Q-Chem User's Manual

Version 4.2
May, 2014

Version 4.2

May, 2014

Q-CHEM User's Guide

This version was edited by:
Prof. Anna Krylov

Version 4.1 was edited by:
Prof. Anna Krylov and Prof. Martin Head-Gordon

Version 4.0.1 was edited by:
Dr. Andrew Gilbert, Dr. Emil Proynov, and Prof. Anna Krylov

Version 4.0 was edited by:
Dr. Emil Proynov, Dr. Jing Kong, and Prof. John Herbert

Version 3.2 was edited by:
Dr. Yihan Shao

Version 3.1 was edited by:
Dr. Andrew Gilbert

Versions 3.1 and 3.2 are revisions and expansions based on version 2.1, which was written by:
Dr. Jeremy Dombroski
Prof. Martin Head-Gordon
Dr. Andrew Gilbert

The contributions of individual developers to each version are highlighted in the New Features Section

Published by:
Q-Chem, Inc.
6601 Owens Dr.
Suite 105
Pleasanton, CA 94588

Customer Support:

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

The information in this document applies to version 4.2 of Q-CHEM.
This document version generated on December 4, 2014.

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 environment, an ASCII file editor and a basic understanding of quantum chemistry.

The manual is divided into 12 chapters and 3 appendices, which are briefly summarized below. 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 chapter 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, while recommended, is not necessary.

1.2 Chapter Summaries

Chapter 1: General overview of the Q-CHEM program, its features and capabilities, the people behind it, and contact information.

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

Chapter 3: Basic attributes of the Q-CHEM command line input.

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

Chapter 5: Running wavefunction-based correlation methods for ground states.

Chapter 6: Running calculations for excited states and open-shell species.

Chapter 7: Using Q-CHEM's built-in basis sets and running user-defined basis sets.

Chapter 8: Using Q-CHEM's effective core potential capabilities.

Chapter 9: Options available for exploring potential energy surfaces, such as determining critical points (transition states and local minima) as well as molecular dynamics options.

Chapter 10: Techniques available for computing molecular properties and performing wavefunction analysis.

Chapter 11: Techniques for molecules in complex environments (e.g., bulk solution, proteins) and intermolecular interactions; Chemical solvent models, QM/MM and Effective Fragment Potential method, density embedding.

Chapter 12: Methods based on absolutely-localized molecular orbitals and fragment-based methods.

Appendix A: OPTIMIZE package used in Q-CHEM for determining molecular geometry critical points.

Appendix B: Q-CHEM's AOINTS library, which contains some of the fastest two-electron integral code currently available.

Appendix C: Quick reference section.

1.3 Contact Information

For general information regarding broad aspects and features of the Q-CHEM program, see Q-CHEM's home page (http://www.q-chem.com).

1.3.1 Customer Support

Full customer support is promptly provided though telephone or email for those customers who have purchased Q-CHEM's maintenance contract. The maintenance contract offers free customer support and discounts on future releases and updates. For details of the maintenance contract please see Q-CHEM's home page (http://www.q-chem.com).

1.4 Q-CHEM, Inc.

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 (OSU), 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. The current group of developers consist of more than 100 people in 9 countries. A brief history of Q-CHEM is given in a recent Software Focus article[1], *"Q-Chem: An Engine for Innovation"*.

1.5 Company Mission

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

1.6 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 far large molecular structures than previously possible and with no loss in accuracy, thereby bringing the power of quantum chemistry to critical research projects for which this tool was previously unavailable.

1.6.1 New Features in Q-CHEM 4.2

Q-CHEM 4.2 provides several bug fixes, performance enhancements, and the following new features:

- Input file changes
  - New keyword METHOD simplifies input in most cases by replacing the pair of keywords EXCHANGE and CORRELATION (see Section 3.8).
  - Keywords for requesting excited state calculations have changed. Consult Chapter 6 for more details.
  - Keywords for solvation models have changed. Consult Section 11.2 for details.
- New features for NMR calculations including spin-spin couplings, see Section 10.15 (Kussmann, Luenser, Ochsenfeld).
- New built-in basis sets, see Chapter 7.
- New features and performance improvements in EOM-CC:

  - EOM-CC methods extended to treat metastable electronic states (resonances) via complex-scaling and complex absorbing potential approaches, see section 6.7.5 (Zuev, Bravaya, Jagau, Shao, Krylov).
  - New features are added to EOM-CC iterative solvers, such as methods for interior eigenvalues and user-specified guesses, See section 6.7.8 (Zuev).
  - Multicore parallel code for EOM-CC/CC gradients; improved (T) performance.

- New features and performance improvements in ADC methods (Wormit, Dreuw):
  - ADC methods can now employ RI approximations that allows one to tackle larger system and reduces the costs, see Section 6.8.2.
  - SOS-ADC methods have been implemented, see Section 6.8.3.
  - State-to-state properties for ADC methods have been implemented, see Section 6.8.4.
- SM12 implicit solvation model, see Section 11.2.7.1 (Marenich, Truhlar, Shao).
- NBO 6 interface, see Section 10.3.
- MECP optimization is now available with SOS-CIS(D) and TD-DFT, see Section 9.4 (Zhang, Herbert).
- ROKS method for Δ-SCF calculations of excited states, see Section 6.5 (Kowalczyk, Van Voorhis).
- PARI-K fast exchange algorithm, see section 4.6.9 (Manzer, Head-Gordon).
- FRAGMO initial guess for SCF methods, see Section 12.3.
- Pseudo-fractional occupation number method for improved SCF convergence in small-gap systems, see Section 4.5.8 (Lambrecht).
- Density embedding scheme, see Section 11.6 (Albrecht, Berquist, Lambrecht).
- New features and enhancements in fragment-based many-body expansion methods:
  - XSAPT(KS)+D: A dispersion corrected version of symmetry-adapted perturbation theory for fast and accurate calculation of interaction energies in non-covalent clusters, see Section 12.9 (Lao and Herbert).
  - A preliminary implementation of the many-body body expansion and the fragment molecular orbital methods for clusters, see Section 12.10 (Lao and Herbert).
  - Periodic boundary conditions with proper Ewald summation (energies only), see Section 11.3 (Holden and Herbert).

1.6.2 New Features in Q-CHEM 4.1

Q-CHEM 4.1 provides several bug fixes, performance enhancements, and the following new features:

- Fundamental algorithms
  - Improved parallel performance at all levels including new OpenMP capabilities for SCF/DFT, MP2, integral transformation and coupled cluster theory (Zhengting Gan, Evgeny Epifanovsky, Matt Goldey, Yihan Shao; see Section 2.7.1).

  - Significantly enhanced ECP capabilities, including availability of gradients and frequencies in all basis sets for which the energy can be evaluated (Yihan Shao and Martin Head-Gordon; see Chapter 8).

- Self-Consistent Field and Density Functional Theory capabilities

  - Numerous DFT enhancements and new features: TD-DFT energy with M06, M08, and M11-series of functionals; XYGJ-OS analytical energy gradient;
  - TD-DFT/C-PCM excitation energies, gradient, and Hessian (Jie Liu, W. Liang; Section 6.3.4).
  - Additional features in the Maximum Overlap Method (MOM) approach for converging difficult DFT and SCF calculations (Nick Besley; Section 4.5.6).

- Wave function correlation capabilities

  - Resolution-of-identity/Cholesky decomposition implementation of all coupled-cluster and equation-of-motion methods enabling applications to larger systems, reducing disk/memory requirements, and improving performance (see Sections 5.7.5 and 5.7.6; Evgeny Epifanovsky, Xintian Feng, Dmitri Zuev, Yihan Shao, Anna Krylov).
  - Attenuated MP2 theory in the aug-cc-pVDZ and aug-cc-pVTZ basis sets, which truncate 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 sets (Matt Goldey and Martin Head-Gordon, Section 5.6.1).
  - Extended RAS-nSF methodology for ground and excited states involving strong non-dynamical correlations (see Section 6.9, Paul Zimmerman, David Casanova, Martin Head-Gordon).
  - Coupled cluster valence bond (CCVB) method for describing molecules with strong spin correlations (David Small and Martin Head-Gordon; see Section 5.14.2).

- Walking on potential energy surfaces

  - Potential energy surface scans (Yihan Shao, Section 9.3).
  - Improvements in automatic transition structure search algorithms by the freezing string method, including the ability to perform such calculations without a Hessian calculation (Shaama Sharada, Martin Head-Gordon, Section 9.7).
  - Enhancements to partial Hessian vibrational analysis (Nick Besley; Section 10.12.1).

- Calculating and characterizing inter- and intra-molecular interactions

  - Extension of EFP to macromolecules: fEFP approach (Adele Laurent, Debashree Ghosh, Anna Krylov, Lyudmila Slipchenko, see Section 11.5.3).
  - 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. (Leif Jacobson, John Herbert; Section 12.8). Q-CHEM features an efficient resolution-of-identity (RI) version of the "SAPT0" approximation, based on 2nd-order perturbation theory for the intermolecular interaction.
  - The "explicit polarization" (XPol) monomer-based SCF calculations to compute many-body polarization effects in linear-scaling time via charge embedding (Section 12.7), 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 (Leif Jacobson, John Herbert; Section 12.9).

  - Extension of the absolutely-localized-molecular-orbital-based energy decomposition analysis to unrestricted cases (Section 12.5, Paul Horn and Martin Head-Gordon)
  - Calculations of populations of effectively unpaired electrons in low-spin state within 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 (Emil Proynov; Section 10.18).

- Calculations of quantum transport including electron transmission functions and electron tunneling currents under applied bias voltage (Section 10.19, Barry Dunietz, Nicolai Sergueev).
- Searchable online version of our pdf manual.

1.6.3 New Features in Q-CHEM 4.0.1

Q-CHEM 4.0.1 provides several bug fixes, performance enhancements, and the following new features:

- Remote submission capability in IQMOL (Andrew Gilbert).
- Scaled nuclear charge and charged cage stabilization capabilities (Tomasz Kuś, Anna Krylov, Section 6.7.6).
- 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 (see Section 10.17).

1.6.4 New Features in Q-CHEM 4.0

Q-CHEM 4.0 provides the following new features and upgrades:

- Exchange-Correlation Functionals
  - Density functional dispersion with implementation of the efficient Becke and Johnson's XDM model in the analytic form (Zhengting Gan, Emil Proynov, Jing Kong; Section 4.3.7).
  - Implementation of the van der Waals density functionals vdW-DF-04 and vdW-DF-10 of Langreth and co-workers (Oleg Vydrov; Section 4.3.5).
  - VV09 and VV10, new analytic dispersion functionals (Oleg Vydrov, Troy Van Voorhis; Section 4.3.5).
  - Implementation of DFT-D3 Methods for improved noncovalent interactions (Shan-Ping Mao, Jeng-Da Chai; Section 4.3.8).
  - ωB97X-2, a double-hybrid functional based on long range corrected B97 functional with improved account for medium and long range interactions (Jeng-Da Chai, Martin Head-Gordon; Section 4.3.9).
  - XYGJ-OS, a double-hybrid functional for predictions of nonbonding interactions and thermochemistry at nearly chemical accuracy (Igor Zhang, Xin Xu, William A. Goddard, III, Yousung Jung; Section 4.3.9).
  - Calculation of near-edge X-ray absorption with short-range corrected DFT (Nick Besley).
  - Improved TDDFT prediction with implementation of asymptotically corrected exchange-correlation potential (TDDFT/TDA with LB94) (Yu-Chuan Su, Jeng-Da Chai; Section 4.3.10).

  - Nondynamic correlation in DFT with efficient RI implementation of Becke-05 model in fully analytic formulation (Emil Proynov, Yihan Shao, Fenglai Liu, Jing Kong; Section 4.3.3).
  - Implementation of meta-GGA functionals TPSS and its hybrid version TPSSh (Fenglai Liu) and the rPW86 GGA functional (Oleg Vydrov).
  - Implementation of double hybrid functional B2PLYP-D (Jeng-Da Chai).
  - Implementation of Mori-Sánchez–Cohen–Yang (MCY2) hyper-GGA functional (Fenglai Liu).
  - SOGGA, SOGGA11 and SOGGA11-X family of GGA functionals (Roberto Peverati, Yan Zhao, Don Truhlar).
  - M08-HX and M08-SO suites of high HF exchange meta-GGA functionals (Yan Zhao, Don Truhlar).
  - M11-L and M11 suites of meta-GGA functionals (Roberto Peverati, Yan Zhao, Don Truhlar).

- DFT Algorithms
  - Fast numerical integration of exchange-correlation with mrXC (multiresolution exchange-correlation) (Shawn Brown, Laszlo Fusti-Molnar, Nicholas J. Russ, Chun-Min Chang, Jing Kong; Section 4.6.7).
  - Efficient computation of the exchange-correlation part of the dual basis DFT (Zhengting Gan, Jing Kong; Section 4.4.5).
  - Fast DFT calculation with "triple jumps" between different sizes of basis set and grid and different levels of functional (Jia Deng, Andrew Gilbert, Peter M. W. Gill; Section 4.8).
  - Faster DFT and HF calculation with atomic resolution of the identity (ARI) algorithms (Alex Sodt, Martin Head-Gordon).

- POST-HF: Coupled Cluster, Equation of Motion, Configuration Interaction, and Algebraic Diagrammatic Construction Methods
  - Significantly enhanced coupled-cluster code rewritten for better performance and multicore systems for many modules (energy and gradient for CCSD, EOM-EE/SF/IP/EA-CCSD, CCSD(T) energy) (Evgeny Epifanovsky, Michael Wormit, Tomasz Kus, Arik Landau, Dmitri Zuev, Kirill Khistyaev, Ilya Kaliman, Anna Krylov, Andreas Dreuw; Chapters 5 and 6). This new code is named CCMAN2.
  - Fast and accurate coupled-cluster calculations with frozen natural orbitals (Arik Landau, Dmitri Zuev, Anna Krylov; Section 5.10).
  - Correlated excited states with the perturbation-theory based, size consistent ADC scheme of second order (Michael Wormit, Andreas Dreuw; Section 6.8).
  - Restricted active space spin flip method for multiconfigurational ground states and multi-electron excited states (Paul Zimmerman, Franziska Bell, David Casanova, Martin Head-Gordon, Section 6.2.4).

- POST-HF: Strong Correlation
  - Perfect Quadruples and Perfect Hextuples methods for strong correlation problems (John Parkhill, Martin Head-Gordon, Section 5.9.4).
  - Coupled Cluster Valence Bond (CCVB) and related methods for multiple bond breaking (David Small, Keith Lawler, Martin Head-Gordon, Section 5.14).

