.

[100] Q. Yu, F. Pavošević, and S. Hammes-Schiffer. *J. Chem. Phys.*, 152:244123, 2020. DOI: 10.1063/5.0009233.

[101] Q. Yu, S. Roy, and S. Hammes-Schiffer. *J. Chem. Theory Comput.*, 18:7132, 2022. DOI: 10.1021/acs.jctc.2c00938.

[102] Q. Yu, P. E. Schneider, and S. Hammes-Schiffer. *J. Chem. Phys.*, 156:114115, 2022. DOI: 10.1063/5.0085344.

[103] L. Zhao, Z. Tao, F. Pavošević, A. Wildman, S. Hammes-Schiffer, and X. Li. *J. Phys. Chem. Lett.*, 11:4052, 2020. DOI: 10.1021/acs.jpclett.0c00701.

[104] L. Zhao, A. Wildman, Z. Tao, P. Schneider, S. Hammes-Schiffer, and X. Li. *J. Chem. Phys.*, 153:224111, 2020. DOI: 10.1063/5.0031019.

Appendix A

AOINTS: Q-CHEM's Integrals Engine

A.1 Historical Overview

Within the Q-CHEM program, an Atomic Orbital integrals (AOINTS) package has been developed which, while relatively invisible to the user, is one of the keys to the overall speed and efficiency of the Q-CHEM program. The key challenge was summarized by Gill: [17]

> Ever since Boys' introduction of Gaussian basis sets to quantum chemistry in 1950, the calculation and handling of the notorious two-electron repulsion integrals (ERIs) over Gaussian functions has been an important avenue of research for practicing computational chemists. Indeed, the emergence of practically useful computer programs has been fueled in no small part by the development of sophisticated algorithms to compute the very large number of ERIs that are involved in calculations on molecular systems of even modest size.

The ERI engine of any competitive quantum chemistry software package will be one of the most complicated aspects of the package as whole. Coupled with the importance of such an engine's efficiency, a useful yardstick of a program's anticipated performance can be quickly measured by considering the components of its ERI engine. In recent times, developers at Q-CHEM, Inc. have made significant contributions to the advancement of ERI algorithm technology (for example, see Refs. 1,14–20,22,23), and it is not surprising that Q-CHEM's AOINTS package is considered the most advanced of its kind.

Prior to the 1950s, the most difficult step in the systematic application of Schrödinger wave mechanics to chemistry was the calculation of the notorious two-electron integrals that measure the repulsion between electrons. Boys [5] showed that this step can be made easier (although still time consuming) if Gaussian, rather than Slater, orbitals are used in the basis set. Following the landmark paper of computational chemistry [6] (again due to Boys) programs were constructed that could calculate all the ERIs that arise in the treatment of a general polyatomic molecule with s and p orbitals. However, the programs were painfully slow and could only be applied to the smallest of molecular systems.

In 1969, Pople constructed a breakthrough ERI algorithm, a hundred time faster than its predecessors. The algorithm remains the fastest available for its associated integral classes and is now referred to as the Pople-Hehre axis-switch method. [29]

Over the two decades following Pople's initial development, an enormous amount of research effort into the construction of ERIs was documented, which built on Pople's original success. Essentially, the advances of the newer algorithms could be identified as either better coping with angular momentum (L) or, contraction (K); each new method increasing the speed and application of quantum mechanics to solving real chemical problems.

By 1990, another barrier had been reached. The contemporary programs had become sophisticated and both academia and industry had begun to recognize and use the power of *ab initio* quantum chemistry, but the software was struggling with "dusty deck syndrome" and it had become increasingly difficult for it to keep up with the rapid advances in hardware development. Vector processors, parallel architectures and the advent of the graphical user interface were all demanding radically different approaches to programming and it had become clear that a fresh start, with a clean slate, was both inevitable and desirable. Furthermore, the integral bottleneck had re-emerged in a new guise and the standard programs were now hitting the N^2 wall. Irrespective of the speed at which ERIs could be computed, the unforgiving fact remained that the number of ERIs required scaled quadratically with the size of the system.

The Q-CHEM project was established to tackle this problem and to seek new methods that circumvent the N^2 wall. Fundamentally new approaches to integral theory were sought and the ongoing advances that have resulted[3,7,12,30,33] have now placed Q-CHEM firmly at the vanguard of the field. It should be emphasized, however, that the $\mathcal{O}(N)$ methods that we have developed still require short-range ERIs to treat interactions between nearby electrons, thus the importance of contemporary ERI code remains.

The chronological development and evolution of integral methods can be summarized by considering a time line showing the years in which important new algorithms were first introduced. These are best discussed in terms of the type of ERI or matrix elements that the algorithm can compute efficiently.

1950	Boys	5	ERIs with low L and low K
1969	Pople	29	ERIs with low L and high K
1976	Dupuis	13	Integrals with any L and low K
1978	McMurchie	25	Integrals with any L and low K
1982	Almlöf	4	Introduction of the direct SCF approach
1986	Obara	26	Integrals with any L and low K
1988	Head-Gordon	20	Integrals with any L and low K
1991	Gill	16,17	Integrals with any L and any K
1994	White	33	J matrix in linear work
1996	Schwegler	30,31	HF exchange matrix in linear work
1997	Challacombe	7	Fock matrix in linear work

A.2 Calculating Electron Repulsion Integrals (ERIs)

A.2.1 Overview

The area of molecular integrals with respect to Gaussian basis functions has recently been reviewed[15] and the user is referred to this review for deeper discussions and further references to the general area. The purpose of this short account is to present the basic approach, and in particular, the implementation of ERI algorithms and aspects of interest to the user in the AOINTS package which underlies the Q-CHEM program.

We begin by observing that all of the integrals encountered in an *ab initio* calculation, of which overlap, kinetic energy, multipole moment, internuclear repulsion, nuclear-electron attraction and inter electron repulsion are the best known, can be written in the general form

$$(\mathbf{ab}|\mathbf{cd}) = \int \phi_{\mathbf{a}}(\mathbf{r}_1)\,\phi_{\mathbf{b}}(\mathbf{r}_1)\,\theta(r_{12})\,\phi_{\mathbf{c}}(\mathbf{r}_2)\,\phi_{\mathbf{d}}(\mathbf{r}_2)\;d\mathbf{r}_1 d\mathbf{r}_2 \tag{A.1}$$

where the basis functions are contracted Gaussians (CGTF)

$$\phi_{\mathbf{a}}(\mathbf{r}) = (x-A_x)^{a_x}(y-A_y)^{a_y}(z-A_z)^{a_z}\sum_{i=1}^{K_a} D_{ai}\,e^{-\alpha_i|\mathbf{r}-\mathbf{A}|^2} \tag{A.2}$$

and the operator θ is a two-electron operator. Of the two-electron operators (Coulomb, CASE, anti-Coulomb and delta-function) used in the Q-CHEM program, the most significant is the Coulomb, which leads us to the ERIs.

An ERI is the classical Coulomb interaction, $\theta(x) = 1/x$ in Eq. (A.1), between two charge distributions referred to as bras $(\mathbf{ab}|$ and kets $|\mathbf{cd})$.

A.2.2 Data Structures: Shell Pairs and Quartets

A.2.2.1 Shell-Pair Data

It is common to characterize a bra, a ket and a bra-ket by their degree of contraction and angular momentum. In general, it is more convenient to compile data for shell-pairs rather than basis-function pairs. A shell is defined as that sharing common exponents and centers. For example, in the case of a number of Pople derived basis sets, four basis functions, encompassing a range of angular momentum types (*i.e.*, s, p_x, p_y, p_z on the same atomic center sharing the same exponents constitute a single shell.

The shell-pair data set is central to the success of any modern integral program for three main reasons. First, in the formation of shell-pairs, all pairs of shells in the basis set are considered and categorized as either significant or negligible. A shell-pair is considered negligible if the shells involved are so far apart, relative to their diffuseness, that their overlap is negligible. Given the rate of decay of Gaussian basis functions, it is not surprising that most of the shell-pairs in a large molecule are negligible, that is, the number of significant shell-pairs increases linearly with the size of the molecule. Second, a number of useful intermediates which are frequently required within ERI algorithms should be computed once in shell-pair formation and stored as part of the shell-pair information, particularly those which require costly divisions. This prevents re-evaluating simple quantities. Third, it is useful to sort the shell-pair information by type (*i.e.*, angular momentum and degree of contraction). The reasons for this are discussed below.

Q-CHEM's shell-pair formation offers the option of two basic integral shell-pair cutoff criteria; one based on the integral threshold (*$rem* variable THRESH) and the other relative to machine precision.

Intelligent construction of shell-pair data scales linearly with the size of the basis set, requires a relative amount of CPU time which is almost entirely negligible for large direct SCF calculations, and for small jobs, constitutes approximately 10% of the job time.

A.2.2.2 Shell-Quartets and Integral Classes

Given a sorted list of shell-pair data, it is possible to construct all potentially important shell-quartets by pairing of the shell-pairs with one another. Because the shell-pairs have been sorted, it is possible to deal with batches of integrals of the same type or class (*e.g.*, $(ss|ss)$, $(sp|sp)$, $(dd|dd)$, *etc.*) where an integral class is characterized by both angular momentum (L) and degree of contraction (K). Such an approach is advantageous for vector processors and for semi-direct integral algorithms where the most expensive (high K or L integral classes can be computed once, stored in memory (or disk) and only less expensive classes rebuilt on each iteration.

While the shell-pairs may have been carefully screened, it is possible for a pair of significant shell-pairs to form a shell-quartet which need not be computed directly. Three cases are:

- The quartet is equivalent, by point group symmetry, to another quartet already treated.
- The quartet can be ignored on the basis of cheaply computed ERI bounds[19] on the largest quartet bra-ket.
- On the basis of an incremental Fock matrix build, the largest density matrix element which will multiply any of the bra-kets associated with the quartet may be negligibly small.

Note: Significance and negligibility is always based on the level of integral threshold set by the *$rem* variable THRESH.

A.2.3 Survey of ERI Evaluation

Various aspects of the ERI evaluation problem are described in the following subsections.

A.2.3.1 Fundamental ERI

The fundamental ERI $[ss|ss]^{(0)} \equiv [\mathbf{0}]^{(0)}$, which is the basis of all ERI algorithms, is usually represented as [15]

$$[\mathbf{0}]^{(0)} = D_A D_B D_C D_D \int e^{-\alpha|\mathbf{r}_1-\mathbf{A}|^2} e^{-\beta|\mathbf{r}_1-\mathbf{B}|^2} \left(\frac{1}{r_{12}}\right) e^{-\gamma|\mathbf{r}_2-\mathbf{C}|^2} e^{-\delta|\mathbf{r}_2-\mathbf{D}|^2}\, d\mathbf{r}_1 d\mathbf{r}_2 \tag{A.3}$$

which can be reduced to a one-dimensional integral of the form

$$[\mathbf{0}]^{(0)} = U(2\,\vartheta^2)^{1/2} \left(\frac{2}{\pi}\right)^{1/2} \int_0^1 e^{-Tu^2} du \tag{A.4}$$

and can be efficiently computed using a modified Chebyshev interpolation scheme.[18] Equation (A.4) can also be adapted for the general case $[\mathbf{0}]^{(m)}$ integrals required for most calculations. Following the fundamental ERI, building up to the full bra-ket ERI (or intermediary matrix elements, see later) are the problems of angular momentum and contraction.

Note: Square brackets denote primitive integrals and parentheses denote fully-contracted integrals.

A.2.3.2 Angular Momentum Problem

The fundamental integral is essentially an integral without angular momentum (*i.e.*, it is an integral of the type $[ss|ss]$). Angular momentum, usually depicted by L, has been problematic for efficient ERI formation, evident in the above time line. Initially, angular momentum was calculated by taking derivatives of the fundamental ERI with respect to one of the Cartesian coordinates of the nuclear center. This is an extremely inefficient route, but it works and was appropriate in the early development of ERI methods. Recursion relations[26,27] and the tensor equations[1] are the basis for the modern approaches.

A.2.3.3 Contraction Problem

The contraction problem may be described by considering a general contracted ERI of s-type functions derived from the STO-3G basis set. Each basis function has degree of contraction $K = 3$. Thus, the ERI may be written

$$\begin{aligned}(ss|ss) &= \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{\ell=1}^{3} D_{Ai} D_{Bj} D_{Ck} D_{D\ell} \\ &\quad \times \int e^{-\alpha_i|\mathbf{r}_1-\mathbf{A}|^2} e^{-\beta_j|\mathbf{r}_1-\mathbf{B}|^2} \left(\frac{1}{r_{12}}\right) e^{-\gamma_k|\mathbf{r}_2-\mathbf{C}|^2} e^{-\delta_\ell|\mathbf{r}_2-\mathbf{D}|^2} d\mathbf{r}_1 d\mathbf{r}_2 \\ &= \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{\ell=1}^{3} [s_i s_j|s_k s_\ell]\end{aligned} \tag{A.5}$$

and requires 81 primitive integrals for the single ERI. The problem escalates dramatically for more highly contracted sets (STO-6G, 6-311G) and has been the motivation for the development of techniques for shell-pair modeling,[2] in which a second shell-pair is constructed with fewer primitives that the first, but introduces no extra error relative to the integral threshold sought.

The Pople-Hehre axis-switch method[29] is excellent for high contraction low angular momentum integral classes.

A.2.4 Efficiency of ERI Evaluation

It is often stated that the cost of SCF calculations scales with size as $\mathcal{O}(N^4)$, but the veracity of this assertion depends critically on what is meant by N. For fixed molecule size, the cost may exhibit quartic scaling due to the growth in the number of ERIs, $(\mu\nu|\lambda\sigma)$. However, what is more often intended in a discussion of the asymptotic cost of an electronic structure calculation is the scaling with respect to *molecular* size, for a fixed choice of basis set. This corresponds to measuring size (N) in terms of the number of atoms, in which case the asymptotic cost should be $\mathcal{O}(N^2)$, as discussed below.

A.2.4.1 Quadratic Scaling

The success of quantitative modern quantum chemistry, relative to its primitive, qualitative beginnings, can be traced to two sources: better algorithms and better computers. While the two technologies continue to improve rapidly, efforts are heavily thwarted by the fact that the total number of ERIs increases quadratically with the size of the molecular system. Even large increases in ERI algorithm efficiency yield only moderate increases in applicability, hindering the more widespread application of *ab initio* methods to areas of, perhaps, biochemical significance where semi-empirical techniques[10,11] have already proven so valuable.

Thus, the elimination of quadratic scaling algorithms has been the theme of many research efforts in quantum chemistry throughout the 1990s and has seen the construction of many alternative algorithms to alleviate the problem. Johnson was the first to implement DFT exchange/correlation functionals whose computational cost scaled linearly with system size.[21] This paved the way for the most significant breakthrough in the area with the linear scaling CFMM algorithm[33] leading to linear scaling DFT calculations.[34] Further breakthroughs have been made with traditional theory in the form of the QCTC[7–9] and ONX[30,31] algorithms, while more radical approaches[3,12] may lead to entirely new approaches to *ab initio* calculations. Investigations into the quadratic Coulomb problem has not only yielded linear scaling algorithms, but is also providing large insights into the significance of many molecular energy components.

Linear scaling Coulomb and SCF exchange/correlation algorithms are not the end of the story as the $\mathcal{O}(N^3)$ diagonalization step has been rate limiting in semi-empirical techniques and, been predicted to become rate limiting in *ab initio* approaches in the medium term.[32] However, divide-and-conquer techniques[24,35–37] and the recently developed quadratically convergent SCF algorithm[28] show great promise for reducing this problem.

A.2.4.2 Algorithm Selection

No single ERI algorithm is available to efficiently handle all integral classes; rather, each tends to have specific integral classes where the specific algorithm outperforms the alternatives. The PRISM algorithm[16] is an intricate collection of pathways and steps in which the path chosen is that which is the most efficient for a given class. It appears that the most appropriate path for a given integral class depends on the relative position of the contraction step (lowly contracted bra-kets prefer late contraction, highly contracted bra-kets are most efficient with early contraction steps).

Careful studies have provided FLOP counts which are the current basis of integral algorithm selection, although care must be taken to ensure that algorithms are not rate limited by MOPs.[14] Future algorithm selection criteria will take greater account of memory, disk, chip architecture, cache size, vectorization and parallelization characteristics of the hardware, many of which are already exist within Q-CHEM.

A.2.4.3 More Efficient Hartree–Fock Gradient and Hessian Evaluations

Q-CHEM combines the Head-Gordon–Pople (HGP) method[20] and the COLD prism method[1] for Hartree-Fock gradient and Hessian evaluations. All two-electron four-center integrals are classified according to their angular momentum

types and degrees of contraction. For each type of integrals, the program chooses one with a lower cost. In practice, the HGP method is chosen for most integral classes in a gradient or Hessian calculation, and thus it dominates the total CPU time.

Recently the HGP codes within Q-CHEM were completely rewritten for the evaluation of the P Π^x P term in the gradient evaluation, and the P Π^{xy} P term in the Hessian evaluation. Our emphasis is to improve code efficiency by reducing cache misses rather than by reducing FLOP counts. Some timing results from a Hartree-Fock calculation on the azidothymidine (AZT) molecule are shown below.

Basis Set	**AIX**			**Linux**		
	Gradient Evaluation: P Π^x P Term					
	Old	New	New/Old	Old	New	New/Old
3-21G	34 s	20 s	0.58	25 s	14 s	0.56
6-31G**	259 s	147 s	0.57	212 s	120 s	0.57
DZ	128 s	118 s	0.92	72 s	62 s	0.86
cc-pVDZ	398 s	274 s	0.69	308 s	185 s	0.60
	Hessian Evaluation: P Π^{xy} P term					
	Old	New	New/Old	Old	New	New/Old
3-21G	294 s	136 s	0.46	238 s	100 s	0.42
6-31G**	2520 s	976 s	0.39	2065 s	828 s	0.40
DZ	631 s	332 s	0.53	600 s	230 s	0.38
cc-pVDZ	3202 s	1192 s	0.37	2715 s	866 s	0.32

Table A.1: AIX timings were obtained on an IBM RS/6000 workstation with AIX4 operating system, and Linux timings on an Opteron cluster where the Q-CHEM executable was compiled with an Intel 32-bit compiler.

A.3 User-Controllable Variables

AOINTS has been optimally constructed so that the fastest integral algorithm for ERI calculation is chosen for the given integral class and batch. Thus, the user has not been provided with the necessary variables for overriding the program's selection process. The user is, however, able to control the accuracy of the cutoff used during shell-pair formation (METECO) and the integral threshold (THRESH). In addition, the user can force the use of the direct SCF algorithm (DIRECT_SCF) and increase the default size of the integrals storage buffer (INCORE_INTS_BUFFER).

Currently, some of Q-CHEM's linear scaling algorithms, such as QCTC and ONX algorithms, require the user to specify their use. It is anticipated that further research developments will lead to the identification of situations in which these, or combinations of these and other algorithms, will be selected automatically by Q-CHEM in much the same way that PRISM algorithms choose the most efficient pathway for given integral classes.

References and Further Reading

[1] T. R. Adams, R. D. Adamson, and P. M. W. Gill. *J. Chem. Phys.*, 107:124, 1997. DOI: 10.1063/1.474359.

[2] R. D. Adamson. Shell-pair economisation. Master's thesis, Massey University, Palmerston North, New Zealand, 1995.

[3] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill. *Chem. Phys. Lett.*, 254:329, 1996. DOI: 10.1016/0009-2614(96)00280-1.

[4] J. Almlöf, K. Fægri Jr., and K. Korsell. *J. Comput. Chem.*, 3:385, 1982. DOI: 10.1002/jcc.540030314.

[5] S. F. Boys. *Proc. Roy. Soc. Ser. A*, 200:542, 1950. DOI: 10.1098/rspa.1950.0036.

[6] S. F. Boys, G. B. Cook, C. M. Reeves, and I. Shavitt. *Nature*, 178:1207, 1956. DOI: 10.1038/1781207a0.

[7] M. Challacombe and E. Schwegler. *J. Chem. Phys.*, 106:5526, 1997. DOI: 10.1063/1.473575.

[8] M. Challacombe, E. Schwegler, and J. Almlöf. Modern developments in hartree-fock theory: Fast methods for computing the coulomb matrix. Technical report, 1996.

[9] M. Challacombe, E. Schwegler, and J. Almlöf. *J. Chem. Phys.*, 104:4685, 1996. DOI: 10.1063/1.471163.

[10] M. J. S. Dewar. *The Molecular Orbital Theory of Organic Chemistry*. McGraw-Hill, New York, 1969.

[11] M. J. S. Dewar. *Org. Mass. Spect.*, 28:305, 1993. DOI: 10.1002/oms.1210280407.

[12] J. P. Dombroski, S. W. Taylor, and P. M. W. Gill. *J. Phys. Chem.*, 100:6272, 1996. DOI: 10.1021/jp952841b.

[13] M. Dupuis, J. Rys, and H. F. King. *J. Chem. Phys.*, 65:111, 1976. DOI: 10.1063/1.432807.

[14] M. J. Frisch, B. G. Johnson, P. M. W. Gill, D. J. Fox, and R. H. Nobes. *Chem. Phys. Lett.*, 206:225, 1993. DOI: 10.1016/0009-2614(93)85545-Y.

[15] P. M. W. Gill. *Adv. Quantum Chem.*, 25:141, 1994. DOI: 10.1016/S0065-3276(08)60019-2.

[16] P. M. W. Gill and J. A. Pople. *Int. J. Quantum Chem.*, 40:753, 1991. DOI: 10.1002/qua.560400605.

[17] P. M. W. Gill, M. Head-Gordon, and J. A. Pople. *J. Phys. Chem.*, 94:5564, 1990. DOI: 10.1021/j100377a031.

[18] P. M. W. Gill, B. G. Johnson, and J. A. Pople. *Int. J. Quantum Chem.*, 40:745, 1991. DOI: 10.1002/qua.560400604.

[19] P. M. W. Gill, B. G. Johnson, and J. A. Pople. *Chem. Phys. Lett.*, 217:65, 1994. DOI: 10.1016/0009-2614(93)E1340-M.

[20] M. Head-Gordon and J. A. Pople. *J. Chem. Phys.*, 89:5777, 1988. DOI: 10.1063/1.455553.

[21] B. G. Johnson. *Development, Implementation, and Performance of Efficient Methodologies for Density Functional Calculations*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1993.

[22] B. G. Johnson, P. M. W. Gill, and J. A. Pople. *Chem. Phys. Lett.*, 206:229, 1993. DOI: 10.1016/0009-2614(93)85546-Z.

[23] B. G. Johnson, P. M. W. Gill, and J. A. Pople. *Chem. Phys. Lett.*, 206:239, 1993. DOI: 10.1016/0009-2614(93)85547-2.

[24] T.-S. Lee, D. M. York, and W. Yang. *J. Chem. Phys.*, 105:2744, 1996. DOI: 10.1063/1.472136.

[25] L. E. McMurchie and E. R. Davidson. *J. Comput. Phys.*, 26:218, 1978. DOI: 10.1016/0021-9991(78)90092-X.

[26] S. Obara and A. Saika. *J. Chem. Phys.*, 84:3963, 1986. DOI: 10.1063/1.450106.

[27] S. Obara and A. Saika. *J. Chem. Phys.*, 89:1540, 1988. DOI: 10.1063/1.455717.

[28] C. Ochsenfeld and M. Head-Gordon. *Chem. Phys. Lett.*, 270:399, 1997. DOI: 10.1016/S0009-2614(97)00402-8.

[29] J. A. Pople and W. J. Hehre. *J. Comput. Phys.*, 27:161, 1978. DOI: 10.1016/0021-9991(78)90001-3.

[30] E. Schwegler and M. Challacombe. *J. Chem. Phys.*, 105:2726, 1996. DOI: 10.1063/1.472135.

[31] E. Schwegler and M. Challacombe. *J. Chem. Phys.*, 106:9708, 1996. DOI: 10.1063/1.473833.

[32] D. L. Strout and G. E. Scuseria. *J. Chem. Phys.*, 102:8448, 1995. DOI: 10.1063/1.468836.

[33] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon. *Chem. Phys. Lett.*, 230:8, 1994. DOI: 10.1016/0009-2614(94)01128-1.

[34] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon. *Chem. Phys. Lett.*, 253:268, 1996. DOI: 10.1016/0009-2614(96)00175-3.

[35] W. Yang. *Phys. Rev. A*, 44:7823, 1991. DOI: 10.1103/PhysRevA.44.7823.

[36] W. Yang. *Phys. Rev. Lett.*, 66:1438, 1991. DOI: 10.1103/PhysRevLett.66.1438.

[37] W. Yang and T.-S. Lee. *J. Chem. Phys.*, 103:5674, 1995. DOI: 10.1063/1.470549.

Appendix B

Q-CHEM Quick Reference

B.1 Text Input Summary

- Users are able to enter input sections in any order; see Table 3.1 for a complete list of input sections.
- Each input section must be terminated with *$end*.
- Not all input sections are required, but *$rem* and *$molecule* are compulsory.
- The entire Q-CHEM input is case-insensitive.
- Multiple jobs are separated by the string `@@@` on a single line.

B.1.1 Descriptions of Some Q-CHEM Input Sections

B.1.1.1 *$molecule*

Four methods are available for inputing geometry information:

- *Z*-matrix (Ångstroms and degrees):

```
$molecule
    [charge] [multiplicity]
    [Z-matrix]
    [blank line, if parameters are being used]
    [Z-matrix parameters, if used]
$end
```

- Cartesian Coordinates (Ångstroms):

```
$molecule
    [charge] [multiplicity]
    [Cartesian coordinates]
    [blank line, if parameter are being used]
    [Coordinate parameters, if used]
$end
```

- Read from a previous calculation:

```
$molecule
   read
$end
```

- Read from a file:

```
$molecule
   read filename
$end
```

B.1.1.2 *$rem*

See also the list of *$rem* variables at the end of this Appendix. The general format is:

```
$rem
   REM_VARIABLE   VALUE   [optional comment]
$end
```

although specifying "REM_VARIABLE = VALUE" is also acceptable, *i.e.*, the equals sign is ignored.

B.1.1.3 *$basis*

The format for the user–defined basis section is as follows:

```
$basis
```

$$
\begin{array}{lllll}
X & 0 & & & \\
L & K & scale & & \\
\alpha_1 & C_1^{L_{\min}} & C_1^{L_{\min}+1} & \ldots & C_1^{L_{\max}} \\
\alpha_2 & C_2^{L_{\min}} & C_2^{L_{\min}+1} & \ldots & C_2^{L_{\max}} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\alpha_K & C_K^{L_{\min}} & C_K^{L_{\min}+1} & \ldots & C_K^{L_{\max}}
\end{array}
$$

```
****
$end
```

where

X	Atomic symbol of the atom (atomic number not accepted)
L	Angular momentum symbol (S, P, SP, D, F, G)
K	Degree of contraction of the shell (integer)
$scale$	Scaling to be applied to exponents (default is 1.00)
a_i	Gaussian primitive exponent (positive real number)
C_i^L	Contraction coefficient for each angular momentum (non–zero real numbers).

Atoms are terminated with $\star\star\star\star$ and the complete basis set is terminated with the *$end* keyword terminator. No blank lines can be incorporated within the general basis set input. Note that more than one contraction coefficient per line is one required for compound shells like SP. As with all Q-CHEM input deck information, all input is case–insensitive.

B.1.1.4 *$comment*

Users are able to add comments to the input file outside keyword input sections, which will be ignored by the program. This can be useful as reminders to the user, or perhaps, when teaching another user to set up inputs. Comments can also be provided in a *$comment* block, which is actually redundant given that the entire input deck is copied to the output file.

```
$comment
   User comments - copied to output file
$end
```

B.1.1.5 *$ecp*

```
$ecp
```
For each atom that will bear an ECP

Chemical symbol for the atom

ECP name; the L value for the ECP; number of core electrons removed

For each ECP component (in the order unprojected, $\hat{P}_0$, $\hat{P}_1$, , $\hat{P}_{L-1}$

The component name

The number of Gaussians in the component

For each Gaussian in the component

The power of r; the exponent; the contraction coefficient

```
****
$end
```

Note: (1) All of the information in the *$ecp* block is case–insensitive.

(2) The L value may not exceed 4. That is, nothing beyond G projectors is allowed.

(3) The power of r (which includes the Jacobian r^2 factor) must be 0, 1 or 2.

B.1.1.6 *$empirical_dispersion*

```
$empirical_dispersion
    S6 S6_value
    D D_value
    C6 element_1 C6_value_for_element_1 element_2 C6_value_for_element_2
    VDW_RADII element_1 radii_for_element_1 element_2 radii_for_element_2
$end
```

Note: This section is only for values that the user wants to change from the default values recommended by Grimme.

B.1.1.7 *$external_charges*

If the *$external_charges* keyword is present, Q-CHEM scans for a set of external charges to be incorporated into a calculation. The format is shown below and consists of Cartesian coordinates and the value of the point charge, with one charge per line. The charge is in atomic units and the coordinates are in Ångstroms, unless bohrs are selected by

setting the *$rem* keyword INPUT_BOHR to TRUE. The external charges are rotated with the molecule into the standard nuclear orientation and are specified in the following format:

```
$external_charges
   x-coord1  y-coord1  z-coord1  charge1
   x-coord2  y-coord2  z-coord2  charge2
   x-coord3  y-coord3  z-coord3  charge3
$end
```

In addition, the user can request to add a charged cage around the molecule (for so-called "charge stabilization" calculations) using the keyword ADD_CHARGED_CAGE. See Section 7.10.12 for details.

B.1.1.8 *$intracule*

```
$intracule
```

int_type	0	Compute $P(u)$ only
	1	Compute $M(v)$ only
	2	Compute $W(u,v)$ only
	3	Compute $P(u)$, $M(v)$ and $W(u,v)$
	4	Compute $P(u)$ and $M(v)$
	5	Compute $P(u)$ and $W(u,v)$
	6	Compute $M(v)$ and $W(u,v)$
u_points		Number of points, start, end.
v_points		Number of points, start, end.
moments	0–4	Order of moments to be computed ($P(u)$ only).
derivs	0–4	order of derivatives to be computed ($P(u)$ only).
accuracy	n	(10^{-n}) specify accuracy of intracule interpolation table ($P(u)$ only).

```
$end
```

B.1.1.9 *$isotopes*

Note that masses should be given in atomic units.

```
$isotopes
   number_extra_loops  tp_flag
      number_of_atoms  [temp pressure]
      atom_number1 mass1
      atom_number2 mass2
   ...
$end
```

B.1.1.10 *$multipole_field*

A multipole field can be applied to the molecule under investigation by specifying the *$multipole_field* input section. Each line in this section consists of a single component of the applied field, in the following format:

```
$multipole_field
   field_component_1   value_1
   field_component_2   value_2
$end
```

Each *field_component* is stipulated using the Cartesian representation: x, y, or z for dipole field components; xx, xy, etc. for quadrupole field components, ... The value of each field component should be provided in atomic units. Contrary to the convention in many textbooks, within Q-CHEM electric field lines are defined to run from negative charge to positive charge.[1,2]

A dipole field component can also be specified using the indices of two atoms followed by the field magnitude. The field component will always be aligned parallel to the vector connecting *atom1* and *atom2*, following the coordinates as the atoms move.

```
$multipole_field
   atom1 atom2 value
$end
```

B.1.1.11 *$nbo*

Refer to Chapter 10 and the NBO manual for further information. Note that the NBO *$rem* variable must be set to ON to initiate the NBO package.

```
$nbo
   [ NBO options ]
$end
```

B.1.1.12 *$occupied*

```
$occupied
   1  2  3  4 ...  nalpha
   1  2  3  4 ...  nbeta
$end
```

B.1.1.13 *$opt*

Note that units are in Ångstroms and degrees. Also see the summary in the next section of this Appendix.

```
$opt
CONSTRAINT
stre  atom1  atom2  value
...
bend  atom1  atom2  atom3  value
...
```

```
    outp  atom1  atom2  atom3  atom4  value
    ...
    tors  atom1  atom2  atom3  atom4  value
    ...
    linc  atom1  atom2  atom3  atom4  value
    ...
    linp  atom1  atom2  atom3  atom4  value
    ...
    ENDCONSTRAINT

    FIXED
    atom   coordinate_reference
    ...
    ENDFIXED

    DUMMY
    idum   type   list_length   defining_list
    ...
    ENDDUMMY

    CONNECT
    atom   list_length   list
    ...
    ENDCONNECT
    $end
```

B.1.1.14 *$svp*

```
$svp
<KEYWORD>=<VALUE>, <KEYWORD>=<VALUE>,...
<KEYWORD>=<VALUE>
$end
```

For example, the section may look like this:

```
$svp
    RHOISO=0.001, DIELST=78.39, NPTLEB=110
$end
```

B.1.1.15 *$svpirf*

```
$svpirf
    <# point> <x point> <y point> <z point> <charge> <grid weight>
    <# point> <x normal> <y normal> <z normal>
$end
```

B.1.1.16 *$plots*

`$plots`

One comment line

Specification of the 3–D mesh of points on 3 lines:

$N_x \quad x_{\min} \quad x_{\max}$

$N_y \quad y_{\min} \quad y_{\max}$

$N_z \quad z_{\min} \quad z_{\max}$

A line with 4 integers indicating how many things to plot:

$N_{\mathrm{MO}} \quad N_{\mathrm{Rho}} \quad N_{\mathrm{Trans}} \quad N_{\mathrm{DA}}$

An optional line with the integer list of MO's to evaluate (only if $N_{\mathrm{MO}} > 0$)

MO(1) MO(2) ... MO(N_{MO})

An optional line with the integer list of densities to evaluate (only if $N_{\mathrm{Rho}} > 0$)

Rho(1) Rho(2) ... Rho(N_{Rho})

An optional line with the integer list of transition densities (only if $N_{\mathrm{Trans}} > 0$)

Trans(1) Trans(2) ... Trans(N_{Trans})

An optional line with states for detachment/attachment densities (if $N_{\mathrm{DA}} > 0$)

DA(1) DA(2) ... DA(N_{DA})

`$end`

B.1.1.17 *$localized_diabatization*

`$localized_diabatization`

One comment line.

One line with an an array of adiabatic states to mix together.

$< adiabat1 > \quad < adiabat2 > \quad < adiabat3 > \ldots$

`$end`

Note: We count adiabatic states such that the first excited state is $< adiabat >= 1$, the fifth is $< adiabat >= 5$, and so forth.

B.1.1.18 *$van_der_waals*

Note: All radii are given in Ångstroms.

```
$van_der_waals
   1
   atomic_number   VdW_radius
$end
```

(alternative format)

```
$van_der_waals
   2
   sequential_atom_number   VdW_radius
$end
```

B.1.1.19 *$xc_functional*

```
$xc_functional
   X  exchange_symbol  coefficient
   X  exchange_symbol  coefficient
   ...
   C  correlation_symbol  coefficient
   C  correlation_symbol  coefficient
   ...
   K  coefficient
$end
```

B.1.2 *$rem* Variable List

B.1.2.1 Overview

The general format of the *$rem* input for Q-CHEM text input files is simply as follows:

```
$rem
   rem_variable   rem_option   [comment]
   rem_variable   rem_option   [comment]
$end
```

This input is not case sensitive. The following sections contain the names and options of available *$rem* variables for users. The format for describing each *$rem* variable is as follows:

REM_VARIABLE
A short description of what the variable controls
TYPE:
Defines the variable as either INTEGER, LOGICAL or STRING.
DEFAULT:
Describes Q-CHEM's internal default, if any exists.
OPTIONS:
Lists options available for the user
RECOMMENDATION:
Gives a quick recommendation.

B.1.2.2 General

BASIS	BASIS_LIN_DEP_THRESH
EXCHANGE	CORRELATION
ECP	JOBTYPE
METHOD	PURECART

B.1.2.3 SCF Control

BASIS2	BASISPROJTYPE
DIIS_PRINT	DIIS_SUBSPACE_SIZE
DIRECT_SCF	INCFOCK
MAX_DIIS_CYCLES	MAX_SCF_CYCLES
PSEUDO_CANONICAL	SCF_ALGORITHM
SCF_CONVERGENCE	SCF_FINAL_PRINT
SCF_GUESS	SCF_GUESS_MIX
SCF_GUESS_PRINT	SCF_PRINT
THRESH	THRESH_DIIS_SWITCH
UNRESTRICTED	VARTHRESH
S2THRESH	

B.1.2.4 DFT Options

CORRELATION	EXCHANGE
FAST_XC	INC_DFT
INCDFT_DENDIFF_THRESH	INCDFT_GRIDDIFF_THRESH
INCDFT_DENDIFF_VARTHRESH	INCDFT_GRIDDIFF_VARTHRESH
XC_GRID	XC_SMART_GRID

B.1.2.5 Large Molecules

CFMM_ORDER	DIRECT_SCF
EPAO_ITERATE	EPAO_WEIGHTS
GRAIN	INCFOCK
INTEGRAL_2E_OPR	INTEGRALS_BUFFER
LIN_K	MEM_STATIC
MEM_TOTAL	METECO
OMEGA	PAO_ALGORITHM
PAO_METHOD	THRESH
VARTHRESH	RI_J
RI_K	ARI
ARI_R0	ARI_R1
S2THRESH	

B.1.2.6 Correlated Methods

AO2MO_DISK	CD_ALGORITHM
CORE_CHARACTER	CORRELATION
MEM_STATIC	MEM_TOTAL
N_FROZEN_CORE	N_FROZEN_VIRTUAL
PRINT_CORE_CHARACTER	

B.1.2.7 Correlated Methods Handled by CCMAN and CCMAN2

Most of these *$rem* variables that start CC_.

These are relevant for CCSD and other CC methods (OD, VOD, CCD, QCCD, CCVB-SD, etc).

CC_CANONIZE	CC_RESTART_NO_SCF
CC_T_CONV	CC_DIIS_SIZE
CC_DIIS_FREQ	CC_DIIS_START
CC_DIIS_MAX_OVERLAP	CC_DIIS_MIN_OVERLAP
CC_RESTART	CC_SAVEAMPL

These options are only relevant to methods involving orbital optimization (OOCD, VOD, QCCD, VQCCD):

CC_MP2NO_GUESS	CC_MP2NO_GRAD
CC_DIIS	CC_DIIS12_SWITCH
CC_THETA_CONV	CC_THETA_GRAD_CONV
CC_THETA_STEPSIZE	CC_RESET_THETA
CC_THETA_GRAD_THRESH	CC_HESS_THRESH
CC_ED_CCD	CC_QCCD_THETA_SWITCH
CC_PRECONV_T2Z	CC_PRECONV_T2Z_EACH
CC_PRECONV_FZ	CC_ITERATE_OV
CC_CANONIZE_FREQ	CC_CANONIZE_FINAL

Properties and optimization:

CC_REF_PROP	CC_REF_PROP_TE
CC_FULLRESPONSE	

B.1.2.8 Perfect Pairing, Coupled-Cluster Valence-Bond, and Related Methods

CCVB_METHOD	CCVB_GUESS
GVB_N_PAIRS	GVB_LOCAL
GVB_ORB_MAX_ITER	GVB_RESTART
GVB_ORB_CONV	GVB_ORB_SCALE
GVB_AMP_SCALE	GVB_DO_SANO
GVB_PRINT	

B.1.2.9 Excited States: CIS, TDDFT, SF-XCIS and SOS-CIS(D)

CIS_CONVERGENCE	CIS_GUESS_DISK
CIS_GUESS_DISK_TYPE	CIS_N_ROOTS
CIS_RELAXED_DENSITY	CIS_SINGLETS
CIS_STATE_DERIV	CIS_TRIPLETS
MAX_CIS_CYCLES	RPA
XCIS	SPIN_FLIP_XCIS

B.1.2.10 Excited States: EOM-CC and CI Methods

Those are keywords relevant to EOM-CC and CI methods handled by CCMAN/CCMAN2. Most of these *$rem* variables that start CC_ and EOM_.

EOM_DAVIDSON_CONVERGENCE
EOM_DAVIDSON_MAXVECTORS
EOM_DAVIDSON_THRESHOLD
EOM_DAVIDSON_MAX_ITER
EOM_NGUESS_DOUBLES
EOM_NGUESS_SINGLES
EOM_DOEXDIAG
EOM_PRECONV_DOUBLES
EOM_PRECONV_SINGLES
EOM_PRECONV_SD
EOM_IPEA_FILTER
EOM_FAKE_IPEA
CC_REST_AMPL
CC_REST_TRIPLES
CC_EOM_PROP
CC_TRANS_PROP
CC_STATE_TO_OPT
CC_EOM_PROP
CC_EOM_PROP_TE
CC_FULLRESPONSE

B.2 Geometry Optimization, Frequencies, & Properties

B.2.1 Survey of Job Control Options

B.2.1.1 General

CIS_STATE_DERIV
FDIFF_STEPSIZE
GEOM_OPT_COORDS
GEOM_OPT_DMAX
GEOM_OPTHESSIAN
GEOM_OPT_LINEAR_ANGLE
GEOM_OPT_MAX_CYCLES
GEOM_OPT_MAX_DIIS
GEOM_OPT_MODE
GEOM_OPT_PRINT
GEOM_OPTSYMFLAG
GEOM_OPT_PRINT
GEOM_OPTTOL_ENERGY
GEOM_OPT_TOL_DISPLACEMENT
GEOM_OPT_TOL_ENERGY
GEOM_OPT_TOL_GRADIENT
GEOMP_OPT_UPDATE
IDERIV
JOBTYPE
SCF_GUESS_ALWAYS
CC_STATE_TO_OPT

B.2.1.2 Vibrational Analysis

DORAMAN
CPSCF_NSEG
FDIFF_STEPSIZE
IDERIV
ISOTOPES
JOBTYPE
VIBMAN_PRINT
ANHAR
VCI
FDIFF_DER
MODE_COUPLING
IGNORE_LOW_FREQ
FDIFF_STEPSIZE_QFF

B.2.1.3 Reaction Coordinate Following

JOBTYPE
RPATH_COORDS
RPATH_DIRECTION
RPATH_MAX_CYCLES
RPATH_MAX_STEPSIZE
RPATH_PRINT
RPATH_TOL_DISPLACEMENT

B.2.1.4 NMR Calculations

D_CPSCF_PERTNUM	D_SCF_CONV_1
D_SCF_CONV_2	D_SCF_DIIS
D_SCF_MAX_1	D_SCF_MAX_2
JOBTYPE	

B.2.1.5 Wave Function Analysis and Properties

CHEMSOL	CHEMSOL_EFIELD
CHEMSOL_NN	CHEM_SOL_PRINT
CIS_RELAXED_DENSITY	ESP_GRID
INTRACULE	MAGNET
MULTIPOLE_ORDER	NBO
POP_MULLIKEN	PRINT_DIST_MATRIX
PRINT_ORBITALS	READ_VDW
RESPONSE	SOLUTE_RADIUS
SOLVENT_DIELECTRIC	STABILITY_ANALYSIS
WAVEFUNCTION_ANALYSIS	WRITE_WFN

B.2.1.6 Symmetry

CC_SYMMETRY	
POINT_GROUP_SYMMETRY	INTEGRAL_SYMMETRY
SYMMETRY_DECOMPOSITION	SYM_TOL

B.2.1.7 Printing Options

CC_PRINT	CHEMSOL_PRINT
DIIS_PRINT	GEOM_OPT_PRINT
MOM_PRINT	PRINT_CORE_CHARACTER
PRINT_DIST_MATRIX	PRINT_GENERAL_BASIS
PRINT_ORBITALS	RPATH_PRINT
SCF_FINAL_PRINT	SCF_GUESS_PRINT
SCF_PRINT	VIBMAN_PRINT
WRITE_WFN	

B.2.1.8 Resource Control

MEM_TOTAL	MEM_STATIC
AO2MO_DISK	CC_MEMORY
INTEGRALS_BUFFER	MAX_SUB_FILE_NUM
DIRECT_SCF	

B.2.2 Geometry Optimization with General Constraints

CONSTRAINT and ENDCONSTRAINT define the beginning and end, respectively, of the constraint section of *$opt* within which users may specify up to six different types of constraints:

interatomic distances
Values in Ångstroms; *value* > 0:
```
stre    atom1    atom2    value
```

angles
Values in degrees, $0 \leq value \leq 180$; *atom2* is the middle atom of the bend:
```
bend    atom1    atom2    atom3    value
```

out–of–plane–bends
Values in degrees, $-180 \leq value \leq 180$ *atom2*; angle between *atom4* and the *atom1–atom2–atom3* plane:
```
outp    atom1    atom2    atom3    atom4    value
```

dihedral angles
Values in degrees, $-180 \leq value \leq 180$; angle the plane *atom1–atom2–atom3* makes with the plane *atom2–atom3–atom4*:
```
tors    atom1    atom2    atom3    atom4    value
```

coplanar bends
Values in degrees, $-180 \leq value \leq 180$; bending of *atom1–atom2–atom3* in the plane *atom2–atom3–atom4*:
```
linc    atom1    atom2    atom3    atom4    value
```

perpendicular bends
Values in degrees, $-180 \leq value \leq 180$; bending of *atom1–atom2–atom3* perpendicular to the plane *atom2–atom3–atom4*:
```
linp    atom1    atom2    atom3    atom4    value
```

Absolute atom positions can be frozen with the *FIXED* section. The section starts with the *FIXED* keyword as the first line and ends with the *ENDFIXED* keyword on the last. The format to fix a coordinate or coordinates of an atom is:

```
atom    coordinate_reference
```

coordinate_reference can be any combination of up to three characters X, Y and Z to specify the coordinate(s) to be fixed: X, Y, Z, *XY*, *XZ*, *YZ*, *XYZ*. The fixing characters must be next to each other. *e.g.*,

```
FIXED
2 XY
ENDFIXED
```

References and Further Reading

[1] S. Shaik, R. Ramanan, D. Danovich, and D. Mandal. *Chem. Soc. Rev.*, 47:5125, 2018. DOI: 10.1039/C8CS00354H.

[2] S. Shaik, D. Danovich, J. Joy, Z. Wang, and T. Stuyver. *J. Am. Chem. Soc.*, 142:12551, 2020. DOI: 10.1021/jacs.0c05128.

B.3 Alphabetical Listing of *$rem* Variables

BASIS_LIN_DEP_THRESH

Sets the threshold for determining linear dependence in the basis set

TYPE:

INTEGER

DEFAULT:

6 Corresponding to a threshold of 10^{-6}

OPTIONS:

n Sets the threshold to 10^{-n}

RECOMMENDATION:

Set to 5 or smaller if you have a poorly behaved SCF and you suspect linear dependence in you basis set. Lower values (larger thresholds) may affect the accuracy of the calculation.

BASIS

Specifies the electronic basis sets to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen	User defined (*$basis* keyword required).
Symbol	Use standard basis sets as per Chapter 8.
Mixed	Use a mixture of basis sets (see Chapter 8).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

CNEO

Enable a CNEO calculation. Using CNEO with multiple quantum protons automatically activates the nuclear Hartree product approximation.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE No CNEO calculation.

OPTIONS:

TRUE (or 1)	Enable a CNEO calculation.
FALSE (or 0)	Disable a CNEO calculation.

RECOMMENDATION:

None.

CONCENTRIC_REF_BASIS

Specify the projection basis (PB) in the concentric localization procedure

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Parsed in the same way as BASIS; if unspecified, the working basis (WB) will be used as PB.

RECOMMENDATION:

WB is usually a good choice; a smaller basis can chosen with caution to further reduce the computational cost.

CONCENTRIC_VIRTS_ZETA

Specify the size of the truncated virtual space

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

m The total number of the CL-truncated virtuals is $m \times n_{\text{occ}}^{\text{active}}$

RECOMMENDATION:

Use the default; set it to a larger value if higher accuracy is requested.

CONCENTRIC_VIRTS

Use the concentric localization (CL) scheme to truncate the virtual space

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use the CL scheme to truncate the virtual space

FALSE Leave the virtual space untruncated

RECOMMENDATION:

Use CL truncation for WFT-in-DFT calculations.

CVS_IP_ALPHA

Sets the number of ionized target states derived by removing α electron ($M_S = -\frac{1}{2}$).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/α states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_IP_BETA

Sets the number of ionized target states derived by removing β electron ($M_S = \frac{1}{2}$, default for CVS-IP).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/β states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_IP_STATES

Sets the number of core-ionized states to find. By default, β electron will be removed.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP states.

OPTIONS:

[i,j,k...] Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DFT_D3_3BODY

Controls whether the three-body interaction in Grimme's DFT-D3 method should be applied (see Eq. (14) in Ref. 15).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply the three-body interaction term

TRUE Apply the three-body interaction term

RECOMMENDATION:

NONE

DFT_D3_A1

The nonlinear parameter α_1 in Eqs. (5.29), (5.30), (5.31), and (5.32). Used in DFT-D3(BJ), DFT-D3(CSO), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_1 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A2

The nonlinear parameter α_2 in Eqs. (5.29) and (5.32). Used in DFT-D3(BJ), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_2 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_POWER

The nonlinear parameter β_6 in Eq. (5.32). Used in DFT-D3(op). Must be greater than or equal to 6 to avoid divergence.

TYPE:

INTEGER

DEFAULT:

600000

OPTIONS:

n Corresponding to $\beta_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS6

The nonlinear parameter $s_{r,6}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,6} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS8

The nonlinear parameter $s_{r,8}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,8} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_S6

The linear parameter s_6 in eq. (5.27). Used in all forms of DFT-D3.

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_S8

The linear parameter s_8 in Eq. (5.27). Used in DFT-D3(0), DFT-D3(BJ), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_8 = n/100000$.

RECOMMENDATION:

NONE

DFT_D4_A1

The nonlinear parameter α_1. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_1 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_A2

The nonlinear parameter α_2. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_2 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_GA

Charge scaling

TYPE:

INTEGER

DEFAULT:

300000000

OPTIONS:

n Corresponding to $ga = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_GC

Charge scaling

TYPE:

INTEGER

DEFAULT:

200000000

OPTIONS:

n Corresponding to $gc = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_S10

The linear parameter s_{10}. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_{10} = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S6

The linear parameter s_6. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_6 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S8

The linear parameter s_8. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_8 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S9

The linear parameter s_9. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_9 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_WF

Weighting factor for Gaussian weighting.

TYPE:

INTEGER

DEFAULT:

600000000

OPTIONS:

n Corresponding to $wf = n/100000000$.

RECOMMENDATION:

Use default

DFT_D_A

Controls the strength of dispersion corrections in the Chai–Head-Gordon DFT-D scheme, Eq. (5.26).

TYPE:

INTEGER

DEFAULT:

600

OPTIONS:

n Corresponding to $a = n/100$.

RECOMMENDATION:

Use the default.

DIIS_ERR_RMS

Changes the DIIS convergence metric from the maximum to the RMS error.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Use the default, the maximum error provides a more reliable criterion.

DIIS_SUBSPACE_SIZE

Controls the size of the DIIS subspace during the SCF.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

User-defined

RECOMMENDATION:

None.

DIRECT_DIAG

Perform direct diagonalization to obtain all the NEO excitation energies.

TYPE:

INTEGER

DEFAULT:

0 Use Davidson algorithm.

OPTIONS:

1 Do the direct diagonalization.

0 Use Davidson algorithm.

RECOMMENDATION:

Only use this option when Davidson solutions are not stable.

DISTORT

Specifies whether to apply pressure or external force to a chemical system

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use pressure or force

True Use pressure or force

RECOMMENDATION:

Set to true to apply pressure or force.

EDA2_MOM

Perform ALMO-EDA calculation with non-*aufbau* electronic configurations using MOM

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard ALMO-EDA calculation

TRUE ALMO-EDA for non-*aufbau* states

RECOMMENDATION:

None

EDA_ALIGN_FRGM_SPIN

Turn on the fragment spin alignment procedure

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not performed the spin alignment procedure (turned on by default in unrestricted cases)

1 Perform fragment spin alignment; use GDM for the polarization step preceding the MOM calculations

2 Perform fragment spin alignment; use GDM and perform stability analysis for the polarization step

RECOMMENDATION:

Use 1 or 2 when the radical is of highly symmetric structure

EDA_NOCV

Perform the NOCV analysis and plot the significant NOCVs

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform NOCV analysis

1 Plot NOCV pair contributions to density deformation

2 Plot both NOCV pair contribution to density deformation and NOCV orbitals

RECOMMENDATION:

None

EDA_PLOT_DIFF_DEN

Plot changes in electron density due to POL and CT

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not make EDD plots

TRUE Make EDD plots

RECOMMENDATION:

None

EIGSLV_METH

Control the method for solving the ALMO-CIS eigen-equation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Explicitly build the Hamiltonian then diagonalize (full-spectrum)

1 Use the Davidson method (currently only available for restricted cases)

RECOMMENDATION:

None; use 1 for ALMO-TDA calculations (0 unavailable)

ENV_METHOD

Specify the low-level theory in a projection-based embedding calculation

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Parsed in the same way as *$rem* variable "METHOD"

RECOMMENDATION:

A mean-field method (pure or hybrid density functional) should be chosen.

EP_FACTOR

Controls the strength of the electron-proton component of the NEO-MP2 correlation energy.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n Corresponding to $c_{\mathrm{ep}} = n/10^6$.

RECOMMENDATION:

NONE

ESP_EFIELD

Triggers the calculation of ESP and/or electric field at nuclear positions or on a given grid of points

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Compute ESP only

1 Compute both ESP and electric field

2 Compute electric field only

RECOMMENDATION:

None

EX_EDA

Perform an ALMO-EDA calculation with one or more fragments excited.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform EDA with excited-state molecule(s) taken into account.

FALSE

RECOMMENDATION:

None

FIXING_V_EMBED

Invoke the linearized approximation for the energy functional used for embedding calculations

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Use the linearized approximation for energy functional [Eq. (11.106)]

FALSE Use the original energy functional [Eq. (11.100)]

RECOMMENDATION:

Use the default to achieve savings in computational costs

FODFT_DONOR

Specify the donor fragment in FODFT calculation

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 First fragment as donor

2 Second fragment as donor

RECOMMENDATION:

With FODFT_METHOD = 1, the charged fragment needs to be the donor fragment

FODFT_METHOD

Specify the flavor of FODFT method

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 FODFT$(2\mathrm{n}-1)@D^{+}A$ (HT) / FODFT$(2\mathrm{n}+1)@D^{-}A$ (ET)

2 FODFT$(2\mathrm{n})@DA$

3 FODFT$(2\mathrm{n}-1)@DA$ (HT) / FODFT$(2\mathrm{n}+1)@D^{-}A^{-}$ (ET)

RECOMMENDATION:

The default approach shows the best overall performance

FRAG_DIABAT_DOHT

Specify whether hole or electron transfer is considered

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE	Do hole transfer
FALSE	Do electron transfer

RECOMMENDATION:

Need to be specified for POD and FODFT calculations

FRAG_DIABAT_METHOD

Specify fragment based diabatization method

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

ALMO_MSDFT	Perform ALMO(MSDFT) diabatization
POD	Perform projection operator diabatization (the original method)
POD2_L	Perform POD2 with Löwdin orthogonalization
POD2_GS	Perform POD2 with Grad-Schmidt orthogonalization
ESID	The energy-split-in-dimer method,[40] which is equivalent to the FMO approach introduced in Section 10.14.2.5
FODFT	Calculate electronic coupling using fragment orbital DFT

RECOMMENDATION:

NONE

FRAG_DIABAT_PRINT

Specify the print level for fragment based diabatization calculations

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	No additional prints
≥ 1	Print additional details

RECOMMENDATION:

Use the default unless debug information is needed

GAP_TOL

HOMO/LUMO gap threshold to control whether to shift the diagonal elements of the virtual block of the Fock matrix or not. If the HOMO/LUMO gap is less than this threshold, at a given SCF iteration, then the diagonal elements of the virtual block of the Fock matrix are shifted. Otherwise no level-shift is applied.

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

User-defined

RECOMMENDATION:

The input number must be an integer between 0 and 9999. The threshold applied is set to GAP_TOL/1000 Eh. The default value is provided to make the level-shifting calculation run and should not be taken as optimal for any specific problem. Trial and error may be required to find the optimal threshold. Larger values of GAP_TOL generally lead to level-shifting being used more frequently during the SCF convergence process.

GEN_SCFMAN_EMBED

Run a projection-based embedding calculation using the implementation based on-GEN_SCFMAN

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a projection-based embedding calculation

FALSE Do not perform an embedding calculation

RECOMMENDATION:

None

GUESS_GRID

Specifies the type of grid to use for SAP guess generation. The options are the same as those of the *$rem* variable XC_GRID.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Use SG-0 for H, C, N, and O; SG-1 for all other atoms.

n Use SG-n for all atoms, $n = 1, 2$, or 3

XY A string of two six-digit integers X and Y, where X is the number of radial points and Y is the number of angular points where possible numbers of Lebedev angular points, which must be an allowed value from Table 5.2 in Section 5.5.

$-XY$ Similar format for Gauss-Legendre grids, with the six-digit integer X corresponding to the number of radial points and the six-digit integer Y providing the number of Gauss-Legendre angular points, $Y = 2N^2$.

RECOMMENDATION:

Larger grids may be required if the SAP guess is poor.

JOBTYPE

Specifies the type of calculation.

TYPE:

STRING

DEFAULT:

SP

OPTIONS:

SP	Single point energy.
FORCE	Analytical force calculation.
OPT	Geometry minimization.
TS	Transition structure search.
FREQ	Frequency calculation.

RECOMMENDATION:

Application-dependent. Always use POINT_GROUP_SYMMETRY = FALSE with geometry optimization.

LEVEL_SHIFT

Determine whether to invoke level-shifting or not together with DIIS.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	Apply level shifting.
FALSE	Do not apply level shifting.

RECOMMENDATION:

Use TRUE if level-shifting is necessary to accelerate SCF convergence.

LSHIFT

Constant shift applied to all diagonal elements of the virtual block of the Fock matrix.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

User-defined

RECOMMENDATION:

The input number must be an integer between 0 and 9999. The actual shift is equal to LSHIFT/1000 Eh. The default value is provided to make the level-shifting calculation run and should not be taken as optimal for any specific problem. Trial and error may be required to find the optimal threshold. Larger level shifts make the SCF process more stable but also slow down convergence, thus requiring more SCF cycles.

MAX_DP_CYCLES

The maximum number of SCF iterations with damping when SCF_ALGORITHM = DP_DIIS and DP_GDM. See also THRESH_DP_SWITCH.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

1 Only a single SCF step with damping, and no damping for the remaining SCF steps.

n n SCF iterations with damping before turning damping off.

RECOMMENDATION:

Increase this number if strong fluctuation continues after damping is turned off.

MAX_LS_CYCLES

The maximum number of DIIS iterations with level-shifting when SCF_ALGORITHM = LS_DIIS. See also THRESH_LS_SWITCH.

TYPE:

INTEGER

DEFAULT:

MAX_SCF_CYCLES

OPTIONS:

1 Only a single DIIS step with level-shifting, and no level-shifting for the remaining DIIS steps.

n n DIIS iterations with level-shifting before turning level-shifting off.

RECOMMENDATION:

None

MAX_SCF_CYCLES

Controls the maximum number of SCF iterations permitted.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

Increase for slowly converging systems such as those containing transition metals.

MEM_STATIC

Sets the memory for AO-integral evaluations and their transformations in Q-CHEM 4.1 or older versions.

TYPE:

INTEGER

DEFAULT:

192 corresponding to 192 MB.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use the default.

MEM_TOTAL

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 2 GB

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use the default, or set to the physical memory of your machine. The minimum requirement is $3X^2$.

METHOD

Specifies the exchange-correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use METHOD = *NAME*, where *NAME* is one of the following: HF for Hartree-Fock theory; one of the DFT methods listed in Section 5.3.5.;

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations for DFT are indicated in bold in Section 5.3.5.

MOM_METHOD

Determines the target orbitals with which to maximize the overlap on each SCF cycle.

TYPE:

INTEGER

DEFAULT:

MOM

OPTIONS:

MOM Maximize overlap with the orbitals from the previous SCF cycle.

IMOM Maximize overlap with the initial guess orbitals.

RECOMMENDATION:

If appropriate guess orbitals can be obtained, then IMOM can provide more reliable convergence to the desired solution.[4]

MSDFT_METHOD

Specify the scheme for ALMO(MSDFT)

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 The original MSDFT scheme [Eq. (10.183)]

2 The ALMO(MSDFT2) approach [Eq. (10.186)]

RECOMMENDATION:

Use the default method. Note that the method will be automatically reset to 1 if a meta-GGA functional is requested.

MSDFT_PINV_THRESH

Set the threshold for pseudo-inverse of the interstate overlap

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n Set the threshold to 10^{-n}

RECOMMENDATION:

Use the default value

NDAMP

Determine the mixing coefficient. α = NDAMP/100.

TYPE:

INTEGER

DEFAULT:

75

OPTIONS:

User-defined. Integers between 0 and 100.

RECOMMENDATION:

Increase NDAMP if strong fluctuations happen during the SCF process.

NEO_BASIS_LIN_DEP_THRESH

This keyword is used to set the liner dependency threshold for nuclear basis sets. It is defined as $10^{-\mathrm{NEO\_BASIS\_LIN\_DEP\_THRESH}}$.

TYPE:

DOUBLE

DEFAULT:

5.0

OPTIONS:

User-defined

RECOMMENDATION:

No recommendation.

NEO_CCSD_CONVERGENCE

NEO-RICCSD is considered converged when the energy error is less than $10^{-\mathrm{NEO\_CCSD\_CONVERGENCE}}$.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_CCSD_MAX_CYCLES

Controls the maximum number of CC iterations permitted.

TYPE:

INTEGER

DEFAULT:

5000

OPTIONS:

n Set the maximum number of iterations to $n > 0$.

RECOMMENDATION:

None

NEO_EPC

Specifies the electron-proton correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use NEO_EPC = *NAME*, where *NAME* can be either epc172 or epc19.

RECOMMENDATION:

Consult the NEO literature to guide your selection.

NEO_E_CONV

Energy convergence criteria in the NEO-SCF calculations so that the difference in energy between consecutive iterations is less than $10^{-\mathrm{NEO\_E\_CONV}}$.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

User-defined

RECOMMENDATION:

Tighter criteria for geometry optimization are recommended. Note that for stepwise calculations, this convergence criteria checks the difference in energy between electronic and protonic microiterations blocks.

NEO_E_MAX_SCF_CYCLES

Controls the maximum number of electronic SCF micoiterations permitted. This variable is only used for stepwise NEO-SCF optimizations.

TYPE:

INTEGER

DEFAULT:

150

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

None.

NEO_ISOTOPE

Enable calculations of different types of isotopes. Only one type of isotope is allowed at present.

TYPE:

INTEGER

DEFAULT:

1 Default is the proton isotope.

OPTIONS:

1 This NEO calculation is using proton isotope.

2 This NEO calculation is using deuterium isotope.

3 This NEO calculation is using tritium isotope.

RECOMMENDATION:

Refer to the NEO literature for the best performance on the isotope effects calculations.

NEO_MSDFT

Enable a NEO-MSDFT calculation

TYPE:

INTEGER

DEFAULT:

0 No NEO-MSDFT calculation.

OPTIONS:

1 Enable a NEO-MSDFT calculation.

0 Disable a NEO-MSDFT calculation.

RECOMMENDATION:

See Section 13.5.2.3 for details on customizing a NEO-MSDFT calculation.

NEO_N_HARTREE_PROD

Enable the nuclear Hartree product approximation.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

TRUE (or 1) Enable a nuclear Hartree product calculation.

FALSE (or 0) Disable a nuclear Hartree product calculation.

RECOMMENDATION:

Use simultaneous NEO-SCF for faster convergence.

NEO_N_MAX_SCF_CYCLES

Controls the maximum number of nuclear SCF micoiterations permitted. This variable is only used for stepwise NEO-SCF optimizations.

TYPE:

INTEGER

DEFAULT:

150

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

None.

NEO_N_SCF_CONVERGENCE

NEO-SCF is considered converged when the nuclear wave function error is less that $10^{-\text{NEO\_N\_SCF\_CONVERGENCE}}$.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

User-defined

RECOMMENDATION:

None.

NEO_PURECART

This keyword is used to specify Cartesian or spherical Gaussians for nuclear basis functions.

TYPE:

INTEGER

DEFAULT:

2222

OPTIONS:

User-defined

RECOMMENDATION:

The default value corresponds to the use of Cartesian Gaussians for all angular momentum classes. The value NEO_PURECART = 1111 would use spherical Gaussians instead, similar to the use of PURECART.

NEO_RICCSD

Enable a NEO-RICCSD calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Enable this option.

0 Disable this option.

RECOMMENDATION:

Both electronic and protonic auxiliary basis sets must be specified.

NEO_RIMP2

Enable a NEO-MP2 or NEO-OOMP2 calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

2 Perform a NEO-OOMP2 calclulation.

1 Perform a NEO-MP2 calculation.

0 Disable this option.

RECOMMENDATION:

Both electronic and protonic auxiliary basis sets must be specified.

NEO_SCFV

Enable a NEO-SCFV calculation

TYPE:

INTEGER

DEFAULT:

0 No NEO-SCFV calculation.

OPTIONS:

1 Enable a NEO-SCFV calculation.

0 Disable a NEO-SCFV calculation.

RECOMMENDATION:

None.

NEO_SCF_GUESS_N

Specifies the initial guess procedure for the nuclear orbitals in a NEO-SCF calculation.

TYPE:

STRING

DEFAULT:

TIGHT Default unless ghost atoms are present.

CORE Default when there are ghost atoms.

OPTIONS:

CORE	Diagonalize the nuclear core Hamiltonian
TIGHT	Occupies the least diffuse nuclear MO
AUTOSAD	On-the-fly superposition of nuclear atomic densities from an atomic NEO calculation
READ	Read previous nuclear MOs from disk

RECOMMENDATION:

Use the default. Note that if NEO_SCF_GUESS_N is set to READ, only the nuclear MOs from a previous NEO-SCF job are read from disk. The starting guess for the electronic MOs in a NEO calculation remains the converged result from the conventional electronic calculation of the current job. To read in both the electronic and nuclear MOs from a previous NEO-SCF job, NEO_SCF_GUESS_N must be set to READ in conjunction with SKIP_SCFMAN set to TRUE.

NEO_SET_ESTATE

This keyword is used to specify for which vibronic excited state with dominant electronic character the gradient or geometry optimization is needed.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

n $n > 0$ Looks to calculate gradient or conduct geometry optimization for the nth NEO vibronic excited state with dominant electronic character.

RECOMMENDATION:

Make sure enough roots are requested by the CIS_N_ROOTS keyword because the vibronic excited states with dominant protonic character usually come before.

NEO_SET_OPT

Enable a NEO excited state geometry optimization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Enable a NEO excited state geometry optimization.

0 Disable a NEO excited state geometry optimization.

RECOMMENDATION:

Need to use with CIS_STATE_DERIV. Consult the keyword NEO_SET_ESTATE if geometry optimization is desired for a vibronic excited state with dominant electronic character.

NEO_SIMULTANEOUS_SCF

Enables simultaneous optimization algorithm.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Use the default unless issues with convergence are observed, in which cause set to FALSE to utilize the traditional stepwise approach.

NEO_STEPWISE_SCF_STEPS

Specifies the number of NEO-SCF stepwise/alternating macro-iterations to perform before switching to simultaneous algorithm.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

The rate of convergence of the NEO-SCF procedure is dependent on the initial guess for the electronic and protonic orbitals. For especially difficult systems, it may be helpful to do stepwise optimization before switching to simultaneous.

NEO_TDKS

Enable a real-time NEO-TDHF or NEO-TDDFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE No RT-NEO calculation.

OPTIONS:

TRUE Enable a RT-NEO calculation.

FALSE Disable a RT-NEO calculation.

RECOMMENDATION:

None.

NEO_VPP

This keyword is used to control whether to remove $J - K$ terms from the nuclear Fock matrix and the corresponding kernel terms for NEO excited state methods. The purpose for the removal of these terms in the case of one quantum proton is to save on computational cost. Note that the ground state NEO-HF or NEO-DFT energy is the same with or without removal of these nuclear $J - K$ terms. The nuclear virtual orbitals will however be different, which may affect the results for perturbative methods (i.e., multicomponent excited-state methods and post-NEO-HF wavefunction methods).

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE

OPTIONS:

TRUE (or 1) Enable this option (include nuclear $J - K$ terms).

FALSE (or 0) Disable this option (remove nuclear $J - K$ terms).

RECOMMENDATION:

Set NEO_VPP = 0 only in the case of one quantum hydrogen.

NEO_ZVEC_CG_CONV

The convergence threshold ($10^{-\mathrm{NEO\_ZVEC\_CG\_CONV}}$) for the iterative gradient solver for NEO Z-vector equations.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n Use $n > 0$ iterations.

RECOMMENDATION:

None.

NEO_ZVEC_CG_MAXITER

Controls the maximum number of iterative gradient solver iterations permitted.

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

n Use $n > 0$ iterations.

RECOMMENDATION:

None.

NEO_ZVEC_LINEAR

Use linear solver for Z-vector equations for NEO excited state gradient.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Use linear solver

0 Use iterative conjugate gradient solver

RECOMMENDATION:

Use the default iterative conjugate gradient solver because it is more memory efficient.

NEO

Enable a NEO-SCF calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Enable a NEO-SCF calculation.

FALSE Disable a NEO-SCF calculation.

RECOMMENDATION:

Set to TRUE if desired.

POD_WINDOW

Specify the number of donor and acceptor orbitals when couplings between multiple pairs are requested

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n Including n frontier occupied orbitals (from HOMO $- n + 1$ to HOMO) and n frontier virtual orbitals (from LUMO to LUMO $+ n - 1$) for both donor and acceptor

RECOMMENDATION:

None

PURECART

INTEGER

TYPE:

Controls the use of pure (spherical harmonic) or Cartesian angular forms

DEFAULT:

1111 Pure h, g, f, d functions

OPTIONS:

$hgfd$ Use 1 for pure and 2 for Cartesian.

RECOMMENDATION:

This is pre-defined for all standard basis sets

RPA

Do a NEO-TDDFT or NEO-TDHF calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do a NEO-TDA or NEO-CIS calculation.

TRUE Do a NEO-TDDFT or NEO-TDHF calculation.

RECOMMENDATION:

Consult the NEO literature to guide your selection.

RR_NO_NORMALISE

Controls whether frequency job calculates resonance Raman intensities

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Normalize RR intensities

True Do not normalize RR intensities

RECOMMENDATION:

False

S2THRESH

Cutoff for neglect of overlap integrals, defined via a two-electron shell-pair threshold of $10^{-\mathrm{S2THRESH}}$ (S2THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

Same as THRESH.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

Increase the value of S2THRESH if the program finds negative eigenvalues for the overlap matrix.

SCFMI_MOM

Perform an SCF-MI calculation with non-*aufbau* electronic configurations using MOM

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard SCF-MI calculation

TRUE SCF-MI calculation with MOM

RECOMMENDATION:

None

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

DIIS Pulay DIIS.

OPTIONS:

DIIS Pulay DIIS.

GDM Geometric Direct Minimization.

DIIS_GDM Use DIIS and then later switch to geometric direct minimization.

RECOMMENDATION:

In the NEO methods, the GDM procedure is recommended. Note that DIIS_GDM is only supported for simultaneous optimizations.

SCF_CONVERGENCE

NEO-SCF is considered converged when the electronic wave function error is less that $10^{-\mathrm{SCF\_CONVERGENCE}}$. Adjust the value of THRESH at the same time. (Starting with Q-CHEM 3.0, the DIIS error is measured by the maximum error rather than the RMS error as in earlier versions.)

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations.

8 For geometry optimizations.

OPTIONS:

User-defined

RECOMMENDATION:

None.

SCS

Set the type of spin-component scaling.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 Turns on spin-component scaling with SCS ($c_{\mathrm{ss}} = 0.33$, $c_{\mathrm{os}} = 1.2$).

2 Turns on spin-component scaling with SOS ($c_{\mathrm{ss}} = 0.0$, $c_{\mathrm{os}} = 1.2$ for MP2, $c_{\mathrm{os}} = 1.3$ for OOMP2).

3 arbitrary SCS (set with SSS_FACTOR and SOS_FACTOR).

0 no spin-component scaling.

RECOMMENDATION:

NONE

SET_SUBSPACE

Specify the number of protonic guess vectors for NEO-TDDFT

TYPE:

INTEGER

DEFAULT:

Number of states desired (as set by CIS_N_ROOTS) if the number is smaller than the size of the protonic subspace (number of protonic occupied orbitals $\times$ number of protonic virtual orbitals) or the size of the protonic subspace

OPTIONS:

n Use $n > 0$ vectors.

RECOMMENDATION:

None.

SKIP_CHARGE_SELF_INTERACT

Ignores the electrostatic interactions among external charges in a QM/MM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE No electrostatic interactions among external charges.

FALSE Computes the electrostatic interactions among external charges.

RECOMMENDATION:

None

SOLVENT_METHOD

Sets the preferred solvent method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0 Do not use a solvation model.

PCM Use an apparent surface charge, polarizable continuum model (Section 11.2.3).

SMD Use SMD (Section 11.2.9.3).

RECOMMENDATION:

No recommendation.

SOS_FACTOR

Controls the strength of the opposite-spin component of the MP2 electron-electron correlation energy.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n Corresponding to $c_{\mathrm{os}} = n/10^6$.

RECOMMENDATION:

NONE

SPADE_PARTITION

Use the SPADE approach to determine the initial set of embedded (active) orbitals

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use SPADE to partition the occupied space

FALSE Use the Pipek-Mezey localization + Mulliken population to assign occupied orbitals

RECOMMENDATION:

Use SPADE if a significant gap in the spectrum of singular values can be detected.

SSS_FACTOR

Controls the strength of the same-spin component of the MP2 electron-electron correlation energy.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n Corresponding to $c_{\mathrm{ss}} = n/10^6$.

RECOMMENDATION:

NONE

THRESH_DP_SWITCH

The threshold for turning off damping in SCF iterations is $10^{-\text{THRESH\_DP\_SWITCH}}$ when SCF_ALGORITHM is set to DP_DIIS or DP_GDM. See also MAX_DP_CYCLES.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined.

RECOMMENDATION:

None

THRESH_LS_SWITCH

The threshold for turning off level-shifting in DIIS is $10^{-\text{THRESH\_LS\_SWITCH}}$ when SCF_ALGORITHM is set to LS_DIIS. See also MAX_LS_CYCLES.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

User-defined.

RECOMMENDATION:

None

THRESH

Cutoff for neglect of two electron integrals. $10^{-\text{THRESH}}$ (THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

8 For single point energies.

10 For optimizations and frequency calculations.

14 For coupled-cluster calculations.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

The value should satisfy THRESH $\geq$ 3 + SCF_CONVERGENCE, although THRESH = 4 + SCF_CONVERGENCE is the default (in most cases) since Q-CHEM v. 6.0. See Ref. 13 for recommended values of THRESH in the presence of diffuse basis functions, where tighter thresholds are often required and too-loose thresholds may lead to slower convergence or convergence failure.

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE Closed-shell systems.

TRUE Open-shell systems.

OPTIONS:

FALSE Constrain the spatial part of the alpha and beta orbitals to be the same.

TRUE Do not Constrain the spatial part of the alpha and beta orbitals.

RECOMMENDATION:

The ROHF method is not available. Note that for unrestricted calculations on systems with an even number of electrons it is usually necessary to break α/β symmetry in the initial guess, by using SCF_GUESS_MIX or providing *$occupied* information (see Section 4.4 on initial guesses).

VFB_CTA

Use the Variational Forward-Backward (VFB) approach to obtain "one-way" CT potential surfaces.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

FORWARD Allow 1→2 CT only (1 and 2 are two fragments).

BACKWARD Allow 2→1 CT only.

RECOMMENDATION:

None

XC_GRID

Specifies the type of grid to use for DFT calculations.

TYPE:

INTEGER

DEFAULT:

Functional-dependent; see Table 5.3.

OPTIONS:

0 Use SG-0 for H, C, N, and O; SG-1 for all other atoms.

n Use SG-n for all atoms, $n = 1, 2$, or 3

XY A string of two six-digit integers X and Y, where X is the number of radial points and Y is the number of angular points where possible numbers of Lebedev angular points, which must be an allowed value from Table 5.2 in Section 5.5.

$-XY$ Similar format for Gauss-Legendre grids, with the six-digit integer X corresponding to the number of radial points and the six-digit integer Y providing the number of Gauss-Legendre angular points, $Y = 2N^2$.

RECOMMENDATION:

We strongly recommend using an unpruned, Euler-Maclaurin-Lebedev grid type for NEO-DFT calculations. Larger grids may be required for optimization and frequency calculations.

FRZN_OPT

Controls whether the job uses zeroed Hessian technique in the frequency calculations

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use the zeroed out Hessian

True Use the zeroed out Hessian

RECOMMENDATION:

False

FRZ_ATOMS

Controls the number of frozen atoms

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

HARM_FORCE

Sets the force constant for harmonic confiner, in units of N/m.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

HARM_OPT

Controls whether the job uses confining potentials

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not use the potential

True Use the potential

RECOMMENDATION:

False

HOATOMS

Controls the number of confined atom

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

ALMO_EFIELD_PROBE_FRGM

Specify the index of the probe fragment in ALMO-based ESP and electric field calculations

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Specify the nth fragment as the probe

RECOMMENDATION:

None

ALMO_EFIELD

Calculate the environment ESP/E-field using ALMO-based partitioning

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE In job 1, it saves the electron density for the environment constructed from ALMOs;
In job 2, it reads in the electron density (must be together with SCF_GUESS = READ_DEN)

FALSE Don't do ALMO-based ESP/field calculations

RECOMMENDATION:

Required for both jobs in ALMO-based electric field calculations

CIS1D_DER_NUMSTATE

Determines the number of states for which derivative couplings are to be calculated. The states are specified in the *$derivative_coupling* section

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not calculate derivative couplings.

n Calculate $n(n-1)/2$ pairs of derivative couplings.

RECOMMENDATION:

None

CIS1D_ED_CONVERGENCE

The first step in TDDFT-1D is to find the optimized orbitals for the double by minimizing the energy of the doubly excited configuration, E_d. This variable controls the convergence criterion for the minimization of E_d. The orbitals are optimized when the error is lower than $10^{-\text{CIS1D\_ED\_CONVERGENCE}}$.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n Convergence achieved when the error is lower than 10^{-n}.

RECOMMENDATION:

The convergence criterion for the roots of the CIS-1D and TDDFT-1D calculations is set by the CIS_CONVERGENCE rem variable, which is set to 9 by default for a CIS-1D/TDDFT-1D calculation. If CIS_CONVERGENCE is increased, then CIS1D_ED_CONVERGENCE should also be increased.

CIS1D_N_ROOTS

Sets the number of CIS-1D and TDDFT-1D states to calculate. The lowest CIS-1D/TDDFT-1D state is the ground state, which is followed by excited states.

TYPE:

INTEGER

DEFAULT:

0 Do not calculate any CIS-1D or TDDFT-1D states.

OPTIONS:

n $n > 0$, Calculate the lowest n CIS-1D or TDDFT-1D states.

RECOMMENDATION:

None

CIS1D_SCALE_GD

The coupling element between the ground and doubly excited configuration in the Hamiltonian in Eq. (7.47), ($\langle\psi_0|H|\psi_{h\bar{h}}^{l\bar{l}}\rangle$) are scaled by this factor.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n $0 \leq n \leq 100$, The coupling element is scaled by $n/100$.

RECOMMENDATION:

Since the KS reference is not a true wave function, there is no rigorous way to determine the coupling elements in the configuration interaction Hamiltonian. It sometimes becomes necessary to scale the coupling elements to obtain accurate potential energy surfaces, and the scaling factor has to be determined with some benchmarking[3].

CIS1D_SCALE_SD

The coupling element between the singles and the double in the Hamiltonian in Eq. (7.47), ($\langle\psi_i^a|H|\psi_{h\bar{h}}^{l\bar{l}}\rangle$) are scaled by this factor.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n $0 \leq n \leq 100$, The coupling element is scaled by $n/100$.

RECOMMENDATION:

Same as CIS1D_SCALE_GD.

CIS1D_STATE_DERIV

Selects the CIS-1D/TDDFT-1D state for which gradients are calculated. This is useful for jobs such as geometry optimizations.

TYPE:

INTEGER

DEFAULT:

-1 Does not select any state

OPTIONS:

n $n \geq 0$, Selects the nth CIS-1D/TDDFT-1D state.

RECOMMENDATION:

None

CIS_SOC

Controls the roots of performing TDDFT/TDA-SOC calculation.

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform the calculation.

N Solve the N lowest spin-adiabatic states of TDDFT/TDA-SOC.

RECOMMENDATION:

Less than or equal to $4\times$CIS_N_ROOTS. TDDFT/TDA-SOC first performs a standard TDDFT/TDA calculation so as to generate an initial guess before rerunning the diagonalization to generate the spin adiabats. Therefore, it is a good idea to perform a stand-alone normal TDDFT/TDA calculation and generate the excited spin-diabats and check the desired range of energies, before generating the excited spin-adiabats.

CLENSHAW_NGRID

Number of grid points for the Curtis-Clenshaw quadrature.

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

RECOMMENDATION:

Use default.

COMPLEX_EXPONENTS

Enable a non-Hermitian calculation with CBFs.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a non-Hermitian calculation with CBFs

RECOMMENDATION:

Set to TRUE if a non-Hermitian calculation using CBFs is desired.

COMPLEX_METSCF

Specify the NH-SCF solver

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Roothaan iterations

1 DIIS

3 ADIIS

21 Newton-MINRES

RECOMMENDATION:

Use the default (DIIS).

COMPLEX_N_ELECTRON

Add electrons for non-Hermitian calculation.

TYPE:

INTEGER

DEFAULT:

0 Perform the non-Hermitian calculation on N-electrons

OPTIONS:

n Perform the non-Hermitian calculation on an $N + n$ electron system

RECOMMENDATION:

None

COMPLEX_SCF_GUESS

Specify the NH-SCF guess

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use a guess from a static-exchange calculation

1 Read real-basis MO coefficients

2 Read real-basis density matrix

1000 Read guess from a previous calculation

RECOMMENDATION:

Use a guess from a static exchange calculation. Note that for temporary anions, this requires the specification of COMPLEX_TARGET.

COMPLEX_SCF

Perform a non-Hermitian SCF calculation with CBFs

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform an NH-SCF calculation

1 Perform a restricted NH-SCF calculation

2 Perform an unrestricted NH-SCF calculation

3 Perform a restricted, open-shell NH-SCF calculation

RECOMMENDATION:

None

COMPLEX_SPIN_STATE

Spin state for non-Hermitian calculation

TYPE:

INTEGER

DEFAULT:

1 Singlet

OPTIONS:

$2S + 1$ A state of spin S

RECOMMENDATION:

None

COMPLEX_STATIC_EXCHANGE

Perform a CBF static-exchange calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a static exchange calculation

FALSE Do not perform a static exchange calculation

RECOMMENDATION:

Set to TRUE if a static-exchange calculation is desired.

COMPLEX_TARGET

Specify the orbital index to be occupied for a temporary anion

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Orbital index (starting at zero) for the additional electron

RECOMMENDATION:

n should always be greater than $N_{\text{occ}} - 1$.

EMBEDDING_EARLY_STOP

Terminate the embedding calculation once the system partition is done (skip the embedded SCF)

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Terminate the embedding calculation once the system partition is done (skip the embedded SCF)

FALSE Doing a normal embedding calculation

RECOMMENDATION:

Turn it on for environment ESP/E-field calculations (see Section 10.6)

EX_ALMO_MSDFT

Enable MSDFT/MSDFT2 calculations using ALMO-based ΔSCF states

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Will calculate ALMO-ΔSCF states as specified in *$scfmi_mom*

FALSE Performing ground-state ALMO(MSDFT) calculations.

RECOMMENDATION:

None

NOCI_DETGEN

Control how the multiple determinants for NOCI are created.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use only the initial reference determinants.

1 Generate CIS excitations from each reference determinant.

2 Generate all FCI excitations from each reference determinant.

3 Generate n multiple determinants using SCF metadynamics, where n is specified using SCF_SAVEMINIMA = n.

4 Generate all CAS excitations from each reference determinant, where the active orbitals are specified using the *$active_orbitals* input section.

RECOMMENDATION:

By default, these multiple determinants are optimized at the SCF level before running NOCI. This behavior can be turned off using by specifying SKIP_SCFMAN = TRUE.

NOCI_NEIGVAL

The number of NOCI eigenvalues to be printed.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Positive integer

RECOMMENDATION:

Increase this to print progressively higher NOCI energies.

NOCI_REFGEN

Control how the initial reference determinants are created.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Generate initial reference determinant from a single SCF calculation.

1 Read (multiple) initial reference determinants from a previous calculation.

RECOMMENDATION:

The specific reference determinants to be read from a previous calculation can be indicated using SCF_READMINIMA.

POD_MULTI_PAIRS

Calculate the couplings between multiple pairs of donor and acceptor orbitals in POD

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate the couplings between multiple pairs of orbitals

FALSE Only calculate the D(HOMO)–A(HOMO) coupling (for hole transfer) or the D(LUMO)–A(LUMO) coupling (for electron transfer)

RECOMMENDATION:

None

REL_SF_X2C_CGHF

Enables X2C scalar relativistic calculation in a cGHF routine.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perform a full relativistic X2C calculation with SOC effects

1 Perform a scalar relativistic X2C calculation

RECOMMENDATION:

Set to 1 if a scalar relativistic X2C calculation in a **cGHF class** is desired. Must be used with REL_X2C = 1. Otherwise, non-relativistic calculation is performed without X2C, and this rem variable is ignored.

REL_SNSO

Enables 1eX2C-SNSO calculation in a cGHF routine.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perform a full relativistic X2C calculation with unadjusted SOC Hamiltonian

1 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by original Boetter coefficients

2 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by modified SNSO coefficients by Yoshizawa and co-workers

3 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by DC-SNSO coefficients from Ehrman and co-workers

4 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by universal DCB-SNSO coefficients from Ehrman and co-workers

5 Perform a full relativistic X2C calculation with SOC Hamiltonian adjusted by row-dependent DCB-SNSO coefficients from Ehrman and co-workers

RECOMMENDATION:

Set to 1 for most general cases. Universal DCB-SNSO coefficients are the most recent contribution to the SNSO family and is reported to give accurate doublet splitting of $L_{2,3}$-edges.

REL_X2C_FD_DISPLACEMENT

Controls finite difference step for calulating W

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n Set finite difference step to $n \times 10^{-6}$

RECOMMENDATION:

None

REL_X2C

Enables X2C scalar relativistic calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perform a regular, non-relativistic SCF calculation

1 Perform a scalar relativistic X2C calculation

RECOMMENDATION:

Set to 1 if a scalar relativistic X2C calculation is desired.

SCF_EESCALE_ARG

Control the phase angle of the complex λ electron-electron scaling.

TYPE:

INTEGER

DEFAULT:

00000 meaning 0.0000

OPTIONS:

$abcde$ corresponding to $a.bcde$

RECOMMENDATION:

A complex phase angle of 00500, meaning 0.0500, is usually sufficient to follow a solution safely past the Coulson-Fischer point and onto its complex holomorphic counterpart.

SCF_EESCALE_MAG

Control the magnitude of the λ electron-electron scaling.

TYPE:

INTEGER

DEFAULT:

10000 meaning 1.0000

OPTIONS:

$abcde$ corresponding to $a.bcde$

RECOMMENDATION:

For holomorphic Hartree-Fock orbitals, only the magnitude of the input is used, while for real Hartree-Fock orbitals, the input sign indicates the sign of λ.

SCF_HOLOMORPHIC

Turn on the use of holomorphic Hartree-Fock orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Holomorphic Hartree-Fock is turned off

TRUE Holomorphic Hartree-Fock is turned on.

RECOMMENDATION:

If TRUE, holomorphic Hartree-Fock complex orbital coefficients will always be used. If FALSE, but COMPLEX = TRUE, complex Hermitian orbitals will be used.

USE_LIBNOCI

Turn on the use of LIBNOCI for running NOCI calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

False Do not use LIBNOCI (uses original Q-CHEM implementation).

True Use the LIBNOCI implementation.

RECOMMENDATION:

The *$rem* variables detailed below are only available in LIBNOCI.

EDA_COVP_THRESH

Specifies the significance above which the COVPs will be saved

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

N COVPs that contributes more than $0.001 \times N$ kJ/mol in energy decrease will be saved

RECOMMENDATION:

None

EDA_NOCV_QUADRATURE

Number of quadratures used to integrate effective fock matrix

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use 1 quadrature

3 Use 3 quadratures

5 Use 5 quadratures

RECOMMENDATION:

Most of the time, one quadrature is enough. However, in cases where the NOCV energy decreases are very different from the corresponding COVP results, it is recommended to increase the quadrature numbers.

EDA_NOCV_THRESH

Specifies the significance above which the NOCVs will be saved

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

N NOCVs that contributes more than $0.001 \times N$ kJ/mol in energy decrease will be saved

RECOMMENDATION:

None

EDA_NOCV

Perform NOCV analysis

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do NOCV analysis

TRUE Do NOCV analysis

RECOMMENDATION:

None

EDA_PCT_A

Perform perturbative CT analysis

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform perturbative CT analysis

TRUE Perform perturbative CT analysis

RECOMMENDATION:

Set to TRUE to perform perturbative CT analysis

EDA_POL_A

Perform EDA for polarization process

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform EDA for polarization process

TRUE Perform EDA for polarization process

RECOMMENDATION:

Set to TRUE to perform EDA for polarization process

EDA_SAVE_COVP

Save significant COVPs or not

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not save significant COVPs

TRUE Save significant COVPs

RECOMMENDATION:

Set to TRUE to save COVPs. Note that REMs for plotting cube files need also be set

EDA_SAVE_NOCV

Save significant COVPs or not

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not save significant NOCVs

1 Save significant NOCVs

RECOMMENDATION:

Set to 1 to save NOCVs. Note REMs for plotting cube files need also be set

EDA_VCT_A

Perform non-perturbative CT analysis

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform non-perturbative CT analysis

TRUE Perform non-perturbative CT analysis.

RECOMMENDATION:

Set to TRUE to perform non-perturbative CT analysis

GEN_SCFMAN_EDA2

Perform ALMO-EDA calculations using the GEN_SCFMAN_EDA2 driver (differing from jobs with EDA2 $>$ 0)

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use the new ALMO-EDA framework

TRUE Use the new ALMO-EDA framework

RECOMMENDATION:

Set to TRUE to perform non-perturbative CT analysis using this driver

GMBE_DM_GUESS_ONLY

Evaluate the total energy of the system directly using GMBE-DM density matrix. Enabling this option skips SCF iterations, preventing further convergence to the supersystem result.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Do not perform SCF iterations and directly evaluate the total energy of the system using the GMBE-DM density matrix.

FALSE Perform SCF iterations with the GMBE-DM density matrix as the initial guess.

RECOMMENDATION:

NONE

GMBE_MAX_PRIM_CARD

Set the maximum cardinality for the primitive fragments.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n represents the maximum cardinality of the primitive fragments.

RECOMMENDATION:

NONE

MBEDM_PURE

Perform McWeeny purification[30] for GMBE density matrix.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform McWeeney purification on the GMBE density matrix.

FALSE Do not perform McWeeney purification on the GMBE density matrix.

RECOMMENDATION:

NONE

PLOT_ALMO_FRZ

Plot ALMOs at the frozen stage of EDA2 calculations

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not plot frozen ALMOs

TRUE Plot frozen ALMOs

RECOMMENDATION:

None

PLOT_ALMO_POL

Plot ALMOs after the polarization calculation

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not plot polarized ALMOs

TRUE Plot polarized ALMOs

RECOMMENDATION:

None

GMBE_ALGO

Specify the cutoff distance criterion used to determine the primitive fragments.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Minimum distance

2 Heavy-atoms distance

RECOMMENDATION:

NONE

GMBE_INT

Perform a GMBE Calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform GMBE.

FALSE Do not perform GMBE.

RECOMMENDATION:

NONE

GMBE_METHOD

Specify whether to perform an energy-based GMBE (GMBE-EB) or a density matrix-based GMBE (GMBE-DM).

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Perform GMBE-EB.

2 Perform GMBE-DM.

RECOMMENDATION:

NONE

GMBE_PRIMARY

Specify the primary cutoff distance used to determine the primitive fragments. The input for GMBE_PRIMARY is the desired cutoff distance in Å, multiplied by 1000. For example, to use a 2.00 Å cutoff distance, set GMBE_PRIMARY to 2000. Similarly, if GMBE_PRIMARY = 3500, the cutoff distance will be 3.5 Å.

TYPE:

INTEGER

DEFAULT:

3000

OPTIONS:

The desired cutoff is $n = 1000d$, where d is the cutoff distance in Å.

RECOMMENDATION:

NONE

GMBE_PRINT_LEVEL

Control the level of details printed in the output file

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Minimal output

2 Detailed output

RECOMMENDATION:

NONE

NEO_MCSCF_CIDEN_PRINT

Print the CI protonic density for MCSCF states to *.cube files.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print density.

TRUE Print density.

RECOMMENDATION:

None

NEO_MCSCF_DEBUG

Run NEO-MCSCF in debug mode.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not debug NEO-MCSCF.

1 Debug NEO-MCSCF

RECOMMENDATION:

0

NEO_MCSCF_EMC_CONV_THRESH

Convergence threshold for NEO-MCSCF energy. Defined as $10^{-\text{NEO\_MCSCF\_EMC\_CONV\_THRESH}}$.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

User-defined

RECOMMENDATION:

$\geq$8

NEO_MCSCF_KSPACE_MAX

Maximum size of the Krylov subspace for NEO-CI microiterations.

TYPE:

INTEGER

DEFAULT:

NEO_MCSCF_KSPACE_MIN + 10

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_KSPACE_MIN

Minimum size of the Krylov subspace for NEO-CI microiterations.

TYPE:

INTEGER

DEFAULT:

NEO_MCSCF_NSTATES + 2

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_MACRO_CONV_THRESH

Orbital gradient convergence tolerance for NEO-MCSCF. Defined as $10^{-\text{NEO\_MCSCF\_MACRO\_CONV\_THRESH}}$.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

User-defined

RECOMMENDATION:

5

NEO_MCSCF_MACRO_MAX_ITER

Maximum number of NEO-MCSCF macroiterations.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

RECOMMENDATION:

Most CAS calculations should require few macroiterations. More complex RASSCF wavefunctions may require more iterations as convergence can be slow.

NEO_MCSCF_MICRO_MAX_ITER

Maximum NEO-CI iterations.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

User-defined

RECOMMENDATION:

Set as high as necessary to converge all roots required.

$\geq$1000

NEO_MCSCF_MICRO_OPT_MAX_ITER

Maximum number of orbital optimizations per macroiteration.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

User-defined

RECOMMENDATION:

5

NEO_MCSCF_NSTATES

Number of NEO-CI roots to compute during microiterations.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_OPT_METHOD

Orbital optimization method for NEO-MCSCF.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 GDM macroiterations. No microiterations.

1 Augmented Hessian macroiterations. Augmented Hessian microiterations.

2 Augmented Hessian macroiterations. GDM microiterations.

RECOMMENDATION:

2

NEO_MCSCF_PRINT_CIVECS

Print the NEO-MCSCF CI vectors to mcvec.dat.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE No printing.

TRUE MCSCF CI vectors are printed to mcvec.dat and wavefunction is printed to mcwf.dat.

RECOMMENDATION:

None

NEO_MCSCF_PRINT_LEVEL

Print level for NEO-MCSCF.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Only convergence information is printed.

1 Macro and micro iteration information is printed.

$\geq$2 Detailed printing.

RECOMMENDATION:

1

NEO_MCSCF_PRINT_MOS

Print MCSCF MO coefficients to mocoefs.electronic.new and mocoefs.protonic.new.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Print MO coefficients.

FALSE MO coefficients are not printed.

RECOMMENDATION:

TRUE

NEO_MCSCF_READ_MOS

Read MCSCF MO coefficients from mocoefs.electronic and mocoefs.protonic.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE NEO-HF MO coefficients used for initial guess.

TRUE MO coefficients from files used as initial guess.

RECOMMENDATION:

None

NEO_MCSCF_REORDER_MOS

Reorder initial guess molecular orbitals. Requires reorder_mo.txt and reorder_nuc_mo.txt.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Orbitals are not reordered.

TRUE Orbitals are reordered.

RECOMMENDATION:

None

NEO_MCSCF_SA_NSTATES

Number of states to include in state-averaging.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MCSCF_SA

Perform state-averaged NEO-MCSCF.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Perform CASSCF.

TRUE Perform SA-MCSCF

RECOMMENDATION:

None

NEO_MCSCF

Run a NEO-MCSCF calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not run a NEO-MCSCF.

TRUE Run a NEO-MCSCF.

RECOMMENDATION:

None

NEO_MRCI_CIDEN_PRINT

Print the NEO CI protonic densities of the converged roots to .cube files

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print protonic densities.

TRUE Print protonic densities.

RECOMMENDATION:

None

NEO_MRCI_CIVEC_ANALYSIS

Perform brief analysis for CI vectors, largest contributing configurations.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE No analysis is performed.

TRUE Analysis is performed.

RECOMMENDATION:

FALSE

NEO_MRCI_CONV_THRESH

Convergence tolerance for roots of NEO-CI. It is defined as $10^{-\text{NEO\_MRCI\_CONV\_THRESH}}$

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

4 Sufficient for energies.

5 Gradients and couplings.

RECOMMENDATION:

4

NEO_MRCI_KSPACE_MAX

Maximum size of Krylov subspace.

TYPE:

INTEGER

DEFAULT:

NEO_MRCI_KSPACE_MIN + 5

OPTIONS:

User-defined

RECOMMENDATION:

Increasing the number of guess vectors before contraction improves CI convergence but has increased memory costs.

NEO_MRCI_KSPACE_MIN

Minimum size of Krylov subspace.

TYPE:

INTEGER

DEFAULT:

NEO_MRCI_NUM_ROOTS + 1

OPTIONS:

User-defined

RECOMMENDATION:

Increasing the number of guess vectors improves CI convergence but has increased memory costs.

NEO_MRCI_MAX_ITER

Maximum NEO CI iterations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_MRCI_NUM_ROOTS

Number of NEO CI states to converge.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

User-defined

RECOMMENDATION:

Due to the spectrum and structure of the NEO CI Hamiltonian, it is advisable to set NEO_MRCI_NUM_ROOTS to be greater than the number of desired roots.

NEO_MRCI_PREDIAG_DIM

Dimension for NEO_MRCI_PREDIAG_ROUTINE

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

User-defined

RECOMMENDATION:

The larger the size, the better the initial guess.

NEO_MRCI_PREDIAG_ROUTINE

Choose how initial guess vectors are created.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Linear combination of lowest NEO_MRCI_PREDIAG_DIM NEO-CI diagonal elements.

1 Eigenvectors of NEO_MRCI_PREDIAG_DIM x NEO_MRCI_PREDIAG_DIM block of NEO-CI Hamiltonian.

2 MCSCF vectors are used for initial guess.

RECOMMENDATION:

0

NEO_MRCI_PRINT_CIVECS

Print the final NEO CI vectors to civec.dat.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MRCI_PRINT_LEVEL

Determines print level for NEO CI calculation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 No convergence information is printed. Only iteration number and timings.

1 Convergence information is printed.

>1 Detailed printing for debugging.

RECOMMENDATION:

1

NEO_MRCI_READ_CIVECS

Read in NEO CI vectors from civec.dat.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

User-defined

RECOMMENDATION:

This can be used to restart from a previously unconverged CI calculation.

NEO_MRCI_READ_MOS

Read in MO coefficients from mocoefs.electronic and mocoefs.protonic.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

User-defined

RECOMMENDATION:

None

NEO_MRCI

Perform NEO Multireference Configuration Interaction

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Run a NEO-MRCI Calculation.

FALSE Do not run a NEO-MRCI Calculation.

RECOMMENDATION:

NONE

NEO_WF_E_AUX_NELECS

Maximum number of electronic *reference* excitations into the AUX orbitals from the RAS and AUX orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_AUX

Number of electronic *reference* auxilliary space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_CAS

Number of electronic *reference* complete active space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_DOCC

Number of electronic *reference* doubly-occupied orbitals *minus* the CI frozen core orbitals. There are **NEO_WF_E_XLEVEL** excitations out of these orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_FROZEN_CORE

Number of electronic frozen core orbitals. Orbitals are doubly occupied in all configurations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

Generally, 1s orbitals can be treated as frozen core orbitals.

NEO_WF_E_FROZEN_VIRT

Number of electronic frozen virtual orbitals. These orbitals are unoccupied in all configurations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

Virtual orbitals contribute to the dynamic correlation recovered by CI calculations,
so it is not recommended that this value is set, except in special cases.

NEO_WF_E_RAS_NHOLES

Maximum Number of electronic *reference* excitations out of the RAS orbitals
and into the CAS and AUX orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_RAS

Number of electronic *reference* restricted active space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_E_XLEVEL

Electronic excitation level of NEO-CI expansion: 1 = single, 2 = single and double, etc.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_N_CAS

Number of nuclear *reference* complete active space orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_N_DOCC

Number of nuclear *reference* doubly-occupied orbitals.

There are **NEO_WF_N_XLEVEL** excitations out of these orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_N_ELECS

Total number of electrons in wavefunction.

TYPE:

INTEGER

DEFAULT:

(Read from $molecule input.)

OPTIONS:

User-defined

RECOMMENDATION:

NONE

NEO_WF_N_FROZEN_VIRT

Number of nuclear frozen virtual orbitals. These orbitals are unoccupied in all configurations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

Virtual orbitals contribute to the dynamic correlation recovered by CI calculations, so it is not recommended that this value is set, except in special cases.

NEO_WF_N_XLEVEL

Nuclear excitation level of NEO-CI expansion: 1 = single, 2 = single and double, etc.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined

RECOMMENDATION:

NONE

RESPONSE_POLAR

Control the use of analytic or numerical polarizabilities.

TYPE:

INTEGER

DEFAULT:

0 or -1 = 0 for HF or DFT, -1 for all other methods

OPTIONS:

0 Perform an analytic polarizability calculation.

-1 Perform a numeric polarizability calculation even when analytic 2nd derivatives are available.

RECOMMENDATION:

None

ADC_CAP

Controls the type of CAP/ADC calculation to be performed.

TYPE:

INTEGER

DEFAULT:

0 Do not perform a CAP/ADC calculation.

OPTIONS:

1 Perform a subspace-projected CAP/ADC calculation.

RECOMMENDATION:

Set to 1 for the computation of CAP/ADC subspace projections.

ADC_CVS

Activates the use of the CVS approximation for the calculation of CVS-ADC core-excited states.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Activates the CVS approximation.

FALSE Do not compute core-excited states using the CVS approximation.

RECOMMENDATION:

Set to TRUE, if to obtain core-excited states for the simulation of X-ray absorption spectra. In the case of TRUE, the *$rem* variable CC_REST_OCC has to be defined as well.

ADC_C_C

Set the spin-opposite scaling parameter c_c for the ADC(2) calculation. The parameter value is obtained by multiplying the given integer by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

1170 Optimized value $c_c = 1.17$ for ADC(2)-s or

1000 $c_c = 1.0$ for ADC(2)-x

OPTIONS:

n Corresponding to $n \cdot 10^{-3}$

RECOMMENDATION:

Use the default.