- DFT Excited States and Charge Transfer
  - Nuclear gradients of excited states with TDDFT (Yihan Shao, Fenglai Liu, Zhengting Gan, Chao-Ping Hsu, Andreas Dreuw, Martin Head-Gordon, Jing Kong; Section 6.3.1).
  - Direct coupling of charged states for study of charge transfer reactions (Zhi-Qiang You, Chao-Ping Hsu, Section 10.17.2.
  - Analytical excited-state Hessian in TDDFT within Tamm-Dancoff approximation (Jie Liu, Wanzhen Liang; Section 6.3.5).
  - Obtaining an excited state self-consistently with MOM, the Maximum Overlap Method (Andrew Gilbert, Nick Besley, Peter M. W. Gill; Section 6.4).
  - Calculation of reactions with configuration interactions of charge-constrained states with constrained DFT (Qin Wu, Benjamin Kaduk, Troy Van Voorhis; Section 4.9).
  - Overlap analysis of the charge transfer in a excited state with TDDFT (Nick Besley; Section 6.3.2).
  - Localizing diabatic states with Boys or Edmiston-Ruedenberg localization scheme for charge or energy transfer (Joe Subotnik, Ryan Steele, Neil Shenvi, Alex Sodt; Section 10.17.1.2).
  - Implementation of non-collinear formulation extends SF-TDDFT to a broader set of functionals and improves its accuracy (Yihan Shao, Yves Bernard, Anna Krylov; Section 6.3).
- Solvation and Condensed Phase Modeling
  - Smooth free energy surface for solvated molecules via switching/Gaussian polarizable continuum models (SWIG-PCMs), for QM and QM/MM calculations, including a linear-scaling QM/MM/PCM algorithm (Adrian Lange and John Herbert; Sections 11.2.2 and 11.2.4).
  - The original COSMO solvation model by Klamt and Schüürmann with DFT energy and gradient (ported by Yihan Shao; Section **??**).
  - Accurate and fast energy computation for large systems including polarizable explicit solvation for ground and excited states with effective fragment potential using DFT/TDDFT, CCSD/EOM-CCSD, as well as CIS and CIS(D); library of effective fragments for common solvents; energy gradient for EFP-EFP systems (Vitalii Vanovschi, Debashree Ghosh, Ilya Kaliman, Dmytro Kosenkov, Chris Williams, John Herbert, Mark Gordon, Michael Schmidt, Yihan Shao, Lyudmila Slipchenko, Anna Krylov; Section 11.5).
- Optimizations, Vibrations, and Dynamics
  - Freezing and Growing String Methods for efficient automatic reaction path finding (Andrew Behn, Paul Zimmerman, Alex Bell, Martin Head-Gordon, Section 9.6).
  - Improved robustness of IRC code (intrinsic reaction coordinate following) (Martin Head-Gordon).
  - Exact, quantum mechanical treatment of nuclear motions at equilibrium with path integral methods (Ryan Steele; Section 9.10).
  - Calculation of local vibrational modes of interest with partial Hessian vibrational analysis (Nick Besley; Section 10.12.1).
  - Ab initio dynamics with extrapolated Z-vector techniques for MP2 and/or dual-basis methods (Ryan Steele; Section 4.7.5).
  - Quasiclassical ab initio molecular dynamics (Daniel Lambrecht, Martin Head-Gordon, Section 9.9.4).

- Fragment-Based Methods
  - Symmetry-adapted perturbation theory (SAPT) for computing and analyzing dimer interaction energies (Leif Jacobson, Mary Rohrdanz, and John Herbert; Section 12.8).
  - A many-body generalization of SAPT, with empirical dispersion corrections for high accuracy and low cost in large clusters (Leif Jacobson, Ka Un Lao, and John Herbert; Section 12.9).
  - Methods based on the a truncated many-body expansion, including the fragment molecular orbital (FMO) method (Ka Un Lao and John Herbert; Section 12.10).
- Properties and Wavefunction Analysis
  - Analysis of metal oxidation states via localized orbital bonding analysis (Alex Thom, Eric Sundstrom, Martin Head-Gordon; Section 10.2.4).
  - Hirshfeld population analysis (Sina Yeganeh; Section 10.2.1).
  - Visualization of noncovalent bonding using Johnson and Yang's algorithm (Yihan Shao; Section 10.5.5).
  - ESP on a grid for transition density (Yihan Shao; Section 10.6).
- Support for Modern Computing Platforms
  - Efficient multicore parallel CC/EOM/ADC methods.
  - Better performance for multicore systems with shared-memory parallel DFT/HF (Zhengting Gan, Yihan Shao, Jing Kong) and RI-MP2 (Matthew Goldey, Martin Head-Gordon; Section 5.13).
  - Accelerating RI-MP2 calculation with GPU (graphic processing unit) (Roberto Olivares-Amaya, Mark Watson, Richard Edgar, Leslie Vogt, Yihan Shao, Alan Aspuru-Guzik; Section 5.5.4).
- Graphical User Interface
  - Input file generation, Q-CHEM job submission, and visualization is supported by IQMOL, a fully integrated GUI developed by Andrew Gilbert from the Australian National University. IQMOL is a free software and does not require purchasing a license. See http://www.iqmol.org for details and installation instructions.
  - Other graphic interfaces are also available.

1.6.5 New Features in Q-CHEM 3.2

Q-CHEM 3.2 provides the following important upgrades:

- Several new DFT options:
  - Long-ranged corrected (LRC) functionals (Dr. Rohrdanz, Prof. Herbert)
  - Baer-Neuhauser-Livshits (BNL) functional (Prof. Baer, Prof. Neuhauser, Dr. Livshits)
  - Variations of ωB97 Functional (Dr. Chai, Prof. Head-Gordon)
  - Constrained DFT (CDFT) (Dr. Wu, Cheng, Prof. Van Voorhis)
  - Grimme's empirical dispersion correction (Prof. Sherrill)

- Default XC grid for DFT:
  - Default xc grid is now SG-1. It used to be SG-0 before this release.
- Solvation models:
  - SM8 model (energy and analytical gradient) for water and organic solvents (Dr. Marenich, Dr. Olson, Dr. Kelly, Prof. Cramer, Prof. Truhlar)
  - Updates to Onsager reaction-field model (Mr. Cheng, Prof. Van Voorhis, Dr. Thanthiriwatte, Prof. Gwaltney)
- Intermolecular interaction analysis (Dr. Khaliullin, Prof. Head-Gordon):
  - SCF with absolutely localized molecular orbitals for molecule interaction (SCF-MI)
  - Roothaan-step (RS) correction following SCF-MI
  - Energy decomposition analysis (EDA)
  - Complimentary occupied-virtual pair (COVP) analysis for charge transfer
  - Automated basis-set superposition error (BSSE) calculation
- Electron transfer analysis (Dr. You, Prof. Hsu)
- Relaxed constraint algorithm (RCA) for converging SCF (Mr. Cheng, Prof. Van Voorhis)
- G3Large basis set for transition metals (Prof. Rassolov)
- New MP2 options:
  - dual-basis RIMP2 energy and analytical gradient (Dr. Steele, Mr. DiStasio Jr., Prof. Head-Gordon)
  - O2 energy and gradient (Dr. Lochan, Prof. Head-Gordon)
- New wavefunction-based methods for efficiently calculating excited state properties (Dr. Casanova, Prof. Rhee, Prof. Head-Gordon):
  - SOS-CIS(D) energy for excited states
  - SOS-CIS(D_0) energy and gradient for excited states
- Coupled-cluster methods (Dr. Pieniazek, Dr. Epifanovsky, Prof. Krylov):
  - IP-CISD energy and gradient
  - EOM-IP-CCSD energy and gradient
  - OpenMP for parallel coupled-cluster calculations
- QM/MM methods (Dr. Woodcock, Ghysels, Dr. Shao, Dr. Kong, Dr. Brooks)
  - QM/MM full Hessian evaluation
  - QM/MM mobile-block Hessian (MBH) evaluation
  - Description for MM atoms with Gaussian-delocalized charges
- Partial Hessian method for vibrational analysis (Dr. Besley)
- Wavefunction analysis tools:
  - Improved algorithms for computing localized orbitals (Dr. Subotnik, Prof. Rhee, Dr. Thom, Mr. Kurlancheek, Prof. Head-Gordon)
  - Distributed multipole analysis (Dr. Vanovschi, Prof. Krylov, Dr. Williams, Prof. Herbert)
  - Analytical Wigner intracule (Dr. Crittenden, Prof. Gill)

1.6.6 New Features in Q-CHEM 3.1

Q-CHEM 3.1 provides the following important upgrades:

- Several new DFT functional options:
  - The nonempirical GGA functional PBE (from the open DF Repository distributed by the QCG CCLRC Daresbury Lab., implemented in Q-CHEM 3.1 by Dr. E. Proynov).
  - M05 and M06 suites of meta-GGA functionals for more accurate predictions of various types of reactions and systems (Dr. Yan Zhao, Dr. Nathan E. Schultz, Prof. Don Truhlar).
- A faster correlated excited state method: RI-CIS(D) (Dr. Young Min Rhee).
- Potential energy surface crossing minimization with CCSD and EOM-CCSD methods (Dr. Evgeny Epifanovsky).
- Dyson orbitals for ionization from the ground and excited states within CCSD and EOM-CCSD methods (Dr. Melania Oana).

1.6.7 New Features in Q-CHEM 3.0

Q-CHEM 3.0 includes many new features, along with many enhancements in performance and robustness over previous versions. Below is a list of some of the main additions, and who is primarily to thank for implementing them. Further details and references can be found in the official citation for Q-CHEM (see Section 1.8).

- Improved two-electron integrals package (Dr. Yihan Shao):
  - Code for the Head-Gordon–Pople algorithm rewritten to avoid cache misses and to take advantage of modern computer architectures.
  - Overall increased in performance, especially for computing derivatives.
- Fourier Transform Coulomb method (Dr. Laszlo Fusti-Molnar):
  - Highly efficient implementation for the calculation of Coulomb matrices and forces for DFT calculations.
  - Linear scaling regime is attained earlier than previous linear algorithms.
  - Present implementation works well for basis sets with high angular momentum and diffuse functions.
- Improved DFT quadrature evaluation:
  - Incremental DFT method avoids calculating negligible contributions from grid points in later SCF cycles (Dr. Shawn Brown).
  - Highly efficient SG-0 quadrature grid with approximately half the accuracy and number of grid points as the SG-1 grid (Siu-Hung Chien).
- Dual basis self-consistent field calculations (Dr. Jing Kong, Ryan Steele):
  - Two stage SCF calculations can reduce computational cost by an order of magnitude.

  - Customized basis subsets designed for optimal projection into larger bases.

- Auxiliary basis expansions for MP2 calculations:
  - RI-MP2 and SOS-MP2 energies (Dr. Yousung Jung) and gradients (Robert A. DiStasio Jr.).
  - RI-TRIM MP2 energies (Robert A. DiStasio Jr.).
  - Scaled opposite spin energies and gradients (Rohini Lochan).

- 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 (Prof. Anna Krylov).
  - Available for RHF/UHF/ROHF references.
  - Analytic gradients and properties calculations (permanent and transition dipoles *etc.*).
  - Full use of Abelian point-group symmetry.
  - Singlet strongly orthogonal geminal (SSG) methods (Dr. Vitaly Rassolov).

- Coupled-cluster perfect-paring methods (Prof. Martin Head-Gordon):
  - Perfect pairing (PP), imperfect pairing (IP) and restricted pairing (RP) models.
  - PP(2) Corrects for some of the worst failures of MP2 theory.
  - Useful in the study of singlet molecules with diradicaloid character.
  - Applicable to systems with more than 100 active electrons.

- Hybrid quantum mechanics /molecular mechanics (QM/MM) methods:
  - Fixed point-charge model based on the AMBER force field.
  - Two-layer ONIOM model (Dr. Yihan Shao).
  - Integration with the MOLARIS simulation package (Dr. Edina Rosta).
  - Q-CHEM/CHARMM interface (Dr. Lee Woodcock)

- New continuum solvation models (Dr. Shawn Brown):
  - Surface and Simulation of Volume Polarization for Electrostatics [SS(V)PE] model.
  - Available for HF and DFT calculations.

- New transition structure search algorithms (Andreas Heyden and Dr. Baron Peters):
  - Growing string method for finding transition states.
  - Dimer Method which does not use the Hessian and is therefore useful for large systems.

- *Ab Initio* Molecular dynamics (Dr. John Herbert):
  - Available for SCF wavefunctions (HF, DFT).
  - Direct Born-Oppenheimer molecular dynamics (BOMD).
  - Extended Lagrangian *ab initio* molecular dynamics (ELMD).

- Linear scaling properties for large systems (Jörg Kussmann and Prof. Christian Ochsenfeld):
  - NMR chemical shifts.

  - Static and dynamic polarizabilities.
  - Static hyperpolarizabilities, optical rectification and electro-optical Pockels effect.

- Anharmonic frequencies (Dr. Ching Yeh Lin):
  - Efficient implementation of high-order derivatives
  - Corrections via perturbation theory (VPT) or configuration interaction (VCI).
  - New transition optimized shifted Hermite (TOSH) method.

- Wavefunction analysis tools:
  - Spin densities at the nuclei (Dr. Vitaly Rassolov).
  - Efficient calculation of localized orbitals.
  - Optimal atomic point-charge models for densities (Andrew Simmonett).
  - Calculation of position, momentum and Wigner intracules (Dr Nick Besley and Dr. Darragh O'Neill).

- Graphical user interface 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 (www.iqmol.org).
  - Support for the public domain version of WEBMO (see www.webmo.net).
  - Seamless integration with the SPARTAN package (see www.wavefun.com).
  - Support for the public domain version of AVOGADRO (see: http://avogadro.openmolecules.net/wiki/Get_Avogadro).
  - Support the MOLDEN molecular orbital viewer (see www.cmbi.ru.nl/molden).
  - Support the JMOL package (see http://sourceforge.net/project/showfiles.php?group_id= 23629&release_id=66897).

1.6.8 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) (Dr. Chris White).
  - Second-generation J-engine and J-force engine (Dr. Yihan Shao).
  - LinK for exchange energies and forces (Dr. Christian Ochsenfeld and Dr. Chris 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 (Dr. Ross Adamson).
  - B3LYP, B3P and user-definable hybrid functionals.
  - Analytical gradients and analytical frequencies.

  - SG-0 standard quadrature grid (Siu-Hung Chien).
  - Lebedev grids up to 5294 points (Dr. Shawn Brown).

- High level wavefunction-based electron correlation methods (Chapter 5):
  - 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 (Dr. Vitaly 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 (Prof. Anna Krylov).
  - Coupled-cluster excited-state geometry optimizations.
  - Coupled-cluster property calculations (dipoles, transition dipoles).
  - Spin-flip calculations for CCSD and TDDFT excited states (Prof. Anna Krylov and Dr. Yihan Shao).

- High performance geometry and transition structure optimization (Jon 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
  - 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 (Robert Doerksen).
  - NMR chemical shifts (Joerg Kussmann).
  - Atoms in Molecules (AIMPAC) support (Jim Ritchie).

  - Stability analysis of SCF wavefunctions (Yihan Shao).
  - Calculation of position and momentum molecular intracules (Aaron Lee, Nick Besley and Darragh 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.

1.7 Current Development and Future Releases

All details of functionality currently under development, information relating to future releases, and patch information are regularly updated on the Q-CHEM web page (http://www.q-chem.com). Users are referred to this page for updates on developments, release information and further information on ordering and licenses. For any additional information, please contact Q-CHEM, Inc. headquarters.