ADC_C_T

Set the spin-opposite scaling parameter c_T for an SOS-ADC(2) calculation. The parameter value is obtained by multiplying the given integer by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

1300 Optimized value $c_T = 1.3$.

OPTIONS:

n Corresponding to $n \cdot 10^{-3}$

RECOMMENDATION:

Use the default.

ADC_C_X

Set the spin-opposite scaling parameter c_x for the ADC(2)-x calculation. The parameter value is obtained by multiplying the given integer by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

1300 Optimized value $c_x = 0.9$ for ADC(2)-x.

OPTIONS:

n Corresponding to $n \cdot 10^{-3}$

RECOMMENDATION:

Use the default.

ADC_DAVIDSON_CONV

Controls the convergence criterion of the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

$n \leq 12$ Corresponding to 10^{-n}.

RECOMMENDATION:

Use the default unless higher accuracy is required or convergence problems are encountered.

ADC_DAVIDSON_MAXITER

Controls the maximum number of iterations of the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

60

OPTIONS:

n Number of iterations

RECOMMENDATION:

Use the default unless convergence problems are encountered.

ADC_DAVIDSON_MAXSUBSPACE

Controls the maximum subspace size for the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

$5\times$ the number of excited states to be calculated.

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Should be at least 2–4$\times$ the number of excited states to calculate. The larger the value the more disk space is required.

ADC_DAVIDSON_THRESH

Controls the threshold for the norm of expansion vectors to be added during the Davidson procedure.

TYPE:

INTEGER

DEFAULT:

Twice the value of ADC_DAVIDSON_CONV, but at maximum 10^{-14}.

OPTIONS:

$n \leq 14$ Corresponding to 10^{-n}

RECOMMENDATION:

Use the default unless convergence problems are encountered. The threshold value 10^{-n} should always be smaller than the convergence criterion ADC_DAVIDSON_CONV.

ADC_DENSITY_MAXITER

When setting ADC_DENSITY_ORDER = 4, this keyword controls the maximum number of DIIS iterations carried out in the $\Sigma(4+)$ procedure.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Use the default value.

ADC_DENSITY_ORDER

Controls the order of the ground state density used for the computation of third-order ADC matrix elements (non-CVS methods only).

TYPE:

INTEGER

DEFAULT:

2 Use strict third-order ADC(3) schemes.

OPTIONS:

3 Use a third-order ground state density computed from the IP-ADC(3) effective transition moments and the corresponding fourth order static self-energy according to the $\Sigma(4)$ scheme

4 Use an improved third-order ground state density and the corresponding improved fourth-order static self-energy computed according to the self-consistent $\Sigma(4+)$ procedure

RECOMMENDATION:

In case of IP-ADC(3) calculations, employing the $\Sigma(4+)$ scheme provides more accurate ionization potentials and ionized state dipole moments.

ADC_DIIS_ECONV

Controls the convergence criterion for the excited state energy during DIIS.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

None

ADC_DIIS_MAXITER

Controls the maximum number of DIIS iterations.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Increase in case of slow convergence.

ADC_DIIS_RCONV

Convergence criterion for the residual vector norm of the excited state during DIIS.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

None

ADC_DIIS_SIZE

Controls the size of the DIIS subspace.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n User-defined integer

RECOMMENDATION:

None

ADC_DIIS_START

Controls the iteration step at which DIIS is turned on.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Set to a large number to switch off DIIS steps.

ADC_DIRECT

For third-order ADC methods, this keyword controls if some large intermediate tensor contractions should be carried out in advance and the result saved in memory for later use or if these quantities should be evaluated directly whenever they are encountered.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Directly evaluate some N^6-scaling tensor contractions. This will reduce the memory requirement by $\sim$10 %.

FALSE Precompute all possible N^6-scaling intermediates. This will speed up ADC(3) calculations considerably (by a factor of $\sim$3 in case of ADC(3) for N-electron excitations and somewhat less for IP- and EA-ADC(3)).

RECOMMENDATION:

Use the default value unless memory is the bottleneck.

ADC_DO_DIIS

Activates the use of the DIIS algorithm for the calculation of ADC(2) excited states.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use DIIS algorithm.

FALSE Do diagonalization using Davidson algorithm.

RECOMMENDATION:

None.

ADC_DO_DYSON

Controls if Dyson orbitals are output in case of IP- and EA-ADC calculations. This keyword only takes effect when used together with STATE_ANALYSIS = TRUE. See Section 10.2.12 for further details.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Output Dyson orbitals as cube files.

FALSE Do not output Dyson orbitals.

RECOMMENDATION:

Set to TRUE if visualization of ionization/electron-attachment processes is desired.

ADC_NGUESS_DOUBLES

Controls the number of excited state guess vectors which are double excitations, two-hole-one-particle ionizations and one-hole-two-particle electron-attachments in case of ADC, IP-ADC and EA-ADC, respectively.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined integer.

RECOMMENDATION:

ADC_NGUESS_SINGLES

Controls the number of excited state guess vectors which are single excitations, one-hole ionizations and one-particle electron-attachments in case of ADC, IP-ADC and EA-ADC, respectively. If the number of requested excited states exceeds the total number of guess vectors (singles and doubles), this parameter is automatically adjusted, so that the number of guess vectors matches the number of requested excited states.

TYPE:

INTEGER

DEFAULT:

Equals to the number of excited states requested.

OPTIONS:

n User-defined integer.

RECOMMENDATION:

Increase if there are convergence problems.

ADC_PRINT

Controls the amount of printing during an ADC calculation.

TYPE:

INTEGER

DEFAULT:

1 Basic status information and results are printed.

OPTIONS:

0 Quiet: almost only results are printed.

1 Normal: basic status information and results are printed.

2 Debug: more status information, extended information on timings.

...

RECOMMENDATION:

Use the default.

ADC_PROP_ES2ES

Controls the calculation of transition properties between excited, ionized or electron-attached states (currently only transition dipole moments and oscillator strengths). For ADC for N-electron excitations, this keyword also controls the computation of two-photon absorption cross-sections of excited states using the sum-over-states expression.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate state-to-state transition properties.

FALSE Do not compute transition properties between excited, ionized or electron-attached states.

RECOMMENDATION:

Set to TRUE, if state-to-state properties (ADC, IP-ADC, EA-ADC) or sum-over-states two-photon absorption cross-sections (only ADC) are required.

ADC_PROP_ES

Controls the calculation of excited, ionized or electron-attached state properties (currently only dipole moments and $\hat{r}^2$ expectation values).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate excited, ionized or electron-attached state properties.

FALSE Do not compute state properties.

RECOMMENDATION:

Set to TRUE, if properties are required.

ADC_PROP_TPA

Controls the calculation of two-photon absorption cross-sections of excited states using matrix inversion techniques.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate two-photon absorption cross-sections.

FALSE Do not compute two-photon absorption cross-sections.

RECOMMENDATION:

Set to TRUE, if to obtain two-photon absorption cross-sections.

ADC_STRICT_ISR

Controls how second-order ground state contributions are treated in the calculation of second- and third-order IP- and EA-ADC state properties using the second-order ISR formalism.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Scale the second-order part of the ground state contribution to one-electron properties of ionized/electron-attached states by the one-hole/one-particle character of the respective states as implied by the strict ISR derivation.

FALSE Use the full second-order ground state contribution for each ionized/electron-attached state property.

RECOMMENDATION:

Use the default value. Both options are, however, valid second-order treatments of ionized/electron-attached state properties and should yield very similar results for states with predominant one-hole/one-particle chaaracter.

ADD_CHARGED_CAGE

Add a point charge cage of a given radius and total charge.

TYPE:

INTEGER

DEFAULT:

0 No cage.

OPTIONS:

0 No cage.

1 Dodecahedral cage.

2 Spherical cage.

RECOMMENDATION:

Spherical cage is expected to yield more accurate results, especially for small radii.

ADIIS_INNER_CONV

Convergence criterion for the ADIIS inner loops (L-BFGS optimization of Eq. 4.43)

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

n Using 10^{-n} as the convergence criterion for the ADIIS inner loops

RECOMMENDATION:

Use the default

AFSSH

Adds decoherence approximation to surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Traditional surface hopping, no decoherence.

1 Use augmented fewest-switches surface hopping (AFSSH).

RECOMMENDATION:

AFSSH will increase the cost of the calculation, but may improve accuracy for some systems. See Refs. 22,36,37 for more detail.

AIFDEM_CTSTATES

Include charge-transfer-like cation/anion pair states in the AIFDEM basis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Include CT states.

FALSE Do not include CT states.

RECOMMENDATION:

Use if CT states are desired in the basis.

AIFDEM_EMBED_RANGE

Specifies the size of the QM region for charge embedding

TYPE:

INTEGER

DEFAULT:

FULL_QM

OPTIONS:

FULL_QM No charge embedding.

0 Treat only excited fragments with QM.

n Range (in Å) from excited fragments within which to treat other fragments with QM.

RECOMMENDATION:

The minimal threshold of zero typically maintains accuracy while significantly reducing computational time.

AIFDEM_FRGM_READ

Skips fragment SCF calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Skips fragment SCF calculations, only computation of matrix elements.

FALSE Regular AIFDEM calculation as specified by other *$rem* variables.

RECOMMENDATION:

Requires a prior calculation that computes fragment SCF data.

AIFDEM_FRGM_WRITE

Fragment SCF calculations only.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Only fragment SCF calculations are carried out, no computation of matrix elements.

FALSE Regular AIFDEM calculation as specified by other *$rem* variables.

RECOMMENDATION:

None

AIFDEM_NTOTHRESH

Controls how many NTOs that are retained in the exciton-site basis states.

TYPE:

INTEGER

DEFAULT:

99

OPTIONS:

n Retain enough NTOs to recover $n\%$ of the norm of the original CIS or TDDFT vectors in Eq. (12.80).

RECOMMENDATION:

A threshold of 85% gives a good trade-off of computational time and accuracy for organic molecules.

AIFDEM_SEGEND

Indicates the index of the last matrix element to be computed.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Last matrix element of thhe chunk to be computed.

RECOMMENDATION:

Needs to be used with AIFDEM_SEGSTART

AIFDEM_SEGSTART

Indicates the index of the first matrix element to be computed.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n First matrix element of the chunk to be computed.

RECOMMENDATION:

Needs to be used with AIFDEM_SEGEND

AIFDEM_SINGFIS

Include multi-exciton states in the AIFDEM basis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Include multi-exciton states.

FALSE Do not include multi-exciton states.

RECOMMENDATION:

Use if multi-exciton states are desired in the basis. This option requires the use of AIFDEM_SEGSTART and AIFDEM_SEGEND in the *$rem* section.

AIFDEM

Perform an AIFDEM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform an AIFDEM calculation.

TRUE Perform an AIFDEM calculation.

RECOMMENDATION:

False

AIMD_FICT_MASS

Specifies the value of the fictitious electronic mass μ, in atomic units, where μ has dimensions of (energy)$\times$(time)^2.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified

RECOMMENDATION:

Values in the range of 50–200 a.u. have been employed in test calculations; consult Ref. 18 for examples and discussion.

AIMD_INIT_VELOC_NANO_RANDOM

Uses a more precise random seed for generating random initial velocities.

TYPE:

LOGICAL

DEFAULT:

TRUE Use a more precise random seed.

OPTIONS:

FALSE Use a less precise random seed.

RECOMMENDATION:

Leave this set to TRUE unless necessary.

This option determines the source of the random seed used for sampling random initial velocities when AIMD_INIT_VELOC requires such. Setting the option to FALSE will have the seed based on the system time in seconds, meaning that two otherwise identical simulations starting in the same second will produce identical initial velocities. With the option set to TRUE, such collisions are virtually impossible.

The option is kept for legacy purposes. There should rarely ever be a need to set it to FALSE.

AIMD_INIT_VELOC

Specifies the method for selecting initial nuclear velocities.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

THERMAL	Random sampling of nuclear velocities from a Maxwell-Boltzmann distribution. The user must specify the temperature in Kelvin via the *$rem* variable AIMD_TEMP.
ZPE	Choose velocities in order to put zero-point vibrational energy into each normal mode, with random signs. This option requires that a frequency job to be run beforehand.
QUASICLASSICAL	Puts vibrational energy into each normal mode. In contrast to the ZPE option, here the vibrational energies are sampled from a Boltzmann distribution at the desired simulation temperature. This also triggers several other options, as described below.
OLD	Use the same initial velocities as the immediately preceding AIMD job.
RESTART	Use the final velocities from a previous AIMD job, reading them from disk.

RECOMMENDATION:

This variable need only be specified in the event that velocities are not specified explicitly in a *$velocity* section.

AIMD_LANGEVIN_TIMESCALE

Sets the timescale (strength) of the Langevin thermostat

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

n Thermostat timescale,asn n fs

RECOMMENDATION:

Smaller values (roughly 100) equate to tighter thermostats but may inhibit rapid sampling. Larger values (≥ 1000) allow for more rapid sampling but may take longer to reach thermal equilibrium.

AIMD_METHOD

Selects an *ab initio* molecular dynamics algorithm.

TYPE:

STRING

DEFAULT:

BOMD

OPTIONS:

BOMD Born-Oppenheimer molecular dynamics.

CURVY Curvy-steps Extended Lagrangian molecular dynamics.

QCMD Meyer-Miller nonadiabatic molecular dynamics.

RECOMMENDATION:

BOMD yields exact classical molecular dynamics, provided that the energy is tolerably conserved. ELMD is an approximation to exact classical dynamics whose validity should be tested for the properties of interest. QCMD initiates Meyer-Miller nonadiabatic dynamics methods including the symmetric quasiclassical approach or the traditional Ehrenfest method.

AIMD_MOMENTS

Requests that multipole moments be output at each time step.

TYPE:

INTEGER

DEFAULT:

0 Do not output multipole moments.

OPTIONS:

n Output the first n multipole moments.

RECOMMENDATION:

None

AIMD_NUCL_DACF_POINTS

Number of time points to use in the dipole auto-correlation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute dipole auto-correlation function.
$1 \leq n \leq$ AIMD_STEPS	Compute dipole auto-correlation function for last n timesteps of the trajectory.

RECOMMENDATION:

If the DACF is desired, set equal to the value of AIMD_STEPS.

AIMD_NUCL_SAMPLE_RATE

The rate at which sampling is performed for the velocity and/or dipole auto-correlation function(s). Specified as a multiple of steps; *i.e.*, sampling every step is 1.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

$1 \leq n \leq$ AIMD_STEPS	Update the velocity/dipole auto-correlation function every n steps.

RECOMMENDATION:

Since the velocity and dipole moment are routinely calculated for *ab initio* methods, this variable should almost always be set to 1 when the VACF or DACF is desired.

AIMD_NUCL_VACF_POINTS

Number of time points to use in the velocity auto-correlation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute velocity auto-correlation function.
$1 \leq n \leq$ AIMD_STEPS	Compute velocity auto-correlation function for last n time steps of the trajectory.

RECOMMENDATION:

If the VACF is desired, set equal to the value of AIMD_STEPS.

AIMD_PRINT

Specifies the verbosity of the output in AIMD.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

User-specified

RECOMMENDATION:

Higher values result in more verbose output.

AIMD_QCT_INITPOS

Chooses the initial geometry in a QCT-MD simulation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Use the equilibrium geometry.
n	Picks a random geometry according to the harmonic vibrational wave function.
$-n$	Generates n random geometries sampled from the harmonic vibrational wave function.

RECOMMENDATION:

None.

AIMD_QCT_WHICH_TRAJECTORY

Picks a set of vibrational quantum numbers from a random distribution.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n	Picks the nth set of random initial velocities.
$-n$	Uses an average over n random initial velocities.

RECOMMENDATION:

Pick a positive number if you want the initial velocities to correspond to a particular set of vibrational occupation numbers and choose a different number for each of your trajectories. If initial velocities are desired that corresponds to an average over n trajectories, pick a negative number.

AIMD_SHORT_TIME_STEP

Specifies a shorter electronic time step for FSSH calculations.

TYPE:

INTEGER

DEFAULT:

TIME_STEP

OPTIONS:

n Specify an electronic time step duration of n/AIMD_TIME_STEP_CONVERSION a.u. If n is less than the nuclear time step variable TIME_STEP, the electronic wave function will be integrated multiple times per nuclear time step, using a linear interpolation of nuclear quantities such as the energy gradient and derivative coupling. Note that n must divide TIME_STEP evenly.

RECOMMENDATION:

Make AIMD_SHORT_TIME_STEP as large as possible while keeping the trace of the density matrix close to unity during long simulations. Note that while specifying an appropriate duration for the electronic time step is essential for maintaining accurate wave function time evolution, the electronic-only time steps employ linear interpolation to estimate important quantities. Consequently, a short electronic time step is not a substitute for a reasonable nuclear time step.

AIMD_STEPS

Specifies the requested number of molecular dynamics steps.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

None.

AIMD_TEMP

Specifies a temperature (in Kelvin) for Maxwell-Boltzmann velocity sampling.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

User-specified number of Kelvin.

RECOMMENDATION:

This variable is only useful in conjunction with AIMD_INIT_VELOC = THERMAL. Note that the simulations are run at constant energy, rather than constant temperature, so the mean nuclear kinetic energy will fluctuate in the course of the simulation.

AIMD_THERMOSTAT

Applies thermostatting to AIMD trajectories.

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

LANGEVIN Stochastic, white-noise Langevin thermostat

NOSE_HOOVER Time-reversible, Nosé-Hoovery chain thermostat

RECOMMENDATION:

Use either thermostat for sampling the canonical (NVT) ensemble.

AIMD_TIME_STEP_CONVERSION

Modifies the molecular dynamics time step to increase granularity.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n The molecular dynamics time step is TIME_STEP/n a.u.

RECOMMENDATION:

None

AIRBED_ALPHA

Sets the value of α.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $\alpha = n/1000$

RECOMMENDATION:

0 or -1200 for hBN surface

AIRBED

Perform an AIRBED calculation.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Perform an AIRBED calculation.

False Don't perform an AIRBED calculation.

RECOMMENDATION:

Set the *$rem* variable DFT_D to EMPIRICAL_GRIMME.

ALMOCIS_FRAGOV

Doing ALMO-CIS/TDA calculations with transitions from occupied orbitals on the 1st fragment and virtuals in the full system

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Doing standard ALMO-CIS/TDA calculations (if LOCAL_CIS > 0)

1 Reading user-specified active fragment O-V pairs from the *$frag_ov_pairs section*

2 Excitations on the *first* fragment only

3 Excitations from the occupied orbitals on the *first* fragment to all virtuals in the system

RECOMMENDATION:

None

ANHAR_SEL

Select a subset of normal modes for subsequent anharmonic frequency analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE Use all normal modes

OPTIONS:

TRUE Select subset of normal modes

RECOMMENDATION:

None

ANHAR

Performing various nuclear vibrational theory (TOSH, VPT2, VCI) calculations to obtain vibrational anharmonic frequencies.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Carry out the anharmonic frequency calculation.

FALSE Do harmonic frequency calculation.

RECOMMENDATION:

Since this calculation involves the third and fourth derivatives at the minimum of the potential energy surface, it is recommended that the GEOM_OPT_TOL_DISPLACEMENT, GEOM_OPT_TOL_GRADIENT and GEOM_OPT_TOL_ENERGY tolerances are set tighter. Note that VPT2 calculations may fail if the system involves accidental degenerate resonances. See the VCI *$rem* variable for more details about increasing the accuracy of anharmonic calculations.

ANTIBOND

Triggers Antibond subroutine to generate antibonding orbitals after a converged SCF

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Does not localize the virtual space.

1 Localizes the virtual space, one antibonding for every bond.

2,3 Fill the virtual space with antibonding orbitals-like guesses.

4 Does Frozen Natural Orbitals and leaves them on scratch for future jobs or visualization.

RECOMMENDATION:

None

ARI_R0

Determines the value of the inner fitting radius (in Ångstroms)

TYPE:

INTEGER

DEFAULT:

4 A value of 4 Å will be added to the atomic van der Waals radius.

OPTIONS:

n User defined radius.

RECOMMENDATION:

For some systems the default value may be too small and the calculation will become unstable.

ARI_R1

Determines the value of the outer fitting radius (in Ångstroms)

TYPE:

INTEGER

DEFAULT:

5 A value of 5 Å will be added to the atomic van der Waals radius.

OPTIONS:

n User defined radius.

RECOMMENDATION:

For some systems the default value may be too small and the calculation will become unstable. This value also determines, in part, the smoothness of the potential energy surface.

ARI

Toggles the use of the atomic resolution-of-the-identity (ARI) approximation.

TYPE:

LOGICAL

DEFAULT:

FALSE ARI will not be used by default for an RI-JK calculation.

OPTIONS:

TRUE Turn on ARI.

RECOMMENDATION:

For large (especially 1D and 2D) molecules the approximation may yield significant improvements in Fock evaluation time.

ASCI_CDETS

Specifies the number of determinants to search over during ASCI wavefunction growth steps.

TYPE:

INTEGER

DEFAULT:

-5

OPTIONS:

$N > 0$ search from the top N determinants

$N < 0$ search from the top determinants whose cumulative weight in the wavefunction corresponds to $1 - 2^N$

RECOMMENDATION:

Using a dynamically determined value ($N < 0$) gives better results.

ASCI_DAVIDSON_GUESS

Specifies the truncated CI guess used for ASCI's Davidson solver.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

N Order of the truncated CI to solve explicitly ASCI Davidson guess.

RECOMMENDATION:

Accurate excited states and rapid convergence of the ground state benefit from a good zero-order guess for the low energy spectrum. The default is often sufficient.

ASCI_DIAG

Specifies the diagonalization procedure.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Davidson solver

2 Eigen sparse matrix solver

RECOMMENDATION:

Use 2 for best trade-off of speed and memory usage. If memory usage becomes to great, switch to 1.

ASCI_NDETS

Specifies the number of determinants to include in the ASCI wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N for a wavefunction with N determinants

RECOMMENDATION:

Typical ASCI expansions range from 50,000 to 2,000,000 determinants depending on active space size, complexity of problem, and desired accuracy

ASCI_RESTART

Specifies whether to initialize the ASCI wavefunction with the `wf_data` file.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE read CI coefficients from the `wf_data` file

FALSE do not read the CI coefficients from disk

RECOMMENDATION:

ASCI_SKIP_PT2

Specifies whether ASCI PT2 correction should be calculated.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE compute ASCI PT2 contribution

TRUE do not compute ASCI PT2 contribution

RECOMMENDATION:

The PT2 correction is essential to obtaining converged ASCI energies.

ASCI_SPIN_PURIFY

Indicates whether or not the ASCI wavefunction should be augmented with missing determinants to ensure a spin-pure state.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE augment the wavefunction with determinants to ensure a spin eigenstate

FALSE do not augment the wavefunction

RECOMMENDATION:

ASCI_USE_NAT_ORBS

Specifies whether rotation to a natural orbital basis should be carried out between growth steps.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE rotate to a natural orbital basis between growth wavefunction growth steps

FALSE do not rotate to a natural orbital basis

RECOMMENDATION:

Natural orbital rotations significantly improve the compactness and therefore accuracy of the ASCI wavefunction.

AUX_BASIS_CORR

Sets the auxiliary basis set for RI-MP2 to be used or invokes RI-MP2 in case of double-hybrid DFT or MP2

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

AUX_BASIS_J

Sets the auxiliary basis set for RI-J to be used or invokes RI-J

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

AUX_BASIS_K

Sets the auxiliary basis set for RI-K or occ-RI-K to be used or invokes occ-RI-K

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

AUX_BASIS

Sets the auxiliary basis set to be used

TYPE:

STRING

DEFAULT:

No default auxiliary basis set

OPTIONS:

General, Gen	User-defined. As for BASIS
Symbol	Use standard auxiliary basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and Basis Set Exchange to aid your selection.

BASIS2

Defines the (small) second basis set.

TYPE:

STRING

DEFAULT:

No default for the second basis set.

OPTIONS:

Symbol	Use standard basis sets as for BASIS.
BASIS2_GEN	General BASIS2
BASIS2_MIXED	Mixed BASIS2

RECOMMENDATION:

BASIS2 should be smaller than BASIS. There is little advantage to using a basis larger than a minimal basis when BASIS2 is used for initial guess purposes. Larger, standardized BASIS2 options are available for dual-basis calculations as discussed in Section 4.7 and summarized in Table 4.7.4.

BASISPROJTYPE

Determines which method to use when projecting the density matrix of BASIS2

TYPE:

STRING

DEFAULT:

OVPROJECTION

OPTIONS:

FOPPROJECTION	Construct the Fock matrix in the second basis
OVPROJECTION	Projects MOs from BASIS2 to BASIS.

RECOMMENDATION:

None

BASIS_LIN_DEP_THRESH

Sets the threshold for determining linear dependence in the basis set

TYPE:

INTEGER

DEFAULT:

6 Corresponding to a threshold of 10^{-6}

OPTIONS:

n Sets the threshold to 10^{-n}

RECOMMENDATION:

Set to 5 or smaller if you have a poorly behaved SCF and you suspect linear dependence in you basis set. Lower values (larger thresholds) may affect the accuracy of the calculation.

BASIS

Sets the basis set to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen	User-defined. See section below
Symbol	Use standard basis sets as in the table below
Mixed	Use a combination of different basis sets

RECOMMENDATION:

Consult literature and reviews to aid your selection.

BECKE_SHIFT

Controls atomic cell shifting in determination of Becke weights.

TYPE:

STRING

DEFAULT:

BRAGG_SLATER

OPTIONS:

UNSHIFTED	Use Becke weighting without atomic size corrections, based on bond midpoints.
BRAGG_SLATER	Use the empirical radii introduced by Bragg and Slater.
UNIVERSAL_DENSITY	Use the *ab initio* radii introduced by Pacios.

RECOMMENDATION:

If interested in the partitioning of the default atomic quadrature, use UNSHIFTED. If using for physical interpretation, choose BRAGG_SLATER or UNIVERSAL_DENSITY. All cDFT calculations and calculations where POP_BECKE = TRUE will default to BRAGG_SLATER radii, otherwise the default grid is UNSHIFTED.

BONDED_EDA

Use the bonded ALMO-EDA.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform bonded ALMO-EDA.

1 Perform ALMO-EDA with non-orthogonal CI.

2 Perform ALMO-EDA with spin-projected formalism.

RECOMMENDATION:

Set to 2 for all cases where the supersystem is closed shell, only use 1 for cases where the fragments have more than one unpaired spin each.

BOYSCALC

Specifies how Boys localized orbitals are to be calculated

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform any Boys localization.

1 Localize core and valence together.

2 Do separate localizations on core and valence.

RECOMMENDATION:

None

BOYS_CIS_NUMSTATE

Define how many states to mix with Boys localized diabatization. These states must be specified in the *$localized_diabatization* section.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Boys localized diabatization.

OPTIONS:

2 to N where N is the number of CIS states requested (CIS_N_ROOTS)

RECOMMENDATION:

It is usually not wise to mix adiabatic states that are separated by more than a few eV or a typical reorganization energy in solvent.

CAGE_CHARGE

Defines the total charge of the cage.

TYPE:

INTEGER

DEFAULT:

400 Add a cage charged +4e.

OPTIONS:

n Total charge of the cage is $n/100$ a.u.

RECOMMENDATION:

None

CAGE_POINTS

Defines number of point charges for the spherical cage.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n Number of point charges to use.

RECOMMENDATION:

None

CAGE_RADIUS

Defines radius of the charged cage.

TYPE:

INTEGER

DEFAULT:

225

OPTIONS:

n radius is $n/100$ Å.

RECOMMENDATION:

None

CALC_NAC

Whether or not nonadiabatic couplings will be calculated for the EOM-CC, CIS, and TDDFT wave functions.

TYPE:

INTEGER

DEFAULT:

0 (do not compute NAC)

OPTIONS:

1 NYI for EOM-CC

2 Compute NACs using Szalay's approach (this what needs to be specified for EOM-CC).

RECOMMENDATION:

Additional response equations will be solved and gradients for all EOM states and for summed states will be computed, which increases the cost of calculations. Request only when needed and do not ask for too many EOM states.

CALC_SOC

Whether or not the spin-orbit couplings between CC/EOM/ADC/CIS/TDDFT electronic states will be calculated. In the CC/EOM-CC suite, by default the couplings are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate couplings between EOM states, CC_STATE_TO_OPT must specify the initial EOM state. If NTO analysis is requested, analysis of spinless transition density matrices will be performed and the spin-orbit integrals over NTO pairs will be printed.

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE (no spin-orbit couplings will be calculated)

OPTIONS:

0/FALSE	(no spin-orbit couplings will be calculated)
1/TRUE	Activates SOC calculation. EOM-CC/EOM-MP2 only: spin-orbit couplings will be computed with the new code with L+/L- averaging
2	EOM-CC/EOM-MP2 only: spin-orbit couplings will be computed with the new code without L+/L- averaging
3	EOM-CC/EOM-MP2 only: spin-orbit couplings will be computed with the legacy code
4	One-electron spin-orbit couplings will be computed with effective nuclear charges (with L+/L- averaging for EOM-CC/MP2)

RECOMMENDATION:

CCMAN2 supports several variants of SOC calculation for EOM-CC/EOM-MP2 methods. One-electron and mean-field two-electron SOCs will be computed by default. To enable full two-electron SOCs, two-particle EOM properties must be turned on (see CC_EOM_PROP_TE).

CALC_SOC

Controls whether to calculate the SOC constants for EOM-CC, RAS-CI, CIS, TDDFT/TDA and TDDFT/RPA.

TYPE:

INTEGER/LOGICAL

DEFAULT:

0

OPTIONS:

0	Do not perform the SOC calculation.
1	Perform the SOC calculation using the one-electron part of the Breit-Pauli Hamiltonian
2	Perform the full SOC calculation using a mean-field treatment of the two-electron contribution
4	Perform the SOC calculation using scaled nuclear charges

RECOMMENDATION:

The final two options are described in Section 7.10.20.4.

CAP_AIMD_SWITCH

Sets CAP_ETA to zero during a CAP-AIMD simulation when the real part of the last alpha occupied orbital's energy is negative

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Set CAP_ETA to zero when the real part of the last alpha occupied orbital's becomes negative.

FALSE Keep user's CAP_ETA constant throughout simulation.

RECOMMENDATION:

Use default.

CAP_X_END

Controls the upper onset limit for a series of CAP onsets, where the lower limit is given by CAP_X. The parameter value in a.u. is obtained by multiplying the given integer by 10^{-3}. Currently only used in ADC methods.

TYPE:

INTEGER

DEFAULT:

CAP_X Do not compute a series of CAP onsets but only use a single CAP with an onset value of CAP_X.

OPTIONS:

$n >$ CAP_X User-defined integer.

RECOMMENDATION:

Use this keyword if CAP onset series are desired.

CAP_X_STEP

Controls the step size for a series of CAP onsets between CAP_X and CAP_X_END. The parameter value in a.u. is obtained by multiplying the given integer by 10^{-3}. Currently only used in ADC methods.

TYPE:

INTEGER

DEFAULT:

500 corresponding to 0.5 a.u.

OPTIONS:

$n > 0$ User-defined integer.

RECOMMENDATION:

None.

CAP_X

For ADC methods, in combination with a smoothed Voronoi-CAP (CAP_TYPE = 2) or a spherical CAP (CAP_TYPE = 0), this keyword controls the lower limit for a series of CAP onsets, where the upper limit is given by CAP_X_END. The parameter value in a.u. is obtained by multiplying the given integer by 10^{-3}. In this case, the onset value defines the region around the molecule with zero CAP strength. In combination with a cuboid CAP (CAP_TYPE = 1) or in general for other electronic structure methods (see 7.10.9 for further details), this keyword controls the CAP onset in x direction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

$n > 0$ User-defined integer.

RECOMMENDATION:

Usually, values of 2000 to 4000 (corresponding to onset values between 2.0 and 4.0 a.u.) give reasonable results.

CAS_DAVIDSON_MAXVECTORS

Specifies the maximum number of vectors to augment the Davidson search space in CAS.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

N sets the maximum Davidson subspace size to N+CAS_N_ROOTS

RECOMMENDATION:

The default should be suitable in most cases

CAS_DAVIDSON_TOL

Specifies the tolerance for the Davidson solver used in CAS.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

The default should be suitable in most cases

CAS_DO_1X

Do perturbative hole (h) and particle (p) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative hole (h) and particle (p) correction

FALSE Do not do perturbative hole (h) and particle (p) correction

RECOMMENDATION:

None.

CAS_DO_2x

Do perturbative 2x (h,p,hp,hh,pp) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative 2x correction

FALSE Do not do perturbative 2x correction

RECOMMENDATION:

None.

CAS_DO_3x

Do perturbative 3x (h,p,hp,hh,pp,hhp,hpp) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative 3x correction

FALSE Do not do perturbative 3x correction

RECOMMENDATION:

None.

CAS_DO_DOUBLES

Do perturbative (h,p,hp,hh,pp,hhp,hpp) correction + MP2 RAS1 $\rightarrow$ RAS3 doubles?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative (h,p,hp,hh,pp,hhp,hpp) + MP2 RAS1$\rightarrow$ RAS3 doubles correction

FALSE Do not do the correction

RECOMMENDATION:

None.

CAS_DO_NDPT

Do non-degenerate perturbation theory?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do non-degenerate perturbation theory.

FALSE Do not use non-degenerate perturbation theory.

RECOMMENDATION:

None.

CAS_DO_SINGLES

Do perturbative singles (h,p,hp) correction?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do perturbative singles correction

FALSE Do not do perturbative singles correction

RECOMMENDATION:

None.

CAS_LEVEL_SHIFT

Use a denominator level-shift?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use the denominator level-shift

FALSE Do not use the denominator level-shift

RECOMMENDATION:

None.

CAS_LOCAL_ALGO

Passed into localizer. Set to 1 if doing Boys localization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No localization

1 Boys localization

2 Pipek-Mezey localization

RECOMMENDATION:

None.

CAS_LOCAL

Determines whether to do localization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No localization

1 Boys localization

2 Pipek-Mezey localization

RECOMMENDATION:

None.

CAS_METHOD

Indicates whether orbital optimization is requested.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Not running a CAS calculation

1 CAS-CI (no orbital optimization)

2 CASSCF (orbital optimization)

RECOMMENDATION:

Use 2 for best accuracy, but such computations may become infeasible for large active spaces.

CAS_M_S

The number of unpaired electrons desired in the CAS wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N for a wavefunction with N unpaired electrons

RECOMMENDATION:

CAS_N_ELEC

Specifies the number of active electrons.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N include N electrons in the active space

-1 include all electrons in the active space

RECOMMENDATION:

Use the smallest active space possible for the given system.

CAS_N_ORB

Specifies the number of active orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

N include N orbitals in the active space

-1 include all orbitals in the active space

RECOMMENDATION:

Use the smallest active space possible for the given system.

CAS_N_ROOTS

Specifies the number of electronic states to determine.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

N solve for N roots of the Hamiltonian

RECOMMENDATION:

CAS_QDPT_ORDER

Order of terms kept in the quasi-degenerate perturbation theory denominator expansion.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

n Keep terms of order n in the denominator expansion.

RECOMMENDATION:

None.

CAS_SAVE_NAT_ORBS

Save the CAS natural orbitals in place of the reference orbitals.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE overwrite the reference orbitals with CAS natural orbitals

FALSE do not save the CAS natural orbitals

RECOMMENDATION:

CAS_SOLVER

Specifies the solver to be used for the active space.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 CAS-CI/CASSCF

2 ASCI (see Section 6.22)

3 Truncated CI (CIS, CISD, CISDT, etc.)

RECOMMENDATION:

CAS_SPARSE

Use a sparse matrix multiply when forming the effective Hamiltonian?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use sparse matrix multiply in forming effective Hamiltonian

FALSE Do not use sparse matrix multiply in forming effective Hamiltonian

RECOMMENDATION:

None. Can be useful for larger numbers of spin-flips.

CAS_THRESH

Specifies the threshold for matrix elements to be included in the CAS Hamiltonian.

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

CAS_USE_RI

Indicates whether the resolution of the identity approximation should be used.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Compute 2-electron integrals analytically

TRUE Use the RI approximation for 2-electron integrals

RECOMMENDATION:

Analytic integrals are more accurate, RI integrals are faster

CCVB_GUESS

Specifies the initial guess for CCVB calculations

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 Standard GVBMAN guess (orbital localization via GVB_LOCAL + Sano procedure).

2 Use orbitals from previous GVBMAN calculation, along with SCF_GUESS = READ.

3 Convert UHF orbitals into pairing VB form.

RECOMMENDATION:

Option 1 is the most useful overall. The success of GVBMAN methods is often dependent on localized orbitals, and this guess shoots for these. Option 2 is useful for comparing results to other GVBMAN methods, or if other GVBMAN methods are able to obtain a desired result more efficiently. Option 3 can be useful for bond-breaking situations when a pertinent UHF solution has been found. It works best for small systems, or if the unrestriction is a local phenomenon within a larger molecule. If the unrestriction is non-local and the system is large, this guess will often produce a solution that is not the global minimum. Any UHF solution has a certain number of pairs that are unrestricted, and this will be output by the program. If GVB_N_PAIRS exceeds this number, the standard GVBMAN initial-guess procedure will be used to obtain a guess for the excess pairs

CCVB_METHOD

Optionally modifies the basic CCVB method

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Standard CCVB model

3 Independent electron pair approximation (IEPA) to CCVB

4 Variational PP (the CCVB reference energy)

RECOMMENDATION:

Option 1 is generally recommended. Option 4 is useful for preconditioning, and for obtaining localized-orbital solutions, which may be used in subsequent calculations. It is also useful for cases in which the regular GVBMAN PP code becomes variationally unstable. Option 3 is a simple independent-amplitude approximation to CCVB. It avoids the cubic-scaling amplitude equations of CCVB, and also is able to reach the correct dissociation energy for any molecular system (unlike regular CCVB which does so only for cases in which UHF can reach a correct dissociate limit). However the IEPA approximation to CCVB is sometimes variationally unstable, which we have yet to observe in regular CCVB.

CC_1HPOL

Specifies the approach for calculating the first hyperpolarizability of the CCSD wave function.

TYPE:

INTEGER

DEFAULT:

0 (CCSD first hyperpolarizability will not be calculated)

OPTIONS:

1 (damped-response expectation-value approach with only first-order response wave functions)

3 (damped-response expectation-value approach with second-order response density matrices for wave-function and natural orbital analyses)

RECOMMENDATION:

CCSD first hyperpolarizabilities are expensive since they require solving a huge number of first- and second-order response equations. Do no request this property unless you need it.

CC_BACKEND

Used to specify the computational back-end of CCMAN2.

TYPE:

STRING

DEFAULT:

VM Default shared-memory disk-based back-end

OPTIONS:

XM *libxm* shared-memory disk-based back-end

INCORE in-core memory back-end

RECOMMENDATION:

Use XM for large jobs with limited memory or when the performance of the default disk-based back-end is not satisfactory, INCORE for small jobs that fit in main memory.

CC_CANONIZE_FINAL

Whether to semi-canonicalize orbitals at the end of the ground state calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE unless required

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Should not normally have to be altered.

CC_CANONIZE_FREQ

The orbitals will be semi-canonicalized every n theta resets. The thetas (orbital rotation angles) are reset every CC_RESET_THETA iterations. The counting of iterations differs for active space (VOD, VQCCD) calculations, where the orbitals are always canonicalized at the first theta-reset.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n User-defined integer

RECOMMENDATION:

Smaller values can be tried in cases that do not converge.

CC_CANONIZE

Whether to semi-canonicalize orbitals at the start of the calculation (*i.e.* Fock matrix is diagonalized in each orbital subspace)

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Should not normally have to be altered.

CC_CONVERGENCE

Overall convergence criterion for the coupled-cluster codes. This is designed to ensure at least n significant digits in the calculated energy, and automatically sets the other convergence-related variables (CC_E_CONV, CC_T_CONV, CC_THETA_CONV, CC_THETA_GRAD_CONV) [10^{-n}].

TYPE:

INTEGER

DEFAULT:

6 Energies.

7 Gradients.

OPTIONS:

n Corresponding to 10^{-n} convergence criterion. Amplitude convergence is set automatically to match energy convergence.

RECOMMENDATION:

Use the default

CC_DIIS12_SWITCH

When to switch from DIIS2 to DIIS1 procedure, or when DIIS2 procedure is required to generate DIIS guesses less frequently. Total value of DIIS error vector must be less than 10^{-n}, where n is the value of this option.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n User-defined integer

RECOMMENDATION:

None

CC_DIIS_FREQ

DIIS extrapolation will be attempted every n iterations. However, DIIS2 will be attempted every iteration while total error vector exceeds CC_DIIS12_SWITCH. DIIS1 cannot generate guesses more frequently than every 2 iterations.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

N User-defined integer

RECOMMENDATION:

None

CC_DIIS_MAX_OVERLAP

DIIS extrapolations will not begin until square root of the maximum element of the error overlap matrix drops below this value.

TYPE:

DOUBLE

DEFAULT:

100 Corresponding to 1.0

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-de}$

RECOMMENDATION:

None

CC_DIIS_MIN_OVERLAP

The DIIS procedure will be halted when the square root of smallest element of the error overlap matrix is less than 10^{-n}, where n is the value of this option. Small values of the B matrix mean it will become near-singular, making the DIIS equations difficult to solve.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

n User-defined integer

RECOMMENDATION:

None

CC_DIIS_SIZE

Specifies the maximum size of the DIIS space.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n User-defined integer

RECOMMENDATION:

Larger values involve larger amounts of disk storage.

CC_DIIS_START

Iteration number when DIIS is turned on. Set to a large number to disable DIIS.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n User-defined

RECOMMENDATION:

Occasionally DIIS can cause optimized orbital coupled-cluster calculations to diverge through large orbital changes. If this is seen, DIIS should be disabled.

CC_DIIS

Specify the version of Pulay's Direct Inversion of the Iterative Subspace (DIIS) convergence accelerator to be used in the coupled-cluster code.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Activates procedure 2 initially, and procedure 1 when gradients are smaller than DIIS12_SWITCH.

1 Uses error vectors defined as differences between parameter vectors from successive iterations. Most efficient near convergence.

2 Error vectors are defined as gradients scaled by square root of the approximate diagonal Hessian. Most efficient far from convergence.

RECOMMENDATION:

DIIS1 can be more stable. If DIIS problems are encountered in the early stages of a calculation (when gradients are large) try DIIS1.

CC_DIRECT_RI

Controls use of RI and Cholesky integrals in conventional (undecomposed) form

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE use all integrals in decomposed format

TRUE transform all RI or Cholesky integral back to conventional format

RECOMMENDATION:

By default all integrals are used in decomposed format allowing significant reduction of memory use. If all integrals are transformed back (TRUE option) no memory reduction is achieved and decomposition error is introduced, however, the integral transformation is performed significantly faster and conventional CC/EOM algorithms are used.

CC_DOV_THRESH

Specifies minimum allowed values for the coupled-cluster energy denominators. Smaller values are replaced by this constant during early iterations only, so the final results are unaffected, but initial convergence is improved when the HOMO-LUMO gap is small or when non-conventional references are used.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

abcde Integer code is mapped to $ab \times 10^{-de}$, *e.g.*, 2501 corresponds to 0.025, 99001 corresponds to 0.99, etc.

RECOMMENDATION:

Increase to 0.25, 0.5 or 0.75 for non convergent coupled-cluster calculations.

CC_DO_DYSON_EE

Whether excited-state or spin-flip state Dyson orbitals will be calculated for EOM-IP/EA-CCSD calculations with CCMAN.

TYPE:

LOGICAL

DEFAULT:

FALSE (the option must be specified to run this calculation)

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

none

CC_DO_DYSON

CCMAN2: starts all types of Dyson orbitals calculations. Desired type is determined by requesting corresponding EOM-XX transitions CCMAN: whether the reference-state Dyson orbitals will be calculated for EOM-IP/EA-CCSD calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE (the option must be specified to run this calculation)

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

none

CC_DO_FESHBACH

Activates calculation of resonance widths using Feshbach-Fano approach.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not invoke Feshbach-Fano calculation

1 invoke Feshbach-Fano calculation of the resonance width

2 invoke Feshbach-Fano calculation of the resonance width and resonance shift

RECOMMENDATION:

Initial and final states should be correctly specified.

CC_DO_NAO

Activates calculation of NAOs within Feshbach–Fano calculation of the decay widths.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not compute NAOs

1 compute NAOs

RECOMMENDATION:

Initial and final states should be correctly specified.

CC_EOM_2PA_SINGLE_PREC

Precision selection for 2PA response equations. Available in CCMAN2 only.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

RECOMMENDATION:

NONE

CC_EOM_2PA_XCONV

Convergence criterion for the response vectors (norm of the difference) of the DIIS solver for damped response equations in 2PA and RIXS calculations.

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion.

RECOMMENDATION:

Use the default in double precision. May reduce in single precision.

CC_EOM_2PA

Whether or not the transition moments and cross-sections for two-photon absorption will be calculated. By default, the transition moments are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition moments between a set of EOM-CCSD states and another EOM-CCSD state, the CC_STATE_TO_OPT must be specified for this state. If 2PA NTO analysis is requested, the CC_EOM_2PA value is redundant as long as CC_EOM_2PA > 0.

TYPE:

INTEGER

DEFAULT:

0 (do not compute 2PA transition moments)

OPTIONS:

1 Compute 2PA using the fastest algorithm (use $\tilde{\sigma}$-intermediates for canonical and σ-intermediates for RI/CD response calculations).

2 Use σ-intermediates for 2PA response equation calculations.

3 Use $\tilde{\sigma}$-intermediates for 2PA response equation calculations.

RECOMMENDATION:

Additional response equations (6 for each target state) will be solved, which increases the cost of calculations. The cost of 2PA moments is about 10 times that of energy calculation. Use the default algorithm. Setting CC_EOM_2PA > 0 turns on CC_TRANS_PROP.

CC_EOM_ECD

Whether or not the ECD transition moments will be calculated. By default, the transition moments are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition moments between a set of EOM-CCSD states and another EOM-CCSD state, the CC_STATE_TO_OPT must be specified for this state.

TYPE:

LOGICAL

DEFAULT:

FALSE (do not compute ECD transition moments)

OPTIONS:

TRUE Compute ECD transition moments.

RECOMMENDATION:

Activate for chiral molecules only.

CC_EOM_PROP_TE

Request for calculation of non-relaxed two-particle EOM-CC properties. The two-particle properties currently include $\langle \hat{S}^2 \rangle$. The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is inferior compared to the cost of $\langle \hat{S}^2 \rangle$. The variable CC_EOM_PROP must be also set to TRUE. Alternatively, CC_CALC_SSQ can be used to request $\langle \hat{S}^2 \rangle$ calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are computationally expensive since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation). Do not request the two-particle properties unless you really need them.

CC_EOM_PROP

Whether or not the non-relaxed (expectation value) one-particle EOM-CCSD target state properties will be calculated. Available properties currently include permanent dipole moment, angular momentum projections, the second moments ($\langle x^2 \rangle$, $\langle y^2 \rangle$, and $\langle z^2 \rangle$) of the electron density along with $\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$. This option is incompatible with JOBTYPE = FORCE, OPT, or FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (EOM-CCSD equations for the left eigenvectors) need to be solved for properties, approximately doubling the cost of calculation for each irrep. The cost of the one-particle properties calculation itself is low. The one-particle density of an EOM-CCSD target state can be analyzed with NBO or LIBWFA packages by specifying the state with CC_STATE_TO_OPT and requesting NBO = TRUE and CC_EOM_PROP = TRUE.

CC_EOM_RIXS

Whether or not the RIXS scattering moments and cross-sections will be calculated.

TYPE:

INTEGER

DEFAULT:

0 do not compute RIXS cross-sections

OPTIONS:

1 Perform RIXS within fc-CVS-EOM-EE-CCSD using the response wave functions of the CCSD reference state only

2 Perform RIXS within fc-CVS-EOM-EE-CCSD response theory along with the wave-function analysis of RIXS transition density matrices

11 Perform RIXS within the standard EOM-EE-CCSD using the response wave functions of the CCSD reference state only

12 Use σ-intermediates for RIXS response calculations within the standard EOM-EE-CCSD

RECOMMENDATION:

Use 1 to deploy fc-CVS-EOM-EE-CCSD with robust convergence

CC_ERASE_DP_INTEGRALS

Controls storage of requisite objects computed with double precision in a single-precision calculation.

TYPE:

INTEGER

DEFAULT:

0 store

OPTIONS:

1 do not store

RECOMMENDATION:

Do not erase integrals if clean-up in double precision is intended.

CC_E_CONV

Convergence desired on the change in total energy, between iterations.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n 10^{-n} convergence criterion.

RECOMMENDATION:

None

CC_FESHBACH_CW

Activates Coulomb wave description of the ejected electron.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use plane wave

1 Use Coulomb wave generated internally

3 Use Coulomb basis provided by the user in the *$coulomb_wave* section

RECOMMENDATION:

Additional details need to be specified in *$coulomb_wave* section.

CC_FESHBACH_DELTA_INTB

Specifies integration limits in calculation of energy shift in Feshbach-Fano calculations.

TYPE:

INTEGER

DEFAULT:

Preset

OPTIONS:

n corresponds to energy limit in eV

RECOMMENDATION:

Use default.

CC_FESHBACH_DELTA_INTC

Specifies integration limits in calculation of energy shift in Feshbach-Fano calculations.

TYPE:

INTEGER

DEFAULT:

Preset

OPTIONS:

n corresponds to energy limit in eV

RECOMMENDATION:

Use default.

CC_FESHBACH_INT_ORDER

Controls k-vector integration grid in calculations of resonance widths using Feshbach-Fano approach.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n corresponds to the Lebedev quadrature order

RECOMMENDATION:

Use default, unless tighter convergence is desired (16 gives fully converged widths).

CC_FNO_THRESH

Initialize the FNO truncation and sets the threshold to be used for both cutoffs (OCCT and POVO)

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

range 0000-10000

$abcd$ Corresponding to $ab.cd\%$

RECOMMENDATION:

None

CC_FNO_USEPOP

Selection of the truncation scheme

TYPE:

INTEGER

DEFAULT:

1 OCCT

OPTIONS:

0 POVO

RECOMMENDATION:

None

CC_FREE_ELECTRON

Specifies whether properties involving free electrons will be computed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

CC_FULLRESPONSE

Fully relaxed properties (including orbital relaxation terms) will be computed. The variable CC_REF_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no orbital response will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Not available for non UHF/RHF references and for the methods that do not have analytic gradients (*e.g.*, QCISD).

CC_HESS_THRESH

Minimum allowed value for the orbital Hessian. Smaller values are replaced by this constant.

TYPE:

DOUBLE

DEFAULT:

102 Corresponding to 0.01

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-de}$

RECOMMENDATION:

None

CC_INCL_CORE_CORR

Whether to include the correlation contribution from frozen core orbitals in non iterative (2) corrections, such as OD(2) and CCSD(2).

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

Use the default unless no core-valence or core correlation is desired (*e.g.*, for comparison with other methods or because the basis used cannot describe core correlation).

CC_ITERATE_ON

In active space calculations, use a "mixed" iteration procedure if the value is greater than 0. Then if the RMS orbital gradient is larger than the value of CC_THETA_GRAD_THRESH, micro-iterations will be performed to converge the occupied-virtual mixing angles for the current active space. The maximum number of space iterations is given by this option.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Up to n occupied-virtual iterations per overall cycle

RECOMMENDATION:

Can be useful for non-convergent active space calculations

CC_ITERATE_OV

In active space calculations, use a "mixed" iteration procedure if the value is greater than 0. Then, if the RMS orbital gradient is larger than the value of CC_THETA_GRAD_THRESH, micro-iterations will be performed to converge the occupied-virtual mixing angles for the current active space. The maximum number of such iterations is given by this option.

TYPE:

INTEGER

DEFAULT:

0 No "mixed" iterations

OPTIONS:

n Up to n occupied-virtual iterations per overall cycle

RECOMMENDATION:

Can be useful for non-convergent active space calculations.

CC_MAX_ITER

Maximum number of iterations to optimize the coupled-cluster energy.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

n up to n iterations to achieve convergence.

RECOMMENDATION:

None

CC_MEMORY

Specifies the maximum size, in MB, of the buffers for in-core storage of block-tensors in CCMAN and CCMAN2.

TYPE:

INTEGER

DEFAULT:

50% of MEM_TOTAL. If MEM_TOTAL is not set, use 1.5 GB. A minimum of 192 MB is hard-coded.

OPTIONS:

n Integer number of MB

RECOMMENDATION:

Larger values can give better I/O performance and are recommended for systems with large memory (add to your `.qchemrc` file. When running CCMAN2 exclusively on a node, CC_MEMORY should be set to 75–80% of the total available RAM.)

CC_MP2NO_GRAD

If CC_MP2NO_GUESS is TRUE, what kind of one-particle density matrix is used to make the guess orbitals?

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE 1 PDM from MP2 gradient theory.

FALSE 1 PDM expanded to 2^{nd} order in perturbation theory.

RECOMMENDATION:

The two definitions give generally similar performance.

CC_MP2NO_GUESS

Will guess orbitals be natural orbitals of the MP2 wave function? Alternatively, it is possible to use an effective one-particle density matrix to define the natural orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use natural orbitals from an MP2 one-particle density matrix (see *CC_MP2NO_GRAD*).

FALSE Use current molecular orbitals from SCF.

RECOMMENDATION:

None

CC_ORBS_PER_BLOCK

Specifies target (and maximum) size of blocks in orbital space.

TYPE:

INTEGER

DEFAULT:

16

OPTIONS:

n Orbital block size of n orbitals.

RECOMMENDATION:

None

CC_OSFNO

Activation of OSFNO. Available only for open-shell references.

TYPE:

LOGICAL

DEFAULT:

FALSE do not activate

OPTIONS:

TRUE activate

RECOMMENDATION:

Use for EOM-SF-CCSD calculations from open-shell references. Available in CCMAN2 only.

CC_POL

Specifies the approach for calculating the polarizability of the CCSD wave function.

TYPE:

INTEGER

DEFAULT:

0 (CCSD polarizability will not be calculated)

OPTIONS:

1 (analytic-derivative or response-theory mixed symmetric-asymmetric approach)

2 (analytic-derivative or response-theory asymmetric approach)

3 (expectation-value approach with right response intermediates)

4 (expectation-value approach with left response intermediates)

13 (damped-response expectation-value approach with right response intermediates)

14 (damped-response expectation-value approach with left response intermediates)

15 (damped-response expectation-value approach with first-order response density matrices)

23 (expectation-value approach with right response intermediates and imaginary frequencies)

RECOMMENDATION:

CCSD polarizabilities are expensive since they require solving three/six (for static) or six/twelve (for dynamic) additional response equations. Do no request this property unless you need it.

CC_PRECONV_FZ

In active space methods, whether to pre-converge other wave function variables for fixed initial guess of active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No pre-iterations before active space optimization begins.

n Maximum number of pre-iterations via this procedure.

RECOMMENDATION:

None

CC_PRECONV_T2Z_EACH

Whether to pre-converge the cluster amplitudes before each change of the orbitals in optimized orbital coupled-cluster methods. The maximum number of iterations in this pre-convergence procedure is given by the value of this parameter.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to n iterations in this pre-convergence procedure.

RECOMMENDATION:

A very slow last resort option for jobs that do not converge.

CC_PRECONV_T2Z

Whether to pre-converge the cluster amplitudes before beginning orbital optimization in optimized orbital cluster methods.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

10 If CC_RESTART, CC_RESTART_NO_SCF or CC_MP2NO_GUESS are TRUE

OPTIONS:

0 No pre-convergence before orbital optimization.

n Up to n iterations in this pre-convergence procedure.

RECOMMENDATION:

Experiment with this option in cases of convergence failure.

CC_PRINT

Controls the output from post-MP2 coupled-cluster module of Q-CHEM

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

$0 - 7$ higher values can lead to deforestation...

RECOMMENDATION:

Increase if you need more output and don't like trees

CC_PW

Activates calculation of partial Auger decay widths via decomposition of the imaginary part of the coupled-cluster energy of a complex-variable CCSD calculation on a core-ionized state. Currently, this is implemented for states resulting from ionization of a β core electron of a closed-shell system.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not invoke energy decomposition into partial Auger decay widths

1 invoke energy decomposition into partial Auger decay widths

RECOMMENDATION:

Use to compute partial widths for a complex-variable calculation on a core-vacant state. An appropriate complex-scaled basis set has to be chosen in order to capture Auger decay and the optimal scaling angle needs to be determined.[28,29]

CC_QCCD_THETA_SWITCH

QCCD calculations switch from OD to QCCD when the rotation gradient is below this threshold $[10^{-n}]$

TYPE:

INTEGER

DEFAULT:

2 10^{-2} switchover

OPTIONS:

n 10^{-n} switchover

RECOMMENDATION:

None

CC_REF_PROP_TE

Request for calculation of non-relaxed two-particle CCSD properties. The two-particle properties currently include $\langle \hat{S}^2 \rangle$. The one-particle properties also will be calculated, since the additional cost of the one-particle properties calculation is small compared to the cost of $\langle \hat{S}^2 \rangle$. The variable CC_REF_PROP must be also set to TRUE.

TYPE:

LOGICAL

DEFAULT:

FALSE (no two-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

The two-particle properties are computationally expensive, since they require calculation and use of the two-particle density matrix (the cost is approximately the same as the cost of an analytic gradient calculation). Do not request the two-particle properties unless you really need them.

CC_REF_PROP

Whether or not the non-relaxed (expectation value) or full response (including orbital relaxation terms) one-particle CCSD properties will be calculated. The properties currently include permanent dipole moment, the second moments ($\langle x^2 \rangle$, $\langle y^2 \rangle$, and $\langle z^2 \rangle$) of the electron density along with $\langle r^2 \rangle = \langle x^2 \rangle + \langle y^2 \rangle + \langle z^2 \rangle$. This option is incompatible with JOBTYPE = FORCE, OPT, or FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations need to be solved (λ-CCSD equations) for properties with the cost approximately the same as CCSD equations. Use the default if you do not need properties. The cost of the properties calculation itself is low. The CCSD one-particle density can be analyzed with NBO package by specifying NBO = TRUE, CC_REF_PROP = TRUE, and JOBTYPE = FORCE.

CC_RESET_THETA

The reference MO coefficient matrix is reset every n iterations to help overcome problems associated with the theta metric as theta becomes large.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

n n iterations between resetting orbital rotations to zero.

RECOMMENDATION:

None

CC_RESTART_NO_SCF

Should an optimized orbital coupled cluster calculation begin with optimized orbitals from a previous calculation? When TRUE, molecular orbitals are initially orthogonalized, and CC_PRECONV_T2Z and CC_CANONIZE are set to TRUE while other guess options are set to FALSE

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

CC_RESTART

Allows an optimized orbital coupled cluster calculation to begin with an initial guess for the orbital transformation matrix U other than the unit vector. The scratch file from a previous run must be available for the U matrix to be read successfully.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use unit initial guess.

TRUE Activates CC_PRECONV_T2Z, CC_CANONIZE, and turns off CC_MP2NO_GUESS

RECOMMENDATION:

Useful for restarting a job that did not converge, if files were saved.

CC_RESTR_AMPL

Controls the restriction on amplitudes is there are restricted orbitals

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 All amplitudes are in the full space

1 Amplitudes are restricted, if there are restricted orbitals

RECOMMENDATION:

None

CC_RESTR_TRIPLES

Controls which space the triples correction is computed in

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Triples are computed in the full space

1 Triples are restricted to the active space

RECOMMENDATION:

None

CC_REST_AMPL

Forces the integrals, T, and R amplitudes to be determined in the full space even though the CC_REST_OCC and CC_REST_VIR keywords are used.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do apply restrictions

TRUE Do not apply restrictions

RECOMMENDATION:

None

CC_REST_OCC

Sets the number of restricted occupied orbitals including active core occupied orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Restrict n energetically lowest occupied orbitals to correspond to the active core space.

RECOMMENDATION:

Example: cytosine with the molecular formula $C_4H_5N_3O$ includes one oxygen atom. To calculate O 1s core-excited states, n has to be set to 1, because the 1s orbital of oxygen is the energetically lowest. To obtain the N 1s core excitations, the integer n has to be set to 4, because the 1s orbital of the oxygen atom is included as well, since it is energetically below the three 1s orbitals of the nitrogen atoms. Accordingly, to simulate the C 1s spectrum of cytosine, n must be set to 8.

CC_REST_TRIPLES

Restricts R_3 amplitudes to the active space, *i.e.*, one electron should be removed from the active occupied orbital and one electron should be added to the active virtual orbital.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Applies the restrictions

RECOMMENDATION:

None

CC_REST_VIR

Sets the number of restricted virtual orbitals including frozen virtual orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Restrict n virtual orbitals.

RECOMMENDATION:

None

CC_SCALE_AMP

If not 0, scales down the step for updating coupled-cluster amplitudes in cases of problematic convergence.

TYPE:

INTEGER

DEFAULT:

0 no scaling

OPTIONS:

$abcd$ Integer code is mapped to $abcd \times 10^{-2}$, *e.g.*, 90 corresponds to 0.9

RECOMMENDATION:

Use 0.9 or 0.8 for non convergent coupled-cluster calculations.

CC_SINGLE_PREC

Precision selection for CCSD calculation. Available in CCMAN2 only.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

2 single-precision calculation followed by double-precision clean-up iterations

RECOMMENDATION:

Do not set too tight convergence thresholds when using single precision

CC_SP_DM

Precision selection for CCSD and EOM-CCSD intermediates, density matrices, gradients, and $\langle \hat{S}^2 \rangle$.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

RECOMMENDATION:

NONE

CC_SP_E_CONV

Energy convergence criterion in single precision in CCSD calculations.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Set 6 to be consistent with the default threshold in double precision in a pure single-precision calculation. When used with clean-up version, it should be smaller than double-precision threshold not to introduce extra iterations.

CC_SP_T_CONV

Amplitude convergence threshold in single precision in CCSD calculations.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Set 4 to be consistent with the default threshold in double precision in a pure single-precision run. When used with clean-up version, it should be smaller than double-precision threshold not to introduce extra iterations.

CC_STATE_TO_OPT

Specifies which state to optimize.

TYPE:

INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[i,j] optimize the jth state of the ith irrep.

RECOMMENDATION:

None

CC_SYMMETRY

Activates point-group symmetry in the ADC calculation.

TYPE:

LOGICAL

DEFAULT:

TRUE If the system possesses any point-group symmetry.

OPTIONS:

TRUE Employ point-group symmetry

FALSE Do not use point-group symmetry

RECOMMENDATION:

None

CC_THETA_CONV

Convergence criterion on the RMS difference between successive sets of orbital rotation angles [10^{-n}].

TYPE:

INTEGER

DEFAULT:

5 Energies

6 Gradients

OPTIONS:

n 10^{-n} convergence criterion.

RECOMMENDATION:

Use default

CC_THETA_GRAD_CONV

Convergence desired on the RMS gradient of the energy with respect to orbital rotation angles $[10^{-n}]$.

TYPE:

INTEGER

DEFAULT:

7 Energies

8 Gradients

OPTIONS:

n 10^{-n} convergence criterion.

RECOMMENDATION:

Use default

CC_THETA_GRAD_THRESH

RMS orbital gradient threshold $[10^{-n}]$ above which "mixed iterations" are performed in active space calculations if CC_ITERATE_OV is TRUE.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n 10^{-n} threshold.

RECOMMENDATION:

Can be made smaller if convergence difficulties are encountered.

CC_THETA_STEPSIZE

Scale factor for the orbital rotation step size. The optimal rotation steps should be approximately equal to the gradient vector.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 1.0

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-de}$
If the initial step is smaller than 0.5, the program will increase step
when gradients are smaller than the value of THETA_GRAD_THRESH,
up to a limit of 0.5.

RECOMMENDATION:

Try a smaller value in cases of poor convergence and very large orbital gradients. For example, a value of 01001 translates to 0.1

CC_TRANS_PROP

Whether or not the transition dipole moment (in atomic units) and oscillator strength and rotatory strength (in atomic units) for the EOM-CCSD target states will be calculated. By default, the transition dipole moment, angular momentum matrix elements, and rotatory strengths are calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition dipole moment, angular momentum matrix elements, and rotatory strengths between a set of EOM-CCSD states and another EOM-CCSD state, the CC_STATE_TO_OPT must be specified for this state.

TYPE:

INTEGER

DEFAULT:

0 (no transition properties will be calculated)

OPTIONS:

1 (calculate transition properties between all computed EOM state and the reference state)

2 (calculate transition properties between all pairs of EOM states)

RECOMMENDATION:

Additional equations (for the left EOM-CCSD eigenvectors plus lambda CCSD equations in case of transition properties between the CCSD reference and EOM-CCSD target states are requested) need to be solved for transition properties, approximately doubling the computational cost. The cost of the transition properties calculation itself is low.

CC_T_CONV

Convergence criterion on the RMS difference between successive sets of coupled-cluster doubles amplitudes $[10^{-n}]$

TYPE:

INTEGER

DEFAULT:

8 energies

10 gradients

OPTIONS:

n 10^{-n} convergence criterion.

RECOMMENDATION:

Use default

CC_Z_CONV

Convergence criterion on the RMS difference between successive doubles Z-vector amplitudes $[10^{-n}]$.

TYPE:

INTEGER

DEFAULT:

8 Energies

10 Gradients

OPTIONS:

n 10^{-n} convergence criterion.

RECOMMENDATION:

Use Default

CDFTCI_PRINT

Controls level of output from CDFT-CI procedure to Q-CHEM output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Only print energies and coefficients of CDFT-CI final states

1 Level 0 plus CDFT-CI overlap, Hamiltonian, and population matrices

2 Level 1 plus eigenvectors and eigenvalues of the CDFT-CI population matrix

3 Level 2 plus promolecule orbital coefficients and energies

RECOMMENDATION:

Level 3 is primarily for program debugging; levels 1 and 2 may be useful for analyzing the coupling elements

CDFTCI_RESTART

To be used in conjunction with CDFTCI_STOP, this variable causes CDFT-CI to read already-converged states from disk and begin SCF convergence on later states. Note that the same *$cdft* section must be used for the stopped calculation and the restarted calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Start calculations on state $n + 1$

RECOMMENDATION:

Use this setting in conjunction with CDFTCI_STOP.

CDFTCI_SKIP_PROMOLECULES

Skips promolecule calculations and allows fractional charge and spin constraints to be specified directly.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard CDFT-CI calculation is performed.

TRUE Use the given charge/spin constraints directly, with no promolecule calculations.

RECOMMENDATION:

Setting to TRUE can be useful for scanning over constraint values.

CDFTCI_STOP

The CDFT-CI procedure involves performing independent SCF calculations on distinct constrained states. It sometimes occurs that the same convergence parameters are not successful for all of the states of interest, so that a CDFT-CI calculation might converge one of these diabatic states but not the next. This variable allows a user to stop a CDFT-CI calculation after a certain number of states have been converged, with the ability to restart later on the next state, with different convergence options.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Stop after converging state n (the first state is state 1)

0 Do not stop early

RECOMMENDATION:

Use this setting if some diabatic states converge but others do not.

CDFTCI_SVD_THRESH

By default, a symmetric orthogonalization is performed on the CDFT-CI matrix before diagonalization. If the CDFT-CI overlap matrix is nearly singular (*i.e.*, some of the diabatic states are nearly degenerate), then this orthogonalization can lead to numerical instability. When computing $\mathbf{S}^{-1/2}$, eigenvalues smaller than $10^{-\mathrm{CDFTCI\_SVD\_THRESH}}$ are discarded.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

Can be decreased if numerical instabilities are encountered in the final diagonalization.

CDFTCI

Initiates a constrained DFT-configuration interaction calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a CDFT-CI Calculation

FALSE No CDFT-CI

RECOMMENDATION:

Set to TRUE if a CDFT-CI calculation is desired.

CDFT_BECKE_POP

Whether the calculation should print the Becke atomic charges at convergence

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Print populations

FALSE Do not print them

RECOMMENDATION:

Use the default. Note that the Mulliken populations printed at the end of an SCF run will not typically add up to the prescribed constraint value. Only the Becke populations are guaranteed to satisfy the user-specified constraints.

CDFT_LAMBDA_MODE

Allows CDFT potentials to be specified directly, instead of being determined as Lagrange multipliers.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Standard CDFT calculations are used.

TRUE Instead of specifying target charge and spin constraints, use the values from the input deck as the value of the Becke weight potential

RECOMMENDATION:

Should usually be set to FALSE. Setting to TRUE can be useful to scan over different strengths of charge or spin localization, as convergence properties are improved compared to regular CDFT(-CI) calculations.

CDFT_MAXITER

Maximum number of iterations for converging the constraint.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

N A maximum of N microiterations will be attempted.

RECOMMENDATION:

Default value is expected to be sufficient in most situations.

CDFT_POP

Sets the charge partitioning scheme for cDFT or cDFT-CI jobs.

TYPE:

STRING

DEFAULT:

BECKE

OPTIONS:

BECKE Linear combination of atomic Becke functions

FBH Fragment-based Hirshfeld partition

RECOMMENDATION:

None

CDFT_POSTDIIS

Controls whether the constraint is enforced after DIIS extrapolation.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Enforce constraint after DIIS

FALSE Do not enforce constraint after DIIS

RECOMMENDATION:

Use the default unless convergence problems arise, in which case it may be beneficial to experiment with setting CDFT_POSTDIIS to FALSE. With this option set to TRUE, energies should be variational after the first iteration.

CDFT_PREDIIS

Controls whether the constraint is enforced before DIIS extrapolation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Enforce constraint before DIIS

FALSE Do not enforce constraint before DIIS

RECOMMENDATION:

Use the default unless convergence problems arise, in which case it may be beneficial to experiment with setting CDFT_PREDIIS to TRUE. Note that it is possible to enforce the constraint both before and after DIIS by setting both CDFT_PREDIIS and CDFT_POSTDIIS to TRUE.

CDFT_PRINT

Whether detailed information about CDFT iterations should be printed in the output file.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print detailed information.

FALSE Do not print detailed information.

RECOMMENDATION:

Use the default and invoke additional printing for troubleshooting.

CDFT_THRESH

Threshold that determines how tightly the constraint must be satisfied.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

N Constraint is satisfied to within 10^{-N}.

RECOMMENDATION:

Default value is set to match SCF_CONVERGENCE. Use the default unless problems occur.

CDFT

Initiates a constrained DFT calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a Constrained DFT Calculation

FALSE No Density Constraint

RECOMMENDATION:

Set to TRUE if a Constrained DFT calculation is desired.

CD_ALGORITHM

Determines the algorithm for MP2 integral transformations.

TYPE:

STRING

DEFAULT:

Program determined.

OPTIONS:

DIRECT Uses fully direct algorithm (energies only).

SEMI_DIRECT Uses disk-based semi-direct algorithm.

LOCAL_OCCUPIED Alternative energy algorithm (see 6.4.1).

RECOMMENDATION:

Semi-direct is usually most efficient, and will normally be chosen by default.

CFMM_ORDER

Controls the order of the multipole expansions in CFMM calculation.

TYPE:

INTEGER

DEFAULT:

15 For single point SCF accuracy

25 For tighter convergence (optimizations)

OPTIONS:

n Use multipole expansions of order n

RECOMMENDATION:

Use the default.

CHARGE_CHARGE_REPULSION

The repulsive Coulomb interaction parameter for YinYang atoms.

TYPE:

INTEGER

DEFAULT:

550

OPTIONS:

n Use Q = $n \times 10^{-3}$

RECOMMENDATION:

The repulsive Coulomb potential maintains bond lengths involving YinYang atoms with the potential $V(r) = Q/r$. The default is parameterized for carbon atoms.

CHELPG_DX

Sets the rectangular grid spacing for the traditional Cartesian ChElPG grid or the spacing between concentric Lebedev shells (when the variables CHELPG_HA and CHELPG_H are specified as well).

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

N Corresponding to a grid space of $N/20$, in Å.

RECOMMENDATION:

Use the default, which corresponds to the "dense grid" of Breneman and Wiberg,[6], unless the cost is prohibitive, in which case a larger value can be selected. Note that this default value is set with the Cartesian grid in mind and not the Lebedev grid. In the Lebedev case, a larger value can typically be used.

CHELPG_HA

Sets the Lebedev grid to use for heavy (non-hydrogen) atoms.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Use N Lebedev points per atom.

RECOMMENDATION:

N must be one of the defined values for a Lebedev grid; see Table 5.2.

CHELPG_HEAD

Sets the "head space"[6] (radial extent) of the ChElPG grid.

TYPE:

INTEGER

DEFAULT:

28

OPTIONS:

N Corresponding to a head space of $N/10$, in Å.

RECOMMENDATION:

Use the default, which is the value recommended by Breneman and Wiberg.[6]

CHELPG_H

Sets the Lebedev grid to use for hydrogen atoms.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Use N Lebedev points per atom.

RECOMMENDATION:

N must be one of the defined values for a Lebedev grid; see Table 5.2. Furthermore, CHELPG_H must always be less than or equal to CHELPG_HA. If it is greater, it will automatically be set to the value of CHELPG_HA.

CHELPG

Controls the calculation of CHELPG charges.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate ChElPG charges.

TRUE Compute ChElPG charges.

RECOMMENDATION:

Set to TRUE if desired. For large molecules, there is some overhead associated with computing ChElPG charges, especially if the number of grid points is large. Note that POP_MULLIKEN must also be set to TRUE (which is the default value) in order to get ChElPG charges.

CHILD_MP_ORDERS

The multipole orders included in the prepared FERFs. The last digit specifies how many multipoles to compute, and the digits in the front specify the multipole orders: 2: dipole (D); 3: quadrupole (Q); 4: octopole (O). Multipole order 1 is reserved for monopole FERFs which can be used to separate the effect of orbital contraction.[27]

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

21 D

232 DQ

2343 DQO

RECOMMENDATION:

Use 232 (DQ) when FERF is needed.

CHILD_MP

Compute FERFs for fragments and use them as the basis for SCF-MI calculations.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute FERFs (use the full AO span of each fragment).

TRUE Compute fragment FERFs.

RECOMMENDATION:

Use FERFs to compute polarization energy when large basis sets are used. In an “EDA2" calculation, this *$rem* variable is set based on the given option automatically.

CHOLESKY_TOL

Tolerance of Cholesky decomposition of two-electron integrals

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Corresponds to a tolerance of 10^{-n}

RECOMMENDATION:

2 - qualitative calculations, 3 - appropriate for most cases, 4 - quantitative (error in total energy typically less than 1 μhartree)

CISTR_PRINT

Controls level of output.

TYPE:

LOGICAL

DEFAULT:

FALSE Minimal output.

OPTIONS:

TRUE Increase output level.

RECOMMENDATION:

None

CIS_AMPL_ANAL

Perform additional analysis of CIS and TDDFT excitation amplitudes, including generation of natural transition orbitals, excited-state multipole moments, and Mulliken analysis of the excited state densities and particle/hole density matrices.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform additional amplitude analysis.

FALSE Do not perform additional analysis.

RECOMMENDATION:

None

CIS_AMPL_PRINT

Sets the threshold for printing CIS and TDDFT excitation amplitudes.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

n Print if $|x_{ia}|$ or $|y_{ia}|$ is larger than $0.01 \times n$.

RECOMMENDATION:

Use the default unless you want to see more amplitudes.

CIS_CONVERGENCE

CIS is considered converged when error is less than $10^{-\text{CIS\_CONVERGENCE}}$

TYPE:

INTEGER

DEFAULT:

6 CIS convergence threshold 10^{-6}

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

Also controls convergence of the CPSCF equations.

CIS_CT_METRICS

Controls whether to compute the charge-separation metrics Λ, Δr, $\Delta\sigma$, and Γ (Section 7.3.2).

TYPE:

INTEGER

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute these metrics.

TRUE Compute all four metrics (for each excited state), in both the canonical and NTO representations.

RECOMMENDATION:

Request if desired. (There is some overhead associated with computing Λ, but it should be small.)

CIS_DER_NUMSTATE

Determines among how many states we calculate nonadiabatic couplings. These states must be specified in the *$derivative_coupling* section.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not calculate nonadiabatic couplings.

n Calculate $n(n-1)/2$ pairs of nonadiabatic couplings.

RECOMMENDATION:

None.

CIS_DIABATH_DECOMPOSE

Decide whether or not to decompose the diabatic coupling into Coulomb, exchange, and one-electron terms.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not decompose the diabatic coupling.

OPTIONS:

TRUE

RECOMMENDATION:

These decompositions are most meaningful for electronic excitation transfer processes. Currently, available only for CIS, not for TDDFT diabatic states.

CIS_DYNAMIC_MEM

Controls whether to use static or dynamic memory in CIS and TDDFT calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Partly use static memory

TRUE Fully use dynamic memory

RECOMMENDATION:

The default control requires static memory (MEM_STATIC) sufficient to hold an array whose size grows by $2 \times OV \times N_{\text{roots}}$ at each CIS iteration, where N_{roots} is the number of unconverged roots ($\leq$ CIS_N_ROOTS). For a large calculation, one has to specify a large value for MEM_STATIC, which is not recommended (see Chapter 2). Therefore, it is recommended to use dynamic memory for large calculations.

CIS_GUESS_DISK_TYPE

Determines the type of guesses to be read from disk

TYPE:

INTEGER

DEFAULT:

Nil

OPTIONS:

0 Read triplets only

1 Read triplets and singlets

2 Read singlets only

RECOMMENDATION:

Must be specified if CIS_GUESS_DISK is TRUE.

CIS_GUESS_DISK

Read the CIS guess from disk (previous calculation).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Create a new guess.

TRUE Read the guess from disk.

RECOMMENDATION:

Requires a guess from previous calculation.

CIS_GUESS_TYPE

Controls how to generate the initial guess excitation vectors in CIS/TDA/RPA calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Generate N (no. of roots requested) occupied $\rightarrow$ virtual single orbital transitions according to their orbital energy difference order (from low to high). This is the common scenario.

1 Generate $N-1$ occupied $\rightarrow$ virtual single orbital transitions according to their orbital energy difference order (from low to high), and generate another guess excitation vector consist of all the remaining single orbital transitions in the occupied $\rightarrow$ virtual transition space with equal weights.

2 Generate N occupied/virtual single orbital transitions according to their orbital energy difference order (from low to high), and generate one more guess excitation vector consist of all the remaining single orbital transitions in the occupied $\rightarrow$ virtual transition space with equal weights.

RECOMMENDATION:

The default setting should work for most of the cases. However, when the number of roots is small, in some CIS/TDA/RPA calculations, low energy excited states could be missing. The options CIS_GUESS_TYPE = 1 or 2 may remedy this root missing issue by sampling more vectors in the transition space. Setting CIS_GUESS_TYPE = 1 or 2 may take more cycles to converge in the Davidson iteration, but the results are expected to be more reliable. Currently, CIS_GUESS_TYPE = 1 or 2 are not supported in SF-XCIS calculations. Setting TRNSS = TRUE also disables the setting of CIS_GUESS_TYPE.

CIS_MOMENTS

Controls calculation of excited-state (CIS or TDDFT) multipole moments.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate excited-state moments.

TRUE Calculate moments for each excited state.

RECOMMENDATION:

Set to TRUE if excited-state moments are desired. (This is a trivial additional calculation.) The MULTIPOLE_ORDER controls how many multipole moments are printed. This option is not available for spin-flip methods.

CIS_MULLIKEN

Controls Mulliken and Löwdin population analyses for excited-state particle and hole density matrices.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform particle/hole population analysis.

TRUE Perform both Mulliken and Löwdin analysis of the particle and hole density matrices for each excited state.

RECOMMENDATION:

Set to TRUE if desired. This represents a trivial additional calculation.

CIS_N_ROOTS

Sets the number of excited state roots to find

TYPE:

INTEGER

DEFAULT:

0 Do not look for any excited states

OPTIONS:

n $n > 0$ Looks for n excited states

RECOMMENDATION:

None

CIS_RELAXED_DENSITY

Use the relaxed CIS density for attachment/detachment density analysis as well as for for the general excited-state analysis of Section 10.2.12.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use the relaxed CIS density in analysis.

TRUE Use the relaxed CIS density in analysis.

RECOMMENDATION:

None

CIS_S2_THRESH

Determines whether a state is a singlet or triplet in unrestricted calculations.

TYPE:

INTEGER

DEFAULT:

120

OPTIONS:

n Sets the $\langle \hat{S}^2 \rangle$ threshold to $n/100$

RECOMMENDATION:

For the default case, states with $\langle \hat{S}^2 \rangle > 1.2$ are treated as triplet states and other states are treated as singlets.

CIS_SINGLETS

Solve for singlet excited states (ignored for spin unrestricted systems)

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for singlet states

FALSE Do not solve for singlet states.

RECOMMENDATION:

None

CIS_STATE_DERIV

Sets the excited state index for analytical gradient calculation for geometry optimizations and vibrational analysis with SOS-CIS(D_0)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Select the nth state.

RECOMMENDATION:

Check to see that the states do no change order during an optimization. For closed-shell systems, either CIS_SINGLETS or CIS_TRIPLETS must be set to false.

CIS_TRIPLETS

Solve for triplet excited states (ignored for spin unrestricted systems)

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Solve for triplet states

FALSE Do not solve for triplet states.

RECOMMENDATION:

None

CM5

Controls running of CM5 population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate CM5 populations.

FALSE Do not calculate CM5 populations.

RECOMMENDATION:

None

COMBINE_K

Controls separate or combined builds for short-range and long-range K

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Build short-range and long-range K separately (twice as expensive as a global hybrid)

TRUE (or 1) Build short-range and long-range K together ($\approx$ as expensive as a global hybrid)

RECOMMENDATION:

Most pre-defined range-separated hybrid functionals in Q-CHEM use this feature by default. However, if a user-specified RSH is desired, it is necessary to manually turn this feature on.

COMPLEX_BASIS

Defines the complex basis.

TYPE:

STRING

DEFAULT:

No default complex basis set

OPTIONS:

Symbol	Use a standard basis set
ZBASIS_GENERAL, ZBASIS_GEN	User-defined. As for BASIS
ZBASIS_MIXED	User-defined mixed basis

RECOMMENDATION:

Consult Ref. 42 and the Basis Set Exchange.

COMPLEX_CCMAN

Requests complex-scaled or CAP-augmented CC/EOM calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Engage complex CC/EOM code.

RECOMMENDATION:

Not available in CCMAN. Need to specify CAP strength or complex-scaling parameter in *$complex_ccman* section.

COMPLEX_MIX

Mix a certain percentage of the real part of the HOMO to the imaginary part of the LUMO.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0–100 The mix angle = π·COMPLEX_MIX/100.

RECOMMENDATION:

It may help find the stable complex solution (similar idea as SCF_GUESS_MIX).

COMPLEX_THETA

Sets the value of θ in degrees for a calculation with complex basis functions.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n $\theta = n/10$ (degrees)

RECOMMENDATION:

Consult Ref. 42. Usually calculations at several different values of θ (a "θ-trajectory") should be performed.

COMPLEX

Run an SCF calculation with complex MOs using GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use complex orbitals.

FALSE Use real orbitals.

RECOMMENDATION:

Set to TRUE if desired.

CORE_CHARACTER

Selects how the core orbitals are determined in the frozen-core approximation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use energy-based definition.

1-4 Use Mulliken-based definition (see Table 6.2 for details).

RECOMMENDATION:

Use the default, unless performing calculations on molecules with heavy elements.

CORE_IONIZE

Indicates how orbitals are specified for reduced excitation spaces.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 all valence orbitals are listed in *$solute* section

2 only hole(s) are specified all other occupations same as ground state

RECOMMENDATION:

For MOM + TDDFT this specifies the input form of the *$solute* section. If set to 1 all occupied orbitals must be specified, 2 only the empty orbitals to ignore must be specified.

CORRELATION

Specifies the correlation level of theory handled by CCMAN/CCMAN2.

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

CCMP2	Regular MP2 handled by CCMAN/CCMAN2
MP3	CCMAN and CCMAN2
MP4SDQ	CCMAN
MP4	CCMAN
CCD	CCMAN and CCMAN2
CCD(2)	CCMAN
CCSD	CCMAN and CCMAN2
CCSDT	CCMAN2
CC2	CCMAN2
CCSD(T)	CCMAN and CCMAN2
CCSD(2)	CCMAN
CCSD(fT)	CCMAN and CCMAN2
CCSD(dT)	CCMAN
CCSDT	CCMAN2
CCVB-SD	CCMAN2
QCISD	CCMAN and CCMAN2
QCISD(T)	CCMAN and CCMAN2
OD	CCMAN
OD(T)	CCMAN
OD(2)	CCMAN
VOD	CCMAN
VOD(2)	CCMAN
QCCD	CCMAN
QCCD(T)	CCMAN
QCCD(2)	CCMAN
VQCCD	CCMAN
VQCCD(T)	CCMAN
VQCCD(2)	CCMAN

RECOMMENDATION:

Consult the literature for guidance.

CPSCF_NSEG

Controls the number of segments used to calculate the CPSCF equations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Determine the number of segments based on the memory request and MEM_TOTAL

n User-defined. Use n segments when solving the CPSCF equations.

RECOMMENDATION:

Use the default.

CS_SCF_FINAL_PRINT

Controls level of output from CAP-SCF procedure.

TYPE:

INTEGER

DEFAULT:

0 No extra print out.

OPTIONS:

1 Print direct breakdown of CAP-SCF energy.

2 Print breakdown of CAP-SCF energy based on the complex coefficient matrix. Also required if the options below are requested.

3 Level 2 plus diagonal elements of complex orbital energy matrix, **F**. Triggered by Level 2.

4 Level 2 plus diagonal elements of complex kinetic energy matrix, **T**. Triggered by Level 2

5 Level 2 plus diagonal elements of complex electron-nuclear Coulomb potential energy matrix, **V**. Triggered by Level 2.

6 Level 2 plus diagonal elements of CAP matrix, **W**. Triggered by Level 2.

7 Level 2 plus diagonal elements of total complex one-electron energy matrix, $\mathbf{T} + \mathbf{V} + \mathbf{W}$. Triggered by Level 2.

8 Level 2 plus diagonal elements of total complex electronic energy matrix, $\mathbf{T} + \mathbf{V} + \mathbf{W} + \mathbf{F}$. Triggered by Level 2.

9 Level 2 to 8. Triggered by Level 2.

RECOMMENDATION:

Level 1 is usually enough. Values for this *$rem* variable are transformed first into a set of distinct values; thus, for example, "1111" is equivalent to "1" and "28224" is equivalent to "248". To request Levels 3–9, please remember to request Level 2 as well.

CS_STRICT

Determines Mulliken charges, multipole moments and complex orbital energies for CAP-HF calculations by reading, when applicable, complex density matrix or complex molecular orbital coefficient file

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE determine Mulliken charges, multipole moments and complex orbital energies for CAP-HF calculations by reading – when applicable – the complex density matrix or complex molecular orbital coefficient file.

FALSE Don't read the complex density matrix or complex molecular orbital coefficient file when determining Mulliken charges, multipole moments and orbital energies for CAP-HF calculations.

RECOMMENDATION:

Set to 'TRUE' for CAP-HF calculations.

CUBEFILE_STATE

Determines which excited state is used to generate cube files

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n Generate cube files for the nth excited state

RECOMMENDATION:

None

CUDA_RI-MP2

Enables GPU implementation of RI-MP2

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE GPU-enabled MGEMM off

TRUE GPU-enabled MGEMM on

RECOMMENDATION:

Necessary to set to 1 in order to run GPU-enabled RI-MP2

CUTOCC

Specifies occupied orbital cutoff.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

0-200 Use a cutoff of CUTOCC/100

RECOMMENDATION:

None

CUTVIR

Specifies virtual orbital cutoff.

TYPE:

INTEGER

DEFAULT:

0 No truncation

OPTIONS:

0-100 CUTOFF = CUTVIR/100

RECOMMENDATION:

None

CVS_EE_SINGLETS

Sets the number of singlet core-excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_EE_TRIPLETS

Sets the number of triplet core-excited state roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i excited states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

CVS_EOM_PRECONV_SINGLES

When not zero, singly excited vectors are converged prior to a full excited states calculation (CVS states only). Sets the maximum number of iterations for pre-converging procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not pre-converge

1 pre-converge singles

RECOMMENDATION:

Sometimes helps with problematic convergence.

CVS_EOM_SHIFT

Specifies energy shift in CVS-EOM calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n corresponds to $n \times 10^{-3}$ hartree shift (*i.e.*, 11000 = 11 hartree); solve for eigenstates around this value.

RECOMMENDATION:

Improves the stability of the calculations.

CVS_SF_STATES

Sets the number of core-level spin-flip target states roots to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any excited states.

OPTIONS:

$[i, j, k \ldots]$ Find i SF states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DALTON_MAXITER

Maximum number of iteration allowed for the DALTON solver for response equations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n User-defined number of iterations.

RECOMMENDATION:

Default is usually sufficient

DALTON_MAXSPACE

Specifies maximum number of vectors in the subspace for the DALTON solver for response equations.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

n Up to n vectors per root before the subspace is reset.

RECOMMENDATION:

Larger values increase disk storage but accelerate and stabilize convergence.

DALTON_PRECOND_START

Specifies the iteration number in the DALTON procedure for response equations from which the preconditioner is applied to the residuals.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n User-defined iteration number.

RECOMMENDATION:

Use default.

DALTON_XCONV

Convergence criterion for the residuals (square norm) of the DALTON solver for response equations.

TYPE:

INTEGER

DEFAULT:

6 Corresponding to 10^{-6}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion.

RECOMMENDATION:

Use the default in double precision. May reduce to 5 in single precision.

DAMPED_DALTON_SOLVER

Boolean for using the new Davidson-like solver (DALTON) for damped (CCSD polarizabilities and hyperpolarizabilities and EOM-CCSD 2PA and RIXS cross sections) response equations.

TYPE:

LOGICAL

DEFAULT:

TRUE (Use the new DALTON solver)

OPTIONS:

FALSE If the old DIIS solver is desired for the above properties.

RECOMMENDATION:

Use the new solver for faster convergence relative to DIIS.

DC_DFT

Controls whether to use DC-DFT.

TYPE:

Boolean

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do DC-DFT.

TRUE Iterate the density to self-consistency at the Hartree-Fock level and then perform evaluate $E_{\mathrm{DFT}}[\rho_{\mathrm{HF}}]$ using the functional specified with METHOD.

RECOMMENDATION:

Use if desired. Analytic gradients are available but are a serial bottleneck in the present implementation.

DEA_AA_STATES

Sets the number of $M_S = 1$ DEA roots (two α electrons) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA $M_S = 1$ transitions.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA $\alpha\alpha$ states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_AB_STATES

Sets the number of $M_S = 0$ DEA roots (one α and one β electron) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA $M_S = 0$ transitions.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA $\alpha\beta$ states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_BB_STATES

Sets the number of $M_S = -1$ DEA roots (two β electrons) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA $M_S = -1$ transitions.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA $\beta\beta$ states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_SINGLETS

Sets the number of singlet DEA roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any singlet DEA states.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA singlet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_STATES

Sets the number of DEA roots to find. For closed-shell reference, defaults into DEA_SINGLETS. For open-shell references, specifies all low-lying states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DEA_TRIPLETS

Sets the number of triplet DEA roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DEA triplet states.

OPTIONS:

$[i, j, k \ldots]$ Find i DEA triplet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DELTA_GRADIENT_SCALE

Scales the gradient of Δ by N/100, which can be useful for cases with troublesome convergence by reducing step size.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

N

RECOMMENDATION:

Use default. For problematic cases, N =50, 25, 10 or even $N = 1$ could be useful.

DEUTERATE

Requests that all hydrogen atoms be replaces with deuterium.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not replace hydrogens.

OPTIONS:

TRUE Replace hydrogens with deuterium.

RECOMMENDATION:

Replacing hydrogen atoms reduces the fastest vibrational frequencies by a factor of 1.4, which allow for a larger fictitious mass and time step in ELMD calculations. There is no reason to replace hydrogens in BOMD calculations.

DFPT_EXCHANGE

Specifies the secondary functional in a HFPC/DFPC calculation.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

None

RECOMMENDATION:

See reference for recommended basis set, functional, and grid pairings.

DFPT_XC_GRID

Specifies the secondary grid in a HFPC/DFPC calculation.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

None

RECOMMENDATION:

See reference for recommended basis set, functional, and grid pairings.

DFTCIS_PARAMS

Selects what variant of DFT/CIS

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do CIS

1 Do B3LYP/CIS

2 Do CAM-B3LYP/CIS

RECOMMENDATION:

Requires DFTCIS to be set to 1 or 2.

DFTCIS

Controls whether to do a DFT/CIS calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do a DFT/CIS calculation.

TRUE Do a DFT/CIS calculation.

RECOMMENDATION:

None

DFT_C

Controls whether the DFT-C empirical BSSE correction should be added.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply the DFT-C correction

TRUE (or 1) Apply the DFT-C correction

RECOMMENDATION:

NONE

DFT_D3_3BODY

Controls whether the three-body interaction in Grimme's DFT-D3 method should be applied (see Eq. (14) in Ref. 15).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply the three-body interaction term

TRUE Apply the three-body interaction term

RECOMMENDATION:

NONE

DFT_D3_A1

The nonlinear parameter α_1 in Eqs. (5.29), (5.30), (5.31), and (5.32). Used in DFT-D3(BJ), DFT-D3(CSO), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_1 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A2

The nonlinear parameter α_2 in Eqs. (5.29) and (5.32). Used in DFT-D3(BJ), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_2 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_POWER

The nonlinear parameter β_6 in Eq. (5.32). Used in DFT-D3(op). Must be greater than or equal to 6 to avoid divergence.

TYPE:

INTEGER

DEFAULT:

600000

OPTIONS:

n Corresponding to $\beta_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS6

The nonlinear parameter $s_{r,6}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,6} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS8

The nonlinear parameter $s_{r,8}$ in Eqs. (5.28) and Eq. (5.31). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,8} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_S6

The linear parameter s_6 in eq. (5.27). Used in all forms of DFT-D3.

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_S8

The linear parameter s_8 in Eq. (5.27). Used in DFT-D3(0), DFT-D3(BJ), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_8 = n/100000$.

RECOMMENDATION:

NONE

DFT_D4_A1

The nonlinear parameter α_1. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_1 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_A2

The nonlinear parameter α_2. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $\alpha_2 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_GA

Charge scaling

TYPE:

INTEGER

DEFAULT:

300000000

OPTIONS:

n Corresponding to $ga = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_GC

Charge scaling

TYPE:

INTEGER

DEFAULT:

200000000

OPTIONS:

n Corresponding to $gc = n/100000000$.

RECOMMENDATION:

Use default

DFT_D4_S10

The linear parameter s_{10}. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_{10} = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S6

The linear parameter s_6. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_6 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S8

The linear parameter s_8. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_8 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_S9

The linear parameter s_9. Used in DFT-D4.

TYPE:

INTEGER

DEFAULT:

Optimized number for the specified functional

OPTIONS:

n Corresponding to $s_9 = n/100000000$.

RECOMMENDATION:

NONE

DFT_D4_WF

Weighting factor for Gaussian weighting.

TYPE:

INTEGER

DEFAULT:

600000000

OPTIONS:

n Corresponding to $wf = n/100000000$.

RECOMMENDATION:

Use default

DFT_D_A

Controls the strength of dispersion corrections in the Chai–Head-Gordon DFT-D scheme, Eq. (5.26).

TYPE:

INTEGER

DEFAULT:

600

OPTIONS:

n Corresponding to $a = n/100$.

RECOMMENDATION:

Use the default.

DFT_D

Controls the empirical dispersion correction to be added to a DFT calculation.

TYPE:

LOGICAL

DEFAULT:

None

OPTIONS:

FALSE	(or 0) Do not apply the DFT-D2, DFT-CHG, or DFT-D3 scheme
EMPIRICAL_GRIMME	DFT-D2 dispersion correction from Grimme [14]
EMPIRICAL_CHG	DFT-CHG dispersion correction from Chai and Head-Gordon [11]
EMPIRICAL_GRIMME3	DFT-D3(0) dispersion correction from Grimme (deprecated as of Q-CHEM 5.0)
D3_ZERO	DFT-D3(0) dispersion correction from Grimme *et al.* [15]
D3S_ZERO	DFT-D3S(0) dispersion correction from Tkachenko and Head-Gordon [39]
D3_BJ	DFT-D3(BJ) dispersion correction from Grimme *et al.* [16]
D3S_BJ	DFT-D3S(BJ) dispersion correction from Tkachenko and Head-Gordon [39]
D3_CSO	DFT-D3(CSO) dispersion correction from Schröder *et al.* [34]
D3_ZEROM	DFT-D3M(0) dispersion correction from Smith *et al.* [35]
D3_BJM	DFT-D3M(BJ) dispersion correction from Smith *et al.* [35]
D3_OP	DFT-D3(op) dispersion correction from Witte *et al.* [43]
D3	Automatically select the "best" available D3 dispersion correction
D4	DFT-D4 dispersion correction from Caldeweyher *et al.* [8–10]

RECOMMENDATION:

Use D4 if the specified functional is avialable. Currently, only a subset of functionals in DFT-D4 is supported. It includes B3LYP, B97, B1LYP, PBE0, PW6B95, M06L, M06, WB97, WB97X, CAMB3LYP, PBE02, PBE0DH, MPW1K, MPWB1K, B1B95, B1PW91, B2GPPLYP, B2PLYP, B3P86, B3PW91, O3LYP, REVPBE, REVPBE0, REVTPSS, REVTPSSH, SCAN, TPSS0, TPSSH, X3LYP, TPSS, BP86, BLYP, BPBE, MPW1PW91, MPW1LYP, PBE, RPBE, and PW91.

DH

Controls the application of DH-DFT scheme.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0)	Do not apply the DH-DFT scheme
TRUE (or 1)	Apply DH-DFT scheme

RECOMMENDATION:

NONE

DIIS_ERR_RMS

Changes the DIIS convergence metric from the maximum to the RMS error.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE, FALSE

RECOMMENDATION:

Use the default, the maximum error provides a more reliable criterion.

DIIS_PRINT

Controls the output from DIIS SCF optimization.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Minimal print out.

1 Chosen method and DIIS coefficients and solutions.

2 Level 1 plus changes in multipole moments.

3 Level 2 plus Multipole moments.

4 Level 3 plus extrapolated Fock matrices.

RECOMMENDATION:

Use the default

DIIS_SEPARATE_ERRVEC

Control optimization of DIIS error vector in unrestricted calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE Use a combined α and β error vector.

OPTIONS:

FALSE Use a combined α and β error vector.

TRUE Use separate error vectors for the α and β spaces.

RECOMMENDATION:

When using DIIS in Q-CHEM a convenient optimization for unrestricted calculations is to sum the α and β error vectors into a single vector which is used for extrapolation. This is often extremely effective, but in some pathological systems with symmetry breaking, can lead to false solutions being detected, where the α and β components of the error vector cancel exactly giving a zero DIIS error. While an extremely uncommon occurrence, if it is suspected, set DIIS_SEPARATE_ERRVEC = TRUE to check.

DIIS_SUBSPACE_SIZE

Controls the size of the DIIS and/or RCA subspace during the SCF.

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

User-defined

RECOMMENDATION:

None

DIP_AA_STATES

Sets the number of $M_S = -1$ DIP roots (remove two α electrons) to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP $M_S = -1$ states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_AB_STATES

Sets the number of $M_S = 0$ DIP roots (remove one α and one β electron) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP $M_S = 0$ states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_BB_STATES

Sets the number of $M_S = +1$ DIP roots (remove two β electrons) to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP $M_S = +1$ states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_SINGLETS

Sets the number of singlet DIP roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any singlet DIP states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP singlet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_STATES

Sets the number of DIP roots to find. For closed-shell reference, defaults into DIP_SINGLETS.

For open-shell references, specifies all low-lying states.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIP_TRIPLETS

Sets the number of triplet DIP roots to find. Valid only for closed-shell references.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DIP triplet states.

OPTIONS:

$[i, j, k \ldots]$ Find i DIP triplet states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DIRECT_SCF

Controls direct SCF.

TYPE:

LOGICAL

DEFAULT:

Determined by program.

OPTIONS:

TRUE Forces direct SCF.

FALSE Do not use direct SCF.

RECOMMENDATION:

Use the default; direct SCF switches off in-core integrals.

DISP_FREE_C

Specify the employed “dispersion-free" correlation functional.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

Correlation functionals supported by Q-CHEM.

RECOMMENDATION:

Put the appropriate correlation functional paired with the chosen exchange functional (*e.g.* put PBE if DISP_FREE_X is revPBE); put NONE if DISP_FREE_X is set to an exchange-correlation functional.

DISP_FREE_X

Specify the employed “dispersion-free" exchange functional.

TYPE:

STRING

DEFAULT:

HF

OPTIONS:

Exchange functionals (*e.g.* revPBE) or exchange-correlation functionals (*e.g.* B3LYP) supported by Q-CHEM.

RECOMMENDATION:

HF is recommended for hybrid (primary) functionals (*e.g.*ωB97X-V) and revPBE for semi-local ones (*e.g.*B97M-V). Other reasonable options (*e.g.* B3LYP for B3LYP-D3) can also be applied.

DOMODSANO

Specifies whether to do modified Sano or the original one

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Does original Sano procedure (similar to GVBMAN).

1 Does an improved Sano procedure that’s more localized.

2 Does another variation of Sano.

RECOMMENDATION:

1 is always better

DORAMAN

Controls calculation of Raman intensities. Requires JOBTYPE to be set to FREQ

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate Raman intensities.

TRUE Do calculate Raman intensities.

RECOMMENDATION:

None

DO_ATOMIC_MULTIPOLES

Enables atomic multipole calculation

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate IAO atomic multipoles

TRUE Calculate IAO atomic multipoles

RECOMMENDATION:

None

DO_IBO

Enables IBO procedure

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate IBOs

TRUE Run the IBO procedure

RECOMMENDATION:

None

DSF_STATES

Sets the number of doubly spin-flipped target states roots to find.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any DSF states.

OPTIONS:

$[i, j, k \ldots]$ Find i doubly spin-flipped states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

DUAL_BASIS_ENERGY

Activates dual-basis SCF (HF or DFT) energy correction.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Analytic first derivative available for HF and DFT (see JOBTYPE)

Can be used in conjunction with MP2 or RI-MP2

See BASIS, BASIS2, BASISPROJTYPE

RECOMMENDATION:

Use dual-basis to capture large-basis effects at smaller basis cost. Particularly useful with RI-MP2, in which HF often dominates. Use only proper subsets for small-basis calculation.

DYSON_PW_COUPLING

Calculate overlap between Dyson orbital and plane wave.

TYPE:

LOGICAL

DEFAULT:

FALSE (do not compute coupling)

OPTIONS:

TRUE/FLSE

RECOMMENDATION:

Activate when needed. CC_DO_DYSON must be turned on.

D_CPSCF_PERTNUM

Specifies whether to do the perturbations one at a time, or all together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Perturbed densities to be calculated all together.

1 Perturbed densities to be calculated one at a time.

RECOMMENDATION:

None

D_SCF_CONV_1

Sets the convergence criterion for the level-1 iterations. This preconditions the density for the level-2 calculation, and does not include any two-electron integrals.

TYPE:

INTEGER

DEFAULT:

4 corresponding to a threshold of 10^{-4}.

OPTIONS:

$n < 10$ Sets convergence threshold to 10^{-n}.

RECOMMENDATION:

The criterion for level-1 convergence must be less than or equal to the level-2 criterion, otherwise the D-CPSCF will not converge.

D_SCF_CONV_2

Sets the convergence criterion for the level-2 iterations.