1.8 Citing Q-CHEM

The most recent official citation for Q-CHEM is a journal article that has been written describing the main technical features of Q-CHEM3.0 version of the program. The full citation for this article is:

"Advances in quantum chemical methods and algorithms in the Q-CHEM 3.0 program package",

Yihan Shao, Laszlo Fusti-Molnar, Yousung Jung, Jürg Kussmann, Christian Ochsenfeld, Shawn T. Brown, Andrew T.B. Gilbert, Lyudmila V. Slipchenko, Sergey V. Levchenko, Darragh P. O'Neill, Robert A. DiStasio Jr., Rohini C. Lochan, Tao Wang, Gregory J.O. Beran, Nicholas A. Besley, John M. Herbert, Ching Yeh Lin, Troy Van Voorhis, Siu Hung Chien, Alex Sodt, Ryan P. Steele, Vitaly A. Rassolov, Paul E. Maslen, Prakashan P. Korambath, Ross D. Adamson, Brian Austin, Jon Baker, Edward F. C. Byrd, Holger Daschel, Robert J. Doerksen, Andreas Dreuw, Barry D. Dunietz, Anthony D. Dutoi, Thomas R. Furlani, Steven R. Gwaltney, Andreas Heyden, So Hirata, Chao-Ping Hsu, Gary Kedziora, Rustam Z. Khaliullin, Phil Klunzinger, Aaron M. Lee, Michael S. Lee, WanZhen Liang, Itay Lotan, Nikhil Nair, Baron Peters, Emil I. Proynov, Piotr A. Pieniazek, Young Min Rhee, Jim Ritchie, Edina Rosta, C. David Sherrill, Andrew C. Simmonett, Joseph E. Subotnik, H. Lee Woodcock III, Weimin Zhang, Alexis T. Bell, Arup K. Chakraborty, Daniel M. Chipman, Frerich J. Keil, Arieh Warshel, Warren J. Hehre, Henry F. Schaefer III, Jing Kong, Anna I. Krylov, Peter M.W. Gill and Martin Head-Gordon. *Phys. Chem. Chem. Phys.*, **8**, 3172 (2006).

This citation will soon be replaced by an official Q-CHEM 4 paper, which has been submitted for publication[2]. The full list of current authors can be found on Q-CHEM's website.

Meanwhile, one can acknowledge using Q-CHEM 4.0 by additionally citing a more recent feature article describing Q-CHEM:

“Q-Chem: An engine for innovation”,

A.I. Krylov and P.M.W. Gill. *WIREs Comput. Mol. Sci.*, **3**, 317-326 (2013).

References and Further Reading

[1] A. I. Krylov and P. Gill, *WIREs Comput Mol Sci* **3**, 317 (2013).

[2] 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.Kuś, 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. Alguires, 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 Laog, A. Laurent, K.V. Lawler, S.V. Levchenko, C. Yeh 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, C.M. Oana, R. Olivares-Amaya, D.P. O'Neill, J.A. Parkhill, T.M. Perrine, R. Peverati, P.A. Pieniazek, A. Prociuk, D.R. Rehn, E. Rosta, N.J. Russ, N. Sergueev, S.M. Sharada, S. Sharmaa, D.W. Small, A. Sodt, T. Stein, D. Stück, Y.-C. Su, A.J. W. Thom, T. Tsuchimochi, L. Vogt, O. Vydrov, T. Wang, M.A. Watson, J. Wenzel, A. White, C.F. Williams, V. Vanovschi, S. Yeganeh, S.R. Yost, Z.-Q. You, I.Y. Zhang, X. Zhang, Y. Zhou, 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, Arieh Warshel, X. Xu, 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.Z. 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, M. Head-Gordon, *Mol. Phys.* (2014), in press; doi: 10.1080/00268976.2014.952696.

Chapter 2

Installation, Customization, and Execution

2.1 Q-CHEM Installation Requirements

2.1.1 Execution Environment

Q-CHEM is shipped as a single executable along with several scripts for the computer system you will run Q-CHEM on. 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).
- Perl, version 5.
- Vendor implementation of MPI or MPICH libraries (MPI-parallel version only).

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

2.1.2 Hardware Platforms and Operating Systems

Q-CHEM runs on a wide varieties of computer systems, ranging from Intel and AMD microprocessor based PCs and workstations to high performance server nodes used in clusters and supercomputers. Currently Q-CHEM support Linux, Mac, Windows, and IBM AIX systems. For the availability of a specific platform/operating system, please contact support@q-chem.com for details.

2.1.3 Memory and Hard Disk

Memory

Q-CHEM, Inc. has endeavored to minimize memory requirements and maximize the efficiency of its use. Still, the larger the structure or the higher the level of theory, the more random access memory (RAM) is needed. Although Q-CHEM can be run with very small memory, we recommend 1 GB as a minimum. Q-CHEM also offers the ability for user control of important memory intensive aspects of the program, an important consideration for non-batch constrained multi-user systems. 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* word MEM_STATIC and has a default value of 64 MB.

The *$rem* word MEM_TOTAL specifies the limit of the total memory the user's job can use. The default value is sufficiently large that on most machines it will allow Q-CHEM to use all the available memory. This value should be reduced on machines where this is undesirable (for example if the machine is used by multiple users). The limit for the dynamic memory allocation is given by (MEM_TOTAL − MEM_STATIC). The amount of MEM_STATIC needed depends on the size of the user's particular job. Please note that one should not specify an excessively large value for MEM_STATIC, otherwise it will reduce the available memory for dynamic allocation. Memory settings in CC/EOM/ADC calculations are described in Section 5.13. The use of *$rem* words will be discussed in the next Chapter.

Disk

The Q-CHEM executables, shell scripts, auxiliary files, samples and documentation require between 360 to 400 MB 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 KBs. 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 on 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 large amount of scratch disk space. Wavefunction-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 that will be feasible and in the case of some jobs, like MP2, the jobs will also run faster as two-electron integrals are computed less often.

Although the size of any one of the Q-CHEM temporary files will not exceed 2 Gb, a user's job will not be limited by this. Q-CHEM writes large temporary data sets to multiple files so that it is not bounded by the 2 Gb file size limitation on some operating systems.

2.2 Installing Q-CHEM

Users are referred to the detailed installation instructions distributed with your copy of Q-CHEM.

An encrypted license file, *qchem.license.dat*, must be obtained from your vendor before you will be able to use Q-Chem. This file should be placed in the directory *$QCAUX/license* and must be able to be read by all users of the software. This file is node-locked, *i.e.*, it will only operate correctly on the machine for which it was generated. Further details about obtaining this file, can be found in the installation instructions.

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

2.3 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 guess (see Chapter 4), library of standard effective fragments (see Section 11.5), and instructions for the AOINTS package for generating two-electron integrals efficiently.

2.4 Q-Chem Runtime 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 2.7 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.

2.5 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 runtime 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 tcsh/csh] 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/.tcshrc* for csh/tcsh or *.bashrc* for bash.

For users using the csh shell (or equivalent), add the following lines to their home directory *.cshrc* file:

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

For users using the Bourne-again shell (bash), add the following lines to their home directory *.bashrc* file:

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

2.6 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:

- User input file
- *$HOME/.qchemrc* file
- *$QC/config/preferences* file
- Program defaults

The order of preference of initialization is as above, where the higher placed input mechanism overrides the lower.

Details of the requirements for the Q-CHEM input file are discussed in detail in this manual. In reviewing the *$rem* variables and their defaults, users may identify some variable defaults that they find too limiting or variables which they find repeatedly need to be set within their input files to make the most of Q-CHEM's features. Rather than having to remember to place such variables into the Q-CHEM input file, users are able to set long-term defaults which are read each time the user runs a Q-CHEM job. This is done by placing these defaults into the file *.qchemrc* stored in the users home directory. Additionally, system administrators can override Q-CHEM defaults with an additional *preferences* file in the *$QC/config* directory achieving a hierarchy of input as illustrated above.

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

2.6.1 *.qchemrc* and Preferences File Format

The format of the *.qchemrc* and *preferences* files is similar to that for the input file, except that only a *$rem* keyword section may be entered, terminated with the usual *$end* keyword. Any other keyword sections will be ignored. So that jobs may easily be reproduced, a copy of the *.qchemrc* file (if present) is now included near the top of the job output file.

It is important that the *.qchemrc* and *preferences* files have appropriate file permissions so 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 2.1 An example of a *.qchemrc* file to apply program default override *$rem* settings to all of the user's Q-CHEM jobs.

```
$rem
   INCORE_INTS_BUFFER   4000000   More integrals in memory
   DIIS_SUBSPACE_SIZE   5         Modify max DIIS subspace size
   THRESH               10
$end
```

2.6.2 Recommendations

As mentioned, the customization files are specifically suited for placing long-term machine specific defaults as clearly some of the defaults placed by Q-CHEM will not be optimal on large or very small machines. The following *$rem* variables are examples of those which should be considered, but the user is free to include as few or as many as desired:

AO2MO_DISK
INCORE_INTS_BUFFER
MEM_STATIC
SCF_CONVERGENCE
THRESH
NBO

Q-CHEM will print a warning message to advise the user if a *$rem* keyword section has been detected in either *.qchemrc* or *preferences*.

2.7 Running Q-CHEM

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 (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 GUI. The two most popular ones are:
    - A general web-based interface for electronic structure software, WEBMO (see www.webmo.net).
    - Wavefunction's SPARTAN user interface on some platforms. Contact Wavefunction (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
```

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/*.

Note: A checkpoint file can be requested by setting GUI=2 in the *$rem* section of the input. The checkpoint file name is determined by the GUIFILE environment variable which by default is set to *${input}.fchk*

2.7.1 Running Q-CHEM in parallel

The parallel execution of Q-CHEM can be based on OpenMP threads on a single node, MPI processes over multiple cores or across multiple nodes, and a hybrid MPI+OpenMP scheme where each spawned MPI process creates OpenMP threads for computing parallel tasks. In the hybrid MPI+OpenMP scheme cross-node communication is handled by MPI protocol and intra-node communication is done implicitly using OpenMP threading for efficient utilization of shared memory SMP systems, which maps well with the current multi-core architecture in cluster computing.

As of the 4.2 release, the OpenMP parallelization is fully supported by HF/DFT, RIMP2, CC, EOM-CC, and ADC methods. The MPI parallel capability is available for SCF, DFT, CIS, and TDDFT methods. The hybrid MPI+OpenMP parallelization is introduced in 4.2 for HF/DFT energy and gradient calculations only. Table 2.1 summarizes the parallel capabilities of Q-CHEM 4.2.

To run Q-CHEM calculation with OpenMP threads specify the number of threads (*nthreads*) using *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).

Method	OpenMP	MPI	MPI+OpenMP
HF energy & gradient	yes	yes	yes
DFT energy & gradient	yes	yes	yes
CDFT/CDFT-CI	no	no	no
RI-MP2 energy	yes	no	no
Attenuated RI-MP2 energy	yes	no	no
Integral transformation	yes	no	no
CCMAN & CCMAN2 methods	yes	no	no
ADC methods	yes	no	no
CIS energy & gradient	no	yes	no
TDDFT energy & gradient	no	yes	no
HF & DFT analytical Hessian	no	yes	no

Table 2.1: Parallel capabilities of Q-CHEM 4.2

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

Similarly, to run parallel calculations with MPI use the option *-np* to specify the number of MPI processes to be spawned.

```
qchem -np n infile outfile
qchem -np n infile outfile savename
qchem -save -np n infile outfile savename
```

where n 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 HF/DFT calculations use combined options *-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.

```
qchem -np n -nt nthreads infile outfile
qchem -np n -nt nthreads infile outfile savename
qchem -save -np n -nt nthreads infile outfile savename
```

When the additional argument *savename* is specified, the temporary files for MPI-parallel Q-CHEM are stored in *$QCSCRATCH/savename.0* At the start of a job, any existing files will be copied into this directory, and on successful completion of the job, be copied to *$QCSCRATCH/savename/* for future use. If the job terminates abnormally, the files will not be copied.

To run parallel Q-CHEM using a batch scheduler such as PBS, users may need to set QCMPIRUN environment variable to point to the *mpirun* command used in the system. For further details users should read the *$QC/README.Parallel* file, and contact Q-CHEM if any problems are encountered (email: support@q-chem.com).

2.8 IQMOL Installation Requirements

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 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 analyze the resulting output. Before Q-CHEM can be launched from IQMOL an appropriate server must be configured. 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 (see http://www.debian.org/devel/passwordlessssh 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 Vault** This allows a single password (the vault key) to be used to unlock the passwords for all the configured servers. The server passwords are salted with 64 random bits and encrypted using the AES algorithm before being stored on disk. The vault key is not stored permanently and must be re-entered 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.

2.9 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 (suffixed *.out*) of the same name.

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 (http://www.q-chem.com) and the README files in the *$QC/bin* 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 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 2.8 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 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 *$end* keyword. A short description of all Q-CHEM keywords is provided in Table 3.2 and the following sections. The user **must** understand the function and format of the *$molecule* (Section 3.3) and *$rem* (Section 3.6) keywords, as these keyword sections are where the user places the molecular geometry information and job specification details.

The keywords *$rem* and *$molecule* are requisites of Q-CHEM input files

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 C). 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 start of the input file.

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 2.6 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*)

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 must be between -50 and 50, inclusive (0 for neutral molecules, 1 for cations, -1 for anions, *etc.*). The multiplicity must be between 1 and 10, inclusive (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), irrespective 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 Appendix A which discusses the OPTIMIZE package 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 SYM_IGNORE=TRUE.

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

Section Name	**Description**
$molecule	Contains the molecular coordinate input (input file requisite).
$rem	Job specification and customization parameters (input file requisite).
$basis	User-defined basis set information (see Chapter 7).
$cdft	Options for the constrained DFT method (see Section 4.9).
$chem_sol	Job control for the Q-CHEM/CHEMSOL interface (Langevin dipoles model; see Section 11.2.8).
$comment	User comments for inclusion into output file.
$complex_ccman	Contains parameters for complex-scaled and CAP-augmented EOM-CC calculations (see Chapter 6.7).
$ecp	User-defined effective core potentials (see Chapter 8).
$efp_fragments	Specifies labels and positions of EFP fragments (see Section 11.5).
$efp_params	Contains user-defined parameters for effective fragments (see Section 11.5).
$empirical_dispersion	User-defined van der Waals parameters for DFT dispersion correction.
$eom_user_guess	User-defined guess for EOM-CC calculations (see Chapter 6.7).
$external_charges	Specifies external point charges and their positions.
$force_field_params	Force field parameters for QM/MM calculations (see Section 11.3).
$intracule	Intracule parameters (see Section 10.10).
$isotopes	Isotopic substitutions for vibrational calculations (see Section 10.12.6).
$localized_diabatization	Information for mixing together multiple adiabatic states into diabatic states (see Chapter 10).
$multipole_field	Details of an external multipole field.
$nbo	Options for the Natural Bond Orbital package.
$occupied	Guess orbitals to be occupied.
$opt	Constraint definitions for geometry optimizations.
$pcm	Job control for polarizable continuum models (see Section 11.2.3).
$plots	Generate plotting information over a grid of points (see Section 10.5).
$qm_atoms	Specify the QM region for QM/MM calculations (see Section 11.3).
$solvent	Additional parameters and variables for implicit solvent models (see Section 11.2).
$smx	Job control for SMx implicit solvent models.
$swap_occupied_virtual	Guess orbitals to be swapped.
$svp	Special parameters for the iso-density SS(V)PE module (see Section 11.2.5).
$svpirf	Initial guess for the iso-density SS(V)PE module.
$van_der_waals	User-defined atomic radii for Langevin dipoles solvation (Section 11.2.8) and PCMs (Section 11.2.2).
$xc_functional	User-defined DFT exchange-correlation functional.

Table 3.1: A 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 ("*$whatever*") should be terminated with *$end*. See the *$QC/samples* directory that is included with your release for specific examples of Q-CHEM input files using these keywords.

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 C.
(6) The entire Q-CHEM input is case-insensitive.

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

Example 3.1 A molecule in *Z*-matrix coordinates. Note that the *$molecule* input begins with the charge and multiplicity.