TYPE:

INTEGER

DEFAULT:

4 Corresponding to a threshold of 10^{-4}.

OPTIONS:

$n < 10$ Sets convergence threshold to 10^{-n}.

RECOMMENDATION:

None

D_SCF_DIIS

Specifies the number of matrices to use in the DIIS extrapolation in the D-CPSCF.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

n n = 0 specifies no DIIS extrapolation is to be used.

RECOMMENDATION:

Use the default.

D_SCF_MAX_1

Sets the maximum number of level-1 iterations.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n User defined.

RECOMMENDATION:

Use the default.

D_SCF_MAX_2

Sets the maximum number of level-2 iterations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User defined.

RECOMMENDATION:

Use the default.

EA_ALPHA

Sets the number of attached target states derived by attaching α electron ($M_S = \frac{1}{2}$, default in EOM-EA).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any EA states.

OPTIONS:

$[i, j, k \ldots]$ Find i EA states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EA_BETA

Sets the number of attached target states derived by attaching β electron ($M_S = \frac{1}{2}$, EA-SF).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any EA states.

OPTIONS:

$[i, j, k \ldots]$ Find i EA states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

EA_STATES

Controls the number of electron-attached states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not perform an EA-ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of electron-attached states in case of restricted calculations.

ECP

Defines the effective core potential and associated basis set to be used

TYPE:

STRING

DEFAULT:

No ECP

OPTIONS:

General, Gen User defined. (*$ecp* keyword required)

Symbol Use standard ECPs discussed above.

RECOMMENDATION:

ECPs are recommended for first row transition metals and heavier elements. Consult the reviews for more details.

EDA2

Switch on EDA2 and specify the option set number.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 Do not run through EDA2.

1 Frozen energy decomposition + nMDQ-uFERF polarization
(the standard EDA2 option)

2 Frozen energy decomposition + (AO-block-based) ALMO polarization
(old scheme with the addition of frozen decomposition)

3 Frozen energy decomposition + oDQ-FERF polarization
(NOT commonly used)

4 Frozen wave function relaxation + Frozen energy decomposition + nDQ-FERF polarization
(NOT commonly used)

5 Frozen energy decomposition + polMO polarization
(NOT commonly used).

10 No preset. Completely controlled by user's *$rem* input
(for developers only)

RECOMMENDATION:

Turn on EDA2 for Q-CHEM's ALMO-EDA jobs unless CTA with the old scheme is desired. Option 1 is recommended in general, especially when substantially large basis sets are employed. The original ALMO scheme (option 2) can be used when the employed basis set is of small or medium size (arguably no larger than augmented triple-ζ). The other options are rarely used for routine applications.

EDA_BSSE

Calculates the BSSE correction when performing the energy decomposition analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set to TRUE unless a very large basis set is used.

EDA_CLS_DISP

Compute the DISP contribution without performing the orthogonal decomposition, which will then be subtracted from the classical PAULI term.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Use the DISP term computed with orthogonal decomposition (if available).

TRUE Use the DISP term computed using undistorted monomer densities.

RECOMMENDATION:

Set it to TRUE when orthogonal decomposition is not performed.

EDA_CLS_ELEC

Perform the classical decomposition of the frozen term.

TYPE:

BOOLEAN

DEFAULT:

FALSE (automatically set to TRUE by EDA2 options 1–5)

OPTIONS:

FALSE Do not compute the classical ELEC and PAULI terms.

TRUE Perform the classical decomposition.

RECOMMENDATION:

TRUE

EDA_CONTRACTION_ANAL

Perform analysis separating orbital contraction from the rest of POL.

TYPE:

BOOLEAN

DEFAULT:

0

OPTIONS:

FALSE Do not perform contraction analysis.

TRUE Perform contraction analysis.

RECOMMENDATION:

No recommendation

EDA_COVP

Perform COVP analysis when evaluating the RS or ARS charge-transfer correction. COVP analysis is currently implemented only for systems of two fragments.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set to TRUE to perform COVP analysis in an EDA or SCF-MI(RS) job.

EDA_PRINT_COVP

Replace the final MOs with the CVOP orbitals in the end of the run.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set to TRUE to print COVP orbitals instead of conventional MOs.

EDA_UFERF

Using uncoupled-FERFs (uFERFs)[1] instead of FERFs

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Use uFERFs

FALSE Use FERFs

RECOMMENDATION:

Use the default uFERFs, use FALSE when FERFs are desired.

EE_SINGLETS

Controls the number of singlet excited states to calculate.

TYPE:

INTEGER/ARRAY

DEFAULT:

0 Do not perform an ADC calculation of singlet excited states

OPTIONS:

$n > 0$ Number of singlet states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in case of restricted calculations of singlet states. In unrestricted calculations it can also be used, if EE_STATES not set. Then, it has the same effect as setting EE_STATES.

EE_STATES

Controls the number of excited states to calculate.

TYPE:

INTEGER/ARRAY

DEFAULT:

0 Do not perform an ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in case of unrestricted or open-shell calculations. In restricted calculations it can also be used, if neither EE_SINGLETS nor EE_TRIPLETS is given. Then, it has the same effect as setting EE_SINGLETS.

EE_TRIPLETS

Controls the number of triplet excited states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not perform an ADC calculation of triplet excited states

OPTIONS:

$n > 0$ Number of triplet states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in case of restricted calculations of triplet states.

EFP_COORD_XYZ

Use coordinates of three atoms instead of Euler angles to specify position and orientation of the fragments

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

None

EFP_DIRECT_POLARIZATION_DRIVER

Use direct solver for EFP polarization

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

Direct polarization solver provides stable convergence of induced dipoles which may otherwise become problematic in case of closely lying or highly polar or charged fragments. The computational cost of direct polarization versus iterative polarization becomes higher for systems containing more than 10000 polarizable points.

EFP_DISP_DAMP

Controls fragment-fragment dispersion screening in EFP

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 switch off dispersion screening

1 use Tang-Toennies screening, with fixed parameter $b = 1.5$

2 use overlap-based damping

RECOMMENDATION:

None

EFP_DISP

Controls fragment-fragment dispersion in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on dispersion

FALSE switch off dispersion

RECOMMENDATION:

None

EFP_ELEC_DAMP

Controls fragment-fragment electrostatic screening in EFP

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 switch off electrostatic screening

1 use overlap-based damping correction

2 use exponential damping correction if SCREEN2 screening parameters are provided in the EFP potential

RECOMMENDATION:

Overlap-based damping is recommended

EFP_ELEC

Controls fragment-fragment electrostatics in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on electrostatics

FALSE switch off electrostatics

RECOMMENDATION:

None

EFP_ENABLE_LINKS

Enable fragment links in EFP region

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

None

EFP_EXREP

Controls fragment-fragment exchange repulsion in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on exchange repulsion

FALSE switch off exchange repulsion

RECOMMENDATION:

None

EFP_FRAGMENTS_ONLY

Specifies whether there is a QM part

TYPE:

LOGICAL

DEFAULT:

FALSE QM part is present

OPTIONS:

TRUE Only MM part is present: all fragments are treated by EFP

FALSE QM part is present: do QM/MM EFP calculation

RECOMMENDATION:

None

EFP_INPUT

Specifies the format of EFP input

TYPE:

LOGICAL

DEFAULT:

FALSE Dummy atom (*e.g.*, He) in *$molecule* section should be present

OPTIONS:

TRUE A format without dummy atom in *$molecule* section

FALSE A format with dummy atom in *$molecule* section

RECOMMENDATION:

None

EFP_ORDER

Controls QM-EFP pairwise fragment energy decomposition analysis

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

1 the first step of energy decomposition is performed

2 the second step of energy decomposition is performed

RECOMMENDATION:

The EFP_PAIRWISE keyword should be turned on to activate the energy decomposition analysis.

EFP_PAIRWISE

Controls QM-EFP pairwise fragment energy decomposition analysis

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 energy decomposition is turned off

1 energy decomposition is turned on

RECOMMENDATION:

None

EFP_POL_DAMP_TT_VALUE

Controls the value of the fragment–fragment polarization Tang–Toennies screening factor in EFP

TYPE:

INTEGER

DEFAULT:

600

OPTIONS:

n corresponds to n/1000.

RECOMMENDATION:

Use the default value of 0.6 for this screening factor, which was hard coded in the original EFP implementation.

EFP_POL_DAMP

Controls fragment-fragment polarization screening in EFP

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 switch off polarization screening

1 use Tang-Toennies screening

RECOMMENDATION:

None

EFP_POL

Controls fragment-fragment polarization in EFP

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on polarization

FALSE switch off polarization

RECOMMENDATION:

None

EFP_QM_DISP

Controls QM-EFP dispersion

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE switch on QM-EFP dispersion

FALSE switch off QM-EFP dispersion

RECOMMENDATION:

None

EFP_QM_ELEC_DAMP

Controls QM-EFP electrostatics screening in EFP

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 switch off electrostatic screening

1 use QM-EFP electrostatic damping if SCREEN screening parameters are provided in the EFP potential

RECOMMENDATION:

None

EFP_QM_ELEC

Controls QM-EFP electrostatics

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on QM-EFP electrostatics

FALSE switch off QM-EFP electrostatics

RECOMMENDATION:

None

EFP_QM_EXREP

Controls QM-EFP exchange-repulsion

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE switch on QM-EFP exchange-repulsion

FALSE switch off QM-EFP exchange-repulsion

RECOMMENDATION:

None

EFP_QM_POL

Controls QM-EFP polarization

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE switch on QM-EFP polarization

FALSE switch off QM-EFP polarization

RECOMMENDATION:

None

EFP

Specifies that EFP calculation is requested

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

The keyword should be present if excited state calculation is requested

EMBEDMAN

Turns density embedding on.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use density embedding.

1 Turn on density embedding.

RECOMMENDATION:

Use EMBEDMAN for QM/QM density embedded calculations.

EMBED_MU

Specifies exponent value of projection operator scaling factor, μ [Eqs. (11.118) and (11.120)].

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n $\mu = 10^n$.

RECOMMENDATION:

Values of 2 - 7 are recommended. A higher value of μ leads to better orthogonality of the fragment MOs but $\mu > 10^7$ introduces numerical noise. $\mu < 10^2$ results in non-additive terms becoming too large. Energy corrections are fairly insensitive to changes in μ within the range of $10^2 - 10^7$.

EMBED_THEORY

Specifies post-DFT method performed on fragment one.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No post HF method, only DFT on fragment one.

1 Perform CCSD(T) calculation on fragment one.

2 Perform MP2 calculation on fragment one.

RECOMMENDATION:

This should be 1 or 2 for the high-level QM calculation of fragment 1-in-2, and 0 for fragment 2-in-1 low-level QM calculation.

EMBED_THRESH

Specifies threshold cutoff for AO contribution used to determine which MOs belong to which fragments

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Threshold $= n/1000$

RECOMMENDATION:

Acceptable values range from 0 to 1000. Should only need to be tuned for non-highly localized MOs

EOM_ACP

Activates calculation of partial Auger decay widths by recomputation of the EOM-CCSD state with an Auger Channel Projector applied. Currently, this is implemented for EOMIP-CCSD calculations with a closed-shell reference.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not run ACP-EOM-CCSD calculations

1 determine partial Auger decay widths by running ACP-EOM-CCSD calculations

RECOMMENDATION:

Use to compute partial widths for a complex-variable calculation which produces a core-vacant state. An appropriate complex-scaled basis set has to be chosen in order to capture Auger decay and the optimal scaling angle needs to be determined.[28,29]

EOM_CORR

Specifies the correlation level.

TYPE:

STRING

DEFAULT:

None No correction will be computed

OPTIONS:

SD(DT) EOM-CCSD(dT), available for EE, SF, and IP

SD(FT) EOM-CCSD(fT), available for EE, SF, IP, and EA

SD(ST) EOM-CCSD(sT), available for IP

RECOMMENDATION:

None

EOM_DAVIDSON_CONVERGENCE

Convergence criterion for the RMS residuals (square of the norm) of excited-state vectors.

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Use the default. Normally this value be the same as EOM_DAVIDSON_THRESHOLD.

EOM_DAVIDSON_MAXVECTORS

Specifies maximum number of vectors in the subspace for the Davidson diagonalization.

TYPE:

INTEGER

DEFAULT:

60

OPTIONS:

n Up to n vectors per root before the subspace is reset

RECOMMENDATION:

Larger values increase disk storage but accelerate and stabilize convergence.

EOM_DAVIDSON_MAX_ITER

Maximum number of iteration allowed for Davidson diagonalization procedure.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User-defined number of iterations

RECOMMENDATION:

Default is usually sufficient

EOM_DAVIDSON_THRESHOLD

Specifies threshold for including a new expansion vector in the iterative Davidson diagonalization. Their norm must be above this threshold.

TYPE:

INTEGER

DEFAULT:

00103 Corresponding to 0.00001

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-(de+2)}$, *i.e.*, 02505->2.5×10^{-6}

RECOMMENDATION:

Use the default unless converge problems are encountered. Should normally be set to the same values as EOM_DAVIDSON_CONVERGENCE, if convergence problems arise try setting to a value slightly larger than EOM_DAVIDSON_CONVERGENCE.

EOM_EA_ALPHA

Controls the number of α-electron-attached states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute α-electron-attached states

OPTIONS:

$n > 0$ Number of α-electron-attached states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 α-electron-attached states for the first irrep, n_2 α-electron-attached states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of α-electron-attached states in case of unrestricted or open-shell calculations.

EOM_EA_BETA

Controls the number of β-electron-attached states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute β-electron-attached states

OPTIONS:

$n > 0$ Number of β-electron-attached states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 β-electron-attached states for the first irrep, n_2 β-electron-attached states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of β-electron-attached states in case of unrestricted or open-shell calculations.

EOM_FAKE_IPEA

If TRUE, calculates fake EOM-IP or EOM-EA energies and properties using the diffuse orbital trick. Default for EOM-EA and Dyson orbital calculations in CCMAN.

TYPE:

LOGICAL

DEFAULT:

FALSE (use proper EOM-IP code)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None. This feature only works for CCMAN.

EOM_IPEA_FILTER

If TRUE, filters the EOM-IP/EA amplitudes obtained using the diffuse orbital implementation (see EOM_FAKE_IPEA). Helps with convergence.

TYPE:

LOGICAL

DEFAULT:

FALSE (EOM-IP or EOM-EA amplitudes will not be filtered)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None

EOM_IP_ALPHA

Controls the number of α-ionized states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute α-ionized states

OPTIONS:

$n > 0$ Number of α-ionized states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 α-ionized states for the first irrep, n_2 α-ionized states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of α-ionized states in case of unrestricted or open-shell calculations.

EOM_IP_BETA

Controls the number of β-ionized states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not compute β-ionized states

OPTIONS:

$n > 0$ Number of β-ionized states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 β-ionized states for the first irrep, n_2 β-ionized states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of β-ionized states in case of unrestricted or open-shell calculations.

EOM_NGUESS_DOUBLES

Specifies number of excited state guess vectors which are double excitations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Include n guess vectors that are double excitations

RECOMMENDATION:

This should be set to the expected number of doubly excited states, otherwise they may not be found.

EOM_NGUESS_SINGLES

Specifies number of excited state guess vectors that are single excitations.

TYPE:

INTEGER

DEFAULT:

Equal to the number of excited states requested

OPTIONS:

n Include n guess vectors that are single excitations

RECOMMENDATION:

Should be greater or equal than the number of excited states requested, unless .

EOM_POL

Specifies the approach for calculating the polarizability of the EOM-CCSD wave function.

TYPE:

INTEGER

DEFAULT:

0 (EOM-CCSD polarizability will not be calculated)

OPTIONS:

1 (analytic-derivative or response-theory mixed symmetric-asymmetric approach)

2 (analytic-derivative or response-theory asymmetric approach)

3 (expectation-value approach with right response intermediates)

4 (expectation-value approach with left response intermediates)

23 (expectation-value approach with right response intermediates and imaginary frequencies)

RECOMMENDATION:

EOM-CCSD polarizabilities are expensive since they require solving three/nine (for static) or six/eighteen (for dynamic) additional response equations. Do no request this property unless you need it.

EOM_PRECONV_DOUBLES

When not zero, doubly excited vectors are converged prior to a full excited states calculation. Sets the maximum number of iterations for pre-converging procedure

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not pre-converge

N Perform N Davidson iterations pre-converging doubles.

RECOMMENDATION:

Occasionally necessary to ensure a doubly excited state is found. Also used in DSF, DIP, and DEA calculations instead of EOM_PRECONV_SINGLES

EOM_PRECONV_SD

When not zero, EOM vectors are pre-converged prior to a full excited states calculation. Sets the maximum number of iterations for pre-converging procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not pre-converge

N perform N Davidson iterations pre-converging singles and doubles.

RECOMMENDATION:

Occasionally necessary to ensure that all low-lying states are found. Also, very useful in EOM(2,3) calculations.

None

EOM_PRECONV_SINGLES

When not zero, singly excited vectors are converged prior to a full excited states calculation. Sets the maximum number of iterations for pre-converging procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 do not pre-converge

1 pre-converge singles

RECOMMENDATION:

Sometimes helps with problematic convergence.

EOM_SHIFT

Specifies energy shift in EOM calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n corresponds to $n \cdot 10^{-3}$ hartree shift (*i.e.*, 11000 = 11 hartree); solve for eigenstates around this value.

RECOMMENDATION:

Not available in CCMAN.

EOM_SINGLE_PREC

Precision selection for EOM-CC/MP2 calculations. Available in CCMAN2 only.

TYPE:

INTEGER

DEFAULT:

0 double-precision calculation

OPTIONS:

1 single-precision calculation

2 single-precision calculation is followed by double-precision clean-up iterations

RECOMMENDATION:

Do not set too tight convergence criteria when use single precision

EOM_USER_GUESS

Specifies if user-defined guess will be used in EOM calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Solve for a state that has maximum overlap with a trans-n specified in *$eom_user_guess*.

RECOMMENDATION:

The orbitals are ordered by energy, as printed in the beginning of the CCMAN2 output. Not available in CCMAN.

EPAO_ITERATE

Controls iterations for EPAO calculations (see PAO_METHOD).

TYPE:

INTEGER

DEFAULT:

0 Use non-iterated EPAOs based on atomic blocks of SPS.

OPTIONS:

n Optimize the EPAOs for up to n iterations.

RECOMMENDATION:

Use the default. For molecules that are not too large, one can test the sensitivity of the results to the type of minimal functions by the use of optimized EPAOs in which case a value of $n = 500$ is reasonable.

EPAO_WEIGHTS

Controls algorithm and weights for EPAO calculations (see PAO_METHOD).

TYPE:

INTEGER

DEFAULT:

115 Standard weights, use 1^{st} and 2^{nd} order optimization

OPTIONS:

15 Standard weights, with 1^{st} order optimization only.

RECOMMENDATION:

Use the default, unless convergence failure is encountered.

ERCALC

Specifies how Edmiston-Ruedenberg localized orbitals are to be calculated

TYPE:

INTEGER

DEFAULT:

06000

OPTIONS:

$aabcd$

aa specifies the convergence threshold.
If $aa > 3$, the threshold is set to 10^{-aa}. The default is 6.
If $aa = 1$, the calculation is aborted after the guess, allowing Pipek-Mezey orbitals to be extracted.

b specifies the guess:
0 Boys localized orbitals. This is the default
1 Pipek-Mezey localized orbitals.

c specifies restart options (if restarting from an ER calculation):
0 No restart. This is the default
1 Read in MOs from last ER calculation.
2 Read in MOs and RI integrals from last ER calculation.

d specifies how to treat core orbitals
0 Do not perform ER localization. This is the default.
1 Localize core and valence together.
2 Do separate localizations on core and valence.
3 Localize only the valence electrons.
4 Use the *$localize* section.

RECOMMENDATION:

ERCALC 1 will usually suffice, which uses threshold 10^{-6}.

ER_CIS_NUMSTATE

Define how many states to mix with ER localized diabatization. These states must be specified in the *$localized_diabatization* section.

TYPE:

INTEGER

DEFAULT:

0 Do not perform ER localized diabatization.

OPTIONS:

2 to N where N is the number of CIS states requested (CIS_N_ROOTS)

RECOMMENDATION:

It is usually not wise to mix adiabatic states that are separated by more than a few eV or a typical reorganization energy in solvent.

ESP_CHARGES

Controls the calculations of Merz-Kollman ESP-derived charges.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 Use Lebedev grid points around each atom.

2 Use spherical harmonics grid points around each atom.

RECOMMENDATION:

NONE

ESP_EFIELD

Triggers the calculation of the electrostatic potential (ESP) and/or the electric field at the positions of the MM charges.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Computes ESP only.

1 Computes ESP and electric field.

2 Computes electric field only.

RECOMMENDATION:

None.

ESP_GRID

Controls evaluation of the electrostatic potential on a grid of points. If enabled, the output is in an ASCII file, `plot.esp`, in the format $x, y, z, \phi(x, y, z)$ for each point, where ϕ is the ESP.

TYPE:

INTEGER

DEFAULT:

-4

OPTIONS:

-1 read grid input via the *$plots* section of the input deck

-2 same as the option -1, plus evaluate the ESP of the *$external_charges*

-3 same as the option -1 but in connection with STATE_ANALYSIS = TRUE. This computes the ESP for all excited-state densities, transition densities, and electron/hole densities.

-4 No ESP evaluation

0 Generate the ESP values at all nuclear positions

$+n$ read n grid points **in bohr** from the ASCII file `ESPGrid`

RECOMMENDATION:

For the $+n$ option, the user should also specify NO_REORIENT = TRUE. This forces Q-CHEM to use the user's coordinate frame (as established in the *$molecule* section), which should be the same coordinate frame used in the `ESPGrid` file.

ESP_SURFACE_DENSITY

Controls the spacing between grid points on vdW surfaces.

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Spacing of $0.001 \times n$ (in Å)

RECOMMENDATION:

The default corresponds to 0.5 Å spacing.

ESP_TRANS

Controls the calculation of the electrostatic potential of the transition density

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE compute the electrostatic potential of the excited state transition density

FALSE compute the electrostatic potential of the excited state electronic density

RECOMMENDATION:

NONE

EXCHANGE

Specifies the exchange functional (or most exchange-correlation functionals for backwards compatibility).

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use EXCHANGE = *NAME*, where *NAME* is either:

1) One of the exchange functionals listed in Section 5.3.3

2) One of the XC functionals listed in Section 5.3.5 that is not marked with an asterisk.

3) GEN, for a user-defined functional (see Section 5.3.7).

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations are indicated in bold in Sections 5.3.5 and 5.3.3.

EXCIT_ENERGY_COMPONENTS

Compute individual compnents of the CIS/TDDFT excitation energy.[33] The output is divided into the one-electron components (H); Fock-matrix type components representing the Coulomb (J1), non-local exchange (K1), and xc potentials (XC1); and true two-electron components (J2, K2, XC2). Note that H+J1+K1+XC1 is equivalent to a weighted sum of MO energy differences whereas J2, K2, and XC2 represent the post-MO terms. For more information see Ref. 21.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute excitation energy components.

FALSE Do not compute excitation energy components.

RECOMMENDATION:

Use if more detailed insight into excitation energies is required.

FAST_XAS

Controls whether fast TDDFT for core excitations is used.

TYPE:

LOGICAL

DEFAULT:

FALSE Normal TDDFT calculation.

OPTIONS:

TRUE Use fast TDDFT.

RECOMMENDATION:

None

FAST_XC

Controls direct variable thresholds to accelerate exchange-correlation (XC) in DFT.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Turn FAST_XC on.

FALSE Do not use FAST_XC.

RECOMMENDATION:

Caution: FAST_XC improves the speed of a DFT calculation, but may occasionally cause the SCF calculation to diverge.

FD2ND_BLOCK_INDEX

The flag that species which atom block is being processed in a given finite-difference calculation.

TYPE:

INTEGER

DEFAULT:

Uninitialized

OPTIONS:

$0, 1, ..., N_{\text{frag}} - 1$	The index of the first fragment is 0 and the last should be $N_{\text{frag}} - 1$
-1	Indicator of the job that wraps up the result.

RECOMMENDATION:

None

FD2ND_BLOCK_SIZE

Controls the number of atoms in each segment of finite-difference Hessian calculation.

TYPE:

INTEGER

DEFAULT:

Uninitialized

OPTIONS:

n	Having n atoms in each segment

RECOMMENDATION:

The value can be estimated based on the desired number of segments. It should be significantly smaller than the total number of atoms in the molecule.

FDA

Decompose intermolecular forces

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE	Does a regular force job.
TRUE	Invokes the force decomposition analysis method

RECOMMENDATION:

Set it to TRUE with jobtype=force to decompose the force.

FDE

Turns density embedding on.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True	Perform an FDET calculation.
False	Don't perform FDET calculation.

RECOMMENDATION:

Set the *$rem* variable FDE to TRUE to start a FDET calculation.

FDIFF_DER

Controls what types of information are used to compute higher derivatives. The default uses a combination of energy, gradient and Hessian information, which makes the force field calculation faster.

TYPE:

INTEGER

DEFAULT:

3 for jobs where analytical 2nd derivatives are available.

0 for jobs with ECP.

OPTIONS:

0 Use energy information only.

1 Use gradient information only.

2 Use Hessian information only.

3 Use energy, gradient, and Hessian information.

RECOMMENDATION:

When the molecule is larger than benzene with small basis set, FDIFF_DER = 2 may be faster. Note that FDIFF_DER will be set lower if analytic derivatives of the requested order are not available. Please refers to IDERIV.

FDIFF_STEPSIZE_QFF

Displacement used for calculating third and fourth derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

5291 Corresponding to 0.1 bohr. For calculating third and fourth derivatives.

OPTIONS:

n Use a step size of $n \times 10^{-5}$.

RECOMMENDATION:

Use the default, unless the potential surface is very flat, in which case a larger value should be used.

FDIFF_STEPSIZE

Displacement used for calculating derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

100 Corresponding to 0.001 Å. For calculating second derivatives.

OPTIONS:

n Use a step size of $n \times 10^{-5}$.

RECOMMENDATION:

Use the default except in cases where the potential surface is very flat, in which case a larger value should be used. See FDIFF_STEPSIZE_QFF for third and fourth derivatives.

FD_MAT_VEC_PROD

Compute Hessian-vector product using the finite difference technique.

TYPE:

BOOLEAN

DEFAULT:

FALSE (TRUE when the employed functional contains non-local correlation (except VV10))

OPTIONS:

FALSE Compute Hessian-vector product analytically.

TRUE Use finite difference to compute Hessian-vector product.

RECOMMENDATION:

Set it to TRUE when analytical Hessian is not available.

Note: For simple R and USCF calculations, it can always be set to FALSE, which indicates that only the NLC part will be computed with finite difference (if its analytic orbital hessian is unavailable).

FEFP_EFP

Specifies that fEFP_EFP calculation is requested to compute the total interaction energies between a ligand (the last fragment in the $efp_fragments section) and the protein (represented by fEFP)

TYPE:

STRING

DEFAULT:

OFF

OPTIONS:

OFF disables fEFP

LA enables fEFP with the Link Atom (HLA or CLA) scheme (only electrostatics and polarization)

MFCC enables fEFP with MFCC (only electrostatics)

RECOMMENDATION:

The keyword should be invoked if EFP/fEFP is requested (interaction energy calculations). This keyword has to be employed with EFP_FRAGMENT_ONLY = TRUE. To switch on/off electrostatics or polarzation interactions, the usual EFP controls are employed.

FEFP_QM

Specifies that fEFP_QM calculation is requested to perform a QM/fEFPcompute computation. The fEFP part is a fractionated macromolecule.

TYPE:

STRING

DEFAULT:

OFF

OPTIONS:

OFF disables fEFP_QM and performs a QM/EFP calculation

LA enables fEFP_QM with the Link Atom scheme

RECOMMENDATION:

The keyword should be invoked if QM/fEFP is requested. This keyword has to be employed with efp_fragment_only false. Only electrostatics is available.

FOA_FUNDGAP

Compute the frozen-orbital approximation of the fundamental gap.

TYPE:

Boolean

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute FOA derivative discontinuity and fundamental gap.

TRUE Compute and print FOA fundamental gap information. Implies KS_GAP_PRINT.

RECOMMENDATION:

Use in conjunction with KS_GAP_UNIT if true.

FOCK_EXTRAP_ORDER

Specifies the polynomial order N for Fock matrix extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

N Extrapolate using an Nth-order polynomial ($N > 0$).

RECOMMENDATION:

None

FOCK_EXTRAP_POINTS

Specifies the number M of old Fock matrices that are retained for use in extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Fock matrix extrapolation.

OPTIONS:

M Save M Fock matrices for use in extrapolation $(M > N)$

RECOMMENDATION:

Higher-order extrapolations with more saved Fock matrices are faster and conserve energy better than low-order extrapolations, up to a point. In many cases, the scheme (N = 6, M = 12), in conjunction with SCF_CONVERGENCE = 6, is found to provide about a 50% savings in computational cost while still conserving energy.

FOLLOW_ENERGY

Adjusts the energy window for near states

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use dynamic thresholds, based on energy difference between steps.

n Search over selected state $E_{\mathrm{est}} \pm n \times 10^{-6}\ E_h$.

RECOMMENDATION:

Use a wider energy window to follow a state diabatically, smaller window to remain on the adiabatic state most of the time.

FOLLOW_OVERLAP

Adjusts the threshold for states of similar character.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use dynamic thresholds, based on energy difference between steps.

n Percentage overlap for previous step and current step.

RECOMMENDATION:

Use a higher value to require states have higher degree of similarity to be considered the same (more often selected based on energy).

FON_E_THRESH

DIIS error below which occupations will be kept constant.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n freeze occupations below DIIS error of 10^{-n}

RECOMMENDATION:

This should be one or two numbers bigger than the desired SCF convergence threshold.

FON_NORB

Number of orbitals above and below the Fermi level that are allowed to have fractional occupancies.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n number of active orbitals

RECOMMENDATION:

The number of valence orbitals is a reasonable choice.

FON_T_END

Final electronic temperature for FON calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

Any desired final temperature.

RECOMMENDATION:

Pick the temperature to either reproduce experimental conditions (*e.g.* room temperature) or as low as possible to approach zero-temperature.

FON_T_METHOD

Selects cooling algorithm.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 temperature is scaled by a factor in each cycle

2 temperature is decreased by a constant number in each cycle

RECOMMENDATION:

We have made slightly better experience with a constant cooling rate. However, choose constant temperature when in doubt.

FON_T_SCALE

Determines the step size for the cooling.

TYPE:

INTEGER

DEFAULT:

90

OPTIONS:

n temperature is scaled by $0.01 \cdot n$ in each cycle (cooling method 1)

n temperature is decreased by n K in each cycle (cooling method 2)

RECOMMENDATION:

The cooling rate should be neither too slow nor too fast. Too slow may lead to final energies that are at undesirably high temperatures. Too fast may lead to convergence issues. Reasonable choices for methods 1 and 2 are 98 and 50, respectively. When in doubt, use constant temperature.

FON_T_START

Initial electronic temperature (in K) for FON calculation.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

Any desired initial temperature.

RECOMMENDATION:

Pick the temperature to either reproduce experimental conditions (*e.g.* room temperature) or as low as possible to approach zero-temperature.

FORCEMAN_PRINT

Controls printing level for MM (and thus QM/MM) jobs.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Minimal printing, as befits a large MM job that runs for many steps.

1 Additional information including MM gradient.

2 Print individual terms in the MM energy expression.

3 Print connectivity information.

4 Print individual terms in the MM gradient.

RECOMMENDATION:

Use the default unless trying to diagnose a problem.

FORCE_FIELD

Specifies the force field for MM energies in QM/MM calculations.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

AMBER99 AMBER99 force field

CHARMM27 CHARMM27 force field

OPLSAA OPLSAA force field

RECOMMENDATION:

None.

FORCE_SYMMETRY_ON

Overrides turning off symmetry in calculations using ghost atoms.

TYPE:

LOGICAL

DEFAULT:

FALSE Turn symmetry off when using ghost atoms.

OPTIONS:

TRUE Force symmetry.

FALSE Do not use symmetry.

RECOMMENDATION:

Use the default unless you know what you are doing.

FRACTIONAL_ELECTRON

Add or subtract a fraction of an electron.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use an integer number of electrons.

n Add $n/1000$ electrons to the system.

RECOMMENDATION:

Use only if trying to generate $E(N)$ plots. If $n < 0$, a fraction of an electron is removed from the system.

FRAC_ELEC_ORB

Specify the occupied orbital from which the fractional electron should be removed.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Remove from ϕ_n.

RECOMMENDATION:

None

FRAC_VIR_ELEC_ORB

Specify the virtual orbital to which the fractional electron should be added.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Add to ϕ_n.

RECOMMENDATION:

Use this only if FRAC_VIR_ELEC > 0.

FRAC_VIR_ELEC

Specify the fraction of an electron to place into the occupied space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Add $n/1000$ of an electron.

RECOMMENDATION:

A value > 0 should be used for excitation (XAS or XES), whereas the default is appropriate for ionization (XPS).

FRAGMO_GUESS_MODE

Decide what to do regarding the FRAGMO guess in the present job (for gen_scfman only)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Spawn fragment jobs sequentially and collect the results as the FRAGMO guess at the end.

1 Generate fragment inputs in folders “FrgX" under the scratch directory of the present job and then terminate. Users can then take advantage of a queuing system to run these jobs simultaneously using “FrgX" as their scratch folders (should be handled with scripting).

2 Read in the available fragment data.

RECOMMENDATION:

Consider using “1" if the fragment calculations are evenly expensive. Use “2" when FRAGMO guess is pre-computed.

FRAG_CPSCF_CONV

The convergence threshold for the CPSCF equation (using the RMS error of the residual vector)

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

n Convergence is reached when the RMS error is below 10^{-n}

RECOMMENDATION:

Use the default

FRAG_CPSCF_MAXITER

The maximum number of iterations executed by the conjugate-gradient solver before switching to the MINRES algorithm. The maximum number of MINRES iterations is set to twice of the value of FRAG_CPSCF_MAXITER. Note that this rem variable is also used for the CPSCF equations involved in the adiabatic EDA (Sec. 12.7.3) and force decomposition analysis (Sec. 12.7.4) calculations.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

RECOMMENDATION:

Use the default

FRGM_LPCORR

Specifies a correction method performed after the locally-projected equations are converged.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

ARS	Approximate Roothaan-step perturbative correction.
RS	Single Roothaan-step perturbative correction.
EXACT_SCF	Full SCF variational correction.
ARS_EXACT_SCF	Both ARS and EXACT_SCF in a single job.
RS_EXACT_SCF	Both RS and EXACT_SCF in a single job.

RECOMMENDATION:

For large basis sets use ARS, use RS if ARS fails.

FRGM_METHOD

Specifies a locally-projected method.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

STOLL	Locally-projected SCF equations of Stoll are solved.
GIA	Locally-projected SCF equations of Gianinetti are solved.
NOSCF_RS	Single Roothaan-step correction to the FRAGMO initial guess.
NOSCF_ARS	Approximate single Roothaan-step correction to the FRAGMO initial guess.
NOSCF_DRS	Double Roothaan-step correction to the FRAGMO initial guess.
NOSCF_RS_FOCK	Non-converged SCF energy of the single Roothaan-step MOs.

RECOMMENDATION:

STOLL and GIA are for variational optimization of the ALMOs. NOSCF options are for computationally fast corrections of the FRAGMO initial guess.

FRZ_GEOM

Compute forces on the frozen potential surface.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE	Do not compute forces on the frozen potential surface.
TRUE	Compute forces on the frozen potential surface.

RECOMMENDATION:

Set it to TRUE when optimized geometry or vibrational frequencies on the frozen potential surface are desired.

FRZ_ORTHO_DECOMP_CONV

Convergence criterion for the minimization problem that gives the orthogonal fragment densities.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n 10^{-n}

RECOMMENDATION:

Use the default unless tighter convergence is preferred.

FRZ_ORTHO_DECOMP

Perform the decomposition of frozen interaction energy based on the orthogonal decomposition of the 1PDM associated with the frozen wave function.

TYPE:

BOOLEAN

DEFAULT:

FALSE (automatically set to TRUE by EDA2 options 1–5)

OPTIONS:

FALSE Do not perform the orthogonal decomposition.

TRUE Perform the frozen energy decomposition using orthogonal fragment densities.

RECOMMENDATION:

Use default value automatically set by "EDA2". Note that users are allowed to turn off the orthogonal decomposition by setting FRZ_ORTHO_DECOMP to -1. Also, for calculations that involve ECPs, it is automatically set to FALSE since unreasonable results will be produced otherwise.

FSM_MODE

Specifies the method of interpolation

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Cartesian

2 LST

RECOMMENDATION:

In most cases, LST is superior to Cartesian interpolation.

FSM_NGRAD

Specifies the number of perpendicular gradient steps used to optimize each node

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

N Number of perpendicular gradients per node

RECOMMENDATION:

Anything between 2 and 6 should work, where increasing the number is only needed for difficult reaction paths.

FSM_NNODE

Specifies the number of nodes along the string

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

N number of nodes in FSM calculation

RECOMMENDATION:

$N = 15$. Use 10 to 20 nodes for a typical calculation. Reaction paths that connect multiple elementary steps should be separated into individual elementary steps, and one FSM job run for each pair of intermediates. Use a higher number when the FSM is followed by an approximate-Hessian based transition state search (Section 9.3.3).

FSM_OPT_MODE

Specifies the method of optimization

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Conjugate gradients

2 Quasi-Newton method with BFGS Hessian update

RECOMMENDATION:

The quasi-Newton method is more efficient when the number of nodes is high.

FSSH_CONTINUE

Restart a FSSH calculation from a previous run, using the file 396.0. When this is enabled, the initial conditions of the surface hopping calculation will be set, including the correct wave function amplitudes, initial surface, and position/momentum moments (if AFSSH) from the final step of some prior calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Start fresh calculation.

1 Restart from previous run.

RECOMMENDATION:

None

FSSH_INITIALSURFACE

Specifies the initial state in a surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n An integer between FSSH_LOWESTSURFACE and FSSH_LOWESTSURFACE + FSSH_NSURFACES -1.

RECOMMENDATION:

None

FSSH_LOWESTSURFACE

Specifies the lowest-energy state considered in a surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n Only states n and above are considered in a FSSH calculation.

RECOMMENDATION:

None

FSSH_NSURFACES

Specifies the number of states considered in a surface hopping calculation.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n n states are considered in the surface hopping calculation.

RECOMMENDATION:

Any states which may come close in energy to the active surface should be included in the surface hopping calculation.

FTC_CLASS_THRESH_MULT

Together with FTC_CLASS_THRESH_ORDER, determines the cutoff threshold for included a shell-pair in the dd class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Multiplicative part of the FTC classification threshold. Together with the default value of the FTC_CLASS_THRESH_ORDER this leads to the 5×10^{-5} threshold value.

OPTIONS:

n User specified.

RECOMMENDATION:

Use the default. If diffuse basis sets are used and the molecule is relatively big then tighter FTC classification threshold has to be used. According to our experiments using Pople-type diffuse basis sets, the default 5×10^{-5} value provides accurate result for an alanine5 molecule while 1×10^{-5} threshold value for alanine10 and 5×10^{-6} value for alanine15 has to be used.

FTC_CLASS_THRESH_ORDER

Together with FTC_CLASS_THRESH_MULT, determines the cutoff threshold for included a shell-pair in the dd class, *i.e.*, the class that is expanded in terms of plane waves.

TYPE:

INTEGER

DEFAULT:

5 Logarithmic part of the FTC classification threshold. Corresponds to 10^{-5}

OPTIONS:

n User specified

RECOMMENDATION:

Use the default.

FTC_SMALLMOL

Controls whether or not the operator is evaluated on a large grid and stored in memory to speed up the calculation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use a big pre-calculated array to speed up the FTC calculations

0 Use this option to save some memory

RECOMMENDATION:

Use the default if possible and use 0 (or buy some more memory) when needed.

FTC

Controls the overall use of the FTC.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use FTC in the Coulomb part

1 Use FTC in the Coulomb part

RECOMMENDATION:

Use FTC when bigger and/or diffuse basis sets are used.

GAUSSIAN_BLUR

Enables the use of Gaussian-delocalized external charges in a QM/MM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Delocalizes external charges with Gaussian functions.

FALSE Point charges

RECOMMENDATION:

None

GAUSS_BLUR_WIDTH

Delocalization width for external MM Gaussian charges in a Janus calculations.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Use a width of $n \times 10^{-4}$ Å.

RECOMMENDATION:

Blur all MM external charges in a QM/MM calculation with the specified width. Gaussian blurring is currently incompatible with PCM calculations. Values of 1.0–2.0 Å are recommended in Ref. 12.

GEN_SCFMAN_ALGO_1

The first algorithm to be used in a hybrid-algorithm calculation.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

All the available SCF_ALGORITHM options, including the GEN_SCFMAN additions (Section 4.3).

RECOMMENDATION:

None

GEN_SCFMAN_CONV_1

The convergence criterion given to the first algorithm. If reached, switch to the next algorithm.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n 10^{-n}

RECOMMENDATION:

None

GEN_SCFMAN_HYBRID_ALGO

Use multiple algorithms in an SCF calculation based on GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Use a single SCF algorithm (given by SCF_ALGORITHM).

TRUE Use multiple SCF algorithms (to be specified).

RECOMMENDATION:

Set it to TRUE when the use of more than one algorithm is desired.

GEN_SCFMAN_ITER_1

Maximum number of iterations given to the first algorithm. If used up, switch to the next algorithm.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

User-defined

RECOMMENDATION:

None

GEN_SCFMAN

Use GEN_SCFMAN for the present SCF calculation.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

FALSE Use the previous SCF code.

TRUE Use GEN_SCFMAN.

RECOMMENDATION:

Set to FALSE in cases where features not yet supported by GEN_SCFMAN are needed.

GEOM_OPT_CHARAC_CONV

Overide the built-in convergence criterion for the Davidson solver.

TYPE:

INTEGER

DEFAULT:

0 (use the built-in default value 10^{-5})

OPTIONS:

n Set the convergence criterion to 10^{-n}.

RECOMMENDATION:

Use the default. If it fails to converge, consider loosening the criterion with caution.

GEOM_OPT_CHARAC

Use the finite difference Davidson method to characterize the resulting energy minimum/transition state.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE do not characterize the resulting stationary point.

TRUE perform a characterization of the stationary point.

RECOMMENDATION:

Set it to TRUE when the character of a stationary point needs to be verified, especially for a transition structure.

GEOM_OPT_COORDS

Controls the type of optimization coordinates.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

0 Optimize in Cartesian coordinates.

1 Generate and optimize in internal coordinates, if this fails abort.

-1 Generate and optimize in internal coordinates, if this fails at any stage of the optimization, switch to Cartesian and continue.

2 Optimize in Z-matrix coordinates, if this fails abort.

-2 Optimize in Z-matrix coordinates, if this fails during any stage of the optimization switch to Cartesians and continue.

RECOMMENDATION:

Use the default, as delocalized internals are more efficient. Note that optimization in Z-matrix coordinates requires that the input be specified in Z-matrix format.

GEOM_OPT_DMAX

Maximum allowed step size. Value supplied is multiplied by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

300 = 0.3

OPTIONS:

n User-defined cutoff.

RECOMMENDATION:

Use the default.

GEOM_OPT_DRIVER

Controls the geometry optimization driver.

TYPE:

STRING

DEFAULT:

LIBOPT3

OPTIONS:

OPTIMIZE Use the original optimization original driver.

LIBOPT3 Use the new driver.

RECOMMENDATION:

Note that the new driver is still under active development.

GEOM_OPT_HESSIAN

Determines the initial Hessian status.

TYPE:

STRING/INTEGER

DEFAULT:

-1

OPTIONS:

-1 Approximate Hessian based on the force constant matrix (same as the "Model" option in LIBOPT3)

READ Have exact or initial Hessian. Use as is if Cartesian, or transform if internals.

RECOMMENDATION:

An accurate initial Hessian will improve the performance of the optimizer, but it may be expensive to compute. Note that more options for the initial Hessian are available in LIBOPT3, controlled by INITIAL_HESSIAN under the *$geom_opt* section (see Sec. 9.2.5).

GEOM_OPT_LINEAR_ANGLE

Threshold for near linear bond angles (degrees).

TYPE:

INTEGER

DEFAULT:

165 degrees.

OPTIONS:

n User-defined level.

RECOMMENDATION:

Use the default.

GEOM_OPT_MAX_CYCLES

Maximum number of optimization cycles.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n User defined positive integer.

RECOMMENDATION:

The default should be sufficient for most cases. Increase if the initial guess geometry is poor, or for systems with shallow potential wells.

GEOM_OPT_MAX_DIIS

Controls maximum size of subspace for GDIIS.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use GDIIS.

-1 Default size = min(NDEG, NATOMS, 4) NDEG = number of molecular degrees of freedom.

n Size specified by user.

RECOMMENDATION:

Use the default or do not set n too large.

GEOM_OPT_MODE

Determines Hessian mode followed during a transition state search.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Mode following off.

n Maximize along mode n.

RECOMMENDATION:

Use the default, for geometry optimizations.

GEOM_OPT_PRINT

Controls the amount of OPTIMIZE print output.

TYPE:

INTEGER

DEFAULT:

3 Error messages, summary, warning, standard information and gradient print out.

OPTIONS:

0 Error messages only.

1 Level 0 plus summary and warning print out.

2 Level 1 plus standard information.

3 Level 2 plus gradient print out.

4 Level 3 plus Hessian print out.

5 Level 4 plus iterative print out.

6 Level 5 plus internal generation print out.

7 Debug print out.

RECOMMENDATION:

Use the default.

GEOM_OPT_SYMFLAG

Controls the use of symmetry in OPTIMIZE.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Make use of point group symmetry.

FALSE Do not make use of point group symmetry.

RECOMMENDATION:

Use the default.

GEOM_OPT_TOL_DISPLACEMENT

Convergence on maximum atomic displacement.

TYPE:

INTEGER

DEFAULT:

$1200 \equiv 1200 \times 10^{-6}$ tolerance on maximum atomic displacement.

OPTIONS:

n Integer value (tolerance = $n \times 10^{-6}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_TOL_ENERGY

Convergence on energy change of successive optimization cycles.

TYPE:

INTEGER

DEFAULT:

$100 \equiv 100 \times 10^{-8}$ tolerance on maximum (absolute) energy change.

OPTIONS:

n Integer value (tolerance = value $n \times 10^{-8}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_TOL_GRADIENT

Convergence on maximum gradient component.

TYPE:

INTEGER

DEFAULT:

$300 \equiv 300 \times 10^{-6}$ tolerance on maximum gradient component.

OPTIONS:

n Integer value (tolerance = $n \times 10^{-6}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_UPDATE

Controls the Hessian update algorithm.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

-1 Use the default update algorithm.
0 Do not update the Hessian (not recommended).
1 Murtagh-Sargent update.
2 Powell update.
3 Powell/Murtagh-Sargent update (TS default).
4 BFGS update (OPT default).
5 BFGS with safeguards to ensure retention of positive definiteness (GDIIS default).

RECOMMENDATION:

Use the default.

GEOM_PRINT

Controls the amount of geometric information printed at each step.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Prints out all geometric information; bond distances, angles, torsions.

FALSE Normal printing of distance matrix.

RECOMMENDATION:

Use if you want to be able to quickly examine geometric parameters at the beginning and end of optimizations. Only prints in the beginning of single point energy calculations.

GHF

Run a generalized Hartree-Fock calculation with GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Run a GHF calculation.

FALSE Do not use GHF.

RECOMMENDATION:

Set to TRUE if desired.

GRAIN

Controls the number of lowest-level boxes in one dimension for CFMM.

TYPE:

INTEGER

DEFAULT:

-1 Program decides best value, turning on CFMM when useful

OPTIONS:

-1 Program decides best value, turning on CFMM when useful

1 Do not use CFMM

$n \geq 8$ Use CFMM with n lowest-level boxes in one dimension

RECOMMENDATION:

This is an expert option; either use the default, or use a value of 1 if CFMM is not desired.

GVB_AMP_SCALE

Scales the default orbital amplitude iteration step size by n/1000 for IP/RCC. PP amplitude equations are solved analytically, so this parameter does not affect PP.

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

n User-defined, 0–1000

RECOMMENDATION:

Default is usually fine, but in some highly-correlated systems it can help with convergence to use smaller values.

GVB_DO_ROHF

Sets the number of Unrestricted-in-Active Pairs to be kept restricted.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-Defined

RECOMMENDATION:

If n is the same value as GVB_N_PAIRS returns the ROHF solution for GVB, only works with the UNRESTRICTED = TRUE implementation of GVB with GVB_OLD_UPP = 0 (its default value)

GVB_DO_SANO

Sets the scheme used in determining the active virtual orbitals in a Unrestricted-in-Active Pairs GVB calculation.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 No localization or Sano procedure

1 Only localizes the active virtual orbitals

2 Uses the Sano procedure

RECOMMENDATION:

Different initial guesses can sometimes lead to different solutions. Disabling sometimes can aid in finding more non-local solutions for the orbitals.

GVB_GUESS_MIX

Similar to SCF_GUESS_MIX, it breaks alpha/beta symmetry for UPP by mixing the alpha HOMO and LUMO orbitals according to the user-defined fraction of LUMO to add the HOMO. 100 corresponds to a 1:1 ratio of HOMO and LUMO in the mixed orbitals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined, $0 \leq n \leq 100$

RECOMMENDATION:

25 often works well to break symmetry without overly impeding convergence.

GVB_LOCAL

Sets the localization scheme used in the initial guess wave function.

TYPE:

INTEGER

DEFAULT:

2 Pipek-Mezey orbitals

OPTIONS:

0 No Localization

1 Boys localized orbitals

2 Pipek-Mezey orbitals

RECOMMENDATION:

Different initial guesses can sometimes lead to different solutions. It can be helpful to try both to ensure the global minimum has been found.

GVB_N_PAIRS

Alternative to CC_REST_OCC and CC_REST_VIR for setting active space size in GVB and valence coupled cluster methods.

TYPE:

INTEGER

DEFAULT:

PP active space (1 occ and 1 virt for each valence electron pair)

OPTIONS:

n user-defined

RECOMMENDATION:

Use the default unless one wants to study a special active space. When using small active spaces, it is important to ensure that the proper orbitals are incorporated in the active space. If not, use the *$reorder_mo* feature to adjust the SCF orbitals appropriately.

GVB_OLD_UPP

Which unrestricted algorithm to use for GVB.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use Unrestricted-in-Active Pairs described in Ref. 24

1 Use Unrestricted Implementation described in Ref. 5

RECOMMENDATION:

Only works for Unrestricted PP and no other GVB model.

GVB_ORB_CONV

The GVB-CC wave function is considered converged when the root-mean-square orbital gradient and orbital step sizes are less than $10^{-\mathrm{GVB\_ORB\_CONV}}$. Adjust THRESH simultaneously.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n User-defined

RECOMMENDATION:

Use 6 for PP(2) jobs or geometry optimizations. Tighter convergence (*i.e.* 7 or higher) cannot always be reliably achieved.

GVB_ORB_MAX_ITER

Controls the number of orbital iterations allowed in GVB-CC calculations. Some jobs, particularly unrestricted PP jobs can require 500–1000 iterations.

TYPE:

INTEGER

DEFAULT:

256

OPTIONS:

User-defined number of iterations.

RECOMMENDATION:

Default is typically adequate, but some jobs, particularly UPP jobs, can require 500–1000 iterations if converged tightly.

GVB_ORB_SCALE

Scales the default orbital step size by n/1000.

TYPE:

INTEGER

DEFAULT:

1000 Corresponding to 100%

OPTIONS:

n User-defined, 0–1000

RECOMMENDATION:

Default is usually fine, but for some stretched geometries it can help with convergence to use smaller values.