```
$molecule
   0 1
   O
   H1 O distance
   H2 O distance H1 theta

   distance = 1.0
   theta = 104.5
$end
```

3.3.1 Reading Molecular Coordinates From a Previous Calculation

Often users wish to perform several calculations in quick succession, whereby 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 later in this Chapter.

Example 3.2 Reading a geometry from a prior calculation.

```
$molecule
   READ
$end
```

```
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*.

Users are also able to use the *READ* function for molecular coordinate input using Q-CHEM's batch job file (see later in this Chapter).

3.3.2 Reading Molecular Coordinates from Another File

Users are able to use the *READ* function 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 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 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 Angstroms) is:

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

Note: The geometry can by specified in bohr; to do so, set the INPUT_BOHR *$rem* variable to TRUE.

3.4.1 Examples

Example 3.4 Atomic number Cartesian coordinate input for H_2O.

```
$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 3.5 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.

3.5 *Z*-matrix Coordinates

Z-matrix notation is one of the most common molecular coordinate input forms. 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 Angstroms and degrees.

Note: As with the Cartesian coordinate input method, Q-CHEM begins a calculation by taking the user-defined coordinates and translating and rotating them into a Standard Nuclear Orientation.

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 3.6 *Z*-matrix 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 3.7 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 3.8 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 3.9 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 3.10 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.

3.5.1 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.

3.6 Job Specification: The *$rem* Array Concept

The *$rem* array is the means by which users convey to Q-CHEM the type of calculation they wish to perform (level of theory, basis set, convergence criteria, *etc.*). The keyword *$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 format for entering *$rem* variables within the *$rem* keyword section of the input is shown in the following example shown in the following example:

Example 3.11 Format for declaring *$rem* variables in the *$rem* keyword section of the Q-CHEM input file. Note, Q-CHEM only reads the first two arguments on each line of *$rem*. All other text is ignored and can be used for placing short user comments.

```
REM_VARIABLE     VALUE    [comment]
```

The *$rem* array stores all details required to perform the calculation, and details of output requirements. It provides the flexibility to customize a calculation to specific user requirements. If a default *$rem* variable setting is indicated in this manual, the user does not have to declare the variable in order for the default to be initiated (*e.g.*, the default JOBTYPE is a single point energy, SP). Thus, to perform a single point energy calculation, the user does **not** need to set the *$rem* variable JOBTYPE to SP. However, to perform an optimization, for example, it is necessary to override the program default by setting JOBTYPE to OPT.

A number of the *$rem* variables have been set aside for internal program use, as they represent variables automatically determined by Q-CHEM (*e.g.*, the number of atoms, the number of basis functions). These need not concern the user.

User communication to the internal program *$rem* array comes in two general forms: (1) long term, machine-specific customization *via* the *.qchemrc* and *preferences* files (Section 2.6) and, (2) the Q-CHEM input deck. There are many defaults already set within the Q-CHEM program many of which can be overridden by the user. Checks are made to ensure that the user specifications are permissible (*e.g.* integral accuracy is confined to 10^{-12} and adjusted, if necessary. If adjustment is not possible, an error message is returned. Details of these checks and defaults will be given as they arise.

The user need not know all elements, options and details of the *$rem* array in order to fully exploit the Q-CHEM program. Many of the necessary elements and options are determined automatically by the program, or the optimized default parameters, supplied according to the user's basic requirements, available disk and memory, and the operating system and platform.

3.7 *$rem* Array Format in Q-CHEM Input

All data between the *$rem* keyword and the next appearance of *$end* is assumed to be user *$rem* array input. On a single line for each *$rem* variable, the user declares the *$rem* variable, followed by a blank space (tab stop inclusive) and then the *$rem* variable option. It is recommended that a comment be placed following a space after the *$rem* variable option. *$rem* variables are case insensitive and a full listing is supplied in Appendix C. Depending on the particular *$rem* variable, *$rem* options are entered either as a case-insensitive keyword, an integer value or logical identifier (true/false). The format for describing each *$rem* variable in this manual is as follows:

REM_VARIABLE
A short description of what the variable controls.
TYPE:
The type of variable, *i.e.* either INTEGER, LOGICAL or STRING
DEFAULT:
The default value, if any.
OPTIONS:
A list of the options available to the user.
RECOMMENDATION:
A quick recommendation, where appropriate.

Example 3.12 General format of the *$rem* section of the text input file.

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

Note: (1) Erroneous lines will terminate the calculation.
(2) Tab stops can be used to format input.
(3) A line prefixed with an exclamation mark '!' is treated as a comment and will be ignored by the program.

3.8 Minimum *$rem* Array Requirements

Although Q-CHEM provides defaults for most *$rem* variables, the user will always have to stipulate a few others. For example, in a single point energy calculation, 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 3.13 Example of minimum *$rem* requirements to run an MP2/6-31G* energy calculation.

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

Alternatively, the level of theory can be specified by two other *$rem* variables, EXCHANGE (defining the level of theory to treat exchange) and CORRELATION (defining the level of theory to treat correlation, if required). For excited states, there is a third *$rem* variable, EOM_CORR, which specifies the level of correlation for the target states.

To invoke DFT calculations, METHOD specifies an exchange-correlation functional (see Section 4.3.15 for the supported functionals).

Supported values for the METHOD keyword for different wave function based correlated methods are listed in Section 5.1; for excited-state calculations, see Section 6.1.

If a wavefunction-based correlation treatment (such as MP2 or CC) is requested by using CORRELATION keyword, HF is taken as the default for EXCHANGE.

3.9 Comments (*$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, although currently the entire input deck is copied to the output file, rendering this redundant.

3.10 User-Defined Basis Sets (*$basis* and *$aux_basis*)

The *$rem* variable BASIS allows the user to indicate that the basis set is being user-defined. The user-defined basis set is entered in the *$basis* section of the input. For further details of entering a user-defined basis set, see Chapter 7. Similarly, a user-defined auxiliary basis set may be entered in a *$aux_basis* section of the input if the *$rem* list includes AUX_BASIS = GEN.

3.11 User-Defined Pseudopotentials (*$ecp*)

The *$rem* variable ECP allows the user to indicate that pseudopotentials (effective core potentials) are being user-defined. The user-defined effective core potential is entered in the *$ecp* section of the input. For further details, see Chapter 8.

3.12 User-Defined Exchange-Correlation Density Functionals (*$xc_functional*)

The EXCHANGE and CORRELATION *$rem* variables (Chapter 4) allow the user to indicate that the exchange-correlation density functional will be user-defined. The user defined exchange-correlation is to be entered in the *$xc_functional* part of the input. The format is:

```
$xc_functional
   X  exchange_symbol  coefficient
   X  exchange_symbol  coefficient
   ...
   C  correlation_symbol  coefficient
   C  correlation_symbol  coefficient
   ...
   K  coefficient
$end
```

Note: Coefficients are real numbers.

3.13 User-defined Parameters for DFT Dispersion Correction (*$empirical_dispersion*)

If a user wants to change from the default values recommended by Grimme, the user-defined dispersion parameters can be entered in the *$empirical_dispersion* section of the input. For further details, see Section 4.3.6.

3.14 Addition of External Charges (*$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 for a set of external charges is the Cartesian coordinates, followed by the charge size, one charge per line. Charges are in atomic units, and coordinates are in angstroms (unless atomic units are specifically selected, see INPUT_BOHR). The external charges are rotated with the molecule into the standard nuclear orientation.

Example 3.14 General format for incorporating a set of external charges.

```
$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 by using ADD_CHARGED_CAGE keyword The cage parameters are controlled by CAGE_RADIUS, CAGE_POINTS, and CAGE_CHARGE. More details are given in Section 6.7.6.

3.15 Applying a Multipole Field (*$multipole_field*)

Q-CHEM has the capability to apply a multipole field to the molecule under investigation. Q-CHEM scans the input deck for the *$multipole_field* keyword, and reads each line (up to the terminator keyword, *$end*) as a single component of the applied field.

Example 3.15 General format for imposing a multipole field.

```
$multipole_field
   field_component_1   value_1
   field_component_2   value_2
$end
```

The *field_component* is simply stipulated using the Cartesian representation *e.g.* X, Y, Z, (dipole), XX, XY, YY (quadrupole) XXX, *etc.*, and the value or size of the imposed field is in atomic units.

3.16 User-Defined Occupied Guess Orbitals (*$occupied* and *$swap_occupied_virtual*)

It is sometimes useful for the occupied guess orbitals to be other than the lowest N_α (or N_β) orbitals. Q-CHEM allows the occupied guess orbitals to be defined using the *$occupied* keyword. The user defines occupied guess orbitals by listing the alpha orbitals to be occupied on the first line, and beta on the second. Alternatively, orbital choice can be controlled by the *$swap_occupied_virtual*keyword. See Section 4.4.4.

3.17 Polarizable Continuum Solvation Models (*$pcm*)

The *$pcm* section is available to provide special parameters for polarizable continuum models (PCMs). These include the C-PCM and IEF-PCM models, which share a common set of parameters. Details are provided in Section 11.2.2.

3.18 SS(V)PE Solvation Modeling (*$svp* and *$svpirf*)

The *$svp* section is available to specify special parameters to the solvation module such as cavity grid parameters and modifications to the numerical integration procedure. The *$svpirf* section allows the user to specify an initial guess for the solution of the cavity charges. As discussed in section 11.2.5, the *$svp* and *$svpirf* input sections are used to specify parameters for the iso-density implementation of SS(V)PE. An alternative implementation of the SS(V)PE mode, based on a more empirical definition of the solute cavity, is available within the PCM code (Section 11.2.2).

3.19 User-Defined van der Waals Radii (*$van_der_waals*)

The *$van_der_waals* section of the input enables the user to customize the Van der Waals radii that are important parameters in the Langevin dipoles solvation model. For more details, see Section 11.2.

3.20 Effective Fragment Potential calculations (*$efp_fragments* and *$efp_params*)

These keywords are used to specify positions and parameters for effective fragments in EFP calculations. Details are provided in Chapter **??**.

3.21 Natural Bond Orbital Package (*$nbo*)

The default action in Q-CHEM is not to run the NBO package. To turn the NBO package on, set the *$rem* variable NBO to ON. To access further features of NBO, place standard NBO package parameters into a keyword section in the input file headed with the *$nbo* keyword. Terminate the section with the termination string *$end*.

3.22 Orbitals, Densities and ESPs on a Mesh (*$plots*)

The *$plots* part of the input permits the evaluation of molecular orbitals, densities, electrostatic potentials, transition densities, electron attachment and detachment densities on a user-defined mesh of points. For more details, see Section 10.5.

3.23 Intracules (*$intracule*)

The *$intracule* section allows the user to enter options to customize the calculation of molecular intracules. The INTRACULE *$rem* variable must also be set to TRUE before this section takes effect. For further details see Section 10.10.

3.24 Geometry Optimization with General Constraints (*$opt*)

When a user defines the JOBTYPE to be a molecular geometry optimization, Q-CHEM scans the input deck for the *$opt* keyword. Distance, angle, dihedral and out-of-plane bend constraints imposed on any atom declared by the user in this section, are then imposed on the optimization procedure. See Chapter 9 for details.

3.25 Isotopic Substitutions (*$isotopes*)

By default Q-CHEM uses atomic masses that correspond to the most abundant naturally occurring isotopes. Alternative masses for any or all of the atoms in a molecule can be specified using the *$isotopes* keyword. The ISOTOPES *$rem* variable must be set to TRUE for this section to take effect. See Section 10.12.6 for details.

3.26 Multiple Jobs in a Single File: Q-CHEM Batch Job Files

It is sometimes useful to place a series of jobs into a single ASCII file. This feature is supported by Q-CHEM and is invoked by separating jobs with the string `@@@` on a single line. All output is subsequently appended to the same output file for each job within the file.

Note: 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.

In general, multiple jobs are placed in a single file for two reasons:

1. To use information from a prior job in a later job
2. To keep projects together in a single file

The `@@@` feature allows these objectives to be met, but the following points should be noted:

- Q-CHEM reads all the jobs from the input file on initiation and stores them. The user cannot make changes to the details of jobs which have not been run post command line initiation.
- If any single job fails, Q-CHEM proceeds to the next job in the batch file.

- No check is made to ensure that dependencies are satisfied, or that information is consistent (*e.g.* an optimization job followed by a frequency job; reading in the new geometry from the optimization for the frequency). No check is made to ensure that the optimization was successful. Similarly, it is assumed that both jobs use the same basis set when reading in MO coefficients from a previous job.

- Scratch files are saved between multi-job/single files runs (*i.e.*, using a batch file with `@@@` separators), but are deleted on completion unless a third *qchem* command line argument is supplied (see Chapter 2).

Using batch files with the `@@@` separator is clearly most useful for cases relating to point 1 above. The alternative would be to cut and paste output, and/or use a third command line argument to save scratch files between separate runs.

For example, the following input file will optimize the geometry of H_2 at HF/6-31G*, calculate vibrational frequencies at HF/6-31G* using the optimized geometry and the self-consistent MO coefficients from the optimization and, finally, perform a single point energy using the optimized geometry at the MP2/6-311G(d,p) level of theory. Each job will use the same scratch area, reading files from previous runs as instructed.

Example 3.16 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
```

Note: (1) Output is concatenated into the same output file.
(2) Only two arguments are necessarily supplied to the command line interface.

3.27 Q-CHEM Output File

The Q-CHEM output file is the file to which details of the job invoked by the user are printed. The type of information printed to this files depends on the type of job (single point energy, geometry optimization *etc.*) and the *$rem* variable print levels. The general and default form is as follows:

- Q-CHEM citation
- User input
- Molecular geometry in Cartesian coordinates
- Molecular point group, nuclear repulsion energy, number of alpha and beta electrons
- Basis set information (number of functions, shells and function pairs)
- SCF details (method, guess, optimization procedure)
- SCF iterations (for each iteration, energy and DIIS error is reported)
- {depends on job type}
- Molecular orbital symmetries
- Mulliken population analysis
- Cartesian multipole moments
- Job completion

Note: Q-CHEM overwrites any existing output files in the working directory when it is invoked with an existing file as the output file parameter.

3.28 Q-CHEM Scratch Files

The directory set by the environment variable *$QCSCRATCH* is the location Q-CHEM places scratch files it creates on execution. Users may wish to use the information created for subsequent calculations. See Chapter 2 for information on saving files.

Chapter 4

Self-Consistent Field Ground State Methods

4.1 Introduction

4.1.1 Overview of Chapter

Theoretical chemical models [5] involve two principal approximations. One must specify the type of atomic orbital basis set used (see Chapters 7 and 8), and one must specify the way in which the instantaneous interactions (or correlations) between electrons are treated. Self-consistent field (SCF) methods are the simplest and most widely used electron correlation treatments, and contain as special cases all Kohn-Sham density functional methods and the Hartree-Fock method. This Chapter summarizes Q-CHEM's SCF capabilities, while the next Chapter discusses more complex (and computationally expensive!) wavefunction-based methods for describing electron correlation. If you are new to quantum chemistry, we recommend that you also purchase an introductory textbook on the physical content and practical performance of standard methods [5–7].

This Chapter is organized so that the earlier sections provide a mixture of basic theoretical background, and a description of the minimum number of program input options that must be specified to run SCF jobs. Specifically, this includes the sections on:

- Hartree-Fock theory
- Density functional theory. Note that all basic input options described in the Hartree-Fock also apply to density functional calculations.

Later sections introduce more specialized options that can be consulted as needed:

- Large molecules and linear scaling methods. A short overview of the ideas behind methods for very large systems and the options that control them.
- Initial guesses for SCF calculations. Changing the default initial guess is sometimes important for SCF calculations that do not converge.

- Converging the SCF calculation. This section describes the iterative methods available to control SCF calculations in Q-CHEM. Altering the standard options is essential for SCF jobs that have failed to converge with the default options.
- Unconventional SCF calculations. Some nonstandard SCF methods with novel physical and mathematical features. Explore further if you are interested!
- SCF Metadynamics. This can be used to locate multiple solutions to the SCF equations and help check that your solution is the lowest minimum.

4.1.2 Theoretical Background

In 1926, Schrödinger [8] combined the wave nature of the electron with the statistical knowledge of the electron *viz.* Heisenberg's Uncertainty Principle [9] to formulate an eigenvalue equation for the total energy of a molecular system. If we focus on stationary states and ignore the effects of relativity, we have the time-independent, non-relativistic equation

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

where the coordinates $\mathbf{R}$ and $\mathbf{r}$ refer to nuclei and electron position vectors respectively and H is the Hamiltonian operator. In atomic units,

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

where ∇^2 is the Laplacian operator,

$$\nabla^2 \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{4.3}$$

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 the Ath and Bth nucleus, $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ 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. E is an eigenvalue of H, equal to the total energy, and the wave function Ψ, is an eigenfunction of H.

Separating the motions of the electrons from that of the nuclei, an idea originally due to Born and Oppenheimer [10], yields the electronic Hamiltonian operator:

$$H_{\text{elec}} = -\frac{1}{2}\sum_{i=1}^{N}\nabla_i^2 - \sum_{i=1}^{N}\sum_{A=1}^{M}\frac{Z_A}{r_{iA}} + \sum_{i=1}^{N}\sum_{j>i}^{N}\frac{1}{r_{ij}} \tag{4.4}$$

The solution of the corresponding electronic Schrödinger equation,

$$H_{\text{elec}}\Psi_{\text{elec}} = E_{\text{elec}}\Psi_{\text{elec}} \tag{4.5}$$

gives the total electronic energy, E_{elec}, and electronic wave function, Ψ_{elec}, which describes the motion of the electrons for a fixed nuclear position. 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, E_1, $E_2 \ldots$) where $E_0 \leq E_1 \leq E_2 \leq \ldots$.

Our interest lies in determining the lowest eigenvalue and associated eigenfunction which correspond to the ground state energy and wavefunction 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 that electrons move independently within molecular orbitals (MO), each of which describes the probability distribution of a single electron. Each MO is determined by considering the electron as moving within an average field of all the other electrons. Ensuring that the wavefunction is antisymmetric upon electron interchange, yields the well known Slater-determinant wavefunction [11, 12],

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

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 approximation, but these SCF methods also include Kohn-Sham Density Functional Theories (see Section 4.3). All SCF methods lead to equations of the form

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

where the Fock operator $f(i)$ can be written

$$f(i) = -\frac{1}{2}\nabla_i^2 + \upsilon^{\text{eff}}(i) \tag{4.9}$$

Here $\mathbf{x}_i$ are spin and spatial coordinates of the ith electron, χ are the 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 and will be elaborated on in further sections.

The second approximation usually introduced when solving Eq. (4.5), is the introduction of an Atomic Orbital (AO) basis. AOs (ϕ_μ) are usually combined linearly to approximate the true MOs. There are many standardized, atom-centered basis sets and details of these are discussed in Chapter 7.

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,

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

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

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

and ε is a diagonal matrix of 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 [13] equations

$$\begin{aligned} \mathbf{F}^\alpha\mathbf{C}^\alpha &= \varepsilon^\alpha\mathbf{SC}^\alpha \\ \mathbf{F}^\beta\mathbf{C}^\beta &= \varepsilon^\beta\mathbf{SC}^\beta \end{aligned} \tag{4.13}$$

Solving Eq. (4.11) or Eq. (4.13) yields the restricted or unrestricted finite basis Hartree-Fock approximation. This approximation inherently neglects the instantaneous electron-electron correlations which 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 SCF methodologies are able to capture all the correlation energy (the difference in energy between the HF energy and the true energy). In practice, the best currently available density functionals perform well, but not perfectly and conventional HF-based approaches to calculating the correlation energy are still often required. They are discussed separately in the following Chapter.

In self-consistent field methods, an initial guess is calculated for the MOs 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. This procedure is repeated until the new MOs differ negligibly from those of the previous iteration.

Because they often yield acceptably accurate chemical predictions at a reasonable computational cost, self-consistent field methods are the corner stone 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 the size, N, of the system. This is slower than the growth of the cheapest conventional correlated methods but recent work by Q-CHEM, Inc. and its collaborators has dramatically reduced it to $\mathcal{O}(N)$, an improvement that now allows SCF methods to be applied to molecules previously considered beyond the scope of *ab initio* treatment.

In order to carry out an SCF calculation using Q-CHEM, two *$rem* variables need to be set:

BASIS	to specify the basis set (see Chapter 7).
METHOD	SCF method: HF or a density functional.

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

Calculation	***$rem* Variable JOBTYPE**
Single point energy (default)	SINGLE_POINT, SP
Force	FORCE
Equilibrium Structure Search	OPTIMIZATION, OPT
Transition Structure Search	TS
Intrinsic reaction pathway	RPATH
Frequency	FREQUENCY, FREQ
NMR Chemical Shift	NMR
Indirect nuclear spin–spin coupling	ISSC

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

4.2 Hartree–Fock Calculations

4.2.1 The Hartree-Fock Equations

As with much of the theory underlying modern quantum chemistry, the Hartree-Fock 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 the integro–differential Eq. (4.8) in which the effective potential υ^{eff} depends on the SCF methodology. In a restricted HF (RHF) formalism, the effective potential can be written as

$$\upsilon^{\text{eff}} = \sum_{a}^{N/2} [2J_a(1) - K_a(1)] - \sum_{A=1}^{M} \frac{Z_A}{r_{1A}} \tag{4.14}$$

where the Coulomb and exchange operators are defined as

$$J_a(1) = \int \psi_a^*(2) \frac{1}{r_{12}} \psi_a(2) d\mathbf{r}_2 \tag{4.15}$$

and

$$K_a(1)\psi_i(1) = \left[\int \psi_a^*(2) \frac{1}{r_{12}} \psi_i(2) d\mathbf{r}_2 \right] \psi_a(1) \tag{4.16}$$

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

$$F_{\mu\nu} = H_{\mu\nu}^{\text{core}} + J_{\mu\nu} - K_{\mu\nu} \tag{4.17}$$

where the core Hamiltonian matrix elements

$$H_{\mu\nu}^{\text{core}} = T_{\mu\nu} + V_{\mu\nu} \tag{4.18}$$

consist of kinetic energy elements

$$T_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left[-\frac{1}{2}\nabla^2 \right] \phi_\nu(\mathbf{r}) d\mathbf{r} \tag{4.19}$$

and nuclear attraction elements

$$V_{\mu\nu} = \int \phi_\mu(\mathbf{r}) \left[-\sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} \right] \phi_\nu(\mathbf{r}) d\mathbf{r} \tag{4.20}$$

The Coulomb and Exchange elements are given by

$$J_{\mu\nu} = \sum_{\lambda\sigma} P_{\lambda\sigma} \left(\mu\nu|\lambda\sigma \right) \tag{4.21}$$

and

$$K_{\mu\nu} = \frac{1}{2} \sum_{\lambda\sigma} P_{\lambda\sigma} \left(\mu\lambda|\nu\sigma \right) \tag{4.22}$$

respectively, where the density matrix elements are

$$P_{\mu\nu} = 2 \sum_{a=1}^{N/2} C_{\mu a} C_{\nu a} \tag{4.23}$$

and the two electron integrals are

$$(\mu\nu|\lambda\sigma) = \int \int \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_1) \left[\frac{1}{r_{12}} \right] \phi_\lambda(\mathbf{r}_2)\phi_\sigma(\mathbf{r_2}) d\mathbf{r}_1 d\mathbf{r}_2 \tag{4.24}$$

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 B for details of Q-CHEM's AOINTS package.

Substituting the matrix element in Eq. (4.17) back into the Roothaan-Hall equations, Eq. (4.11), and iterating until self-consistency is achieved will yield the Restricted Hartree-Fock (RHF) energy and wavefunction. Alternatively, one could have adopted the unrestricted form of the wavefunction by defining an alpha and beta density matrix:

$$\begin{aligned} P^{\alpha}_{\mu\nu} &= \sum_{a=1}^{n_\alpha} C^{\alpha}_{\mu a} C^{\alpha}_{\nu a} \\ P^{\beta}_{\mu\nu} &= \sum_{a=1}^{n_\beta} C^{\beta}_{\mu a} C^{\beta}_{\nu a} \end{aligned} \tag{4.25}$$

The total electron density matrix P^{T} is simply the sum of the alpha and beta density matrices. The unrestricted alpha Fock matrix,

$$F^{\alpha}_{\mu\nu} = H^{\mathrm{core}}_{\mu\nu} + J_{\mu\nu} - K^{\alpha}_{\mu\nu} \tag{4.26}$$

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

$$K^{\alpha}_{\mu\nu} = \sum_{\lambda}^{N} \sum_{\sigma}^{N} P^{\alpha}_{\lambda\sigma} \left(\mu\lambda|\nu\sigma\right) \tag{4.27}$$

4.2.2 Wavefunction Stability Analysis

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 wavefunction is said to be unstable. Even if the wavefunction is at a minimum, this minimum may be an artifact of the constraints placed on the form of the wavefunction. For example, an unrestricted calculation will usually give a lower energy than the corresponding restricted calculation, and this can give rise to a RHF→UHF instability.

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

$$\chi_i(\mathbf{r},\zeta) = \psi^{\alpha}_i(\mathbf{r})\alpha(\zeta) + \psi^{\beta}_i(\mathbf{r})\beta(\zeta) \tag{4.28}$$

Here, the ψ's are complex 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

$$\chi_i(\mathbf{r},\zeta) = \psi^{\alpha}_i(\mathbf{r})\alpha(\zeta) \quad \text{or} \quad \chi_i(\mathbf{r},\zeta) = \psi^{\beta}_i(\mathbf{r})\beta(\zeta) \tag{4.29}$$

where the ψ's are still complex functions. Most SCF packages, including Q-CHEM's, deal only with real functions, and this places an additional constraint on the form of the wavefunction. If there exists a complex solution to the SCF equations that has a lower energy, the wavefunction will exhibit either a RHF → CRHF or a UHF → CUHF instability. The final constraint that is commonly placed on the spin-functions is that $\psi^{\alpha}_i = \psi^{\beta}_i$, *i.e.*, the spatial parts of the spin-up and spin-down orbitals are the same. This gives the familiar restricted formalism and can lead to a RHF→UHF instability as mentioned above. Further details about the possible instabilities can be found in Ref. 14.

Wavefunction 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.

If a wavefunction exhibits an instability, the seriousness of it can be judged from the magnitude of the negative eigenvalues of the stability matrices. These matrices and eigenvalues are computed by Q-CHEM's Stability Analysis package, which was implemented by Dr Yihan Shao. The package is invoked by setting the STABILITY_ANALYSIS *$rem* variable is set to TRUE. In order to compute these stability matrices Q-CHEM must first perform a CIS calculation. This will be performed automatically, and does not require any further input from the user. By default Q-CHEM computes only the lowest eigenvalue of the stability matrix. This is usually sufficient to determine if there is a negative eigenvalue, and therefore an instability. Users wishing to calculate additional eigenvalues can do so by setting the CIS_N_ROOTS *$rem* variable to a number larger than 1.

Q-CHEM's Stability Analysis package also seeks to correct internal instabilities (RHF→RHF or UHF→UHF). Then, if such an instability is detected, Q-CHEM automatically performs a unitary transformation of the molecular orbitals following the directions of the lowest eigenvector, and writes a new set of MOs to disk. One can read in these MOs as an initial guess in a second SCF calculation (set the SCF_GUESS *$rem* variable to READ), it might also be desirable to set the SCF_ALGORITHM to GDM. In cases where the lowest-energy SCF solution breaks the molecular point-group symmetry, the SYM_IGNORE *$rem* should be set to TRUE.

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

4.2.3 Basic Hartree-Fock Job Control

In brief, Q-CHEM supports the three main variants of the Hartree-Fock method. They are:

- Restricted Hartree-Fock (RHF) for closed shell molecules. It is typically appropriate for closed shell molecules at their equilibrium geometry, where electrons occupy orbitals in pairs.
- Unrestricted Hartree-Fock (UHF) for open shell molecules. Appropriate for radicals with an odd number of electrons, and also for molecules with even numbers of electrons where not all electrons are paired (for example stretched bonds and diradicaloids).
- Restricted open shell Hartree-Fock (ROHF) for open shell molecules, where the alpha and beta orbitals are constrained to be identical.

Only two *$rem* variables are required in order to run Hartree-Fock (HF) calculations:

- METHOD must be set to HF.

- A valid keyword for BASIS must be specified (see Chapter 7).

In slightly more detail, here is a list of basic *$rem* variables associated with running Hartree-Fock calculations. See Chapter 7 for further detail on basis sets available and Chapter 8 for specifying effective core potentials.

JOBTYPE

Specifies the type of calculation.

TYPE:

STRING

DEFAULT:

SP

OPTIONS:

SP	Single point energy.
OPT	Geometry Minimization.
TS	Transition Structure Search.
FREQ	Frequency Calculation.
FORCE	Analytical Force calculation.
RPATH	Intrinsic Reaction Coordinate calculation.
NMR	NMR chemical shift calculation.
ISSC	Indirect nuclear spin–spin coupling calculation.
BSSE	BSSE calculation.
EDA	Energy decomposition analysis.

RECOMMENDATION:

Job dependent

METHOD

Specifies the level of theory.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

HF	Exact (Hartree-Fock).

RECOMMENDATION:

Use HF for Hartree-Fock calculations.

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 7.
Mixed	Use a mixture of basis sets (see Chapter 7).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

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 virtuals.
NVIRT	Number of virtuals to print.

RECOMMENDATION:

Use TRUE unless more virtuals are desired.

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:

Should be at least three greater than SCF_CONVERGENCE. Increase for more significant figures, at greater computational cost.

SCF_CONVERGENCE

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

TYPE:

INTEGER

DEFAULT:

5	For single point energy calculations.
8	For geometry optimizations and vibrational analysis.
8	For SSG calculations, see Chapter 5.

OPTIONS:

User-defined

RECOMMENDATION:

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

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE	(Restricted) Closed-shell systems.
TRUE	(Unrestricted) Open-shell systems.

OPTIONS:

TRUE	(Unrestricted) Open-shell systems.
FALSE	Restricted open-shell HF (ROHF).

RECOMMENDATION:

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

4.2.4 Additional Hartree-Fock Job Control Options

Listed below are a number of useful options to customize a Hartree-Fock 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 C. A number of other specialized topics (large molecules, customizing initial guesses, and converging the calculation) are discussed separately in Sections 4.6, 4.4, and 4.5, respectively.

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 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 $10^{-\text{THRESH}}$.

OPTIONS:

1	Discard shell-pairs four orders of magnitude below machine precision.
2	Discard shell-pairs below $10^{-\text{THRESH}}$.

RECOMMENDATION:

Use default.

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 and DFT exchange-correlation matrices) on each cycle.

RECOMMENDATION:

Proceed with care; can result in *extremely* large output files at level 2 or higher. 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 and density matrices.

RECOMMENDATION:

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

DIIS_SEPARATE_ERRVEC

Control optimization of DIIS error vector in unrestricted calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE	Use a combined alpha and beta error vector.

OPTIONS:

FALSE	Use a combined alpha and beta error vector.
TRUE	Use separate error vectors for the alpha and beta spaces.

RECOMMENDATION:

When using DIIS in Q-CHEM a convenient optimization for unrestricted calculations is to sum the alpha and beta 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 alpha and beta 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 to TRUE to check.

4.2.5 Examples

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

Example 4.1 Example Q-CHEM input for a single point energy calculation on water. Note that the declaration of the single point *$rem* variable and level of theory to treat correlation are redundant because they are the same as the Q-CHEM defaults.