GVB_POWER

Coefficient for GVB_IP exchange type amplitude regularization to improve the convergence of the amplitude equations especially for spin-unrestricted amplitudes near dissociation. This is the leading coefficient for an amplitude dampening term included in the energy denominator: -(c/10000)$(e^{t_{ij}^p} - 1)/(e^1 - 1)$

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

p User-defined

RECOMMENDATION:

Should be decreased if unrestricted amplitudes do not converge or converge slowly at dissociation, and should be kept even valued.

GVB_PRINT

Controls the amount of information printed during a GVB-CC job.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

Should never need to go above 0 or 1.

GVB_REGULARIZE

Coefficient for GVB_IP exchange type amplitude regularization to improve the convergence of the amplitude equations especially for spin-unrestricted amplitudes near dissociation. This is the leading coefficient for an amplitude dampening term $-(c/10000)(e^{t_{ij}^p} - 1)/(e^1 - 1)$

TYPE:

INTEGER

DEFAULT:

0 For restricted

1 For unrestricted

OPTIONS:

c User-defined

RECOMMENDATION:

Should be increased if unrestricted amplitudes do not converge or converge slowly at dissociation. Set this to zero to remove all dynamically-valued amplitude regularization.

GVB_REORDER_1

Tells the code which two pairs to swap first.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 1.

GVB_REORDER_2

Tells the code which two pairs to swap second.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 2.

GVB_REORDER_3

Tells the code which two pairs to swap third.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 3.

GVB_REORDER_4

Tells the code which two pairs to swap fourth.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 4.

GVB_REORDER_5

Tells the code which two pairs to swap fifth.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined XXXYYY

RECOMMENDATION:

This is in the format of two 3-digit pair indices that tell the code to swap pair XXX with YYY, for example swapping pair 1 and 2 would get the input 001002. Must be specified in GVB_REORDER_PAIRS ≥ 5.

GVB_REORDER_PAIRS

Tells the code how many GVB pairs to switch around.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n $0 \leq n \leq 5$

RECOMMENDATION:

This allows for the user to change the order the active pairs are placed in after the orbitals are read in or are guessed using localization and the Sano procedure. Up to 5 sequential pair swaps can be made, but it is best to leave this alone.

GVB_RESTART

Restart a job from previously-converged GVB-CC orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Useful when trying to converge to the same GVB solution at slightly different geometries, for example.

GVB_SHIFT

Value for a statically valued energy shift in the energy denominator used to solve the coupled cluster amplitude equations, n/10000.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

Default is fine, can be used in lieu of the dynamically valued amplitude regularization if it does not aid convergence.

GVB_SYMFIX

Should GVB use a symmetry breaking fix.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 no symmetry breaking fix

1 symmetry breaking fix with virtual orbitals spanning the active space

2 symmetry breaking fix with virtual orbitals spanning the whole virtual space

RECOMMENDATION:

It is best to stick with type 1 to get a symmetry breaking correction with the best results coming from CORRELATION = NP and GVB_SYMFIX = 1.

GVB_SYMPEN

Sets the pre-factor for the amplitude regularization term for the SB amplitudes.

TYPE:

INTEGER

DEFAULT:

160

OPTIONS:

γ User-defined

RECOMMENDATION:

Sets the pre-factor for the amplitude regularization term for the SB amplitudes: $-(\gamma/1000)(e^{(c*100)*t^2}-1)$.

GVB_SYMSCA

Sets the weight for the amplitude regularization term for the SB amplitudes.

TYPE:

INTEGER

DEFAULT:

125

OPTIONS:

c User-defined

RECOMMENDATION:

Sets the weight for the amplitude regularization term for the SB amplitudes: $-(\gamma/1000)(e^{(c*100)*t^2}-1)$.

GVB_TRUNC_OCC

Controls how many pairs' occupied orbitals are truncated from the GVB active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

This allows for asymmetric GVB active spaces removing the n lowest energy occupied orbitals from the GVB active space while leaving their paired virtual orbitals in the active space. Only the models including the SIP and DIP amplitudes (*i.e.* NP and 2P) benefit from this all other models this equivalent to just reducing the total number of pairs.

GVB_TRUNC_VIR

Controls how many pairs' virtual orbitals are truncated from the GVB active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined

RECOMMENDATION:

This allows for asymmetric GVB active spaces removing the n highest energy occupied orbitals from the GVB active space while leaving their paired virtual orbitals in the active space. Only the models including the SIP and DIP amplitudes (*i.e.* NP and 2P) benefit from this all other models this equivalent to just reducing the total number of pairs.

GVB_UNRESTRICTED

Controls restricted versus unrestricted PP jobs. Usually handled automatically.

TYPE:

LOGICAL

DEFAULT:

same value as UNRESTRICTED

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Set this variable explicitly only to do a UPP job from an RHF or ROHF initial guess. Leave this variable alone and specify UNRESTRICTED = TRUE to access the new unrestricted-in-active-pairs GVB code which can return an RHF or ROHF solution if used with GVB_DO_ROHF

G_TENSOR

Activates g-tensor calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Don't calculate g-tensor

TRUE (or 1) Calculate g-tensor.

RECOMMENDATION:

None.

HBCI_EPS1

Determines dimension of HBCI space.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

N HBCI ε_1 in μ $\mathrm{E_h}$

RECOMMENDATION:

Use default or 500 for tighter convergence.

HESS_AND_GRAD

Enables the evaluation of both analytical gradient and Hessian in a single job

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Evaluates both gradient and Hessian.

FALSE Evaluates Hessian only.

RECOMMENDATION:

Use only in a frequency (and thus Hessian) evaluation.

HFK_LR_COEF

Sets the coefficient for long-range HF exchange

TYPE:

INTEGER

DEFAULT:

100000000

OPTIONS:

n Corresponding to $n/100000000$

RECOMMENDATION:

None

HFK_SR_COEF

Sets the coefficient for short-range HF exchange

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $n/100000000$

RECOMMENDATION:

None

HFPT_BASIS

Specifies the secondary basis in a HFPC/DFPC calculation.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

None

RECOMMENDATION:

See reference for recommended basis set, functional, and grid pairings.

HFPT

Activates HFPC/DFPC calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Single-point energy only

RECOMMENDATION:

Use Dual-Basis to capture large-basis effects at smaller basis cost. See reference for recommended basis set, functional, and grid pairings.

HF_LR

Sets the fraction of Hartree-Fock exchange at $r_{12} = \infty$.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to HF_LR = $n/1000$

RECOMMENDATION:

None

HF_SR

Sets the fraction of Hartree-Fock exchange at $r_{12} = 0$.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to HF_SR = $n/1000$

RECOMMENDATION:

None

HIRSHFELD_CONV

Set different SCF convergence criterion for the calculation of the single-atom Hirshfeld calculations

TYPE:

INTEGER

DEFAULT:

same as SCF_CONVERGENCE

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

5

HIRSHFELD_READ

Switch to force reading in of isolated atomic densities.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Read in isolated atomic densities from previous Hirshfeld calculation from disk.

FALSE Generate new isolated atomic densities.

RECOMMENDATION:

Use the default unless system is large. Note, atoms should be in the same order with same basis set used as in the previous Hirshfeld calculation (although coordinates can change). The previous calculation should be run with the -save switch.

HIRSHFELD_SPHAVG

Controls whether atomic densities should be spherically averaged in pro-molecule.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Spherically average atomic densities.

FALSE Do not spherically average.

RECOMMENDATION:

Use the default.

HIRSHFELD

Controls running of Hirshfeld population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate Hirshfeld populations.

FALSE Do not calculate Hirshfeld populations.

RECOMMENDATION:

None

HIRSHITER_THRESH

Controls the convergence criterion of iterative Hirshfeld population analysis.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

N Corresponding to the convergence criterion of $N/10000$, in e.

RECOMMENDATION:

Use the default, which is the value recommended in Ref. 7

HIRSHITER

Controls running of iterative Hirshfeld population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate iterative Hirshfeld populations.

FALSE Do not calculate iterative Hirshfeld populations.

RECOMMENDATION:

None

HIRSHMOD

Apply modifiers to the free-atom volumes used in the calculation of the scaled TS-vdW parameters

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

0 Do not apply modifiers to the Hirshfeld volumes.

1 Apply built-in modifier to H.

2 Apply built-in modifier to H and C.

3 Apply built-in modifier to H, C and N.

4 Apply built-in modifier to H, C, N and O

RECOMMENDATION:

Use the default

IDERIV

Controls the order of derivatives that are evaluated analytically. The user is not normally required to specify a value, unless numerical derivatives are desired. The derivatives will be evaluated numerically if IDERIV is set lower than JOBTYPE requires.

TYPE:

INTEGER

DEFAULT:

Set to the order of derivative that JOBTYPE requires

OPTIONS:

2 Analytic second derivatives of the energy (Hessian)

1 Analytic first derivatives of the energy.

0 Analytic energies only.

RECOMMENDATION:

Usually set to the maximum possible for efficiency. Note that IDERIV will be set lower if analytic derivatives of the requested order are not available.

IFCI_NO_THRESH

Equivalent to HBCI ε_1 for increment-specific NO generation step.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n in $\mu\mathrm{E_h}$

RECOMMENDATION:

Set to equal HBCI_EPS1.

IFCI_OCC

Specifies the number of active occupied orbitals.

TYPE:

INTEGER

DEFAULT:

Full valence.

OPTIONS:

n Include n orbitals in the active space

-1 Full valence

RECOMMENDATION:

Use full valence active space.

IFCI_PRINT

Larger number gives more output.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Minimal output

2 Readable output

3 Extra output

4 Excessive output

5+ Bug testing output

RECOMMENDATION:

2 is recommended, 1-3 is appropriate, larger than 4 is unnecessary (consider yourself warned).

IFCI_QUAD_SCREEN

Cutoff ($\mathcal{C}_4$) for determining if a 4-body term is significant.

TYPE:

INTEGER

DEFAULT:

IFCI_TRIPLES_SCREEN

OPTIONS:

n where $\mathcal{C}_4 = 10^{-\zeta} \times n$ in $\mathrm{E_h}$

RECOMMENDATION:

Same as IFCI_TRIPLES_SCREEN but note that 4-body terms are significantly more costly.

IFCI_READ

Restarts iFCI with existing `TUPLES_1E_DATA file`, if it exists.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Start from scratch

1 Restart from previous file

RECOMMENDATION:

Use 0 if no previous run files exist. Use 1 if intending to restart from previous data.

IFCI_REF_ITER

Use HF or PP reference density.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 HF

1 PP

RECOMMENDATION:

Use 0.

IFCI_STATE_ADD

Adds additional states to HBCI solver when there is degeneracy amongst states.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Add states within n $\mathrm{mE_h}$

RECOMMENDATION:

Use default unless it is known that degenerate states are present.

IFCI_TRIPLES_SCREEN

Cutoff ($\mathcal{C}_3$) for determining if a 3-body term is significant.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n where $\mathcal{C}_3 = 10^{-\zeta} \times n$ in $\mathrm{E_h}$

RECOMMENDATION:

Use the default unless looser (higher n) or tighter (lower n) consideration of triads for a given system is desired. Setting to 0 computes all triads (costly).

IFCI_TRIPLETS

Set state to solve.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Singlet

1 Triplet

2 Quintet

RECOMMENDATION:

None

IFCI_TUPLES

Level of n-body expansion to solve. Note that $n > 2$ can be computationally costly.

TYPE:

INTEGER

DEFAULT:

Must be set.

OPTIONS:

1 $n = 1$

2 $n = 2$

3 $n = 3$

4 $n = 4$

RECOMMENDATION:

Use $n = 2$ for initial system analysis, $n > 2$ for higher accuracy.

IFCI_TUPLE_THRESH

Collapse near-degenerate geminals within threshold into one body.

TYPE:

INTEGER

DEFAULT:

2500

OPTIONS:

n in $\mu\mathrm{E_h}$

RECOMMENDATION:

Use default unless there are sets of highly correlating occupied orbitals.

IFCI_ZETA

Convergence for each iFCI increment. Note that the format is $\zeta = \text{IFCI\_ZETA}/10$.

TYPE:

INTEGER

DEFAULT:

55

OPTIONS:

45 Loose

55 Moderate

65 Tight

75 Tighter

85 Quite tight

95 Maximum

RECOMMENDATION:

Use 65 and increase to 75 to check convergence.

IGNORE_LOW_FREQ

Low frequencies that should be treated as rotation can be ignored during anharmonic correction calculation.

TYPE:

INTEGER

DEFAULT:

300 Corresponding to 300 cm^{-1}.

OPTIONS:

n Any mode with harmonic frequency less than n will be ignored.

RECOMMENDATION:

Use the default.

INCDFT_DENDIFF_THRESH

Sets the threshold for screening density matrix values in the IncDFT procedure.

TYPE:

INTEGER

DEFAULT:

SCF_CONVERGENCE + 3

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

INCDFT_DENDIFF_VARTHRESH

Sets the lower bound for the variable threshold for screening density matrix values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT_DENDIFF_THRESH is reached. This means this value must be set lower than INCDFT_DENDIFF_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

INCDFT_GRIDDIFF_THRESH

Sets the threshold for screening functional values in the IncDFT procedure

TYPE:

INTEGER

DEFAULT:

SCF_CONVERGENCE + 3

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten the threshold.

INCDFT_GRIDDIFF_VARTHRESH

Sets the lower bound for the variable threshold for screening the functional values in the IncDFT procedure. The threshold will begin at this value and then vary depending on the error in the current SCF iteration until the value specified by INCDFT_GRIDDIFF_THRESH is reached. This means that this value must be set lower than INCDFT_GRIDDIFF_THRESH.

TYPE:

INTEGER

DEFAULT:

0 Variable threshold is not used.

OPTIONS:

n Corresponding to a threshold of 10^{-n}.

RECOMMENDATION:

If the default value causes convergence problems, set this value higher to tighten accuracy. If this fails, set to 0 and use a static threshold.

INCDFT

Toggles the use of the IncDFT procedure for DFT energy calculations.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not use IncDFT

TRUE Use IncDFT

RECOMMENDATION:

Turning this option on can lead to faster SCF calculations, particularly towards the end of the SCF. Please note that for some systems use of this option may lead to convergence problems.

INCFOCK

Iteration number after which the incremental Fock matrix algorithm is initiated

TYPE:

INTEGER

DEFAULT:

1 Start INCFOCK after iteration number 1

OPTIONS:

User-defined (0 switches INCFOCK off)

RECOMMENDATION:

May be necessary to allow several iterations before switching on INCFOCK.

INTEGRALS_BUFFER

Controls the size of in-core integral storage buffer.

TYPE:

INTEGER

DEFAULT:

15 15 Megabytes.

OPTIONS:

User defined size.

RECOMMENDATION:

Use the default, or consult your systems administrator for hardware limits.

INTEGRAL_2E_OPR

Determines the two-electron operator.

TYPE:

INTEGER

DEFAULT:

-2 Coulomb Operator.

OPTIONS:

-1 Apply the CASE approximation.

-2 Coulomb Operator.

RECOMMENDATION:

Use the default unless the CASE operator is desired.

INTEGRAL_SYMMETRY

Controls the efficiency through the use of point group symmetry for calculating integrals.

TYPE:

LOGICAL

DEFAULT:

False Do not use symmetry for computing integrals.

OPTIONS:

TRUE Use symmetry when available.

FALSE Do not use symmetry. This is always the case for RIMP2 jobs

RECOMMENDATION:

Use the default unless benchmarking. Note that symmetry usage is disabled for RIMP2, FFT, and QM/MM jobs.

INTERNAL_STABILITY_CONV

Convergence criterion for the Davidson solver (for the lowest eigenvalues).

TYPE:

INTEGER

DEFAULT:

4 (3 when FD_MAT_VEC_PROD = TRUE)

OPTIONS:

n Terminate Davidson iterations when the norm of the residual vector is below 10^{-n}.

RECOMMENDATION:

Use the default.

INTERNAL_STABILITY_DAVIDSON_ITER

Maximum number of Davidson iterations allowed in one stability analysis.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n Perform up to n Davidson iterations.

RECOMMENDATION:

Use the default.

INTERNAL_STABILITY_ITER

Maximum number of new SCF calculations permitted after the first stability analysis is performed.

TYPE:

INTEGER

DEFAULT:

0 (automatically set to 1 if INTERNAL_STABILITY = TRUE)

OPTIONS:

n n new SCF calculations permitted.

RECOMMENDATION:

Give a larger number if 1 is not enough (still unstable).

INTERNAL_STABILITY_ROOTS

Number of lowest Hessian eigenvalues to solve for.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n Solve for n lowest eigenvalues.

RECOMMENDATION:

Use the default.

INTERNAL_STABILITY

Perform internal stability analysis in GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform internal stability analysis after convergence.

TRUE Perform internal stability analysis and generate the corrected MOs.

RECOMMENDATION:

Turn it on when the SCF solution is prone to unstable solutions, especially for open-shell species.

INTRACULE

Controls whether intracule properties are calculated (see also the *$intracule* section).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE No intracule properties.

TRUE Evaluate intracule properties.

RECOMMENDATION:

None

IP_ALPHA

Sets the number of ionized target states derived by removing α electron ($M_S = \frac{1}{2}$).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/α states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

IP_BETA

Sets the number of ionized target states derived by removing β electron ($M_S = \frac{1}{2}$, default for EOM-IP).

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not look for any IP/β states.

OPTIONS:

$[i, j, k \ldots]$ Find i ionized states in the first irrep, j states in the second irrep *etc.*

RECOMMENDATION:

None

IP_STATES

Controls the number of ionized states to calculate.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

0 Do not perform an IP-ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of ionized states in case of restricted calculations.

IQMOL_FCHK

Controls printing of a formatted checkpoint file that can be read by the IQMOL program.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not generate the checkpoint file.

OPTIONS:

TRUE Generate a checkpoint file named `inputfilename.fchk`.

RECOMMENDATION:

For many Q-CHEM jobs there is no reason *not* to generate the checkpoint file. Note that GUI = 2 (used by IQMOL) is synonymous with IQMOL_FCHK = TRUE.

ISOTOPES

Specifies if non-default masses are to be used in the frequency calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use default masses only.

TRUE Read isotope masses from *$isotopes* section.

RECOMMENDATION:

None

JOBTYPE

Specifies the calculation.

TYPE:

STRING

DEFAULT:

Default is single-point, which should be changed to one of the following options.

OPTIONS:

OPT Equilibrium structure optimization.

TS Transition structure optimization.

RPATH Intrinsic reaction path following.

RECOMMENDATION:

Application-dependent.

KS_GAP_PRINT

Control printing of (generalized Kohn-Sham) HOMO-LUMO gap information.

TYPE:

Boolean

DEFAULT:

false

OPTIONS:

false (default) do not print gap information

true print gap information

RECOMMENDATION:

Use in conjunction with KS_GAP_UNIT if true.

KS_GAP_UNIT

Unit for KS_GAP_PRINT and FOA_FUNDGAP (see Section 5.12.2)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 (default) hartrees

1 eV

RECOMMENDATION:

none

LB94_BETA

Sets the β parameter for the LB94 XC potential

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Corresponding to $\beta = n/10000$.

RECOMMENDATION:

Use the default.

LIBPT_MIXED_PRECISION

Deploys single-precision evaluation of (T) and (fT) within LIBPT.

TYPE:

INTEGER

DEFAULT:

0 do not use single precision

OPTIONS:

1 use single precision

RECOMMENDATION:

Use in combination with USE_LIBPT.

LINK_ATOM_PROJECTION

Controls whether to perform a link-atom projection

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Performs the projection

FALSE No projection

RECOMMENDATION:

Necessary in a full QM/MM Hessian evaluation on a system with link atoms

LIN_K

Controls whether linear scaling evaluation of exact exchange (LinK) is used.

TYPE:

LOGICAL

DEFAULT:

Program chooses, switching on LinK whenever CFMM is used.

OPTIONS:

TRUE Use LinK

FALSE Do not use LinK

RECOMMENDATION:

Use for HF and hybrid DFT calculations with large numbers of atoms.

LOBA_THRESH

Specifies the thresholds to use for LOBA

TYPE:

INTEGER

DEFAULT:

6015

OPTIONS:

aabb aa specifies the threshold to use for localization
bb specifies the threshold to use for occupation
Both are given as percentages.

RECOMMENDATION:

Decrease bb to see the smaller contributions to orbitals. Values of aa between 40 and 75 have been shown to given meaningful results.

LOBA

Specifies the methods to use for LOBA

TYPE:

INTEGER

DEFAULT:

00

OPTIONS:

ab

a specifies the localization method

0 Perform Boys localization.

1 Perform PM localization.

2 Perform ER localization.

b specifies the population analysis method

0 Do not perform LOBA. This is the default.

1 Use Mulliken population analysis.

2 Use Löwdin population analysis.

RECOMMENDATION:

Boys Localization is the fastest. ER will require an auxiliary basis set.

LOBA 12 provides a reasonable speed/accuracy compromise.

LOCALFREQ_GROUP1

Select the number of modes to include in the first subset of modes to localize independently when the keyword LOCALFREQ_GROUPS > 0.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n User-specified integer.

RECOMMENDATION:

Modes will be included starting with the lowest frequency mode and then in ascending energy order up to the defined value.

LOCALFREQ_GROUPS

Select the number of groups of frequencies to be localized separately within a localized mode calculation. The size of the groups are then controlled using the LOCALFREQ_GROUP1, LOCALFREQ_GROUP2, and LOCALFREQ_GROUP3 keywords.

TYPE:

INTEGER

DEFAULT:

0 Localize all normal modes together.

OPTIONS:

1 Define one subset of modes to localize independently.

2 Define two subsets of modes to localize independently.

3 Define three subsets of modes to localize independently.

RECOMMENDATION:

None

LOCALFREQ_MAX_ITER

Controls the maximum number of mode localization sweeps permitted.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

n User-specified integer.

RECOMMENDATION:

None

LOCALFREQ_SELECT

Select a subset of normal modes for subsequent anharmonic frequency analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE Use all normal modes.

OPTIONS:

TRUE Select a subset of normal modes.

RECOMMENDATION:

None

LOCALFREQ_THRESH

Mode localization is considered converged when the change in the localization criterion is less than $10^{-\mathrm{LOCALFREQ\_THRESH}}$.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n User-specified integer.

RECOMMENDATION:

None

LOCALFREQ

Controls whether a vibrational mode localization calculation is performed.

TYPE:

INTEGER

DEFAULT:

0 Normal mode calculation.

OPTIONS:

1 Localized mode calculation with a Pipek-Mezey like criterion.

2 Localized mode calculation with a Boys like criterion.

RECOMMENDATION:

None

LOCAL_CIS

Invoke ALMO-CIS/TDA or ALMO-CIS/TDA+CT calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Regular CIS/TDDFT calculations

1 ALMO-CIS/TDA without RI

2 ALMO-CIS with RI

RECOMMENDATION:

Use 2 when running full-spectrum ALMO-CIS calculations (EIGSLV_METH = 0)

Use 1 when running the iterative version of ALMO-CIS/TDA (EIGSLV_METH = 1)

LOCAL_INTERP_ORDER

Controls the order of the B-spline

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n An integer

RECOMMENDATION:

The default value is sufficiently accurate

LOC_CIS_OV_SEPARATE

Decide whether or not to localized the "occupied" and "virtual" components of the localized diabatization function, *i.e.*, whether to localize the electron attachments and detachments separately.

TYPE:

LOGICAL

DEFAULT:

FALSE Do not separately localize electron attachments and detachments.

OPTIONS:

TRUE

RECOMMENDATION:

If one wants to use Boys localized diabatization for energy transfer (as opposed to electron transfer) , this is a necessary option. ER is more rigorous technique, and does not require this OV feature, but will be somewhat slower.

LOWDIN_POPULATION

Run Löwdin population analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate Löwdin populations.

TRUE Run Löwdin population analysis.

RECOMMENDATION:

None

LRC_DFT

Controls the application of long-range-corrected DFT

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply long-range correction.

TRUE (or 1) Add 100% long-range Hartree-Fock exchange to the requested functional.

RECOMMENDATION:

The *$rem* variable OMEGA must also be specified, in order to set the range-separation parameter.

MAGNET

Activate the magnetic property module.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Don't activate the magnetic property module.

TRUE (or 1) Activate the magnetic property module.

RECOMMENDATION:

None.

MAKE_CUBE_FILES

Requests generation of cube files for MOs, NTOs, or NBOs.

TYPE:

LOGICAL/STRING

DEFAULT:

FALSE

OPTIONS:

FALSE Do not generate cube files.

TRUE Generate cube files for MOs and densities.

NTOS Generate cube files for NTOs.

NBOS Generate cube files for NBOs.

RECOMMENDATION:

None

MANY_BODY_INT

Perform a MBE calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a MBE calculation.

FALSE Do not perform a MBE calculation.

RECOMMENDATION:

NONE

MAXBOX

Sets the size of the box which the molecules are kept within.

TYPE:

INTEGER

DEFAULT:

20000

OPTIONS:

n Corresponding to MAXBOX = $n/1000$ bohr.

RECOMMENDATION:

Need to ensure that the cluster can fit within this box.

MAX_ADIIS_CYCLES

The maximum number of ADIIS cycles before switching to DIIS in a ADIIS-DIIS calculations

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

N Doing at most N ADIIS iterations before switching to DIIS

RECOMMENDATION:

Use the default; typically there is no benefit of doing ADIIS for too many iterations

MAX_CASSCF_CYCLES

Maximum number of orbital optimization cycles for CASSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

N set maximum number of optimization cycles to N

RECOMMENDATION:

MAX_CIS_CYCLES

Maximum number of CIS iterative cycles allowed.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User-defined number of cycles.

RECOMMENDATION:

Default is usually sufficient.

MAX_CIS_SUBSPACE

Maximum number of subspace vectors allowed in the CIS iterations

TYPE:

INTEGER

DEFAULT:

As many as required to converge all roots

OPTIONS:

n User-defined number of subspace vectors

RECOMMENDATION:

The default is usually appropriate, unless a large number of states are requested for a large molecule. The total memory required to store the subspace vectors is bounded above by $2nOV$, where O and V represent the number of occupied and virtual orbitals, respectively. n can be reduced to save memory, at the cost of a larger number of CIS iterations. Convergence may be impaired if n is not much larger than CIS_N_ROOTS.

MAX_DIIS_CYCLES

The maximum number of DIIS iterations before switching to (geometric) direct minimization when SCF_ALGORITHM is DIIS_GDM or DIIS_DM. See also THRESH_DIIS_SWITCH.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

1 Only a single Roothaan step before switching to (G)DM

n n DIIS iterations before switching to (G)DM.

RECOMMENDATION:

None

MAX_DISPLACE

Sets the maximum distance a molecule will be moved during a translation.

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n Corresponding to MAX_DISPLACE = $n/100$ bohr.

RECOMMENDATION:

None.

MAX_JUMP

INTEGER

TYPE:

Sets the number of moves accepted on jumping.

DEFAULT:

10

OPTIONS:

User defined.

RECOMMENDATION:

None

MAX_RCA_CYCLES

The maximum number of RCA iterations before switching to DIIS when SCF_ALGORITHM is RCA_DIIS.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

N N RCA iterations before switching to DIIS

RECOMMENDATION:

None

MAX_SCF_CYCLES

Controls the maximum number of SCF iterations permitted.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

Increase for slowly converging systems such as those containing transition metals.

MBDVDW_BETA

Set custom value of the s_R (β) damping parameter

TYPE:

INTEGER

DEFAULT:

no default value defined

OPTIONS:

n Corresponding to $n \cdot 10^{-4}$

RECOMMENDATION:

Use predefined values for supported functionals, otherwise consult Ref. 2 and other relevant literature.

MBDVDW

Flag to switch on the MBD-vdW method

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0	Do not calculate MBD.
MBD_SP	Calculate the MBD-vdW contribution to the energy.
MBD_FORCES	Calculate the MBD-vdW contribution to the energy and the nuclear gradient.
MBD_SCF	Calculate the MBD-vdW contribution to the energy, the nuclear as well as the SCF gradient.

RECOMMENDATION:

NONE

MC_CYCLES

INTEGER

TYPE:

Sets the number of cycles in a basin hopping search.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

MC_STEPS

INTEGER

TYPE:

Sets the number of Monte Carlo steps in each MC_CYCLES. After MC_STEPS jumping is initiated.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

MC_TEMP

INTEGER

TYPE:

Sets the temperature (in Kelvin).

DEFAULT:

300

OPTIONS:

User defined.

RECOMMENDATION:

None

MECP_METHODS

Determines which method to be used.

TYPE:

STRING

DEFAULT:

BRANCHING_PLANE

OPTIONS:

BRANCHING_PLANE	Use the branching-plane updating method.
MECP_DIRECT	Use the direct method.
PENALTY_FUNCTION	Use the penalty-constrained method.

RECOMMENDATION:

The direct method is stable for small molecules or molecules with high symmetry. The branching-plane updating method is more efficient for larger molecules but does not work if the two states have different symmetries. If using the branching-plane updating method, GEOM_OPT_COORDS must be set to 0 in the *$rem* section, as this algorithm is available in Cartesian coordinates only. The penalty-constrained method converges slowly and is suggested only if other methods fail.

MECP_OPT

Determines whether we are doing MECP optimizations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	Do MECP optimization.
FALSE	Do not do MECP optimization.

RECOMMENDATION:

None.

MECP_PROJ_HESS

Determines whether to project out the coupling vector from the Hessian when using branching plane updating method.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE

FALSE

RECOMMENDATION:

Use the default.

MECP_STATE1

Sets the first Born-Oppenheimer state for MECP optimization.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[i,j] Find the jth excited state with the total spin i; $j = 0$ means the SCF ground state.

RECOMMENDATION:

i is ignored for restricted calculations; for unrestricted calculations, i can only be 0 or 1.

MECP_STATE2

Sets the second Born-Oppenheimer state for MECP optimization.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

[i,j] Find the jth excited state with the total spin i; $j = 0$ means the SCF ground state.

RECOMMENDATION:

i is ignored for restricted calculations; for unrestricted calculations, i can only be 0 or 1.

MEM_STATIC

Sets the memory for AO-integral evaluations and their transformations in Q-CHEM 4.1 or older versions.

TYPE:

INTEGER

DEFAULT:

192 corresponding to 192 MB.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

For RI-MP2 calculations using Q-CHEM 4.1 or older versions, $150(ON + V)$ of MEM_STATIC is required. Because a number of matrices with N^2 size also need to be stored, 32–160 MB of additional MEM_STATIC is needed.

MEM_TOTAL

Sets the total memory available to Q-CHEM, in megabytes.

TYPE:

INTEGER

DEFAULT:

2000 2 GB

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use the default, or set to the physical memory of your machine. The minimum requirement is $3X^2$.

METECO

Sets the threshold criteria for discarding shell-pairs.

TYPE:

INTEGER

DEFAULT:

2 Discard shell-pairs below $10^{-\mathrm{THRESH}}$.

OPTIONS:

1 Discard shell-pairs four orders of magnitude below machine precision.

2 Discard shell-pairs below $10^{-\mathrm{THRESH}}$.

RECOMMENDATION:

Use the default.

METHOD

Specifies the level of theory, either DFT or wave function-based.

TYPE:

STRING

DEFAULT:

HF No correlation, Hartree-Fock exchange

OPTIONS:

MP2	Sections 6.3 and 6.4
RI-MP2	Section 6.6
Local_MP2	Section 6.5
RILMP2	Section 6.6.2
ATTMP2	Section 6.7
ATTRIMP2	Section 6.7
ZAPT2	A more efficient restricted open-shell MP2 method. [19]
MP3	Section 6.3
MP4SDQ	Section 6.3
MP4	Section 6.3
CCD	Section 6.12
CCD(2)	Section 6.13
CCSD	Section 6.12
CCSDT	Section 6.12.3
CC2	Section 6.12
CCSD(T)	Section 6.13
CCSD(2)	Section 6.13
CCSD(fT)	Section 6.13.3
CCSD(dT)	Section 6.13.3
QCISD	Section 6.12
QCISD(T)	Section 6.13
OD	Section 6.12
OD(T)	Section 6.13
OD(2)	Section 6.13
VOD	Section 6.14
VOD(2)	Section 6.14
QCCD	Section 6.12
QCCD(T)	
QCCD(2)	
VQCCD	Section 6.14

RECOMMENDATION:

Consult the literature for guidance.

MGC_AMODEL

Choice of approximate cluster model.

TYPE:

INTEGER

DEFAULT:

Determines how the CC equations are approximated:

OPTIONS:

0 Local Active-Space Amplitude iterations (pre-calculate GVB orbitals with your method of choice (RPP is good)).

7 Optimize-Orbitals using the VOD 2-step solver. (Experimental-only use with MGC_AMPS = 2, 24 ,246)

8 Traditional Coupled Cluster up to CCSDTQPH.

9 MR-CC version of the Pair-Models. (Experimental)

RECOMMENDATION:

None

MGC_AMPS

Choice of Amplitude Truncation

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

$2 \leq n \leq 123456$, a sorted list of integers for every amplitude which will be iterated. Choose 1234 for PQ and 123456 for PH

RECOMMENDATION:

None

MGC_LOCALINTER

Pair filter on an intermediate.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Any nonzero value enforces the pair constraint on intermediates, significantly reducing computational cost. Not recommended for ≤ 2 pair locality

RECOMMENDATION:

None

MGC_LOCALINTS

Pair filter on an integrals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

Enforces a pair filter on the 2-electron integrals, significantly
reducing computational cost. Generally useful for more than 1 pair locality.

RECOMMENDATION:

None

MGC_NLPAIRS

Number of local pairs on an amplitude.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

Must be greater than 1, which corresponds to the PP model. 2 for PQ, and 3 for PH.

RECOMMENDATION:

None

MGEMM_THRESH

Sets MGEMM threshold to determine the separation between "large" and "small" matrix elements. A larger threshold value will result in a value closer to the single-precision result. Note that the desired factor should be multiplied by 10000 to ensure an integer value.

TYPE:

INTEGER

DEFAULT:

10000 (corresponds to 1)

OPTIONS:

n User-specified threshold

RECOMMENDATION:

For small molecules and basis sets up to triple-ζ, the default value suffices to not deviate too much from the double-precision values. Care should be taken to reduce this number for larger molecules and also larger basis-sets.

MGGA_GINV

Controls whether to add gauge invariance correction to meta-GGA functionals.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No correction.

1 Add gauge invariance correction to meta-GGA functionals.

RECOMMENDATION:

Not recommended when TDA is used because the TDA violates gauge invariance.

MIN_SEPARATION

Reject initial structures where the closest approach of molecules is less than this value.

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

n Corresponding to MIN_SEPARATION = $n/100$ bohr.

RECOMMENDATION:

MIN_SEPARATION of approximately 2.5 bohr.

MI_ACTIVE_FRAGMENT

Sets the active fragment

TYPE:

INTEGER

DEFAULT:

NO DEFAULT

OPTIONS:

n Specify the fragment on which the TDDFT calculation is to be performed, for LEA-TDDFT(MI).

RECOMMENDATION:

None

MI_LEA

Controls the LEA-TDDFT(MI) methods

TYPE:

INTEGER

DEFAULT:

NO DEFAULT

OPTIONS:

0 The LEA0 method

1 The LEA-Q method

2 The LEAc method

RECOMMENDATION:

1

MM_CHARGES

Requests the calculation of multipole-derived charges (MDCs).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculates the MDCs and also the traceless form of the multipole moments

RECOMMENDATION:

Set to TRUE if MDCs or the traceless form of the multipole moments are desired. The calculation does not take long.

MM_SUBTRACTIVE

Specifies whether a subtractive scheme is used in the E_{Coul}, Eq. (11.50), portion of the calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Only pairs that are not 1-2, 1-3, or 1-4 pairs are used.

TRUE All pairs are calculated, and then the pairs that are double counted (1-2, 1-3, and 1-4) are subtracted out.

RECOMMENDATION:

When running QM/MM or MM calculations there is not recommendation. When running a QM/MM-Ewald calculation the value must be set to TRUE.

MODEL_SYSTEM_CHARGE

Specifies the QM subsystem charge if different from the *$molecule* section.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n The charge of the QM subsystem.

RECOMMENDATION:

This option only needs to be used if the QM subsystem (model system) has a charge that is different from the total system charge.

MODEL_SYSTEM_MULT

Specifies the QM subsystem multiplicity if different from the *$molecule* section.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n The multiplicity of the QM subsystem.

RECOMMENDATION:

This option only needs to be used if the QM subsystem (model system) has a multiplicity that is different from the total system multiplicity. ONIOM calculations must be closed shell.

MODE_COUPLING

Number of modes coupling in the third and fourth derivatives calculation.

TYPE:

INTEGER

DEFAULT:

2 for two modes coupling.

OPTIONS:

n for n modes coupling, Maximum value is 4.

RECOMMENDATION:

Use the default.

MOLDEN_FORMAT

Sets the output format of NTOs in RASCI2 SOC analysis to MOLDEN format.

TYPE:

Logical

DEFAULT:

False

OPTIONS:

True Append MOLDEN input file at the end of the Q-CHEM output file.

RECOMMENDATION:

Currently, SOC-NTO analysis in RASCI2 only works with MOLDEN. Other visualization tools are not supported at the moment. Please see the Visualizing Orbitals Using MOLDEN section for more information.

MOM_METHOD

Determines the target orbitals with which to maximize the overlap on each SCF cycle.

TYPE:

INTEGER

DEFAULT:

MOM

OPTIONS:

MOM Maximize overlap with the orbitals from the previous SCF cycle.

IMOM Maximize overlap with the initial guess orbitals.

RECOMMENDATION:

If appropriate guess orbitals can be obtained, then IMOM can provide more reliable convergence to the desired solution.[4]

MOM_PRINT

Switches printing on within the MOM procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Printing is turned off

TRUE Printing is turned on.

RECOMMENDATION:

None

MOM_START

Determines when MOM is switched on to preserve orbital occupancies.

TYPE:

INTEGER

DEFAULT:

0 (FALSE)

OPTIONS:

0 (FALSE) MOM is not used

n MOM begins on cycle n.

RECOMMENDATION:

For calculations on excited states, an initial calculation without MOM is usually required to get satisfactory starting orbitals. These orbitals should be read in setting SCF_GUESS = TRUE and MOM_START = 1.

MOPROP_CONV_1ST

Sets the convergence criteria for CPSCF and 1st order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$ Convergence threshold set to 10^{-n}.

RECOMMENDATION:

None

MOPROP_CONV_2ND

Sets the convergence criterion for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

$n < 10$ Convergence threshold set to 10^{-n}.

RECOMMENDATION:

None

MOPROP_DIIS_DIM_SS

Specified the DIIS subspace dimension.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

0 No DIIS.

n Use a subspace of dimension n.

RECOMMENDATION:

None

MOPROP_DIIS

Controls the use of Pulay's DIIS in solving the CPSCF equations.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

0 Turn off DIIS.

5 Turn on DIIS.

RECOMMENDATION:

None

MOPROP_ISSC_PRINT_REDUCED

Specifies whether the isotope-independent reduced coupling tensor $\mathbf{K}$ should be printed in addition to the isotope-dependent $\mathbf{J}$-tensor when calculating indirect nuclear spin-spin couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print $\mathbf{K}$.

TRUE Print $\mathbf{K}$.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_DSO

Specifies whether to skip the calculation of the diamagnetic spin-orbit contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate diamagnetic spin-orbit contribution.

TRUE Skip diamagnetic spin-orbit contribution.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_FC

Specifies whether to skip the calculation of the Fermi contact contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate Fermi contact contribution.

TRUE Skip Fermi contact contribution.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_PSO

Specifies whether to skip the calculation of the paramagnetic spin-orbit contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate paramagnetic spin-orbit contribution.

TRUE Skip paramagnetic spin-orbit contribution.

RECOMMENDATION:

None

MOPROP_ISSC_SKIP_SD

Specifies whether to skip the calculation of the spin-dipole contribution to the indirect nuclear spin-spin coupling tensor.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Calculate spin-dipole contribution.

TRUE Skip spin-dipole contribution.

RECOMMENDATION:

None

MOPROP_MAXITER_1ST

The maximum number of iterations for CPSCF and first-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n Set maximum number of iterations to n.

RECOMMENDATION:

Use the default.

MOPROP_MAXITER_2ND

The maximum number of iterations for second-order TDSCF.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n Set maximum number of iterations to n.

RECOMMENDATION:

Use the default.

MOPROP_PERTNUM

Set the number of perturbed densities that will to be treated together.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 All at once.

n Treat the perturbed densities batch-wise.

RECOMMENDATION:

Use the default. For large systems, limiting this number may be required to avoid memory exhaustion.

MOPROP_RESTART

Specifies the option for restarting MOProp calculations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Not a restart calculation.

1 Restart from a previous calculation using the same scratch directory.

RECOMMENDATION:

Need to also include "SCF_GUESS READ" and "SKIP_SCFMAN TRUE" to ensure the same set of MOs.

MOPROP

Specifies the job number for MOProp module.

TYPE:

STRING

DEFAULT:

0 Do not run the MOProp module.

OPTIONS:

NMR	NMR chemical shielding tensors.
STATIC_POLAR	Static polarizability.
ISSC	Indirect nuclear spin–spin coupling tensors.
DYN_POLAR	Dynamic polarizability.
HYPERPOL	First hyperpolarizability using Wigner's $2n + 1$ rule.

RECOMMENDATION:

None

MO_OVERLAPS_TWO_GEOMS

Specifies whether to compute molecular orbital overlaps at two different geometries.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute these overlaps.

1 Used to indicate the first of two required SCF calculations.

2 Used to indicate the second of two required SCF calculations.

RECOMMENDATION:

The atoms must be ordered in the same way for both calculations. (This is not checked.)

MP2_SCALING

Scales the RI-MP2 correlation energy contribution.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n corresponding to a scaling factor of $n/10^6$

RECOMMENDATION:

Use default.

MP3_SCALING

Scales the RI-MP3 correlation energy contribution.

TYPE:

INTEGER

DEFAULT:

1000000

OPTIONS:

n corresponding to a scaling factor of $n/10^6$

RECOMMENDATION:

Use default.

MRXC_CLASS_THRESH_MULT

Controls the of smoothness precision

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

im An integer

RECOMMENDATION:

A prefactor in the threshold for MRXC error control: $im \times 10^{-io}$

MRXC_CLASS_THRESH_ORDER

Controls the of smoothness precision

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

io An integer

RECOMMENDATION:

The exponent in the threshold of the MRXC error control: $im \times 10^{-io}$

MRXC

Controls the use of MRXC.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use MRXC

1 Use MRXC in the evaluation of the XC part

RECOMMENDATION:

MRXC is very efficient for medium and large molecules, especially when medium and large basis sets are used.

MULTIPOLE_ORDER

Determines highest order of multipole moments to print if wave function analysis requested.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n Calculate moments to nth order.

RECOMMENDATION:

Use the default unless higher multipoles are required.

NBOMEMORY

Controls how much memory is allocated to Q-CHEM's internal NBO (v. 5) package.

TYPE:

INTEGER

DEFAULT:

500000

OPTIONS:

N Allocate N double-precision words (= $8N$ bytes).

RECOMMENDATION:

This memory comes out of MEM_TOTAL.

NBO

Controls the use of the NBO package.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not invoke the NBO package.

1 Do invoke the NBO package, for the ground state.

2 Invoke the NBO package for the ground state, and also each CIS, RPA, or TDDFT excited state.

RECOMMENDATION:

None

NCORE_XES

Specifies how many core levels to use in a Koopmans-type XES calculation.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Compute transition dipoles corresponding to the first (lowest energy) n core orbitals, ϕ_c.

RECOMMENDATION:

None

NL_CORRELATION

Specifies a non-local correlation functional that includes non-empirical dispersion.

TYPE:

STRING

DEFAULT:

None No non-local correlation.

OPTIONS:

None	No non-local correlation
vdW-DF-04	the non-local part of vdW-DF-04
vdW-DF-10	the non-local part of vdW-DF-10 (also known as vdW-DF2)
VV09	the non-local part of VV09
VV10	the non-local part of VV10

RECOMMENDATION:

Do not forget to add the LSDA correlation (PW92 is recommended) when using vdW-DF-04, vdW-DF-10, or VV09. VV10 should be used with PBE correlation. Choose exchange functionals carefully: HF, rPW86, revPBE, and some of the LRC exchange functionals are among the recommended choices.

NL_GRID

Specifies the grid to use for non-local correlation.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

Same as for XC_GRID

RECOMMENDATION:

Use the default unless computational cost becomes prohibitive, in which case SG-0 may be used. XC_GRID should generally be finer than NL_GRID.

NL_VV_B

Sets the parameter b in VV10. This parameter controls the short range behavior of the non-local correlation energy.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $b = n/100$

RECOMMENDATION:

The optimal value depends strongly on the exchange functional used. $b = 5.9$ is recommended for rPW86. For further details see Ref. 41.

NL_VV_C

Sets the parameter C in VV09 and VV10. This parameter is fitted to asymptotic van der Waals C_6 coefficients.

TYPE:

INTEGER

DEFAULT:

89 for VV09

No default for VV10

OPTIONS:

n Corresponding to $C = n/10000$

RECOMMENDATION:

$C = 0.0093$ is recommended when a semi-local exchange functional is used. $C = 0.0089$ is recommended when a long-range corrected (LRC) hybrid functional is used. For further details see Ref. 41.

NMOL1

INTEGER

TYPE:

Sets the number of molecules of type 1.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

NMOL2

INTEGER

TYPE:

Sets the number of molecules of type 2.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

NN_THRESH

The distance cutoff for neighboring fragments (between which CT excitation occurs).

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not include interfragment transitions (ALMO-CIS/TDA)

n Include interfragment excitations between pairs of fragments the distances between whom are smaller than n a_0 (ALMO-CIS/TDA+CT)

RECOMMENDATION:

None

NOCIS

Requests a NOCIS/STEX/1C-NOCIS/EA-TDDFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not run these methods.

TRUE Run one of these methods, options controlled in *$nocis*.

RECOMMENDATION:

None

NOCI_PRINT

Specify the debug print level of NOCI.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n Positive integer

RECOMMENDATION:

Increase this for additional debug information.

NOSE_HOOVER_LENGTH

Sets the chain length for the Nosé-Hoover thermostat

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

n Chain length of n auxiliary variables

RECOMMENDATION:

Typically 3-6

NOSE_HOOVER_TIMESCALE

Sets the timescale (strength) of the Nosé-Hoover thermostat

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

n Thermostat timescale, as n fs

RECOMMENDATION:

Smaller values (roughly 100) equate to tighter thermostats but may inhibit rapid sampling. Larger values (≥ 1000) allow for more rapid sampling but may take longer to reach thermal equilibrium.

NO_REORIENT

Disable rotation into the standard nuclear orientation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use the standard nuclear orientation.

TRUE Use the coordinate system defined by the *$molecule* section.

RECOMMENDATION:

Use the default unless you have a reason to want a different coordinate system.

NSEARCH

INTEGER

TYPE:

Sets the number of structures that are generated and optimized.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

NTO_PAIRS

Controls the writing of hole/particle NTO pairs for SOC transitions calculated within the RASCI2 SOC analysis section.

TYPE:

Integer

DEFAULT:

0

OPTIONS:

N Write N NTO pairs per SOC transition.

RECOMMENDATION:

If activated ($N > 0$), a minimum of two NTO pairs will be printed for each transition. Increase the value of N if additional NTOs are desired. See Section 7.14.3 for information on visualizing NTOs.

NVAL_XES

Specifies how many valence virtual levels to use in a Koopmans-type XES calculation.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Compute transition dipoles corresponding to the highest n occupied orbitals, ϕ_v.

RECOMMENDATION:

Setting $n = 1$ will include the HOMO in the occupied space, $n = 2$ will include HOMO and HOMO $- 1$, etc.

NVO_LIN_CONVERGENCE

Target error factor in the preconditioned conjugate gradient solver of the single-excitation amplitude equations.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n User–defined number.

RECOMMENDATION:

Solution of the single-excitation amplitude equations is considered converged if the maximum residual is less than 10^{-n} multiplied by the current DIIS error. For the ARS correction, n is automatically set to 1 since the locally-projected DIIS error is normally several orders of magnitude smaller than the full DIIS error.

NVO_LIN_MAX_ITE

Maximum number of iterations in the preconditioned conjugate gradient solver of the single-excitation amplitude equations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User–defined number of iterations.

RECOMMENDATION:

None.

NVO_METHOD

Sets method to be used to converge solution of the single-excitation amplitude equations.

TYPE:

INTEGER

DEFAULT:

9

OPTIONS:

n User–defined number.

RECOMMENDATION:

This is an experimental option. Use the default.

NVO_TRUNCATE_DIST

Specifies which atomic blocks of the Fock matrix are used to construct the preconditioner.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

$n > 0$ If distance between a pair of atoms is more than n Ångstroms do not include the atomic block.

-2 Do not use distance threshold, use NVO_TRUNCATE_PRECOND instead.

-1 Include all blocks.

0 Include diagonal blocks only.

RECOMMENDATION:

This option does not affect the final result. However, it affects the rate of the PCG algorithm convergence. For small systems, use the default.

NVO_TRUNCATE_PRECOND

Specifies which atomic blocks of the Fock matrix are used to construct the preconditioner. This variable is used only if NVO_TRUNCATE_DIST is set to -2.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n If the maximum element in an atomic block is less than 10^{-n} do not include the block.

RECOMMENDATION:

Use the default. Increasing n improves convergence of the PCG algorithm but overall may slow down calculations.

NVO_UVV_MAXPWR

Controls convergence of the Taylor series when calculating the U_{vv} block from the single-excitation amplitudes. If the series is not converged at the nth term, more expensive direct inversion is used to calculate the U_{vv} block.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n User–defined number.

RECOMMENDATION:

None.

NVO_UVV_PRECISION

Controls convergence of the Taylor series when calculating the U_{vv} block from the single-excitation amplitudes. Series is considered converged when the maximum element of the term is less than 10^{-n}.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

n User–defined number.

RECOMMENDATION:

NVO_UVV_PRECISION must be the same as or larger than THRESH.

N_ATOM_TYPE_1

INTEGER

TYPE:

Sets the number atoms in molecule type 1.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

N_ATOM_TYPE_2

INTEGER

TYPE:

Sets the number atoms in molecule type 2.

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

None

N_FC_CVS_INACTIVE

Number of frozen-core CVS inactive orbitals

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n, where $0 \leq$ n $\leq$ total FC orbitals

RECOMMENDATION:

Useful in cluster calculations.

N_FROZEN_CORE

Sets the number of frozen core orbitals in a post-Hartree–Fock calculation.

TYPE:

INTEGER

DEFAULT:

FC

OPTIONS:

FC Frozen Core approximation (all core orbitals frozen).

n Freeze n core orbitals (if set to 0, all electrons will be active).

RECOMMENDATION:

Correlated calculations calculations are more efficient with frozen core orbitals. Use default if possible.

N_FROZEN_VIRTUAL

Sets the number of frozen virtual orbitals in a post-Hartree–Fock calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Freeze n virtual orbitals.

RECOMMENDATION:

None

N_I_SERIES

Sets summation limit for series expansion evaluation of $i_n(x)$.

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

N_J_SERIES

Sets summation limit for series expansion evaluation of $j_n(x)$.

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

$n > 0$

RECOMMENDATION:

Lower values speed up the calculation, but may affect accuracy.

N_MOL_TYPE

INTEGER

TYPE:

Sets the number of different atom or molecule types.

DEFAULT:

No default.

OPTIONS:

User defined : can be 1 or 2.

RECOMMENDATION:

None

N_MOVES

INTEGER

TYPE:

Sets the number of structural changes/moves on each step.

DEFAULT:

2

OPTIONS:

User defined.

RECOMMENDATION:

None

N_SOL

Specifies number of atoms or orbitals in the *$solute* or *$alist* section.

TYPE:

INTEGER

DEFAULT:

No default.

OPTIONS:

User defined.

RECOMMENDATION:

Reads from either the *$solute* or *$alist* input section.

N_SWOP

INTEGER

TYPE:

Sets the number atom coordinate swops for atomic cluster search.

DEFAULT:

No default.

OPTIONS:

User defined

RECOMMENDATION:

None

N_WIG_SERIES

Sets summation limit for Wigner integrals.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

$n < 100$

RECOMMENDATION:

Increase n for greater accuracy.

OCCUPATIONS

Activates pFON calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Integer occupation numbers

1 Not yet implemented

2 Pseudo-fractional occupation numbers (pFON)

RECOMMENDATION:

Use pFON to improve convergence for small-gap systems.

OCC_RI_K

Controls the use of the occ-RI-K approximation for constructing the exchange matrix

TYPE:

LOGICAL

DEFAULT:

False Do not use occ-RI-K.

OPTIONS:

True Use occ-RI-K.

RECOMMENDATION:

Larger the system, better the performance

OMEGA2

Sets the Coulomb attenuation parameter for the long-range component.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega_2 = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

OMEGA_GDD_SCALING

Sets the empirical constant C in the GDD tuning procedure.

TYPE:

INTEGER

DEFAULT:

885

OPTIONS:

n Corresponding to $C = n/1000$.

RECOMMENDATION:

The quantity $C = 0.885$ ($n = 885$) was determined in Ref. 23 using LRC-ωPBE/hpTZVPP calculations.

OMEGA_GDD

Controls the application of ω_{GDD} tuning for long-range corrected DFT

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply GDD tuning.

TRUE (or 1) Use GDD tuning.

RECOMMENDATION:

The *$rem* variable OMEGA must also be specified, in order to set the initial range-separation parameter.

OMEGA

Sets the range-separation parameter, ω, also known as μ, in functionals based on Hirao's RSH scheme.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

OMEGA

Sets the Coulomb attenuation parameter for the short-range component.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

OPSING

Controls whether approximate spin purification will be performed according to Eq. (7.68).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Spin purification is not performed.

TRUE Spin purification will be performed.

RECOMMENDATION:

Set to TRUE if spin purification calculation is desired. Make sure that UNRESTRICTED is set to TRUE Gradients are available.

ORBITAL_ENERGY_PREC

Obtain additional digits of precision in the orbital energies.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Standard printout with 4 decimal digits (in Hartree).

1 5 decimal digits.

2 Scientific notation with 10 digits of precision.

RECOMMENDATION:

Set as desired. Has no effect in cases where the orbital symmetry labels are printed; use SYM_IGNORE = TRUE if additional precision is needed in such cases.

OSLO

Triggers OSLO procedure after a converged SCF

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Don't perform OSLO

1 Perform the OSLO procedure

RECOMMENDATION:

None

OS_ROSCF

Run an open-shell singlet ROSCF calculation with GEN_SCFMAN.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE OS_ROSCF calculation is performed.

FALSE Do not run OS_ROSCF (it will run a closed-shell RSCF calculation instead).

RECOMMENDATION:

Set to TRUE if desired.

PAO_ALGORITHM

Algorithm used to optimize polarized atomic orbitals (see PAO_METHOD)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use efficient (and riskier) strategy to converge PAOs.

1 Use conservative (and slower) strategy to converge PAOs.

RECOMMENDATION:

None

PAO_METHOD

Controls the type of PAO calculations requested.

TYPE:

STRING

DEFAULT:

EPAO For local MP2, EPAOs are chosen by default.

OPTIONS:

EPAO Find EPAOs by minimizing delocalization function.

PAO Do SCF in a molecule-optimized minimal basis.

RECOMMENDATION:

None

PARI_K

Controls the use of the PARI-K approximation in the construction of the exchange matrix

TYPE:

LOGICAL

DEFAULT:

FALSE Do not use PARI-K.

OPTIONS:

TRUE Use PARI-K.

RECOMMENDATION:

Use for basis sets aug-cc-pVTZ and larger.

PDFT_CORRELATION

Specifies the correlation functional to be used in MC-PDFT calculation.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

NAME Use PDFT_CORRELATION = *NAME*, where *NAME* is one of the LDA or GGA correlation functionals listed in Section 5.3.4. This keyword is only invoked when method is set to RDM(PDFT).

RECOMMENDATION:

In general, consult the literature to guide your selection.

PDFT_EXCHANGE

Specifies the exchange functional to be used in MC-PDFT calculation.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use PDFT_EXCHANGE = *NAME*, where *NAME* must be one of the LDA or GGA exchange functionals listed in Section 5.3.3. This keyword is only invoked when method is set to RDM(PDFT).

RECOMMENDATION:

In general, consult the literature to guide your selection.

PEQS_SWITCH

Inclusion of solvent effects begins when the SCF error falls below $10^{-\text{PEQS\_SWITCH}}$.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

Use the default unless solvent effects need to be incorporated earlier in the SCF procedure.

PE

Turns PE on.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Perform a PE calculation.

False Don't perform a PE calculation.

RECOMMENDATION:

Set the *$rem* variable PE to TRUE to start a PE calculation.

PHESS

Controls whether partial Hessian calculations are performed.

TYPE:

INTEGER

DEFAULT:

0 Full Hessian calculation

OPTIONS:

1 Partial Hessian calculation.

2 Vibrational subsystem analysis (massless).

3 Vibrational subsystem analysis (weighted).

RECOMMENDATION:

None

PH_FAST

Lowers integral cutoff in partial Hessian calculation is performed.

TYPE:

LOGICAL

DEFAULT:

FALSE Use default cutoffs

OPTIONS:

TRUE Lower integral cutoffs

RECOMMENDATION:

None

PIMC_ACCEPT_RATE

Acceptance rate for MC/PIMC simulations when Cartesian or normal-mode displacements are used.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

$0 < n < 100$ User-specified rate, given as a whole-number percentage.

RECOMMENDATION:

Choose acceptance rate to maximize sampling efficiency, which is typically signified by the mean-square displacement (printed in the job output). Note that the maximum displacement is adjusted during the warm-up run to achieve roughly this acceptance rate.

PIMC_MCMAX

Number of Monte Carlo steps to sample.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified number of steps to sample.

RECOMMENDATION:

This variable dictates the statistical convergence of MC/PIMC simulations. For converged simulations at least 10^5 steps is recommended.

PIMC_MOVETYPE

Selects the type of displacements used in MC/PIMC simulations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Cartesian displacements of all beads, with occasional (1%) center-of-mass moves.

1 Normal-mode displacements of all modes, with occasional (1%) center-of-mass moves.

2 Levy flights without center-of-mass moves.

RECOMMENDATION:

Except for classical sampling (MC) or small bead-number quantum sampling (PIMC), Levy flights should be used. For Cartesian and normal-mode moves, the maximum displacement is adjusted during the warm-up run to the desired acceptance rate (controlled by PIMC_ACCEPT_RATE). For Levy flights, the acceptance is solely controlled by PIMC_SNIP_LENGTH.

PIMC_NBEADSPERATOM

Number of path integral time slices ("beads") used on each atom of a PIMC simulation.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

1 Perform classical Boltzmann sampling.

>1 Perform quantum-mechanical path integral sampling.

RECOMMENDATION:

This variable controls the inherent convergence of the path integral simulation. The one-bead limit represents classical sampling and the infinite-bead limit represents exact quantum-mechanical sampling. Using 32 beads is reasonably converged for room-temperature simulations of molecular systems.

PIMC_SNIP_LENGTH

Number of "beads" to use in the Levy flight movement of the ring polymer.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

$3 \leq n \leq$ PIMC_NBEADSPERATOM User-specified length of snippet.

RECOMMENDATION:

Choose the snip length to maximize sampling efficiency. The efficiency can be estimated by the mean-square displacement between configurations, printed at the end of the output file. This efficiency will typically, however, be a trade-off between the mean-square displacement (length of statistical correlations) and the number of beads moved. Only the moved beads require recomputing the potential, *i.e.*, a call to Q-CHEM for the electronic energy. (Note that the endpoints of the snippet remain fixed during a single move, so $n - 2$ beads are actually moved for a snip length of n. For 1 or 2 beads in the simulation, Cartesian moves should be used instead.)

PIMC_TEMP

Temperature, in Kelvin (K), of path integral simulations.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified number of Kelvin for PIMC or classical MC simulations.

RECOMMENDATION:

None.

PIMC_WARMUP_MCMAX

Number of Monte Carlo steps to sample during an equilibration period of MC/PIMC simulations.

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified number of steps to sample.

RECOMMENDATION:

Use this variable to equilibrate the molecule/ring polymer before collecting production statistics. Usually a short run of roughly 10% of PIMC_MCMAX is sufficient.

PLOT_SPIN_DENSITY

Requests the generation of spin densities, ρ_α and ρ_β.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not generate spin density cube files.

TRUE Generate spin density cube files.

RECOMMENDATION:

Set to TRUE if spin densities are desired in addition to total densities. Requires that MAKE_CUBE_FILES be set to TRUE as well, and that one or more total densities is requested in the *$plots* input section. The corresponding spin densities will then be generated also.

POINT_GROUP_SYMMETRY

Controls whether or not Q-CHEM determines the point group of the molecule and reorients the molecule to the standard orientation.

TYPE:

LOGICAL

DEFAULT:

TRUE Do determine the point group (disabled for RIMP2 jobs).

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Use the default unless you do not want the molecule to be reoriented. Note that symmetry usage is disabled for RIMP2 jobs.

POL_GEOM

Compute forces on the polarized (converged SCF-MI) potential surface.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

FALSE Do not compute forces on the polarized potential surface.

TRUE Compute forces on the polarized potential surface.

RECOMMENDATION:

Set it to TRUE when optimized geometry or vibrational frequencies on the polarized potential surface are desired.

POP_BECKE

Controls the printing of atomic Becke populations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print atomic Becke populations.

FALSE Do not print atomic Becke populations.

RECOMMENDATION:

None

POP_MULLIKEN

Controls running of Mulliken population analysis.

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE (or 1)

OPTIONS:

FALSE (or 0)	Do not calculate Mulliken populations.
TRUE (or 1)	Calculate Mulliken populations.
2	Also calculate shell populations for each occupied orbital.
3	Same output as 2 and also orbital densities at the nuclear centers.
-1	Calculate Mulliken charges for both the ground state and any CIS, RPA, or TDDFT excited states.

RECOMMENDATION:

Leave as TRUE, unless excited-state charges are desired. Mulliken analysis is a trivial additional calculation, for ground or excited states.

PRINT_CORE_CHARACTER

Determines the print level for the CORE_CHARACTER option.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No additional output is printed.

1 Prints core characters of occupied MOs.

2 Print level 1, plus prints the core character of AOs.

RECOMMENDATION:

Use the default, unless you are uncertain about what the core character is.

PRINT_DIST_MATRIX

Controls the printing of the inter-atomic distance matrix

TYPE:

INTEGER

DEFAULT:

15

OPTIONS:

0 Turns off the printing of the distance matrix

n Prints the distance matrix if the number of atoms in the molecule is less than or equal to n.

RECOMMENDATION:

Use default unless distances are required for large systems

PRINT_GENERAL_BASIS

Controls print out of built in basis sets in input format

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print out standard basis set information

FALSE Do not print out standard basis set information

RECOMMENDATION:

Useful for modification of standard basis sets.

PRINT_INPUT

Specifies whether to echo the input file in the output.

TYPE:

INTEGER

DEFAULT:

TRUE

OPTIONS:

TRUE Print the Q-CHEM input file as part of the Q-CHEM output file.

FALSE Do not repeat the input file.

RECOMMENDATION:

This is useful for record-keeping and defaults to TRUE except for QM/MM jobs, where the default value is FALSE due to the size of some input files.

PRINT_ORBITALS

Prints orbital coefficients with atom labels in analysis part of output.

TYPE:

INTEGER/LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not print any orbitals.

TRUE Prints occupied orbitals plus 5 virtual orbitals.

NVIRT Number of virtual orbitals to print.

RECOMMENDATION:

Use true unless more virtual orbitals are desired.

PRINT_QIS

Requests to dump stuff needed for OpenFermion.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print stuff for QIS in user directory.

RECOMMENDATION:

Beware of size of the files.

PRINT_RADII_GYRE

Controls printing of MO centroids and radii of gyration.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

TRUE (or 1) Print the centroid and radius of gyration for each occupied MO and each density.

2 Print centroids and radii of gyration for the virtual MOs as well.

FALSE (or 0) Do not calculate these quantities.

RECOMMENDATION:

None

PROJ_TRANSROT

Removes translational and rotational drift during AIMD trajectories.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not apply translation/rotation corrections.

TRUE Apply translation/rotation corrections.

RECOMMENDATION:

When computing spectra (see AIMD_NUCL_DACF_POINTS, for example), this option can be used to remove artificial, contaminating peaks stemming from translational and/or rotational motion. Recommend setting to TRUE for all dynamics-based spectral simulations.

PSEUDO_CANONICAL

When SCF_ALGORITHM = DM, this controls the way the initial step, and steps after subspace resets are taken.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use Roothaan steps when (re)initializing

TRUE Use a steepest descent step when (re)initializing

RECOMMENDATION:

The default is usually more efficient, but choosing TRUE sometimes avoids problems with orbital reordering.

PURECART

INTEGER

TYPE:

Controls the use of pure (spherical harmonic) or Cartesian angular forms

DEFAULT:

1111 Pure h, g, f, d functions

OPTIONS:

$hgfd$ Use 1 for pure and 2 for Cartesian.

RECOMMENDATION:

This is pre-defined for all standard basis sets

QCMD_INITNUC

Specifies the distribution used when sampling initial nuclear positions and velocities.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

Wigner Wigner distribution.

Boltzmann Boltzmann distribution

RECOMMENDATION:

Used in conjunction with AIMD_TEMP.

QCMD_INITSTATE

Specifies the initially populated electronic state.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

n An integer set less than CIS_N_ROOTS.

RECOMMENDATION:

None

QCMD_METHOD

Specifies the nonadiabatic Meyer-Miller molecular dynamics method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

Ehrenfest Traditional Ehrenfest molecular dynamics.

SQC Symmetric Quasi-Classical Meyer-Miller molecular dynamics

RECOMMENDATION:

None

QCMD_WARMUP

Specifies the number of linearly-interpolated steps between the initial and sampled configurations for accurate state following before the dynamics begin.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n

RECOMMENDATION:

None

QMMM_CHARGES

Controls the printing of QM charges to file.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Writes a charges.dat file with the Mulliken charges from the QM region.

FALSE No file written.

RECOMMENDATION:

Use the default unless running calculations with CHARMM where charges on the QM region need to be saved.

QMMM_FULL_HESSIAN

Trigger the evaluation of the full QM/MM Hessian.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Evaluates full Hessian.

FALSE Hessian for QM-QM block only.

RECOMMENDATION:

None

QMMM_PRINT

Controls the amount of output printed from a QM/MM job.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Limit molecule, point charge, and analysis printing.

FALSE Normal printing.

RECOMMENDATION:

Use the default unless running calculations with CHARMM.

QM_MM_INTERFACE

Enables internal QM/MM calculations.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

MM Molecular mechanics calculation (*i.e.*, no QM region)

ONIOM QM/MM calculation using two-layer mechanical embedding

JANUS QM/MM calculation using electronic embedding

RECOMMENDATION:

The ONIOM model and Janus models are described above. Choosing MM leads to no electronic structure calculation. However, when using MM, one still needs to define the *$rem* variables BASIS and EXCHANGE in order for Q-CHEM to proceed smoothly.

QM_MM

Turns on the Q-CHEM/CHARMM interface.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do QM/MM calculation through the Q-CHEM/CHARMM interface.

FALSE Turn this feature off.

RECOMMENDATION:

Use the default unless running calculations with CHARMM.

QRRHO_ALPHA

Specifies the exponent in the damping function of Chai and Head-Gordon, used for interpolating the vibrational enthalpy and entropy in the qRRHO scheme. Specify MRRHO_ALPHA to change the exponent for the entropy interpolation separately.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

α Dimensionless interpolator exponent used in the qRRHO scheme.

RECOMMENDATION:

Use the default.

QRRHO_OMEGA_CUTOFF

Sets the frequency cutoff in the Chai-Head-Gordon damping function for interpolating the vibrational enthalpy and entropy in the qRRHO scheme. Specify MRRHO_OMEGA_CUTOFF to change the frequency cutoff for the entropy interpolation separately.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

ω_0 Interpolator cutoff frequency used in the qRRHO scheme in cm^{-1}.

RECOMMENDATION:

Use the default.

RASSF_DELTA_ALPHA

Sets the number of alpha electrons to remove relative to the reference.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

0 Remove no alpha electrons (use for EA)

1 Remove one alpha electron (use for 1SF, IP)

2 Remove two alpha electrons (use for 2SF, 1SF-IP)

RECOMMENDATION:

None.

RASSF_DELTA_BETA

Sets the number of beta electrons to add relative to the reference.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

0 Add no beta electrons (use for IP)

1 Add one beta electron (use for 1SF, EA)

2 Add two beta electrons (use for 2SF, 1SF-EA)

RECOMMENDATION:

None.

RASSF_DO_BLOCH

Determines whether to do effective Hamiltonian analysis.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Skip analysis

1 Do effective Hamiltonian analysis

RECOMMENDATION:

None.

RASSF_GUESS

Determines which initial set of guess vectors to use for Davidson.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

0 Random orthonormal guess (default for CAS)

1 Identity guess

2 CAS guess (default for RAS)

RECOMMENDATION:

Starting from a CAS guess is recommended for larger molecules. If Davidson encounters issues with linearly dependent eigenvectors, consider using identity. The random orthonormal guess requires building a large NxN matrix and is therefore only recommended for calculations with fewer determinants.

RASSF_WRITE_EVALS

Determines whether to write eigenvalues to an output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not write eigenvalues to an output file

1 Write eigenvalues to an output file

RECOMMENDATION:

None.

RASSF_WRITE_EVECS

Determines whether to write eigenvectors to an output file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not write eigenvectors to an output file

1 Write eigenvectors to an output file

RECOMMENDATION:

None.

RAS_ACT_DIFF

Sets the number of α versus β electrons and therefore controls the level of excitations used in calculations.

TYPE:

Integer

DEFAULT:

None

OPTIONS:

1	odd number of electrons or cations
0	even number of electrons
-1	anions
n	$n < -99$ triggers RAS2-SF at DDCI level of excitations $n = -451$ and $n = -452$ triggers restart mechanism that restores the last best guess for each state to the number of states requested

RECOMMENDATION:

Set to 0 would be appropriate for most singlet systems. Only works with RASCI2.

RAS_ACT_OCC

Sets the number of occupied orbitals to enter the RAS active space.

TYPE:

Integer

DEFAULT:

None

OPTIONS:

n	user defined integer

RECOMMENDATION:

None. Only works with RASCI2

RAS_ACT_ORB

Sets the user-selected active orbitals (RAS2 orbitals).

TYPE:

INTEGER ARRAY

DEFAULT:

From RAS_OCC + 1 to RAS_OCC + RAS_ACT

OPTIONS:

$[i, j, k...]$	The number of orbitals must be equal to the RAS_ACT variable

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ACT_VIR

Sets the number of virtual orbitals to enter the RAS active space.

TYPE:

Integer

DEFAULT:

None

OPTIONS:

n user defined integer

RECOMMENDATION:

None. Only works with RASCI2.

RAS_ACT

Sets the number of orbitals in RAS2 (active orbitals).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_AMPL_PRINT

Defines the absolute threshold ($\times 10^2$) for the CI amplitudes to be printed.

TYPE:

INTEGER

DEFAULT:

10 0.1 minimum absolute amplitude

OPTIONS:

n User-defined integer, $n \geq 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_CALC_SOC

Controls whether to calculate the SOC constants for RAS2 jobs only.

TYPE:

Integer/Logical

DEFAULT:

False

OPTIONS:

False Do not perform the SOC calculation.

True Perform the SOC calculation.

RECOMMENDATION:

This *$rem* variable is used to control the spin-orbit coupling (SOC) analysis section.

RAS_DO_HOLE

Controls the presence of hole excitations in the RAS-CI wave function.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Include hole configurations (RAS1 to RAS2 excitations)

FALSE Do not include hole configurations

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_DO_PART

Controls the presence of particle excitations in the RAS-CI wave function.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Include particle configurations (RAS2 to RAS3 excitations)

FALSE Do not include particle configurations

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ELEC_ALPHA

Sets the number of spin-α electrons in RAS2 (active electrons).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ELEC_BETA

Sets the number of spin-β electrons in RAS2 (active electrons).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ELEC

Sets the number of electrons in RAS2 (active electrons).

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_FRAG_SETS

Defines the number of orbitals in each disjoint set to perform orbital localization.

TYPE:

INTEGER ARRAY

DEFAULT:

[NOcc,NAct,NVir] Number of orbitals within RAS1, RAS2 and RAS3 spaces

OPTIONS:

$[i, j, k...]$ Defines sets of canonical MOs to be localized into n fragments

RECOMMENDATION:

Setting within RAS1, RAS2 and RAS3 spaces alleviates the computational cost of the localization procedure. It might also result in improved fragment orbitals. Only works with RAS-CI.

RAS_GUESS_CS

Controls the number of closed shell guess configurations in RAS-CI.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Imposes to start with n closed shell guesses

RECOMMENDATION:

Only relevant for the computation of singlet states. Only works with RAS-CI.

RAS_NATORB_STATE

Saves the natural orbitals of the ith RAS-CI computed state into the `.fchk` file.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

i Saves the natural orbitals for the ith state

RECOMMENDATION:

None. Only works with RAS-CI and if GUI = 2.

RAS_NATORB

Controls the computation of the natural orbital occupancies.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute natural orbital occupancies for all states

FALSE Do not compute natural orbital occupancies

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_NFRAG_ATOMS

Sets the number of atoms in each fragment.

TYPE:

INTEGER ARRAY

DEFAULT:

None

OPTIONS:

$[i, j, k...]$ The sum of the numbers must be equal to the total number of atoms in the systems

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_NFRAG

If $n > 0$ activates the excitation analysis in RAS-CI

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Number of fragments to be considered

RECOMMENDATION:

Only for RAS-CI. The printed information level is controlled by RAS_PRINT.

RAS_N_ROOTS

Sets the number of RAS-CI roots to be computed.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n $n > 0$ Compute n RAS-CI states

RECOMMENDATION:

None. Only works with RASCI2

RAS_N_SPIN_FLIP

Sets the number of spin-flips.

TYPE:

INTEGER

DEFAULT:

Maximum number of spin-flips ($n = (\alpha - \beta)/2$)

OPTIONS:

n Do n spin-flips

RECOMMENDATION:

None.

RAS_OCC

Sets the number of orbitals in RAS1

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined integer, $n > 0$

RECOMMENDATION:

These are the initial doubly occupied orbitals (RAS1) before including $hole$ type of excitations. The RAS1 space starts from the lowest orbital up to RAS_OCC, *i.e.* no frozen orbitals option available yet. Only works with RAS-CI.

RAS_OMEGA

Sets the Coulomb range-separation parameter within the RAS-CI-srDFT method.

TYPE:

INTEGER

DEFAULT:

400 ($\omega = 0.4$ bohr^{-1})

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None. Range-separation parameter is typical indicated by ω or μ. Only works with RAS-CI.

RAS_PCM_INIT

Triggers state-specific PCM solvation within RAS-SF.

TYPE:

LOGICAL

DEFAULT:

FALSE no solvation (vacuum boundary conditions)

OPTIONS:

TRUE include solvation

RECOMMENDATION:

This triggers the nonequilibrium version of state-specific solvation. Additional keywords are necessary for the equilibrium version.

RAS_PT2_PARTITION

Specifies the partitioning scheme in RASCI(2)

TYPE:

INTEGER

DEFAULT:

1 Davidson-Kapuy (DK) partitioning

OPTIONS:

2 Epstein-Nesbet (EN) partitioning

0 Do both DK and EN partitionings

RECOMMENDATION:

Only for RAS-CI if RAS_PT2 is set to true.

RAS_PT2_VSHIFT

Defines the energy level shift ($\times 10^3$ au) in RASCI(2)

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n User-defined integer

RECOMMENDATION:

Only for RAS-CI if RAS_PT2 is set to true.

RAS_PT2

Perform second-order perturbative correction to RAS-CI energy

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute RASCI(2) energy corrections

FALSE Do not compute RASCI(2) energy corrections

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_ROOTS

Sets the number of RAS-CI roots to be computed.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

n $n > 0$ Compute n RAS-CI states

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_SOC_2E

Controls whether to compute two-electron mean-field contribution to RAS-CI SOC.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not compute two-electron mean-field contribution.

TRUE Compute two-electron mean-field contribution.

RECOMMENDATION:

None.

RAS_SOC_SYM_DENS

Controls whether to perform averaging of α and β densities.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not average α and β densities .

TRUE Average α and β densities.

RECOMMENDATION:

None.

RAS_SPIN_MULT

Specifies the spin multiplicity of the roots to be computed

TYPE:

INTEGER

DEFAULT:

1 Singlet states

OPTIONS:

0 Compute any spin multiplicity

$2n + 1$ User-defined integer, $n \geq 0$

RECOMMENDATION:

RAS_SPIN_MULT option is only available for $M_S = 0$ systems, that is, with the same number of α and β electrons.

RAS_SRDFT_COR

Define short-range correlation functional

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use RAS_SRDFT_COR = *NAME*, where *NAME* is one of the short-range correlation functionals listed in Section 5.3.4

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_SRDFT_DAMP

Sets damping factor ($\alpha < 1$) in the RAS-CI-srDFT method.

TYPE:

REAL

DEFAULT:

0.5 ($\alpha = 0.5$)

OPTIONS:

α Damping factor $0 < \alpha < 1$

RECOMMENDATION:

Modify in case of convergence issues along the RAS-CI-srDFT iterations. Only works with RAS-CI

RAS_SRDFT_EXC

Define short-range exchange functional

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use RAS_SRDFT_EXC = *NAME*, where *NAME* is one of the short-range exchange functionals listed in Section 5.3.3

RECOMMENDATION:

None. Only works with RAS-CI.

RAS_SRDFT_SA_ROOTS

Sets the list of roots used to build the state averaged reference density in RAS-CI-srDFT.

TYPE:

INTEGER ARRAY

DEFAULT:

All computed states

OPTIONS:

$[i, j, k \ldots]$ List of states.

RECOMMENDATION:

None. Only works with RAS-CI

RAS_SRDFT

Perform short-range density functional RAS-CI calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Compute RASCI-srDFT states and energies

FALSE Do not perform a RASCI-srDFT calculation

RECOMMENDATION:

None. Only works with RAS-CI. RAS_SRDFT_EXC and RAS_SRDFT_COR need to be set.

RATTLE_MAXIT

Specifies the maximum number of iterations in the RATTLE steps.

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

User-defined

RECOMMENDATION:

Use the default unless it does not get converged.

RATTLE_THRESH

Specifies the threshold for the convergence in the RATTLE steps.

TYPE:

INTEGER

DEFAULT:

6

OPTIONS:

n 10^{-n} threshold.

RECOMMENDATION:

Use the default

RCA_PRINT

Controls the output from RCA SCF optimizations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No print out

1 RCA summary information

2 Level 1 plus RCA coefficients

3 Level 2 plus RCA iteration details

RECOMMENDATION:

None

RC_R0

Determines the parameter in the Gaussian weight function used to smooth the density at the nuclei.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Corresponds the traditional delta function spin and charge densities

n corresponding to $n \times 10^{-3}$ a.u.

RECOMMENDATION:

We recommend value of 250 for a typical spit valence basis. For basis sets with increased flexibility in the nuclear vicinity the smaller values of r_0 also yield adequate spin density.

RDM_CG_CONVERGENCE

The minimum threshold for the conjugate gradient solver.

TYPE:

INTEGER

DEFAULT:

12

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

Should be at least (RDM_EPS_CONVERGENCE+2).

RDM_CG_MAXITER

Maximum number of iterations for each conjugate gradient computations in the BPSDP algorithm.

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

$N > 0$

RECOMMENDATION:

Use default unless problems arise.

RDM_CONSTRAIN_SPIN

Indicates if the spin-constraints are enforced.

TYPE:

BOOLEAN

DEFAULT:

TRUE

OPTIONS:

TRUE Enforce spin-constraints.

FALSE Do not enforce spin-constraints.

RECOMMENDATION:

Use default.

RDM_DIAGONALIZER

The algorithm used to diagonalize matrices inside semidefinite programming.

TYPE:

INTEGER

DEFAULT:

11

OPTIONS:

0 Use parallel LAPACK function DSYEV

1 Use parallel LAPACK function DSYEVD

10 Use multiple simultaneous calls to serial LAPACK function DSYEV

11 Use multiple simultaneous calls to serial LAPACK function DSYEVD

RECOMMENDATION:

Use default. Under certain circumstances (*e.g.*, low symmetry), algorithm 1 may be faster.

RDM_EPS_CONVERGENCE

The threshold for the error in the primal and dual constraints.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

Increase for gradient computations.

RDM_E_CONVERGENCE

The threshold for the primal-dual energy gap.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

N for a threshold of 10^{-N}

RECOMMENDATION:

Increase for gradient computations.

RDM_MAXITER

Maximum number of diagonalization steps in the BPSDP solver.

TYPE:

INTEGER

DEFAULT:

50000

OPTIONS:

$N > 0$

RECOMMENDATION:

Increase for computations that are difficult to converge.

RDM_MU_UPDATE_FREQUENCY

The number of v2RDM iterations after which the penalty parameter μ is updated.

TYPE:

INTEGER

DEFAULT:

200

OPTIONS:

$N > 0$

RECOMMENDATION:

Change if convergence problems arise.

RDM_ORBOPT_ENERGY_CONVERGENCE

The threshold for energy convergence during orbital optimization.

TYPE:

INTEGER

DEFAULT:

8

OPTIONS:

N for threshold of 10^{-N}

RECOMMENDATION:

Tighten for gradient computations.

RDM_ORBOPT_FREQUENCY

The number of v2RDM iterations after which the orbital optimization routine is called.

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

$N > 0$

RECOMMENDATION:

Use default unless convergence problems arise.

RDM_ORBOPT_GRADIENT_CONVERGENCE

The threshold for the orbital gradient during orbital optimization.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

N for threshold of 10^{-N}

RECOMMENDATION:

Tighten for gradient computations.

RDM_ORBOPT_MAXITER

The maximum number of orbital optimization steps each time the orbital optimization routine is called.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

$N > 0$

RECOMMENDATION:

Use default unless convergence problems arise.

RDM_POSITIVITY

Indicates positivity conditions enforced in the v2RDM optimization.

TYPE:

STRING

DEFAULT:

DQG

OPTIONS:

DQG,	Two-electron conditions
DQGT1	Two-electron conditions plus the T1 partial three-electron conditions
DQGT2	Two-electron conditions plus the T2 partial three-electron conditions
DQGT1T2	Two-electron conditions plus the T1 and T2 partial three-electron conditions
DQG3POS	Two-electron conditions plus the full three-electron conditions

RECOMMENDATION:

For high-accuracy, use DQG3POS or DQGT2, although such computations become impractical for large active spaces. For large active spaces (*e.g.*, n > 16 for CAS(n, n)), use DQG.

RDM_PRINT

Controls the amount of printing.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Print minimal information.
1	Print information about all iterations.

RECOMMENDATION:

Use 1 to analyze convergence issues.

RDM_TAU

Step-length parameter used in the BPSDP solver.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

N	for a value of 0.1 * N

RECOMMENDATION:

RDM_TAU should range between 10 and 16 for $1.0 \leq \tau \leq 1.6$.

RDM_TPDM_GUESS

Initial guess for the RDMs

TYPE:

STRING

DEFAULT:

HF_GUESS

OPTIONS:

HF_GUESS	Use RDMs from Hartree-Fock calculations as the initial density for the semidefinite solver
RANDOM_GUESS	Use random numbers as the initial density for the semidefinite solver

RECOMMENDATION:

Use default unless convergence problems arise.

REL_SHIFT

Corrects the calculated TDDFT excitation energy for scalar relativistic effects.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

Z Corresponding to the atomic number of the core-ionized element.

RECOMMENDATION:

The relativistic correction is equal to the difference of relativistic (Douglas-Kroll-Hess) and non-relativistic Hartree-Fock/cc-pCVTZ eigenvalues for the $1s$ orbital of the isolated atom.

RESPONSE

Activate the general response property module.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0)	Don't activate the general response property module.
TRUE (or 1)	Activate the general response property module.

RECOMMENDATION:

None.

RESP_CHARGES

Controls the calculations of RESP charges, where chemically equivalent atoms are restricted to have the same atomic charge value.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 Use Lebedev grid points around each atom.

2 Use spherical harmonics grid points around each atom.

RECOMMENDATION:

NONE

RI_J

Toggles the use of the RI algorithm to compute J.

TYPE:

LOGICAL

DEFAULT:

FALSE RI will not be used to compute J.

OPTIONS:

TRUE Turn on RI for J.

RECOMMENDATION:

For large (especially 1D and 2D) molecules the approximation may yield significant improvements in Fock evaluation time when used with ARI.

RI_K_GRAD

Turn on the nuclear gradient calculations

TYPE:

LOGICAL

DEFAULT:

FALSE Do not invoke occ-RI-K based gradient

OPTIONS:

TRUE Use occ-RI-K based gradient

RECOMMENDATION:

Use "RI_J false"

RI_K

Toggles the use of the RI algorithm to compute K.

TYPE:

LOGICAL

DEFAULT:

FALSE RI will not be used to compute K.

OPTIONS:

TRUE Turn on RI for K.

RECOMMENDATION:

For large (especially 1D and 2D) molecules the approximation may yield significant improvements in Fock evaluation time when used with ARI.

ROKS_LEVEL_SHIFT

Introduce a level shift of N/100 hartree to aid DIIS convergence.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No shift

N level shift of N/100 hartree.

RECOMMENDATION:

Use in cases of problematic DIIS convergence. Only available for the ROKS implementation on the old SCF engine (GEN_SCFMAN = FALSE).

ROKS_SS_MIXING

Allow coupling between the two singly-occupied molecular orbitals.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Supress coupling.

1 Allow coupling.

RECOMMENDATION:

Supress coupling when the resuling states are difficult to converge and / or overlap significantly with the ground state. Only for GDM and SGM-based solvers.

ROKS

Controls whether ROKS calculation will be performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE ROKS is not performed.

TRUE ROKS will be performed.

RECOMMENDATION:

Set to TRUE if ROKS calculation is desired. Make sure that UNRESTRICTED is not set to TRUE.

ROSCF_DIAG_SCALE_A

When performing canonical restricted open-shell SCF calculations, scale the open-shell diagonal α Fock matrix block (and, thus, singly occupied α orbital energies) during the SCF iterations in order to mitigate aufbau principle violations in case of poorly converging systems. After convergence, proper canonical orbital energies are restored.

TYPE:

INTEGER

DEFAULT:

50 Scale α orbital energies by a factor of 0.5 which often leads to improved convergence behavior.

OPTIONS:

n Scale α orbital energies by a factor of $n \cdot 10^{-}2$.

RECOMMENDATION:

In case of convergence issues, try series of different values, e.g., ± 800, ± 400, ± 200, ± 100, ± 50, ± 20, ± 10.

ROSCF_DIAG_SCALE_B

The same as ROSCF_DIAG_SCALE_A, except that singly unpccupied β orbital energies are manipulated during the SCF iterations.

TYPE:

INTEGER

DEFAULT:

50 Scale β orbital energies by a factor of 0.5.

OPTIONS:

n Scale β orbital energies by a factor of $n \cdot 10^{-}2$.

RECOMMENDATION:

See ROSCF_DIAG_SCALE_A for recommendations.

ROSCF_DODS

Run a canonical restricted open-shell calculation (also called "ROSCF-DODS", where DODS stands for different orbitals for different spins) with GEN_SCFMAN. Available for high-spin half-filled open-shell systems.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a canonical ROSCF calculation.

FALSE Do not perform a canonical ROSCF calculation (run an open-shell USCF calculation instead).

RECOMMENDATION:

Set to TRUE if desired.

RPATH_COORDS

Determines which coordinate system to use in the IRC search.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Use mass-weighted coordinates.

1 Use Cartesian coordinates.

2 Use *Z*-matrix coordinates.

RECOMMENDATION:

Use the default. Note that use of Z-matrix coordinates requires that geometries be input in Z-matrix format.

RPATH_DIRECTION

Determines the first direction of the eigenmode to follow. This will not usually be known prior to the Hessian diagonalization.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Descend in the positive direction of the eigenmode, then restart in the negative direction.

-1 Descend in the negative direction of the eigenmode, then restart in the positive direction.

RECOMMENDATION:

It is usually not possible to determine in which direction to go *a priori*, so both directions are automatically considered. A job that reads in the final geometry from the reaction path job will use the final step from the second direction.

RPATH_MAX_CYCLES

Specifies the maximum number of points to find on the reaction path.

TYPE:

INTEGER

DEFAULT:

20

OPTIONS:

n User-defined number of cycles.

RECOMMENDATION:

Use more points if the minimum is desired, but not reached using the default.

RPATH_MAX_STEPSIZE

Specifies the maximum step size to be taken (in 0.001 a.u.).

TYPE:

INTEGER

DEFAULT:

150 corresponding to a step size of 0.15 a.u..

OPTIONS:

n Step size = n/1000 a.u.

RECOMMENDATION:

None.

RPATH_PRINT

Specifies the print output level.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n

RECOMMENDATION:

Use the default, as little additional information is printed at higher levels. Most of the output arises from the multiple single point calculations that are performed along the reaction pathway.

RPATH_TOL_DISPLACEMENT

Specifies the convergence threshold for the step. If a step size is chosen by the algorithm that is smaller than this, the path is deemed to have reached the minimum.

TYPE:

INTEGER

DEFAULT:

5000 Corresponding to 0.005 a.u.

OPTIONS:

n User-defined. Tolerance = n/1000000 a.u.

RECOMMENDATION:

Use the default. Note that this option *only* controls the threshold for ending the RPATH job and does nothing to the intermediate steps of the calculation. A smaller value will provide reaction paths that end closer to the true minimum. Use of smaller values without adjusting RPATH_MAX_STEPSIZE, however, can lead to oscillations about the minimum.

RPA

Do an RPA calculation in addition to a CIS or TDDFT/TDA calculation.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an RPA calculation.

TRUE Do an RPA calculation.

2 Do an RPA calculation without running CIS or TDDFT/TDA first.

RECOMMENDATION:

RPA = 2 is not available for restricted open-shell wavefunctions.

S2THRESH

Cutoff for neglect of overlap integrals, defined via a two-electron shell-pair threshold of $10^{-\mathrm{S2THRESH}}$ (S2THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

Same as THRESH.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

Increase the value of S2THRESH if the program finds negative eigenvalues for the overlap matrix.

SASF_CIS

Controls whether to do an SA-SF-CIS/DFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an SA-SF-CIS/DFT calculation.

TRUE Do an SA-SF-CIS/DFT calculation.

RECOMMENDATION:

The SA-SF method requires a restricted open-shell ground-state calculation.

SAVE_LAST_GPX

Save the last $\mathbf{G}[\mathbf{P}^x]$ when calculating dynamic polarizabilities in order to call the MOProp code in a second run, via MOPROP = 104 (which is otherwise the same as MOPROP = HYPERPOL).

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 False

1 True

RECOMMENDATION:

None

SAVE_VIBRONIC_PARAMS

Save information about excited state which is requested in vibronic spectra simulation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE

RECOMMENDATION:

TRUE

SCALE_NUCLEAR_CHARGE

Scale the nuclear charges.

TYPE:

INTEGER

DEFAULT:

0 do not scale (use true atomic numbers)

OPTIONS:

N scale the nuclear charges in a way that adds a charge of N/100 (in a.u.)

RECOMMENDATION:

For EOM methods a perturbative correction can be added in conjunction with this option (as noted above), but for other electronic structure methods once simply gets a traditional calculation but with modified nuclear charges.

SCFMI_FREEZE_SS

Keep the first several fragments unrelaxed in an SCFMI calculation.

TYPE:

INTEGER

DEFAULT:

0 (all fragments are active)

OPTIONS:

n Freeze the first n fragments.

RECOMMENDATION:

None

SCFMI_MODE

Determine whether generalized SCF-MI is used and also the property of the working basis.

TYPE:

INTEGER

DEFAULT:

0 (“1" is used by basic “EDA2" calculations).

OPTIONS:

0 AO-block based SCF-MI (the original definition of ALMOs).

1 Generalized SCF-MI with basis vectors that are non-orthogonal between fragments.

2 Generalized SCF-MI with basis vectors that are orthogonal between fragments.

RECOMMENDATION:

None

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

None

OPTIONS:

SGM

SGM_LS

SGM_QLS

RECOMMENDATION:

SGM should be used for RO-ΔSCF or ROKS calculations only. SGM_LS is recommended for R- or U-ΔSCF, though it can also be used for RO-ΔSCF or ROKS. SGM_QLS is a slower but more robust option for R- and U-ΔSCF calculations.

SCF_CONVERGENCE

SCF is considered converged when the wave function error is less that $10^{-\text{SCF\_CONVERGENCE}}$. Adjust the value of THRESH at the same time.

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations (including BSSE and XSAPT jobs).

7 For job types NMR, STATPOLAR, DYNPOLAR, HYPERPOLAR, and ISSC.

8 For most other job types, including geometry optimization, transition-state search, vibrational analysis, CIS/TDDFT calculations, correlated wavefunction methods, energy decomposition analysis (EDA2), etc.

OPTIONS:

User-defined

RECOMMENDATION:

Tighter criteria are recommended for geometry optimization and vibration analysis. Larger values provide more significant figures, at greater computational cost. Beginning with Q-CHEM 3.0, the DIIS error is measured by the maximum error rather than the RMS error that was used in earlier versions.

SCF_FINAL_PRINT

Controls level of output from SCF procedure to Q-CHEM output file at the end of the SCF.

TYPE:

INTEGER

DEFAULT:

0 No extra print out.

OPTIONS:

0 No extra print out.

1 Orbital energies and break-down of SCF energy.

2 Level 1 plus MOs and density matrices.

3 Level 2 plus Fock matrix.

RECOMMENDATION:

The break-down of energies is often useful (level 1).

SCF_GUESS_ALWAYS

Switch to force the regeneration of a new initial guess for each series of SCF iterations (for use in geometry optimization).

TYPE:

LOGICAL

DEFAULT:

False

OPTIONS:

False Do not generate a new guess for each series of SCF iterations in an optimization; use MOs from the previous SCF calculation for the guess, if available.

True Generate a new guess for each series of SCF iterations in a geometry optimization.

RECOMMENDATION:

Use the default unless SCF convergence issues arise

SCF_GUESS_MIX

Controls mixing of LUMO and HOMO to break symmetry in the initial guess. For unrestricted jobs, the mixing is performed only for the alpha orbitals.

TYPE:

INTEGER

DEFAULT:

0 (FALSE) Do not mix HOMO and LUMO in SCF guess.

OPTIONS:

0 (FALSE) Do not mix HOMO and LUMO in SCF guess.

1 (TRUE) Add 10% of LUMO to HOMO to break symmetry.

n Add $n \times 10\%$ of LUMO to HOMO ($0 < n < 10$).

RECOMMENDATION:

When performing unrestricted calculations on molecules with an even number of electrons, it is often necessary to break alpha/beta symmetry in the initial guess with this option, or by specifying input for *$occupied*.

SCF_GUESS_PRINT

Controls printing of guess MOs, Fock and density matrices.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not print guesses.
SAD	
1	Atomic density matrices and molecular matrix.
2	Level 1 plus density matrices.
CORE and GWH	
1	No extra output.
2	Level 1 plus Fock and density matrices and, MO coefficients and eigenvalues.
READ	
1	No extra output
2	Level 1 plus density matrices, MO coefficients and eigenvalues.

RECOMMENDATION:

None

SCF_GUESS

Specifies the initial guess procedure to use for the SCF.

TYPE:

STRING

DEFAULT:

SAD	Superposition of atomic densities[26] (default for internal basis sets)
AUTOSAD	For internally defined or user-customized general basis sets or mixed basis
GWH	For ROHF jobs with GEN_SCFMAN = FALSE which require a set of orbitals
FRAGMO	For fragment jobs such as ALMO-based calculations
CORE	For special cases that currently can't be handled by the ones above (e.g. mixed basis with ghost atoms)

OPTIONS:

CORE	Diagonalize core Hamiltonian
SAD	Superposition of atomic density[26]
SAP	Superposition of atomic potentials[25] (only available with GEN_SCFMAN = TRUE)
AUTOSAD	On-the-fly superposition of atomic densities
SADMO	Purified superposition of atomic densities (available only with standard basis sets)
GWH	Apply generalized Wolfsberg-Helmholtz approximation
READ	Read previous MOs from disk
FRAGMO	Superimposing converged fragment MOs (see Section 12.3)

RECOMMENDATION:

SAD, AUTOSAD, or SADMO guess for standard basis sets. For either standard or user-customized general basis sets, AUTOSAD is recommended and used as default. If these options fail, use the SAP guess; try the GWH or core Hamiltonian guess only as a last resort. For mixed basis sets, only the AUTOSAD, SAP, GWH, and core Hamiltonian guesses are currently available. For ROHF it can be useful to READ guesses from an SCF calculation on the corresponding cation or anion. Note that because the density is made spherical, this may favor an undesired state for atomic systems, especially transition metals. Use FRAGMO in a fragment MO calculation.

SCF_MINFIND_INCREASEFACTOR

Controls how the height of the penalty function changes when repeatedly trapped at the same solution

TYPE:

INTEGER

DEFAULT:

10100 meaning 1.01

OPTIONS:

$abcde$	corresponding to $a.bcde$

RECOMMENDATION:

If the algorithm converges to a solution which corresponds to a previously located solution, increase both the normalization N and the width lambda of the penalty function there. Then do a restart.

SCF_MINFIND_INITLAMBDA

Control the initial width of the penalty function.

TYPE:

INTEGER

DEFAULT:

02000 meaning 2.000

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The initial inverse-width (*i.e.*, the inverse-variance) of the Gaussian to place to fill solution's well. Measured in electrons$^{(}-1)$. Increasing this will repeatedly converging on the same solution.

SCF_MINFIND_INITNORM

Control the initial height of the penalty function.

TYPE:

INTEGER

DEFAULT:

01000 meaning 1.000

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The initial normalization of the Gaussian to place to fill a well. Measured in hartrees.

SCF_MINFIND_MIXENERGY

Specify the active energy range when doing Active mixing

TYPE:

INTEGER

DEFAULT:

00200 meaning 00.200

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The standard deviation of the Gaussian distribution used to select the orbitals for mixing (centered on the Fermi level). Measured in Hartree. To find less-excited solutions, decrease this value

SCF_MINFIND_MIXMETHOD

Specify how to select orbitals for random mixing

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Random mixing: select from any orbital to any orbital.

1 Active mixing: select based on energy, decaying with distance from the Fermi level.

2 Active Alpha space mixing: select based on energy, decaying with distance from the Fermi level only in the alpha space.

RECOMMENDATION:

Random mixing will often find very high energy solutions. If lower energy solutions are desired, use 1 or 2.

SCF_MINFIND_NRANDOMMIXES

Control how many random mixes to do to generate new orbitals

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Perform n random mixes.

RECOMMENDATION:

This is the number of occupied/virtual pairs to attempt to mix, per separate density (*i.e.*, for unrestricted calculations both alpha and beta space will get this many rotations). If this is negative then only mix the highest 25% occupied and lowest 25% virtuals.

SCF_MINFIND_RANDOMMIXING

Control how to choose new orbitals after locating a solution

TYPE:

INTEGER

DEFAULT:

00200 meaning .02 radians

OPTIONS:

$abcde$ corresponding to $a.bcde$ radians

RECOMMENDATION:

After locating an SCF solution, the orbitals are mixed randomly to move to a new position in orbital space. For each occupied and virtual orbital pair picked at random and rotate between them by a random angle between 0 and this. If this is negative then use exactly this number, *e.g.*, -15708 will almost exactly swap orbitals. Any number< -15708 will cause the orbitals to be swapped exactly.

SCF_MINFIND_READDISTTHRESH

The distance threshold at which to consider two solutions the same

TYPE:

INTEGER

DEFAULT:

00100 meaning 0.1

OPTIONS:

$abcde$ corresponding to $ab.cde$

RECOMMENDATION:

The threshold to regard a minimum as the same as a read in minimum. Measured in electrons. If two minima are closer together than this, reduce the threshold to distinguish them.

SCF_MINFIND_RESTARTSTEPS

Restart with new orbitals if no minima have been found within this many steps

TYPE:

INTEGER

DEFAULT:

300

OPTIONS:

n Restart after n steps.

RECOMMENDATION:

If the SCF calculation spends many steps not finding a solution, lowering this number may speed up solution-finding. If the system converges to solutions very slowly, then this number may need to be raised.

SCF_MINFIND_RUNCORR

Run post-SCF correlated methods on multiple SCF solutions

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

If this is set > 0, then run correlation methods for all found SCF solutions.

RECOMMENDATION:

Post-HF correlation methods should function correctly with excited SCF solutions, but their convergence is often much more difficult owing to intruder states.

SCF_MINFIND_WELLTHRESH

Specify what SCF_MINFIND believes is the basin of a solution

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n for a threshold of 10^{-n}

RECOMMENDATION:

When the DIIS error is less than 10^{-n}, penalties are switched off to see whether it has converged to a new solution.

SCF_NOCRASH

Ensure the calculations continues if the SCF fails to converge for a given structure.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Ensure calculation will continue with next structure.

False Calculation will stop.

RECOMMENDATION:

Use SCF_NOCRASH = TRUE.

SCF_PRINT_FRGM

Controls the output of Q-CHEM jobs on isolated fragments.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE The output is printed to the parent job output file.

FALSE The output is not printed.

RECOMMENDATION:

Use TRUE if details about isolated fragments are important.

SCF_PRINT

Controls level of output from SCF procedure to Q-CHEM output file.

TYPE:

INTEGER

DEFAULT:

0 Minimal, concise, useful and necessary output.

OPTIONS:

0 Minimal, concise, useful and necessary output.

1 Level 0 plus component breakdown of SCF electronic energy.

2 Level 1 plus density, Fock and MO matrices on each cycle.

3 Level 2 plus two-electron Fock matrix components (Coulomb, HF exchange , orbital kinetic energies, and DFT exchange-correlation matrices) on each cycle.

RECOMMENDATION:

Proceed with care; can result in *extremely* large output files at level 2 or higher. Output of all information is only available in scfman (GEN_SCFMAN = FALSE). If GEN_SCFMAN is set to TRUE and SCF_PRINT > 1, only level 1 plus MO matrices are available in the output. These levels are primarily for program debugging.

SCF_READMINIMA

Read in solutions from a previous SCF metadynamics calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Read in n previous solutions and attempt to locate them all.

$-n$ Read in n previous solutions, but only attempt to locate solution n (not available in LIBNOCI).

RECOMMENDATION:

This may not actually locate all solutions required and will probably locate others too. The SCF will also stop when the number of solutions specified in SCF_SAVEMINIMA are found. Solutions from other geometries may also be read in and used as starting orbitals. If a solution is found and matches one that is read in within SCF_MINFIND_READDISTTHRESH, its orbitals are saved in that position for any future calculations. The algorithm works by restarting from the orbitals and density of a the minimum it is attempting to find. After 10 failed restarts (defined by SCF_MINFIND_RESTARTSTEPS), it moves to another previous minimum and attempts to locate that instead. If there are no minima to find, the restart does random mixing (with 10 times the normal random mixing parameter). Note that in LIBNOCI, previous minima are read using NOCI_REFGEN = 1, whilst the exact solutions are specified as described in Section 4.9.3

SCF_ROBUST_MIN_SIGN_CHANGES

Controls the number of sign changes in a window of logarithms of DIIS error or RMS gradients used for detecting oscillations in robust SCF procedures.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Customized.

RECOMMENDATION:

None

SCF_ROBUST_PLATEAU_WINDOWS

Controls the number of windows of logarithms of DIIS error or RMS gradients used for detecting a plateau in robust SCF procedures.