```
$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

$comment
HF/STO-3G water single point calculation
$end
```

Example 4.2 UHF/6-311G calculation on the Lithium 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 alpha and beta electron differ, MOs default to an unrestricted formalism.

```
$molecule
   0,2
   3
$end

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

Example 4.3 ROHF/6-311G calculation on the Lithium atom. Note again that correlation and the job type need not be indicated.

```
$molecule
   0,2
   3
$end

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

Example 4.4 RHF/6-31G stability analysis calculation on the singlet state of the oxygen molecule. The wavefunction is RHF→UHF unstable.

```
$molecule
   0 1
   O
   O  1  1.165
$end

$rem
   METHOD               hf         Hartree-Fock
   UNRESTRICTED         false      Restricted MOs
   BASIS                6-31G(d)   Basis set
   STABILITY_ANALYSIS   true       Perform a stability analysis
$end
```

4.2.6 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 [15–17]. 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. SYM_IGNORE controls symmetry throughout all modules. The default is FALSE. 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 SYM_IGNORE *$rem* is set to TRUE then the coordinates will not be altered from the input, and the point group will be set to C_1.

The SYMMETRY (an alias for ISYM_RQ) keyword controls symmetry in some integral routines. It is set to FALSE 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 5 and 6), although we noticed that sometimes it may mess up the determination of orbital symmetries, possibly due to numeric noise. In some cases, SYMMETRY=TRUE can cause problems (poor convergence and crazy SCF energies) and turning it off can help.

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 irreps in the affected groups are labeled. For example, b_1 and b_2 irreps in C_{2v} are flipped when using different conventions. Q-CHEM uses non-Mulliken symmetry convention. See http://iopenshell.usc.edu/howto/symmetry for detailed explanations.

SYMMETRY

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

TYPE:

LOGICAL

DEFAULT:

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

SYM_IGNORE

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

TYPE:

LOGICAL

DEFAULT:

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

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Use 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.

4.3 Density Functional Theory

4.3.1 Introduction

In recent years, Density Functional Theory [18–21] has emerged as an accurate alternative first-principles approach to quantum mechanical molecular investigations. DFT currently accounts for approximately 90% of all quantum chemical calculations being performed, not only because of its proven chemical accuracy, but also because of its relatively cheap computational expense. 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 functional theories, which make calculations on quite 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 wavefunction $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$ There are a number of distinct similarities and differences to traditional wavefunction approaches and modern DFT methodologies. Firstly, the essential building blocks of the many electron wavefunction are single-electron orbitals are directly analogous to the Kohn-Sham (see below) orbitals in the current DFT framework. Secondly, both the electron density and the many-electron wavefunction tend to be constructed *via* a SCF approach that requires the construction of matrix elements which are remarkably and conveniently very similar.

However, traditional approaches using the many electron wavefunction as a foundation must resort to a post-SCF calculation (Chapter 5) to incorporate correlation effects, whereas DFT approaches do not. Post-SCF methods, such as perturbation theory or coupled cluster theory are extremely expensive relative to the SCF procedure. On the other hand, the DFT approach is, in principle, exact, but in practice relies on modeling the unknown exact 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 an arbitrary level of accuracy, but can be computationally demanding, whereas DFT approaches offer a practical route but the theory is currently incomplete.

4.3.2 Kohn-Sham Density Functional Theory

The Density Functional Theory by Hohenberg, Kohn and Sham [22, 23] stems from the original work of Dirac [24], who found that the exchange energy of a uniform electron gas may be calculated exactly, knowing only the charge density. However, while the more traditional DFT constitutes a direct approach and the necessary equations contain only the electron density, difficulties associated with the kinetic energy functional obstructed the extension of DFT to anything more than a crude level of approximation. Kohn and Sham developed an indirect approach to the kinetic energy functional which transformed DFT into a practical tool for quantum chemical calculations.

Within the Kohn-Sham formalism [23], the ground state electronic energy, E, can be written as

$$E = E_{\rm T} + E_{\rm V} + E_{\rm J} + E_{\rm XC} \tag{4.30}$$

where $E_{\rm T}$ is the kinetic energy, $E_{\rm V}$ is the electron–nuclear interaction energy, $E_{\rm J}$ is the Coulomb self-interaction of the electron density $\rho(\mathbf{r})$ and $E_{\rm XC}$ is the exchange-correlation energy. Adopting an unrestricted format, the alpha and beta total electron densities can be written as

$$\begin{aligned} \rho_\alpha(\mathbf{r}) &= \sum_{i=1}^{n_\alpha} |\psi_i^\alpha|^2 \\ \rho_\beta(\mathbf{r}) &= \sum_{i=1}^{n_\beta} |\psi_i^\beta|^2 \end{aligned} \tag{4.31}$$

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

$$\rho(\mathbf{r}) = \rho_\alpha(\mathbf{r}) + \rho_\beta(\mathbf{r}) \tag{4.32}$$

Within a finite basis set [25], the density is represented by

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}^{\rm T} \phi_\mu(\mathbf{r}) \phi_\nu(\mathbf{r}) \tag{4.33}$$

The components of Eq. (4.28) can now be written as

$$
\begin{aligned}
E_{\mathrm{T}} &= \sum_{i=1}^{n_\alpha}\left\langle\psi_i^\alpha\left|-\frac{1}{2}\nabla^2\right|\psi_i^\alpha\right\rangle+\sum_{i=1}^{n_\beta}\left\langle\psi_i^\beta\left|-\frac{1}{2}\nabla^2\right|\psi_i^\beta\right\rangle \\
&= \sum_{\mu\nu}P_{\mu\nu}^{\mathrm{T}}\left\langle\phi_\mu(\mathbf{r})\left|-\frac{1}{2}\nabla^2\right|\phi_\nu(\mathbf{r})\right\rangle \qquad (4.34)\\
E_{\mathrm{V}} &= -\sum_{A=1}^{M}Z_A\frac{\rho(\mathbf{r})}{|\mathbf{r}-\mathbf{R}_A|}d\mathbf{r} \\
&= -\sum_{\mu\nu}P_{\mu\nu}^{\mathrm{T}}\sum_A\left\langle\phi_\mu(\mathbf{r})\left|\frac{Z_A}{|\mathbf{r}-\mathbf{R}_A|}\right|\phi_\nu(\mathbf{r})\right\rangle \qquad (4.35)\\
E_{\mathrm{J}} &= \frac{1}{2}\left\langle\rho(\mathbf{r}_1)\left|\frac{1}{|\mathbf{r}_1-\mathbf{r}_2|}\right|\rho(\mathbf{r}_2)\right\rangle \\
&= \frac{1}{2}\sum_{\mu\nu}\sum_{\lambda\sigma}P_{\mu\nu}^{\mathrm{T}}P_{\lambda\sigma}^{\mathrm{T}}\left(\mu\nu|\lambda\sigma\right) \qquad (4.36)\\
E_{\mathrm{XC}} &= \int f\left[\rho(\mathbf{r}),\nabla\rho(\mathbf{r}),\ldots\right]d\mathbf{r} \qquad (4.37)
\end{aligned}
$$

Minimizing E with respect to the unknown Kohn-Sham orbital coefficients yields a set of matrix equations exactly analogous to the UHF case

$$\mathbf{F}^\alpha\mathbf{C}^\alpha=\varepsilon^\alpha\mathbf{S}\mathbf{C}^\alpha \qquad (4.38)$$

$$\mathbf{F}^\beta\mathbf{C}^\beta=\varepsilon^\beta\mathbf{S}\mathbf{C}^\beta \qquad (4.39)$$

where the Fock matrix elements are generalized to

$$F_{\mu\nu}^\alpha=H_{\mu\nu}^{\mathrm{core}}+J_{\mu\nu}-F_{\mu\nu}^{\mathrm{XC}\alpha} \qquad (4.40)$$

$$F_{\mu\nu}^\beta=H_{\mu\nu}^{\mathrm{core}}+J_{\mu\nu}-F_{\mu\nu}^{\mathrm{XC}\beta} \qquad (4.41)$$

where $F_{\mu\nu}^{\mathrm{XC}\alpha}$ and $F_{\mu\nu}^{\mathrm{XC}\beta}$ are the exchange-correlation parts of the Fock matrices dependent on the exchange-correlation functional used. The Pople-Nesbet equations are obtained simply by allowing

$$F_{\mu\nu}^{\mathrm{XC}\alpha}=K_{\mu\nu}^\alpha \qquad (4.42)$$

and similarly for the beta equation. Thus, the density and energy are obtained in a manner analogous to that for the Hartree-Fock method. Initial guesses are made for the MO coefficients and an iterative process applied until self consistency is obtained.

4.3.3 Exchange-Correlation Functionals

There are an increasing number of exchange and correlation functionals and hybrid DFT methods available to the quantum chemist, many of which are very effective. In short, there are nowadays five basic working types of functionals (five rungs on the Perdew's "Jacob‘s Ladder"): those based on the local spin density approximation (LSDA) are on the first rung, those based on generalized gradient approximations (GGA) are on the second rung. Functionals that include not only density gradient corrections (as in the GGA functionals), but also a dependence on the electron kinetic energy density and/or the Laplacian of the electron density, occupy the third rung of the Jacob‘s Ladder and are known as "meta-GGA" functionals. The latter lead to a systematic, and often substantial improvement over GGA for thermochemistry and reaction kinetics. Among the meta-GGA functionals, a particular attention deserve the VSXC functional [26], the functional of Becke and Roussel for exchange [27], and for correlation [28] (the BR89B94 meta-GGA combination [28]). The latter functional did not receive enough popularity until recently, mainly because it was not representable in an analytic form. In Q-CHEM, BR89B94 is implemented now self-consistently in a fully analytic form, based on the recent work [29]. The one and only non-empirical meta-GGA functional, the TPSS functional [30], was also implemented recently in Q-CHEM [31]. Each of the above mentioned "pure" functionals can be combined with a fraction of exact (Hartree-Fock) non-local exchange energy replacing a similar fraction from the DFT local exchange energy. When a nonzero amount of Hartree-Fock exchange is used (less than a 100%), the corresponding functional is a hybrid extension (a global hybrid) of the parent "pure" functional. In most cases a hybrid functional would have one or more (up to 21 so far) linear mixing parameters that are fitted to experimental data. An exception is the hybrid extension of the TPSS meta-GGA functional, the non-empirical TPSSh scheme, which is also implemented now in Q-CHEM [31].

The forth rung of functionals ("hyper-GGA" functionals) involve occupied Kohn-Sham orbitals as additional non-local variables [32–35]. This helps tremendously in describing cases of strong inhomogeneity and strong non-dynamic correlation, that are evasive for global hybrids at GGA and meta-GGA levels of the theory. The success is mainly due to one novel feature of these functionals: they incorporate a 100% of exact (or HF) exchange combined with a hyper-GGA model correlation. Employing a 100% of exact exchange has been a long standing dream in DFT, but most previous attempts were unsuccessful. The correlation models used in the hyper-GGA schemes B05 [32] and PSTS [35], properly compensate the spuriously high non-locality of the exact exchange hole, so that cases of strong non-dynamic correlation become treatable.

In addition to some GGA and meta-GGA variables, the B05 scheme employs a new functional variable, namely, the exact-exchange energy density:

$$e_X^{\mathrm{HF}}(r) = -\frac{1}{2}\int dr' \, \frac{|n(r,r^{'})|^2}{|r-r^{'}|}, \tag{4.43}$$

where

$$n(r,r^{'}) = \frac{1}{\rho(r)}\sum_i^{occ} \varphi_i^{\mathrm{ks}}(r)\varphi_i^{\mathrm{ks}}(r^{'}). \tag{4.44}$$

This new variable enters the correlation energy component in a rather sophisticated nonlinear manner [32]: This presents a huge challenge for the practical implementation of such functionals, since they require a Hartree-Fock type of calculation at each grid point, which renders the task impractical. Significant progress in implementing efficiently the B05 functional was reported only recently [36, 37]. This new implementation achieves a speed-up of the B05 calculations by a factor of 100 based on resolution-of-identity (RI) technique (the RI-B05 scheme) and analytical interpolations. Using this methodology, the PSTS hyper-GGA was also implemented in Q-CHEM more recently [31]. For the time being only single-point SCF calculations are available for RI-B05 and RI-PSTS (the energy gradient will be available soon).

In contrast to B05 and PSTS, the forth-rung functional MCY employs a 100% global exact exchange, not only as a separate energy component of the functional, but also as a non-linear variable used the MCY correlation energy expression [33, 34]. Since this variable is the same at each grid point, it has to be calculated only once per SCF iteration. The form of the MCY correlation functional is deduced from known adiabatic connection and coordinate scaling relationships which, together with a few fitting parameters, provides a good correlation match to the exact exchange. The MCY functional [33] in its MCY2 version [34] is now implemented in Q-CHEM, as described in Ref. [31].

The fifth-rung functionals include not only occupied Kohn-Sham orbitals, but also unoccupied orbitals, which improves further the quality of the exchange-correlation energy. The practical application so far of these consists of adding empirically a small fraction of correlation energy obtained from MP2-like post-SCF calculation [38, 39]. Such functionals are known as "double-hybrids". A more detailed description of some these as implemented in Q-CHEM is given in Subsections 4.3.9 and 4.3.4.3. Finally, the so-called range-separated (or long-range corrected, LRC) functionals that employ exact exchange for the long-range part of the functional are showing excellent performance and considerable promise (see Section 4.3.4). In addition, many of the functionals can be augmented by an empirical dispersion correction, "-D" (see Section 4.3.6).

In summary, Q-CHEM includes the following exchange and correlation functionals:

LSDA functionals:

- Slater-Dirac (Exchange) [24]
- Vokso-Wilk-Nusair (Correlation) [40]
- Perdew-Zunger (Correlation) [41]
- Wigner (Correlation) [42]
- Perdew-Wang 92 (Correlation) [43]
- Proynov-Kong 2009 (Correlation) [44]

GGA functionals:

- Becke86 (Exchange) [45]
- Becke88 (Exchange) [46]
- PW86 (Exchange) [47]
- refit PW86 (Exchange) [48]
- Gill96 (Exchange) [49]
- Gilbert-Gill99 (Exchange [50]
- Lee-Yang-Parr (Correlation) [51]
- Perdew86 (Correlation) [52]
- GGA91 (Exchange and correlation) [53]
- mPW1PW91 (Exchange and Correlation) [54]
- mPW1PBE (Exchange and Correlation)
- mPW1LYP (Exchange and Correlation)
- PBE (Exchange and correlation) [55, 56]
- revPBE (Exchange) [57]
- PBE0 (25% Hartree-Fock exchange + 75% PBE exchange + 100% PBE correlation) [58]

- PBE50 (50% Hartree-Fock exchange + 50% PBE exchange + 100% PBE correlation)
- B3LYP (Exchange and correlation within a hybrid scheme) [59]
- B3PW91 (B3 Exchange + PW91 correlation)
- B3P86 (B3 Exchange + PW86 correlation)
- B5050LYP (50% Hartree-Fock exchange + 5% Slater exchange + 42% Becke exchange + 100% LYP correlation) [60]
- BHHLYP (50% Hartree-Fock exchange + 50% Becke exchange + 100% LYP correlation) [59]
- O3LYP (Exchange and correlation) [61]