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

n Customized.

RECOMMENDATION:

None

SCF_ROBUST_SLOPE_THRESH

Controls the slope threshold for a series of windows of logarithms of DIIS error or RMS gradients used for detecting a plateau in robust SCF procedures.

TYPE:

INTEGER

DEFAULT:

3 Corresponds to 10^{-3}

OPTIONS:

n Customized to 10^{-n}.

RECOMMENDATION:

None

SCF_ROBUST_WINDOW_SIZE

Controls the size of the window of logarithms of DIIS error or RMS gradients used for detecting oscillations or a plateau in robust SCF procedures.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n Customized.

RECOMMENDATION:

None

SCF_SAVEMINIMA

Turn on SCF metadynamics and specify how many solutions to locate.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not use SCF metadynamics

n Attempt to find n distinct SCF solutions.

RECOMMENDATION:

Perform SCF Orbital metadynamics and attempt to locate n different SCF solutions. Note that these may not all be minima. Many saddle points are often located. The last one located will be the one used in any post-SCF treatments. In systems where there are infinite point groups, this procedure cannot currently distinguish between spatial rotations of different densities, so will likely converge on these multiply.

SEARCH_ATOMIC

Perform an optimization for atomic cluster.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Atomic cluster search will be performed.

False Molecular clusters search will be performed.

RECOMMENDATION:

Use N_SWOP to specify atomic number of atom swops in structure generation.

SEARCH_MOM

Allows the search to be performed in conjunction with MOM to explore excited states.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True A search with MOM is performed.

False Normal calculation without MOM.

RECOMMENDATION:

None.

SET_QUADRATIC

Determines whether to include full quadratic response contributions for TDDFT.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Include full quadratic response contributions for TDDFT.

FALSE Use pseudo-wave function approach.

RECOMMENDATION:

The pseudo-wave function approach is usually accurate enough and is free of accidental singularities. Consult Refs. 44 and 31 for additional guidance.

SFX_AMP_OCC_A

Defines a custom amplitude guess vector in SF-XCIS method.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n builds a guess amplitude with an α-hole in the nth orbital (requires SFX_AMP_VIR_B).

RECOMMENDATION:

Only use when default guess is not satisfactory.

SFX_AMP_VIR_B

Defines a user-specified amplitude guess vector in SF-XCIS method.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n builds a guess amplitude with a β-particle in the nth orbital (requires SFX_AMP_OCC_A).

RECOMMENDATION:

Only use when default guess is not satisfactory.

SF_STATES

Controls the number of excited spin-flip states to calculate.

TYPE:

INTEGER

DEFAULT:

0 Do not perform a SF-ADC calculation

OPTIONS:

$n > 0$ Number of states to calculate for each irrep or

$[n_1, n_2, ...]$ Compute n_1 states for the first irrep, n_2 states for the second irrep, ...

RECOMMENDATION:

Use this variable to define the number of excited states in the case of a spin-flip calculation. SF-ADC is available for ADC(2)-s, ADC(2)-x and ADC(3).

SKIP_CHARGE_SELF_INTERACT

Ignores the electrostatic interactions among external charges in a QM/MM calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE No electrostatic interactions among external charges.

FALSE Computes the electrostatic interactions among external charges.

RECOMMENDATION:

None

SKIP_CIS_RPA

Skips the solution of the CIS, RPA, TDA or TDDFT equations for wave function analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE / FALSE

RECOMMENDATION:

Set to true to speed up the generation of plot data if the same calculation has been run previously with the scratch files saved.

SKIP_OLD_SCFMAN

Skips *only* old SCF drivers

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Skip *only* old SCF drivers

FALSE Do not skip old SCF drivers

RECOMMENDATION:

When performing CAP calculations on temporary anions, it may help setting this variable to FALSE.

SMX_GAS_PHASE

Converge the gas-phase SCF first before doing calculations with SMx models

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Run SMx calculations directly

TRUE Run gas-phase calculation first

RECOMMENDATION:

Use the default unless solvation free energy is needed. Set it to TRUE if the SCF calculation fails to converge otherwise.

SOLVENT_METHOD

Sets the preferred solvent method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0	Do not use a solvation model.
KIRKWOOD	Use the Kirkwood-Onsager model (Section 11.2.2).
PCM	Use an apparent surface charge, polarizable continuum model (Section 11.2.3).
ISOSVP	Use the isodensity implementation of the SS(V)PE model (Section 11.2.6).
COSMO	Use COSMO (Section 11.2.8).
SM8	Use version 8 of the Cramer-Truhlar SMx model (Section 11.2.9.1).
SM12	Use version 12 of the SMx model (Section 11.2.9.2).
SMD	Use SMD (Section 11.2.9.3).
CHEM_SOL	Use the Langevin Dipoles model (Section 11.2.10).
PEQS	Use the Poisson Equation Solver (Section 11.2.11).

RECOMMENDATION:

Consult the literature (*e.g.*, Ref. 17). PCM is a collective name for a family of models and additional input options may be required in this case, in order to fully specify the model; see Section 11.2.3. Several versions of SM12 are available as well, as discussed in Section 11.2.9.2.

SOS_FACTOR

Controls the strength of the opposite-spin component of PT2 correlation energy.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $c_{\mathrm{os}} = n/10^6$ in Eq. (5.60).

RECOMMENDATION:

NONE

SOS_UFACTOR

Sets the scaling parameter c_U

TYPE:

INTEGER

DEFAULT:

151 For SOS-CIS(D), corresponding to 1.51

140 For SOS-CIS(D_0), corresponding to 1.40

OPTIONS:

n $c_U = n/100$

RECOMMENDATION:

Use the default

SPATIAL_OVERLAP_GRID

Controls the grid that is used to evaluate O_{ia} or $\tilde{O}_{ia}$ in Eq. (7.17).

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Use a EML grid with $N_r = 300$ and $N_\Omega = 302$.

2 Use a EML grid with $N_r = 400$ and $N_\Omega = 434$.

RECOMMENDATION:

None.

SPATIAL_OVERLAP_METRIC

Controls whether O_{ia} or $\tilde{O}_{ia}$ is used to evaluate Λ in Eq. (7.16).

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Compute Λ using O_{ia}.

2 Compute Λ using $\tilde{O}_{ia}$.

RECOMMENDATION:

Option 1 represents the original metric suggested by Tozer and co-workers in Ref. 32.

SPATIAL_OVERLAP_PRINT

Controls whether to print the spatial overlaps O_{ia} or $\tilde{O}_{ia}$.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not print the O_{ia}.

1 Print the frontier overlaps only (5 occupied and 5 virtual orbitals.

2 Print all of the O_{ia}.

RECOMMENDATION:

These may be useful for *a posteriori* analysis of the spatial proximity of various MOs; however, option 2 will engender significant printing for large molecules. Whether it is O_{ia} [Eq. (7.17)] or $\tilde{O}_{ia}$ [Eq. (7.18)] that is printed depends upon the setting of SPATIAL_OVERLAP_ANAL.

SPIN_FLIP_XCIS

Controls whether to do a SF-XCIS calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an SF-XCIS calculation.

TRUE Do an SF-XCIS calculation (requires ROHF triplet ground state).

RECOMMENDATION:

None

SPIN_FLIP

Selects whether to perform a standard excited state calculation, or a spin-flip calculation. Spin multiplicity should be set to 3 for systems with an even number of electrons, and 4 for systems with an odd number of electrons.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

None

SQC_GAMADJUST

Specifies the γ-adjustment protocol.

TYPE:

STRING

DEFAULT:

True

OPTIONS:

True use the γ-adjustment protocol.

False

RECOMMENDATION:

The γ-adjustment protocol is generally recommended.

SRC_DFT

Selects form of the short-range corrected functional.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

1 SRC1 functional.

2 SRC2 functional.

RECOMMENDATION:

None

SSG

Controls the calculation of the SSG wave function.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute the SSG wave function

1 Do compute the SSG wave function

RECOMMENDATION:

See also the UNRESTRICTED and DIIS_SUBSPACE_SIZE *$rem* variables.

SSS_FACTOR

Controls the strength of the same-spin component of PT2 correlation energy.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $c_{\rm ss} = n/10^6$ in Eq. (5.60).

RECOMMENDATION:

NONE

STABILITY_ANALYSIS

Performs stability analysis for a HF or DFT solution.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform stability analysis.

FALSE Do not perform stability analysis.

RECOMMENDATION:

Set to TRUE when a HF or DFT solution is suspected to be unstable.

STATE_ANALYSIS

Controls the analysis and export of excited, ionized or electron-attached state densities and orbitals (see 10.2.12 for details).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform excited state analyses.

FALSE No excited state analyses or export will be performed.

RECOMMENDATION:

Set to TRUE, if detailed analysis of the excited, ionized or electron-attached states is required or if density or orbital plots are needed.

STATE_FOLLOW

Turns on state following.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use state-following.

TRUE Use state-following.

RECOMMENDATION:

None.

STEP_EPSILON

Scales the size of the occupied/virtual gap imposed by the level-shift by $N/100$ Hartree.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

N

RECOMMENDATION:

Use the default unless convergence issues arise, in which case a larger value can be used until the desired state is found. Be aware that increasing the occupied/virtual gap in level-shift algorithms slows convergence so it may be advisable to increase SCF_MAX_CYCLES if large shifts are required.

STEP_PRINT

Controls the print level for STEP algorithm information.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 Do not print any information about STEP between SCF cycles.

1 Print the level-shift applied at each SCF cycle (R- and U-STEP).

2 Print the level-shift for both mixed and triplet states at each SCF cycle (RO-STEP).

RECOMMENDATION:

Use the default. Level shifts of 0 indicate that an *aufbau* criterion is sufficient to determine orbital occupation, and shifts > 0 imply non-*aufbau* selection of the occupied space.

STEP

Activates the STEP procedure.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not apply the STEP level-shift algorithm.

TRUE Apply the STEP level-shift algorithm.

RECOMMENDATION:

None

STS_ACCEPTOR

Define the acceptor molecular fragment.

TYPE:

STRING

DEFAULT:

0 No acceptor fragment is defined.

OPTIONS:

i-j Acceptor fragment is in the ith atom to the jth atom.

RECOMMENDATION:

Note no space between the hyphen and the numbers i and j.

STS_DONOR

Define the donor fragment.

TYPE:

STRING

DEFAULT:

0 No donor fragment is defined.

OPTIONS:

i-j Donor fragment is in the ith atom to the jth atom.

RECOMMENDATION:

Note no space between the hyphen and the numbers i and j.

STS_FCD

Control the calculation of FCD for ET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform an FCD calculation.

TRUE Include an FCD calculation.

RECOMMENDATION:

None

STS_FED

Control the calculation of FED for EET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform a FED calculation.

TRUE Include a FED calculation.

RECOMMENDATION:

None

STS_FSD

Control the calculation of FSD for EET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform a FSD calculation.

TRUE Include a FSD calculation.

RECOMMENDATION:

For RCIS triplets, FSD and FED are equivalent. FSD will be automatically switched off and perform a FED calculation.

STS_GMH

Control the calculation of GMH for ET couplings.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform a GMH calculation.

TRUE Include a GMH calculation.

RECOMMENDATION:

When set to true computes Mulliken-Hush electronic couplings. It yields the generalized Mulliken-Hush couplings as well as the transition dipole moments for each pair of excited states and for each excited state with the ground state.

STS_MOM

Control calculation of the transition moments between excited states in CIS and TDDFT calculations (including spin-flip variants).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate state-to-state transition moments.

TRUE Do calculate state-to-state transition moments.

RECOMMENDATION:

When set to true requests the state-to-state dipole transition moments for all pairs of excited states and for each excited state with the ground state. This is not available for restricted open-shell wavefunctions.

STS_TRANS_ACCEPTOR

Control the range of frontier MOs for the acceptor in the FMO approach to ET couplings.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

o-v Compute couplings between o and v occupied and virtual MOs, respectively.

RECOMMENDATION:

None

STS_TRANS_DONOR

Control the range of frontier MOs for the donor in the FMO approach to ET couplings.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

o-v Compute couplings between o and v occupied and virtual MOs, respectively.

RECOMMENDATION:

None

SVP_CAVITY_CONV

Determines the convergence value of the iterative isodensity cavity procedure.

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

n Convergence threshold set to 10^{-n}.

RECOMMENDATION:

The default value unless convergence problems arise.

SVP_CHARGE_CONV

Determines the convergence value for the charges on the cavity. When the change in charges fall below this value, if the electron density is converged, then the calculation is considered converged.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

n Convergence threshold set to 10^{-n}.

RECOMMENDATION:

The default value unless convergence problems arise.

SVP_GUESS

Specifies how and if the SS(V)PE model will use a given guess for the charges and cavity points.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No guessing.

1 Read a guess from a previous isodensity SS(V)PE calculation.

2 Use a guess specified by the *$svpirf* section from the input.

RECOMMENDATION:

It is helpful to also set SCF_GUESS to READ when using a guess from a previous Q-CHEM run. Set **IPnRF** = 1 in the *$svp* section of the previous Q-CHEM job, in order to save the reaction field for subsequent reading (see Section 11.2.6.2).

SVP_MEMORY

Specifies the amount of memory for use by the solvation module.

TYPE:

INTEGER

DEFAULT:

125

OPTIONS:

n corresponds to the amount of memory in MB.

RECOMMENDATION:

The default should be fine for medium size molecules with the default Lebedev grid, only increase if needed.

SVP_PATH

Specifies whether to run a gas phase computation prior to performing the solvation procedure.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Run a gas-phase calculation and upon convergence run the SS(V)PE computation.

1 Do not run a gas-phase calculation.

RECOMMENDATION:

Running the gas-phase calculation provides a good guess to start the solvation stage and provides a more complete set of solvated properties.

SYMMETRY_DECOMPOSITION

Determines symmetry decompositions to calculate.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 No symmetry decomposition.

1 Calculate MO eigenvalues and symmetry (if available).

2 Perform symmetry decomposition of kinetic energy and nuclear attraction matrices.

RECOMMENDATION:

None

SYM_TOL

Controls the tolerance for determining point group symmetry. Differences in atom locations less than $10^{-\mathrm{SYM\_TOL}}$ are treated as zero.

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}.

OPTIONS:

User defined.

RECOMMENDATION:

Use the default unless the molecule has high symmetry which is not being correctly identified. Note that relaxing this tolerance too much may introduce errors into the calculation. Misidentified point group symmetry can affect rotational symmetry numbers in thermochemical calculations.

TAO_DFT_THETA_NDP

The parameter n (the exponent) for the value of the fictitious temperature $\theta = m \times 10^{-n}\ E_h$ in TAO-DFT.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Customize the exponential power for the fictitious temperature.

RECOMMENDATION:

NONE

TAO_DFT_THETA

The parameter m (the mantissa) for the value of the fictitious temperature $\theta = m \times 10^{-n}\ E_h$ in TAO-DFT.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

m Customize the mantissa for the fictitious temperature.

RECOMMENDATION:

NONE

TAO_DFT

Controls whether to use TAO-DFT.

TYPE:

Boolean

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use TAO-DFT

TRUE Use TAO-DFT

RECOMMENDATION:

NONE

TDDFT_LR_PCM

Controls LR-PCM for TDDFT, *i.e.*, whether or not to add the LR-PCM contributions to the TDDFT eigenvalue problem.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE Do not do LR-PCM (0th-order solvent correction only).

TRUE Perform full LR-PCM.

RECOMMENDATION:

Assuming that PCM solvation is turned on for the ground state (SOLVENT_METHOD = PCM in the *$rem* section), then disabling LR-PCM by setting TDDFT_LR_PCM = FALSE will afford a "0th-order" solvation correction, in which solvent-polarized MOs and energy levels are used in what is otherwise equivalent to a gas-phase TDDFT calculation. This is the first step in more sophisticated "nonequilibrium" TDDFT + PCM methods, which are discussed in Section 11.2.3.3. The LR-PCM correction to the excitation energies has some peculiar properties, such as the fact that it vanishes for optically-forbidden states,[17] and the state-specific approaches that are discussed in Section 11.2.3.3 are likely preferable.

TDDFT_MI

Perform an TDDFT(MI) calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform an TDDFT(MI) calculation

TRUE Perform an TDDFT(MI) calculation

RECOMMENDATION:

False

TDDFT_NVIRT

Specifies the number of virtual orbitals included in the XAS TDDFT calculation.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Corresponding to the lowest energy n virtual orbitals.

RECOMMENDATION:

None

TDKS_RESTART

Restart the calculation by continuing the previous job

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE The TDKS calculation continues from the previous calculation.

FALSE The TDKS calculation starts from the beginning.

RECOMMENDATION:

None.

TDKS

Job control keyword to turn on TDKS calculation

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform a TDKS calculation following a ground-state SCF calculation

FALSE Do not perform a TDKS calculation

RECOMMENDATION:

None.

THRESH_ADIIS_SWITCH

The threshold for switching from ADIIS to DIIS in a ADIIS-DIIS calculations

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Switching from ADIIS to DIIS when the SCF error is below 10^{-n}

RECOMMENDATION:

3 or 4 is suitable

THRESH_DIIS_SWITCH

The threshold for switching between DIIS extrapolation and direct minimization of the SCF energy is $10^{-\text{THRESH\_DIIS\_SWITCH}}$ when SCF_ALGORITHM is DIIS_GDM or DIIS_DM. See also MAX_DIIS_CYCLES.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

User-defined.

RECOMMENDATION:

None

THRESH_RCA_SWITCH

The threshold for switching between RCA and DIIS when SCF_ALGORITHM is RCA_DIIS.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

N Algorithm changes from RCA to DIIS when Error is less than 10^{-N}.

RECOMMENDATION:

None

THRESH

Cutoff for neglect of two electron integrals. $10^{-\mathrm{THRESH}}$ (THRESH ≤ 14).

TYPE:

INTEGER

DEFAULT:

8 For single point energies.

10 For optimizations and frequency calculations.

14 For coupled-cluster calculations.

OPTIONS:

n for a threshold of 10^{-n}.

RECOMMENDATION:

The value should satisfy THRESH $\geq$ 3 + SCF_CONVERGENCE, although THRESH = 4 + SCF_CONVERGENCE is the default (in most cases) since Q-CHEM v. 6.0. See Ref. 13 for recommended values of THRESH in the presence of diffuse basis functions, where tighter thresholds are often required and too-loose thresholds may lead to slower convergence or convergence failure.

TIGHTEN_CONVERG

At the end of the search re-calculate the energies of the optimized structures with tighter SCF convergence criteria.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Additional calculations with tighter SCF convergence performed.

False No additional calculations performed.

RECOMMENDATION:

None.

TIME_STEP

Specifies the molecular dynamics time step, in atomic units (1 a.u. = 0.0242 fs).

TYPE:

INTEGER

DEFAULT:

None.

OPTIONS:

User-specified.

RECOMMENDATION:

Smaller time steps lead to better energy conservation; too large a time step may cause the job to fail entirely. Make the time step as large as possible, consistent with tolerable energy conservation.

TPDFT_ATOM

Activate TP-DFT by specifying the atom from which to remove an electron.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Remove an electron from the lowest-energy orbital on the atom whose index is n.

RECOMMENDATION:

Be sure to set UNRESTRICTED = TRUE for TP-DFT calculations.

TPDFT_FRAC

Specify the fractional value of n_i to be removed.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Remove $n/100$ of an electron from the orbital specified using TPDFT_ATOM.

RECOMMENDATION:

None

TPDFT_LUMO

Specify the fractional value of n_{LUMO} to be added.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Add $n/100$ of an electron to the LUMO.

RECOMMENDATION:

Leave TPDFT_LUMO = 0 for core-level binding energy calculations [Eq. (7.162)] or use TPDFT_LUMO = 50 to implement Eq. (7.163).

TRNSS

Controls whether reduced single excitation space is used.

TYPE:

LOGICAL

DEFAULT:

FALSE Use full excitation space.

OPTIONS:

TRUE Use reduced excitation space.

RECOMMENDATION:

None

TRTYPE

Controls how reduced subspace is specified.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Select orbitals localized on a set of atoms.

2 Specify a set of orbitals.

3 Specify a set of occupied orbitals, include excitations to all virtual orbitals.

RECOMMENDATION:

None

TRUNC_CI_LEVEL

Specifies the order of truncated CI to be used in the active space.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not carry out truncated CI

1 CIS

2 CISD

3 CISDT

4 CISDTQ

etc.

RECOMMENDATION:

TSVDW_SR

Set custom value of the s_R damping parameter

TYPE:

INTEGER

DEFAULT:

no default value defined

OPTIONS:

n Corresponding to $n \cdot 10^{-4}$

RECOMMENDATION:

Use predefined values for supported functionals, otherwise consult Ref. 38 and other relevant literature.

TSVDW

Flag to switch on the TS-vdW method

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not apply TS-vdW.

1 Apply the TS-vdW method to obtain the TS-vdW energy.

2 Apply the TS-vdW method to obtain the TS-vdW energy and corresponding gradients.

RECOMMENDATION:

Since TS-vdW is itself a form of dispersion correction, it should *not* be used in conjunction with any of the dispersion corrections described in Section 5.7.3.

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE Closed-shell systems.

TRUE Open-shell systems.

OPTIONS:

FALSE Constrain the spatial part of the alpha and beta orbitals to be the same.

TRUE Do not Constrain the spatial part of the alpha and beta orbitals.

RECOMMENDATION:

Use the default unless ROHF is desired. Note that for unrestricted calculations on systems with an even number of electrons it is usually necessary to break α/β symmetry in the initial guess, by using SCF_GUESS_MIX or providing *$occupied* information (see Section 4.4 on initial guesses).

USECUBLAS_THRESH

Sets threshold of matrix size sent to GPU (smaller size not worth sending to GPU).

TYPE:

INTEGER

DEFAULT:

250

OPTIONS:

n user-defined threshold

RECOMMENDATION:

Use the default value. Anything less can seriously hinder the GPU acceleration

USER_CONNECT

Enables explicitly defined bonds.

TYPE:

STRING

DEFAULT:

FALSE

OPTIONS:

TRUE Bond connectivity is read from the *$molecule* section

FALSE Bond connectivity is determined by atom proximity

RECOMMENDATION:

Set to TRUE if bond connectivity is known, in which case this connectivity must be specified in the *$molecule* section. This greatly accelerates MM calculations.

USE_INITIAL

Include input structure as part of the search.

TYPE:

BOOLEAN

DEFAULT:

False

OPTIONS:

True Input structure is included in the search.

False Input structure is not included in the search.

RECOMMENDATION:

None.

USE_LIBNLQ

Turn on the use of LIBNLQ for calculating nonlocal correlation funcitonal.

TYPE:

LOGICAL

DEFAULT:

True For VV10.

FALSE For all other nonlocal funcitonals.

OPTIONS:

False

True

RECOMMENDATION:

Use the default

USE_LIBPT

Enable LIBPT for CCSD(T) calculations in CCMAN2.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

LIBPT is now used by default in all real-valued CC/EOM-CC calculations

USE_MGEMM

Use the mixed-precision matrix scheme (MGEMM) if you want to make calculations in your card in single-precision (or if you have a single-precision-only GPU), but leave some parts of the RI-MP2 calculation in double precision)

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE MGEMM disabled

TRUE MGEMM enabled

RECOMMENDATION:

Use when having single-precision cards

USE_RVV10

Used to turn on the rVV10 NLC functional

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Use VV10 NLC (the default for NL_CORRELATION)

TRUE Use rVV10 NLC

RECOMMENDATION:

Set to TRUE if the rVV10 NLC is desired.

VARTHRESH

Controls the temporary integral cut-off threshold, $t = 10^{-\mathrm{VARTHRESH}} \times$ (DIIS error)

TYPE:

INTEGER

DEFAULT:

0 Turns VARTHRESH off

OPTIONS:

n User-defined threshold

RECOMMENDATION:

3 has been found to be a practical level, and can slightly speed up SCF evaluation.

VCD_PRINT

Controls level of extra print out for the VCD calculations.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Standard full information print out.

2 Electronic part of AAT.

RECOMMENDATION:

Use the default.

VCD

Controls calculation of the VCD signals. Requires JOBTYPE to be set to FREQ

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate the VCD properties.

TRUE Do calculate the VCD properties.

RECOMMENDATION:

None

VCI

Specifies the number of quanta involved in the VCI calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

User-defined. Maximum value is 10.

RECOMMENDATION:

The availability depends on the memory of the machine. Memory allocation for VCI calculation is the square of $2(N_{\mathrm{Vib}} + N_{\mathrm{VCI}})/N_{\mathrm{Vib}}N_{\mathrm{VCI}}$ with double precision. For example, a machine with 1.5 GB memory and for molecules with fewer than 4 atoms, VCI(10) can be carried out, for molecule containing fewer than 5 atoms, VCI(6) can be carried out, for molecule containing fewer than 6 atoms, VCI(5) can be carried out. For molecules containing fewer than 50 atoms, VCI(2) is available. VCI(1) and VCI(3) usually overestimated the true energy while VCI(4) usually gives an answer close to the converged energy.

VIBMAN_PRINT

Controls level of extra print out for vibrational analysis.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Standard full information print out.
If VCI is TRUE, overtones and combination bands are also printed.

3 Level 1 plus vibrational frequencies in atomic units.

4 Level 3 plus mass-weighted Hessian matrix, projected mass-weighted Hessian matrix.

6 Level 4 plus vectors for translations and rotations projection matrix.

RECOMMENDATION:

Use the default, unless the Hessian is desired.

VIBRONIC_SPECTRA

Specifies which type of vibronic spectra will be predicted. Should be used in a frequency job (jobtype = Freq).

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No vibronic spectra is predicted.

1 OPA spectra is calculated.

2 OPE spectra is calculated.

3 RRS spectra is calculated.

RECOMMENDATION:

Use the default.

WANG_ZIEGLER_KERNEL

Controls whether to use the Wang-Ziegler non-collinear exchange-correlation kernel in a SF-TDDFT calculation. Set NEW_DFT = TRUE if using a Q-CHEM version older than 5.0.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use non-collinear kernel.

TRUE Use non-collinear kernel.

RECOMMENDATION:

None

WAVEFUNCTION_ANALYSIS

Controls the running of the default wave function analysis tasks.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Perform default wave function analysis.

FALSE Do not perform default wave function analysis.

RECOMMENDATION:

None. This option has no effect on NBO analysis.

WFA_LEVEL

Master variable for controlling the amount of output produced by LIBWFA.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

1 Only perform some population analyses.

2 Also perform exciton analysis and compute natural (transition/difference) orbitals.

3 Also perform charge transfer number analysis.

4 Maximal output including cube files for NTOs (this is needed to reproduce Ref. 20)

RECOMMENDATION:

Reduce if you want less print-out.

WFA_ORB_THRESH

Controls the number of hole/particle NTO pairs and frontier natural orbital pairs and natural difference orbital pairs exported to the Molden files.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

N Export all NTO/NO/NDO pairs with a weight above 10^{-N}.

RECOMMENDATION:

WFA_REF_STATE

Controls the reference state for the transition and difference density matrices used by LIBWFA. This keyword works for CIS/TDDFT/SF-DTDDFT computations. Use CC_STATE_TO_OPT for EOM-CC.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

-1 Use default: ground-state for standard CIS/TDDFT computations, first response state for SF-TDDFT.

0 Reference state

N Nth excited state/response state.

RECOMMENDATION:

NONE

WIG_GRID

Specify angular Lebedev grid for Wigner intracule calculations.

TYPE:

INTEGER

DEFAULT:

194

OPTIONS:

Lebedev grids up to 5810 points.

RECOMMENDATION:

Larger grids if high accuracy required.

WIG_LEB

Use Lebedev quadrature to evaluate Wigner integrals.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Evaluate Wigner integrals through series summation.

TRUE Use quadrature for Wigner integrals.

RECOMMENDATION:

None

WIG_MEM

Reduce memory required in the evaluation of $W(u, v)$.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not use low memory option.

TRUE Use low memory option.

RECOMMENDATION:

The low memory option is slower, so use the default unless memory is limited.

WRITE_WFN

Specifies whether or not a `.wfn` file is created, which is suitable for use with AIMPAC. Note that the output to this file is currently limited to f orbitals, which is the highest angular momentum implemented in AIMPAC.

TYPE:

STRING

DEFAULT:

(NULL) No output file is created.

OPTIONS:

filename Specifies the output file name. The suffix `.wfn` will be appended to this name.

RECOMMENDATION:

None

XAS_EDGE

Specifies the nuclear charge of element being excited.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

n Corresponding to the nuclear charge of element being excited.

RECOMMENDATION:

None

XAS_SCREEN_LEVEL

Sets the integral screening procedure for fast TDDFT.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

1 only evaluate integrals that include the inner core basis function on relevant atom(s).

2 only evaluate integrals that include basis functions on relevant atom(s).

RECOMMENDATION:

1

XCIS

Controls whether to do an XCIS calculation in addition to a CIS calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not do an XCIS calculation.

TRUE Do an XCIS calculation (requires ROHF ground state).

RECOMMENDATION:

None

XC_GRID

Specifies the type of grid to use for DFT calculations.

TYPE:

INTEGER

DEFAULT:

Functional-dependent; see Table 5.3.

OPTIONS:

0	Use SG-0 for H, C, N, and O; SG-1 for all other atoms.
n	Use SG-n for all atoms, $n = 1, 2$, or 3
XY	A string of two six-digit integers X and Y, where X is the number of radial points and Y is the number of angular points where possible numbers of Lebedev angular points, which must be an allowed value from Table 5.2 in Section 5.5.
$-XY$	Similar format for Gauss-Legendre grids, with the six-digit integer X corresponding to the number of radial points and the six-digit integer Y providing the number of Gauss-Legendre angular points, $Y = 2N^2$.

RECOMMENDATION:

Use the default unless numerical integration problems arise. Larger grids may be required for optimization and frequency calculations.

XC_SMART_GRID

Uses SG-0 (where available) for early SCF cycles, and switches to the (larger) target grid specified by XC_GRID for final cycles of the SCF.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE (or 1)	Use the smaller grid for the initial cycles.
FALSE (or 0)	Use the target grid for all SCF cycles.

RECOMMENDATION:

The use of the smart grid can save some time on initial SCF cycles.

XDM

Controls whether to add XDM dispersion to an SCF calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not apply the XDM scheme.
1	Add XDM dispersion as a correction to the SCF energy (and gradient, if appropriate).
2	Add dispersion as a DFT functional and do full SCF.

RECOMMENDATION:

The second (self-consistent) option is only available for XDM6.

XOPT_SEAM_ONLY

Orders an intersection seam search only, no minimization is to be performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Find a point on the intersection seam and stop.

FALSE Perform a minimization of the intersection seam.

RECOMMENDATION:

In systems with a large number of degrees of freedom it might be useful to locate the seam first by setting this option to TRUE and using that geometry as a starting point for the minimization.

XOPT_STATE_1, XOPT_STATE_2

Specify two electronic states the intersection of which will be searched.

TYPE:

[INTEGER, INTEGER, INTEGER]

DEFAULT:

No default value (the option must be specified to run this calculation)

OPTIONS:

[spin, irrep, state]

spin = 0	Addresses states with low spin, see also EE_SINGLETS or IP_STATES,EA_STATES.
spin = 1	Addresses states with high spin, see also EE_TRIPLETS.
irrep	Specifies the irreducible representation to which the state belongs; for example, in the C_{2v} point group, irreps are ordered 1, 2, 3, 4 for A_1, A_2, B_1, and B_2, respectively.
state	Specifies the state number within the irreducible representation, state = 1 means the lowest excited state, state = 2 is the second excited state, *etc.*.
0, 0, -1	Ground state.

RECOMMENDATION:

Only intersections of states with different spin or symmetry can be calculated at this time.

XPOL

Perform a self-consistent XPol calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform an XPol calculation.

FALSE Do not perform an XPol calculation.

RECOMMENDATION:

NONE

Z_EXTRAP_ORDER

Specifies the polynomial order N for Z-vector extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Z-vector extrapolation.

OPTIONS:

N Extrapolate using an Nth-order polynomial ($N > 0$).

RECOMMENDATION:

None

Z_EXTRAP_POINTS

Specifies the number M of old Z-vectors that are retained for use in extrapolation.

TYPE:

INTEGER

DEFAULT:

0 Do not perform response equation extrapolation.

OPTIONS:

M Save M previous Z-vectors for use in extrapolation $(M > N)$

RECOMMENDATION:

Using the default Z-vector convergence settings, a $(M, N) = (4, 2)$ extrapolation was shown to provide the greatest speedup. At this setting, a 2–3-fold reduction in iterations was demonstrated.

References and Further Reading

[1] A. Aldossary, H. Shen, Z. Wang, and M. Head-Gordon. *Chem. Phys. Lett.*, page 141825, 2024. DOI: 10.1016/j.cplett.2024.141825.

[2] A. Ambrosetti, A. M. Reilly, R. A. DiStasio, Jr., and A. Tkatchenko. *J. Chem. Phys.*, 140:18A508, 2014. DOI: 10.1063/1.4865104.

[3] V. Athavale, H.-H. Teh, and J. E. Subotnik. *J. Chem. Phys.*, 155:154105, 2021. DOI: 10.1063/5.0064269.

[4] G. M. J. Barca, A. T. B. Gilbert, and P. M. W. Gill. *J. Chem. Theory Comput.*, 14:1501, 2018. DOI: 10.1021/acs.jctc.7b00994.

[5] G. J. O. Beran, B. Austin, A. Sodt, and M. Head-Gordon. *J. Phys. Chem. A*, 109:9183, 2005. DOI: 10.1021/jp053780c.

[6] C. M. Breneman and K. B. Wiberg. *J. Comput. Chem.*, 11:361, 1990. DOI: 10.1002/jcc.540110311.

[7] P. Bultinck, C. Van Alsenoy, P. W. Ayers, and R. Carbó-Dorca. *J. Chem. Phys.*, 126:144111, 2007. DOI: 10.1063/1.2715563.

[8] E. Caldeweyher, C. Bannwarth, and S. Grimme. *J. Chem. Phys.*, 147:034112, 2017. DOI: 10.1063/1.4993215.

[9] E. Caldeweyher, S. Ehlert, A. Hansen, H. Neugebauer, S. Spicher, C. Bannwarth, and S. Grimme. *J. Chem. Phys.*, 150:154122, 2019. DOI: 10.1063/1.5090222.

[10] E. Caldeweyher, J.-M. Mewes, S. Ehlert, and S. Grimme. *Phys. Chem. Chem. Phys.*, 22:8499, 2020. DOI: 10.1039/d0cp00502a.

[11] J.-D. Chai and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 10:6615, 2008. DOI: 10.1039/b810189b.

[12] D. Das, K. P. Eurenius, E. M. Billings, P. Sherwood, D. C. Chatfield, M. Hodoscek, and B. R. Brooks. *J. Chem. Phys.*, 117:10534, 2002. DOI: 10.1063/1.1520134.

[13] M. Gray, P. E. Bowling, and J. M. Herbert. *J. Phys. Chem. A*, 128:7739, 2024. DOI: 10.1021/acs.jpca.4c00283.

[14] S. Grimme. *J. Comput. Chem.*, 27:1787, 2006. DOI: 10.1002/jcc.20495.

[15] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg. *J. Chem. Phys.*, 132:154104, 2010. DOI: 10.1063/1.3382344.

[16] S. Grimme, S. Ehrlich, and L. Goerigk. *J. Comput. Chem.*, 32:1456, 2011. DOI: 10.1002/jcc.21759.

[17] J. M. Herbert. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 11:e1519, 2021. DOI: 10.1002/wcms.1519.

[18] J. M. Herbert and M. Head-Gordon. *J. Chem. Phys.*, 121:11542, 2004. DOI: 10.1063/1.1814934.

[19] D. Jayatilaka and T. J. Lee. *Chem. Phys. Lett.*, 199:211, 1992. DOI: 10.1016/0009-2614(92)80108-N.

[20] P. Kimber and F. Plasser. *Phys. Chem. Chem. Phys.*, 22:6058, 2020. DOI: 10.1039/D0CP00369G.

[21] P. Kimber and F. Plasser. *J. Chem. Theory Comput.*, 19:2340, 2023. DOI: 10.1021/acs.jctc.3c00125.

[22] B. R. Landry and J. E. Subotnik. *J. Chem. Phys.*, 137:22A513, 2012. DOI: 10.1063/1.4733675.

[23] K. U. Lao and J. M. Herbert. *J. Chem. Theory Comput.*, 14:2955, 2018. DOI: 10.1021/acs.jctc.8b00058.

[24] K. V. Lawler, D. W. Small, and M. Head-Gordon. *J. Phys. Chem. A*, 114:2930, 2010. DOI: 10.1021/jp911009f.

[25] S. Lehtola. *J. Chem. Theory Comput.*, 15:1593, 2019. DOI: 10.1021/acs.jctc.8b01089.

[26] J. H. Van Lenthe, R. Zwaans, H. J. J. Van Dam, and M. F. Guest. *J. Comput. Chem.*, 27:926, 2006. DOI: 10.1002/jcc.20393.

[27] D. S. Levine and M. Head-Gordon. *J. Phys. Chem. Lett.*, 8:1967, 2017. DOI: 10.1021/acs.jpclett.7b00766.

[28] F. Matz and T.-C. Jagau. *J. Chem. Phys.*, 156:114117, 2022. DOI: 10.1063/5.0075646.

[29] F. Matz and T.-C. Jagau. *Mol. Phys.*, 121:e2105270, 2023. DOI: 10.1080/00268976.2022.2105270.

[30] R. McWeeny. *Rev. Mod. Phys.*, 32:335, 1960. DOI: 10.1103/RevModPhys.32.335.

[31] Q. Ou, G. D. Bellchambers, F. Furche, and J. E. Subotnik. *J. Chem. Phys.*, 142:064114, 2015. DOI: 10.1063/1.4906941.

[32] M. J. G. Peach, P. Benfield, T. Helgaker, and D. J. Tozer. *J. Chem. Phys.*, 128:044118, 2008. DOI: 10.1063/1.2831900.

[33] Z. Pei, Q. Ou, Y. Mao, J. Yang, A. de la Lande, F. Plasser W. Liang, Z. Shuai, and Y. Shao. *J. Phys. Chem. Lett.*, 12:2712, 2021. DOI: 10.1021/acs.jpclett.1c00094.

[34] H. Schröder, A. Creon, and T. Schwabe. *J. Chem. Theory Comput.*, 11:3163, 2015. DOI: 10.1021/acs.jctc.5b00400.

[35] D. G. Smith, L. A. Burns, K. Patkowski, and C. D. Sherrill. *J. Phys. Chem. Lett.*, 7:2197, 2016. DOI: 10.1021/acs.jpclett.6b00780.

[36] J. E. Subotnik. *J. Phys. Chem. A*, 114:12083, 2011. DOI: 10.1021/jp206557h.

[37] J. E. Subotnik and N. Shenvi. *J. Chem. Phys.*, 134:024105, 2011. DOI: 10.1063/1.3506779.

[38] A. Tkatchenko and M. Scheffler. *Phys. Rev. Lett.*, 102:073005, 2009. DOI: 10.1103/PhysRevLett.102.073005.

[39] N. V. Tkatchenko and M. Head-Gordon. *J. Chem. Theory Comput.*, 20:9741, 2024. DOI: 10.1021/acs.jctc.4c01105.

[40] E. F. Valeev, V. Coropceanu, D. A. da Silva Filho, S. Salman, and J.-L. Brédas. *J. Am. Chem. Soc.*, 128:9882, 2006. DOI: 10.1021/ja061827h.

[41] O. A. Vydrov and T. Van Voorhis. *J. Chem. Phys.*, 133:244103, 2010. DOI: 10.1063/1.3521275.

[42] A. F. White, M. Head-Gordon, and C. W. McCurdy. *J. Chem. Phys.*, 142:054103, 2015. DOI: 10.1063/1.4906940.

[43] J. Witte, N. Mardirossian, J. B. Neaton, and M. Head-Gordon. *J. Chem. Theory Comput.*, 13:2043, 2017. DOI: 10.1021/acs.jctc.7b00176.

[44] X. Zhang and J. M. Herbert. *J. Chem. Phys.*, 142:064109, 2015. DOI: 10.1063/1.4907376.

Appendix C

Third-party Components

C.1 Introduction

Some Q-CHEM components make use of open-source software, which is listed below along with their respective copyrights and end-user license agreements.

C.2 Armadillo

control with that entity. For the purposes of this definition,
"control" means (i) the power, direct or indirect, to cause the
direction or management of such entity, whether by contract or
otherwise, or (ii) ownership of fifty percent (50%) or more of the
outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity
exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications,
including but not limited to software source code, documentation
source, and configuration files.

"Object" form shall mean any form resulting from mechanical
transformation or translation of a Source form, including but
not limited to compiled object code, generated documentation,
and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or
Object form, made available under the License, as indicated by a
copyright notice that is included in or attached to the work
(an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object
form, that is based on (or derived from) the Work and for which the
editorial revisions, annotations, elaborations, or other modifications
represent, as a whole, an original work of authorship. For the purposes
of this License, Derivative Works shall not include works that remain
separable from, or merely link (or bind by name) to the interfaces of,
the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including
the original version of the Work and any modifications or additions
to that Work or Derivative Works thereof, that is intentionally
submitted to Licensor for inclusion in the Work by the copyright owner
or by an individual or Legal Entity authorized to submit on behalf of
the copyright owner. For the purposes of this definition, "submitted"
means any form of electronic, verbal, or written communication sent
to the Licensor or its representatives, including but not limited to
communication on electronic mailing lists, source code control systems,
and issue tracking systems that are managed by, or on behalf of, the
Licensor for the purpose of discussing and improving the Work, but
excluding communication that is conspicuously marked or otherwise
designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity
on behalf of whom a Contribution has been received by Licensor and
subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of
this License, each Contributor hereby grants to You a perpetual,
worldwide, non-exclusive, no-charge, royalty-free, irrevocable
copyright license to reproduce, prepare Derivative Works of,

publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

```
   END OF TERMS AND CONDITIONS

   APPENDIX: How to apply the Apache License to your work.

      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

C.3 ctx

```
ctx --  Key-value datastructure for organised hierarchical storage
Copyright 2018 Michael F. Herbst

                                 Apache License
                           Version 2.0, January 2004
                        http://www.apache.org/licenses/

   TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

   1. Definitions.

      "License" shall mean the terms and conditions for use, reproduction,
      and distribution as defined by Sections 1 through 9 of this document.

      "Licensor" shall mean the copyright owner or entity authorized by
      the copyright owner that is granting the License.

      "Legal Entity" shall mean the union of the acting entity and all
      other entities that control, are controlled by, or are under common
      control with that entity. For the purposes of this definition,
      "control" means (i) the power, direct or indirect, to cause the
      direction or management of such entity, whether by contract or
```

otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of
   this License, each Contributor hereby grants to You a perpetual,
   worldwide, non-exclusive, no-charge, royalty-free, irrevocable
   (except as stated in this section) patent license to make, have made,
   use, offer to sell, sell, import, and otherwise transfer the Work,
   where such license applies only to those patent claims licensable
   by such Contributor that are necessarily infringed by their
   Contribution(s) alone or by combination of their Contribution(s)
   with the Work to which such Contribution(s) was submitted. If You
   institute patent litigation against any entity (including a
   cross-claim or counterclaim in a lawsuit) alleging that the Work
   or a Contribution incorporated within the Work constitutes direct
   or contributory patent infringement, then any patent licenses
   granted to You under this License for that Work shall terminate
   as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the
   Work or Derivative Works thereof in any medium, with or without
   modifications, and in Source or Object form, provided that You
   meet the following conditions:

   (a) You must give any other recipients of the Work or
       Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices
       stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works
       that You distribute, all copyright, patent, trademark, and
       attribution notices from the Source form of the Work,
       excluding those notices that do not pertain to any part of
       the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its
       distribution, then any Derivative Works that You distribute must
       include a readable copy of the attribution notices contained
       within such NOTICE file, excluding those notices that do not
       pertain to any part of the Derivative Works, in at least one
       of the following places: within a NOTICE text file distributed
       as part of the Derivative Works; within the Source form or
       documentation, if provided along with the Derivative Works; or,
       within a display generated by the Derivative Works, if and
       wherever such third-party notices normally appear. The contents
       of the NOTICE file are for informational purposes only and
       do not modify the License. You may add Your own attribution
       notices within Derivative Works that You distribute, alongside
       or as an addendum to the NOTICE text from the Work, provided
       that such additional attribution notices cannot be construed
       as modifying the License.

   You may add Your own copyright statement to Your modifications and
   may provide additional or different license terms and conditions
   for use, reproduction, or distribution of Your modifications, or

for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

```
      To apply the Apache License to your work, attach the following
      boilerplate notice, with the fields enclosed by brackets "[]"
      replaced with your own identifying information. (Don't include
      the brackets!)  The text should be enclosed in the appropriate
      comment syntax for the file format. We also recommend that a
      file or class name and description of purpose be included on the
      same "printed page" as the copyright notice for easier
      identification within third-party archives.

   Copyright [yyyy] [name of copyright owner]

   Licensed under the Apache License, Version 2.0 (the "License");
   you may not use this file except in compliance with the License.
   You may obtain a copy of the License at

       http://www.apache.org/licenses/LICENSE-2.0

   Unless required by applicable law or agreed to in writing, software
   distributed under the License is distributed on an "AS IS" BASIS,
   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
   See the License for the specific language governing permissions and
   limitations under the License.
```

C.4 dftd4

```
Copyright (C) 2017-2019 Stefan Grimme, Sebastian Ehlert, Eike Caldeweyher

dftd4 is free software: you can redistribute it and/or modify it under
the terms of the GNU Lesser General Public License as published by
the Free Software Foundation, either version 3 of the License, or
(at your option) any later version.

dftd4 is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License
along with dftd4.  If not, see <https://www.gnu.org/licenses/>.

                   GNU LESSER GENERAL PUBLIC LICENSE
                       Version 3, 29 June 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <https://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.


  This version of the GNU Lesser General Public License incorporates
the terms and conditions of version 3 of the GNU General Public
```

License, supplemented by the additional permissions listed below.

0. Additional Definitions.

As used herein, "this License" refers to version 3 of the GNU Lesser General Public License, and the "GNU GPL" refers to version 3 of the GNU General Public License.

"The Library" refers to a covered work governed by this License, other than an Application or a Combined Work as defined below.

An "Application" is any work that makes use of an interface provided by the Library, but which is not otherwise based on the Library. Defining a subclass of a class defined by the Library is deemed a mode of using an interface provided by the Library.

A "Combined Work" is a work produced by combining or linking an Application with the Library. The particular version of the Library with which the Combined Work was made is also called the "Linked Version".

The "Minimal Corresponding Source" for a Combined Work means the Corresponding Source for the Combined Work, excluding any source code for portions of the Combined Work that, considered in isolation, are based on the Application, and not on the Linked Version.

The "Corresponding Application Code" for a Combined Work means the object code and/or source code for the Application, including any data and utility programs needed for reproducing the Combined Work from the Application, but excluding the System Libraries of the Combined Work.

1. Exception to Section 3 of the GNU GPL.

You may convey a covered work under sections 3 and 4 of this License without being bound by section 3 of the GNU GPL.

2. Conveying Modified Versions.

If you modify a copy of the Library, and, in your modifications, a facility refers to a function or data to be supplied by an Application that uses the facility (other than as an argument passed when the facility is invoked), then you may convey a copy of the modified version:

a) under this License, provided that you make a good faith effort to ensure that, in the event an Application does not supply the function or data, the facility still operates, and performs whatever part of its purpose remains meaningful, or

b) under the GNU GPL, with none of the additional permissions of this License applicable to that copy.

3. Object Code Incorporating Material from Library Header Files.

```
  The object code form of an Application may incorporate material from
a header file that is part of the Library.  You may convey such object
code under terms of your choice, provided that, if the incorporated
material is not limited to numerical parameters, data structure
layouts and accessors, or small macros, inline functions and templates
(ten or fewer lines in length), you do both of the following:

   a) Give prominent notice with each copy of the object code that the
   Library is used in it and that the Library and its use are
   covered by this License.

   b) Accompany the object code with a copy of the GNU GPL and this license
   document.

  4. Combined Works.

  You may convey a Combined Work under terms of your choice that,
taken together, effectively do not restrict modification of the
portions of the Library contained in the Combined Work and reverse
engineering for debugging such modifications, if you also do each of
the following:

   a) Give prominent notice with each copy of the Combined Work that
   the Library is used in it and that the Library and its use are
   covered by this License.

   b) Accompany the Combined Work with a copy of the GNU GPL and this license
   document.

   c) For a Combined Work that displays copyright notices during
   execution, include the copyright notice for the Library among
   these notices, as well as a reference directing the user to the
   copies of the GNU GPL and this license document.

   d) Do one of the following:

       0) Convey the Minimal Corresponding Source under the terms of this
       License, and the Corresponding Application Code in a form
       suitable for, and under terms that permit, the user to
       recombine or relink the Application with a modified version of
       the Linked Version to produce a modified Combined Work, in the
       manner specified by section 6 of the GNU GPL for conveying
       Corresponding Source.

       1) Use a suitable shared library mechanism for linking with the
       Library.  A suitable mechanism is one that (a) uses at run time
       a copy of the Library already present on the user's computer
       system, and (b) will operate properly with a modified version
       of the Library that is interface-compatible with the Linked
       Version.

   e) Provide Installation Information, but only if you would otherwise
```

be required to provide such information under section 6 of the
GNU GPL, and only to the extent that such information is
necessary to install and execute a modified version of the
Combined Work produced by recombining or relinking the
Application with a modified version of the Linked Version. (If
you use option 4d0, the Installation Information must accompany
the Minimal Corresponding Source and Corresponding Application
Code. If you use option 4d1, you must provide the Installation
Information in the manner specified by section 6 of the GNU GPL
for conveying Corresponding Source.)

5. Combined Libraries.

You may place library facilities that are a work based on the Library side by side in a single library together with other library facilities that are not Applications and are not covered by this License, and convey such a combined library under terms of your choice, if you do both of the following:

a) Accompany the combined library with a copy of the same work based on the Library, uncombined with any other library facilities, conveyed under the terms of this License.

b) Give prominent notice with the combined library that part of it is a work based on the Library, and explaining where to find the accompanying uncombined form of the same work.

6. Revised Versions of the GNU Lesser General Public License.

The Free Software Foundation may publish revised and/or new versions of the GNU Lesser General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Library as you received it specifies that a certain numbered version of the GNU Lesser General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that published version or of any later version published by the Free Software Foundation. If the Library as you received it does not specify a version number of the GNU Lesser General Public License, you may choose any version of the GNU Lesser General Public License ever published by the Free Software Foundation.

If the Library as you received it specifies that a proxy can decide whether future versions of the GNU Lesser General Public License shall apply, that proxy's public statement of acceptance of any version is permanent authorization for you to choose that version for the Library.

C.5 libecpint

```
MIT License

Copyright (c) 2017 Robert A. Shaw

Permission is hereby granted, free of charge, to any person obtaining a copy
of this software and associated documentation files (the "Software"), to deal
in the Software without restriction, including without limitation the rights
to use, copy, modify, merge, publish, distribute, sublicense, and/or sell
copies of the Software, and to permit persons to whom the Software is
furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all
copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR
IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY,
FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE
AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER
LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM,
OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE
SOFTWARE.
```

C.6 libefp

```
Libefp was written by Ilya Kaliman.
The following people contributed to libefp. Thanks!

Lori A. Burns
Dmitry Morozov

Copyright (c) 2012-2017 Ilya Kaliman

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

1. Redistributions of source code must retain the above copyright
   notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright
   notice, this list of conditions and the following disclaimer in the
   documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS "AS IS" AND
ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE
IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
ARE DISCLAIMED. IN NO EVENT SHALL AUTHOR OR CONTRIBUTORS BE LIABLE
FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
```

C.7 libtensor

C.8 libxm