- X3LYP (Exchange and correlation) [62]
- CAM-B3LYP (Range separated exchange and LYP correlation) [63]
- Becke97 (Exchange and correlation within a hybrid scheme) [56, 64]
- Becke97-1 (Exchange and correlation within a hybrid scheme) [56, 65]
- Becke97-2 (Exchange and correlation within a hybrid scheme) [56, 66]
- B97-D (Exchange and correlation and empirical dispersion correction) [67]
- HCTH (Exchange- correlation within a hybrid scheme) [56, 65]
- HCTH-120 (Exchange- correlation within a hybrid scheme) [56, 68]
- HCTH-147 (Exchange- correlation within a hybrid scheme) [56, 68]
- HCTH-407 (Exchange- correlation within a hybrid scheme) [56, 69]
- The ωB97X functionals developed by Chai and Gordon [70] (Exchange and correlation within a hybrid scheme, with long-range correction, see further in this manual for details)
- BNL (Exchange GGA functional) [71, 72]
- BOP (Becke88 exchange plus the “one-parameter progressive” correlation functional, OP) [73]
- PBEOP (PBE Exchange plus the OP correlation functional) [73]
- SOGGA (Exchange plus the PBE correlation functional) [74]
- SOGGA11 (Exchange and Correlation) [75]
- SOGGA11-X (Exchange and Correlation within a hybrid scheme, with re-optimized SOGGA11 parameters) [76]
- LRC-ωPBEPBE (Long-range corrected PBE exchange and PBE correlation) [77]
- LRC-ωPBEhPBE (Long-range corrected hybrid PBE exchange and PBE correlation) [78]

Note: The OP correlation functional used in BOP has been parameterized for use with Becke88 exchange, whereas in the PBEOP functional, the same correlation ansatz is re-parameterized for use with PBE exchange. These two versions of OP correlation are available as the correlation functionals (B88)OP and (PBE)OP. The BOP functional, for example, consists of (B88)OP correlation combined with Becke88 exchange.

Meta-GGA functionals involving the kinetic energy density (τ), andor the Laplacian of the electron density:

- VSXC (Exchange and Correlation) [26]
- TPSS (Exchange and Correlation in a single non-empirical scheme) [30, 31]
- TPSSh (Exchange and Correlation within a non-empirical hybrid scheme) [79]
- BMK (Exchange and Correlation within a hybrid scheme) [80]
- M05 (Exchange and Correlation within a hybrid scheme) [81, 82]
- M05-2X (Exchange and Correlation within a hybrid scheme) [82, 83]
- M06-L (Exchange and Correlation) [82, 84]
- M06-HF (Exchange and Correlation within a hybrid scheme) [82, 85]
- M06 (Exchange and Correlation within a hybrid scheme) [82, 86]
- M06-2X (Exchange and Correlation within a hybrid scheme) [82, 86]
- M08-HX (Exchange and Correlation within a hybrid scheme) [87]
- M08-SO (Exchange and Correlation within a hybrid scheme) [87]
- M11-L (Exchange and Correlation) [88]
- M11 (Exchange and Correlation within a hybrid scheme, with long-range correction) [89]
- BR89 (Exchange) [27, 29]
- B94 (Correlation) [28, 29]
- B95 (Correlation) [90]
- B1B95 (Exchange and Correlation) [90]
- PK06 (Correlation) [91]

Hyper-GGA functionals:

- B05 (A full exact-exchange Kohn-Sham scheme of Becke that accounts for static correlation via real-space corrections) [32, 36, 37]
- mB05 (Modified B05 method that has simpler functional form and SCF potential) [92]
- PSTS (Hyper-GGA functional of Perdew-Staroverov-Tao-Scuseria) [35]

- MCY2 (The adiabatic connection-based MCY2 functional) [31, 33, 34]

  Fifth-rung, double-hybrid (DH) functionals:

- ωB97X-2 (Exchange and Correlation within a DH generalization of the LC corrected ωB97X scheme) [39]
- B2PLYP (another DH scheme proposed by Grimme, based on GGA exchange and correlation functionals) [67]
- XYG3 and XYGJ-OS (an efficient DH scheme based on generalization of B3LYP) [93]

In addition to the above functional types, Q-CHEM contains the Empirical Density Functional 1 (EDF1), developed by Adamson, Gill and Pople [94]. EDF1 is a combined exchange and correlation functional that is specifically adapted to yield good results with the relatively modest-sized 6-31+G* basis set, by direct fitting to thermochemical data. It has the interesting feature that exact exchange mixing was not found to be helpful with a basis set of this size. Furthermore, for this basis set, the performance substantially exceeded the popular B3LYP functional, while the cost of the calculations is considerably lower because there is no need to evaluate exact (non-local) exchange. We recommend consideration of EDF1 instead of either B3LYP or BLYP for density functional calculations on large molecules, for which basis sets larger than 6-31+G* may be too computationally demanding.

EDF2, another Empirical Density Functional, was developed by Ching Yeh Lin and Peter Gill [95] in a similar vein to EDF1, but is specially designed for harmonic frequency calculations. It was optimized using the cc-pVTZ basis set by fitting into experimental *harmonic* frequencies and is designed to describe the potential energy curvature well. Fortuitously, it also performs better than B3LYP for thermochemical properties.

A few more words deserve the hybrid functionals [59], where several different exchange and correlation functionals can be combined linearly to form a hybrid functional. These have proven successful in a number of reported applications. However, since the hybrid functionals contain HF exchange they are more expensive that pure DFT functionals. Q-CHEM has incorporated two of the most popular hybrid functionals, B3LYP [96] and B3PW91 [27], with the additional option for users to define their own hybrid functionals via the *$xc_functional* keyword (see user-defined functionals in Section 4.3.17, below). Among the latter, a recent new hybrid combination available in Q-CHEM is the 'B3tLap' functional, based on Becke's B88 GGA exchange and the 'tLap' (or 'PK06') meta-GGA correlation [91, 97]. This hybrid combination is on average more accurate than B3LYP, BMK, and M06 functionals for thermochemistry and better than B3LYP for reaction barriers, while involving only five fitting parameters. Another hybrid functional in Q-CHEM that deserves attention is the hybrid extension of the BR89B94 meta-GGA functional [28, 97]. This hybrid functional yields a very good thermochemistry results, yet has only three fitting parameters.

In addition, Q-CHEM now includes the M05 and M06 suites of density functionals. These are designed to be used only with certain definite percentages of Hartree-Fock exchange. In particular, M06-L [84] is designed to be used with no Hartree-Fock exchange (which reduces the cost for large molecules), and M05 [81], M05-2X [83], M06, and M06-2X [86] are designed to be used with 28%, 56%, 27%, and 54% Hartree-Fock exchange. M06-HF [85] is designed to be used with 100% Hartree-Fock exchange, but it still contains some local DFT exchange because the 100% non-local Hartree-Fock exchange replaces only some of the local exchange.

Note: The hybrid functionals are not simply a pairing of an exchange and correlation functional, but are a combined exchange-correlation functional (*i.e.*, B-LYP and B3LYP vary in the correlation contribution in addition to the exchange part).

4.3.4 Long-Range-Corrected DFT

As pointed out in Ref. 98 and elsewhere, the description of charge-transfer excited states within density functional theory (or more precisely, time-dependent DFT, which is discussed in Section 6.3) requires full (100%) non-local Hartree-Fock exchange, at least in the limit of large donor–acceptor distance. Hybrid functionals such as B3LYP [96] and PBE0 [58] that are well-established and in widespread use, however, employ only 20% and 25% Hartree-Fock 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 clusters, where TDDFT often predicts a near-continuum of of spurious, low-lying charge transfer states [99, 100]. The problems with TDDFT's description of charge transfer are not limited to large donor–acceptor distances, but have been observed at $\sim$2 Å separation, in systems as small as uracil–$(\mathrm{H_2O})_4$ [99]. 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% Hartree-Fock exchange. To date, few such functionals exist, and those that do (such as M06-HF) contain a very large number of empirical adjustable parameters. An alternative option is to attempt to preserve the form of common GGAs and hybrid functionals at short range (*i.e.*, keep the 25% Hartree-Fock exchange in PBE0) while incorporating 100% Hartree-Fock exchange at long range. Functionals along these lines are known variously as "Coulomb-attenuated" functionals, "range-separated" functionals, or (our preferred designation) "long-range-corrected" (LRC) density functionals. Whatever the nomenclature, these functionals are all based upon a partition of the electron-electron Coulomb potential into long- and short-range components, using the error function (erf):

$$\frac{1}{r_{12}} \equiv \frac{1-\mathrm{erf}(\omega r_{12})}{r_{12}} + \frac{\mathrm{erf}(\omega r_{12})}{r_{12}} \tag{4.45}$$

The first term on the right in Eq. (4.45) 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. The basic idea of LRC-DFT is to utilize the short-range component of the Coulomb operator in conjunction with standard DFT exchange (including any component of Hartree-Fock exchange, if the functional is a hybrid), while at the same time incorporating full Hartree-Fock exchange using the long-range part of the Coulomb operator. This 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 extra Hartree-Fock exchange.

Consider an exchange-correlation functional of the form

$$E_{\mathrm{XC}} = E_{\mathrm{C}} + E_{\mathrm{X}}^{\mathrm{GGA}} + C_{\mathrm{HF}}\, E_{\mathrm{X}}^{\mathrm{HF}} \tag{4.46}$$

in which E_{C} is the correlation energy, $E_{\mathrm{X}}^{\mathrm{GGA}}$ is the (local) GGA exchange energy, and $E_{\mathrm{X}}^{\mathrm{HF}}$ is the (non-local) Hartree-Fock exchange energy. The constant C_{HF} denotes the fraction of Hartree-Fock exchange in the functional, therefore $C_{\mathrm{HF}} = 0$ for GGAs, $C_{\mathrm{HF}} = 0.20$ for B3LYP, $C_{\mathrm{HF}} = 0.25$ for PBE0, *etc.*. The LRC version of the generic functional in Eq. (4.46) is

$$E_{\mathrm{XC}}^{\mathrm{LRC}} = E_{\mathrm{C}} + E_{\mathrm{X}}^{\mathrm{GGA,SR}} + C_{\mathrm{HF}}\, E_{\mathrm{X}}^{\mathrm{HF,SR}} + E_{\mathrm{X}}^{\mathrm{HF,LR}} \tag{4.47}$$

in which the designations "SR" and "LR" in the various exchange energies indicate that these components of the functional are evaluated using either the short-range (SR) or the long-range (LR) component of the Coulomb operator. (The correlation energy E_{C} is evaluated using the full Coulomb operator.) The LRC functional in Eq. (4.47) incorporates full Hartree-Fock exchange in the asymptotic limit via the final term, $E_{\mathrm{X}}^{\mathrm{HF,LR}}$. To fully specify the LRC functional, one must choose a value for the range separation parameter ω in Eq. (4.45); in the limit $\omega \to 0$, the LRC functional in Eq. (4.47) reduces to the original functional in Eq. (4.46), while the $\omega \to \infty$ limit corresponds to a new functional, $E_{\mathrm{XC}} = E_{\mathrm{C}} + E_{\mathrm{X}}^{\mathrm{HF}}$. It is well known that full Hartree-Fock 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 probably not worth considering [77, 101].

Evaluation of the short- and long-range Hartree-Fock exchange energies is straightforward [102], so the crux of LRC-DFT rests upon the form of the short-range GGA exchange energy. Several different short-range GGA exchange functionals are available in Q-CHEM, including short-range variants of B88 and PBE exchange described by Hirao and co-workers [103, 104], an alternative formulation of short-range PBE exchange proposed by Scuseria and co-workers [105], and several short-range variants of B97 introduced by Chai and Head-Gordon [39, 70, 106, 107]. The reader is referred to these papers for additional methodological details.

These LRC-DFT functionals have been shown to remove the near-continuum of spurious charge-transfer excited states that appear in large-scale TDDFT calculations [101]. However, certain results depend sensitively upon the range-separation parameter ω [77, 78, 100, 101], and the results of LRC-DFT calculations must therefore be interpreted with caution, and probably for a range of ω values. In two recent benchmark studies of several LRC density functionals, Rohrdanz and Herbert [77, 78] have considered the errors engendered, as a function of ω, in both ground-state properties and also TDDFT vertical excitation energies. In Ref. 101, the sensitivity of valence excitations versus charge-transfer excitation energies in TDDFT was considered, again as a function of ω. A careful reading of these references is suggested prior to performing any LRC-DFT calculations.

Within Q-CHEM 3.2, there are three ways to perform LRC-DFT calculations.

4.3.4.1 LRC-DFT with the μB88, μPBE, and ωPBE exchange functionals

The form of $E_{\mathrm{X}}^{\mathrm{GGA,SR}}$ is different for each different GGA exchange functional, and short-range versions of B88 and PBE exchange are available in Q-CHEM through the efforts of the Herbert group. Versions of B88 and PBE, in which the Coulomb attenuation is performed according to the procedure of Hirao [104], are denoted as μB88 and μPBE, respectively (since μ, rather than ω, is the Hirao group's notation for the range-separation parameter). Alternatively, a short-range version of PBE exchange called ωPBE is available, which is constructed according to the prescription of Scuseria and co-workers [105].

These short-range exchange functionals can be used in the absence of long-range Hartree-Fock exchange, and using a combination of ωPBE exchange and PBE correlation, a user could, for example, employ the short-range hybrid functional recently described by Heyd, Scuseria, and Ernzerhof [108]. Short-range hybrids appear to be most appropriate for extended systems, however. Thus, within Q-CHEM, short-range GGAs should be used with long-range Hartree-Fock exchange, as in Eq. 4.47. Long-range Hartree-Fock exchange is requested by setting LRC_DFT to TRUE.

LRC-DFT is thus available for any functional whose exchange component consists of some combination of Hartree-Fock, B88, and PBE exchange (*e.g.*, BLYP, PBE, PBE0, BOP, PBEOP, and various user-specified combinations, but not B3LYP or other functionals whose exchange components are more involved). Having specified such a functional via the EXCHANGE and CORRELATION variables, a user may request the corresponding LRC functional by setting LRC_DFT to TRUE. Long-range-corrected variants of PBE0, BOP, and PBEOP must be obtained through the appropriate user-specified combination of exchange and correlation functionals (as demonstrated in the example below). In any case, the value of ω must also be specified by the user. Analytic energy gradients are available but analytic Hessians are not. TDDFT vertical excitation energies are also available.

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) Use the long-range-corrected version of the requested functional.

RECOMMENDATION:

Long-range correction is available for any combination of Hartree-Fock, B88, and PBE exchange (along with any stand-alone correlation functional).

OMEGA

Sets the Coulomb attenuation parameter ω.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $\omega = n/1000$, in units of bohr^{-1}

RECOMMENDATION:

None

Example 4.5 Application of LRC-BOP to $(\mathrm{H_2O})_2^-$.

```
$comment
   To obtain LRC-BOP, a short-range version of BOP must be specified,
   using muB88 short-range exchange plus (B88)OP correlation, which is
   the version of OP parameterized for use with B88.
$end

$molecule
-1 2
O           1.347338    -.017773    -.071860
H           1.824285     .813088     .117645
H           1.805176    -.695567     .461913
O          -1.523051    -.002159    -.090765
H           -.544777    -.024370    -.165445
H          -1.682218     .174228     .849364
$end

$rem
   EXCHANGE       GEN
   BASIS          6-31(1+,3+)G*
   LRC_DFT        TRUE
   OMEGA          330      ! = 0.330 a.u.
$end
```

```
$xc_functional
   C    (B88)OP    1.0
   X    muB88      1.0
$end
```

Regarding the choice of functionals and ω values, it has been found that the Hirao and Scuseria *ansatz* afford virtually identical TDDFT excitation energies, for all values of ω [78]. Thus, functionals based on μPBE versus ωPBE provide the same excitation energies, as a function of ω. However, the ωPBE functional appears to be somewhat superior in the sense that it can provide accurate TDDFT excitation energies *and* accurate ground-state properties using the *same* value of ω [78], whereas this does not seem to be the case for functionals based on μB88 or μPBE [77].

Recently, Rohrdanz *et al.* [78] have published a thorough benchmark study of both ground- and excited-state properties, using the "LRC-ωPBEh" functional, a hybrid (hence the "h") that contains a fraction of short-range Hartree-Fock exchange in addition to full long-range Hartree-Fock exchange:

$$E_{\rm XC}(\text{LRC-}\omega\text{PBEh}) = E_{\rm C}(\text{PBE}) + E_{\rm X}^{\rm SR}(\omega\text{PBE}) + C_{\rm HF}\, E_{\rm X}^{\rm SR}(\text{HF}) + E_{\rm X}^{\rm LR}(\text{HF}) \tag{4.48}$$

The statistically-optimal parameter set, consider both ground-state properties and TDDFT excitation energies together, was found to be $C_{\rm HF} = 0.2$ and $\omega = 0.2$ bohr^{-1} [78]. With these parameters, the LRC-ωPBEh functional outperforms the traditional hybrid functional PBE0 for ground-state atomization energies and barrier heights. For TDDFT excitation energies corresponding to localized excitations, TD-PBE0 and TD-LRC-ωPBEh show similar statistical errors of $\sim$0.3 eV, but the latter functional also exhibits only $\sim$0.3 eV errors for charge-transfer excitation energies, whereas the statistical error for TD-PBE0 charge-transfer excitation energies is 3.0 eV! Caution is definitely warranted in the case of charge-transfer excited states, however, as these excitation energies are very sensitive to the precise value of ω [78, 100]. It was later found that the parameter set ($C_{\rm HF} = 0$, $\omega = 0.3$ bohr^{-1}) provides similar (statistical) performance to that described above, although the predictions for specific charge-transfer excited states can be somewhat different as compared to the original parameter set [100].

Example 4.6 Application of LRC-ωPBEh to the C_2H_4—C_2F_4 hetero-dimer at 5 Å separation.

```
$comment
    This example uses the "optimal" parameter set discussed above.
    It can also be run by setting "EXCHANGE LRC-WPBEhPBE".
$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
```

4.3.4.2 LRC-DFT with the BNL Functional

The Baer-Neuhauser-Livshits (BNL) functional [71, 72] is also based on the range separation of the Coulomb operator in Eq. 4.45. Its functional form resembles Eq. 4.47:

$$E_{\mathrm{XC}} = E_{\mathrm{C}} + C_{\mathrm{GGA,X}} E_{\mathrm{X}}^{\mathrm{GGA,SR}} + E_{\mathrm{X}}^{\mathrm{HF,LR}} \tag{4.49}$$

where the recommended GGA correlation functional is LYP. The recommended GGA exchange functional is BNL, which is described by a local functional [109]. For ground state properties, the optimized value for $C_{GGA,X}$ (scaling factor for the BNL exchange functional) was found to be 0.9.

The value of ω in BNL calculations can be chosen in several different ways. For example, one can use the optimized value ω=0.5 bohr^{-1}. For calculation of excited states and properties related to orbital energies, it is strongly recommend to tune ω as described below[110, 111].

System-specific optimization of ω is based on Koopmans conditions that would be satisfied for the exact functional[110], that is, ω is varied until the Koopmans IE/EA for the HOMO/LUMO is equal to ΔE IE/EA. Based on published benchmarks [72, 112], this system-specific approach yields the most accurate values of IEs and excitation energies.

The script that optimizes ω is called `OptOmegaIPEA.pl` and is located in the *$QC/bin* directory. The script optimizes ω in the range 0.1-0.8 (100-800). See the script for the instructions how to modify the script to optimize in a broader range. To execute the script, you need to create three inputs for a BNL job using the same geometry and basis set for a neutral molecule (`N.in`), anion (`M.in`), and cation (`P.in`), and then type `'OptOmegaIPEA.pl >& optomega'`. The script will run creating outputs for each step (`N_*`, `P_*`, `M_*`) writing the optimization output into `optomega`.

A similar script, `OptOmegaIP.pl`, will optimize ω to satisfy the Koopmans condition for the IP only. This script minimizes $J = (IP + \epsilon_{HOMO})^2$, not the absolute values.

Note: (i) If the system does not have positive EA, then the tuning should be done according to the IP condition only. The IPEA script will yield a wrong value of ω in such cases.
(ii) 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.
(iii) When tuning omega we recommend taking the amount of X BNL in the XC part as 1.0 and not 0.9.

The *$xc_functional* keyword for a BNL calculation reads:

```
$xc_functional
  X HF 1.0
  X BNL 0.9
  C LYP 1.0
$end
```

and the *$rem* keyword reads

```
$rem
  EXCHANGE       GENERAL
  SEPARATE_JK    TRUE
  OMEGA          500      != 0.5 Bohr$^{-1}$
  DERSCREEN      FALSE    !if performing unrestricted calcn
  IDERIV         0        !if performing unrestricted Hessian evaluation
$end
```

4.3.4.3 LRC-DFT with ωB97, ωB97X, ωB97X-D, and ωB97X-2 Functionals

Also available in Q-CHEM are the ωB97 [70], ωB97X [70], ωB97X-D [106], and ωB97X-2 [39] functionals, recently developed by Chai and Head-Gordon. These authors have proposed a very simple *ansatz* to extend any $\mathrm{E}_{\mathrm{X}}^{\mathrm{GGA}}$ to $\mathrm{E}_{\mathrm{X}}^{\mathrm{GGA,SR}}$, as long as the SR operator has considerable spatial extent [70, 107]. With the use of flexible GGAs, such as Becke97 functional [64], their new LRC hybrid functionals [70, 106, 107] outperform the corresponding global hybrid functionals (*i.e.*, B97) and popular hybrid functionals (*e.g.*, B3LYP) in thermochemistry, kinetics, and non-covalent interactions, which has not been easily achieved by the previous LRC hybrid functionals. In addition, the qualitative failures i of the commonly used hybrid density functionals in some "difficult problems", such as dissociation of symmetric radical cations and long-range charge-transfer excitations, are significantly reduced by these new functionals [70, 106, 107]. Analytical gradients and analytical Hessians are available for ωB97, ωB97X, and ωB97X-D.

Example 4.7 Application of ωB97 functional to nitrogen dimer.

```
$comment
Geometry optimization, followed by a TDDFT calculation.
$end

$molecule
0 1
N1
N2 N1 1.1
$end

$rem
jobtype          opt
exchange         omegaB97
basis            6-31G*
$end

@@@
```

```
$molecule
READ
$end

$rem
jobtype         sp
exchange        omegaB97
basis           6-31G*
scf_guess       READ
cis_n_roots     10
rpa             true
$end
```

Example 4.8 Application of ωB97X functional to nitrogen dimer.

```
$comment
Frequency calculation (with analytical Hessian methods).
$end

$molecule
0 1
N1
N2 N1 1.1
$end

$rem
jobtype         freq
exchange        omegaB97X
basis           6-31G*
$end
```

Among these new LRC hybrid functionals, ωB97X-D is a DFT-D (density functional theory with empirical dispersion corrections) functional, where the total energy is computed as the sum of a DFT part and an empirical atomic-pairwise dispersion correction. Relative to ωB97 and ωB97X, ωB97X-D is significantly superior for non-bonded interactions, and very similar in performance for bonded interactions. However, it should be noted that the remaining short-range self-interaction error is somewhat larger for ωB97X-D than for ωB97X than for ωB97. A careful reading of Refs. 70, 106, 107 is suggested prior to performing any DFT and TDDFT calculations based on variations of ωB97 functional. ωB97X-D functional automatically involves two keywords for the dispersion correction, DFT_D and DFT_D_A, which are described in Section 4.3.6.

Example 4.9 Application of ωB97X-D functional to methane dimer.

```
$comment
Geometry optimization.
$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        omegaB97X-D
basis           6-31G*
$end
```

Similar to the existing double-hybrid density functional theory (DH-DFT) [38, 93, 113–115], which is described in Section 4.3.9, LRC-DFT can be extended to include non-local orbital correlation energy from second-order Møller-Plesset perturbation theory (MP2) [116], that includes a same-spin (ss) component E_c^{ss}, and an opposite-spin (os) component E_c^{os} of PT2 correlation energy. The two scaling parameters, c_{ss} and c_{os}, are introduced to avoid double-counting correlation with the LRC hybrid functional.

$$E_{\text{total}} = E_{\text{LRC-DFT}} + c_{ss}E_c^{ss} + c_{os}E_c^{os} \tag{4.50}$$

Among the ωB97 series, ωB97X-2 [39] is a long-range corrected double-hybrid (DH) functional, which can greatly reduce the self-interaction errors (due to its high fraction of Hartree-Fock exchange), and has been shown significantly superior for systems with 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. ωB97X-2(LP) was parameterized with the 6-311++G(3df,3pd) basis set (the large Pople type basis set), while ωB97X-2(TQZ) was parameterized with the TQ extrapolation to the basis set limit. A careful reading of Ref. 39 is thus highly advised.

ωB97X-2(LP) and ωB97X-2(TQZ) automatically involve three keywords for the PT2 correlation energy, DH, SSS_FACTOR and SOS_FACTOR, which are described in Section 4.3.9. The PT2 correlation energy can also be computed with the efficient resolution-of-identity (RI) methods (see Section 5.5).

Example 4.10 Application of ωB97X-2(LP) functional to LiH molecules.

```
$comment
Geometry optimization and frequency calculation on LiH, followed by
single-point calculations with non-RI and RI approaches.
$end

$molecule
0 1
H
Li H 1.6
$end

$rem
jobtype         opt
exchange        omegaB97X-2(LP)
correlation     mp2
basis           6-311++G(3df,3pd)
$end
```

```
@@@

$molecule
READ
$end

$rem
jobtype         freq
exchange        omegaB97X-2(LP)
correlation     mp2
basis           6-311++G(3df,3pd)
$end

@@@

$molecule
READ
$end

$rem
jobtype         sp
exchange        omegaB97X-2(LP)
correlation     mp2
basis           6-311++G(3df,3pd)
$end

@@@

$molecule
READ
$end

$rem
jobtype          sp
exchange         omegaB97X-2(LP)
correlation      rimp2
basis            6-311++G(3df,3pd)
aux_basis        rimp2-aug-cc-pvtz
$end
```

Example 4.11 Application of ωB97X-2(TQZ) functional to LiH molecules.

```
$comment
Single-point calculations on LiH.
$end

$molecule
0 1
H
Li H 1.6
$end

$rem
jobtype         sp
exchange        omegaB97X-2(TQZ)
correlation     mp2
basis           cc-pvqz
$end
```

```
@@@

$molecule
READ
$end

$rem
jobtype         sp
exchange        omegaB97X-2(TQZ)
correlation     rimp2
basis           cc-pvqz
aux_basis       rimp2-cc-pvqz
$end
```

4.3.4.4 LRC-DFT with the M11 Family of Functionals

The Minnesota family of functional by Truhlar's group has been recently updated by adding two new functionals: M11-L [88] and M11 [89]. The M11 functional is a long-range corrected meta-GGA, obtained by using the LRC scheme of Chai and Head-Gordon (see above), with the successful parameterization of the Minnesota meta-GGA functionals:

$$E_{xc}^{M11} = \left(\frac{X}{100}\right) E_x^{SR-HF} + \left(1 - \frac{X}{100}\right) E_x^{SR-M11} + E_x^{LR-HF} + E_c^{M11} \tag{4.51}$$

with the percentage of Hartree-Fock exchange at short range X being 42.8. An extension of the LRC scheme to local functional (no HF exchange) was introduced in the M11-L functional by means of the dual-range exchange:

$$E_{xc}^{M11-L} = E_x^{SR-M11} + E_x^{LR-M11} + E_c^{M11-L} \tag{4.52}$$

The correct long-range scheme is selected automatically with the input keywords. A careful reading of the references [88, 89] is suggested prior to performing any calculations with the M11 functionals.

Example 4.12 Application of M11 functional to water molecule

```
$comment
Optimization of H2O with M11
$end

$molecule
0 1
O 0.000000 0.000000  0.000000
H 0.000000 0.000000  0.956914
H 0.926363 0.000000 -0.239868
$end

$rem
jobtype opt
exchange m11
basis 6-31+G(d,p)
$end
```

4.3.5 Nonlocal Correlation Functionals

Q-CHEM includes four nonlocal correlation functionals that describe long-range dispersion (i.e. van der Waals) interactions:

- vdW-DF-04, developed by Langreth, Lundqvist, and coworkers [117, 118] and implemented as described in Ref. [119];
- vdW-DF-10 (also known as vdW-DF2), which is a re-parameterization [120] of vdW-DF-04, implemented in the same way as its precursor [119];
- VV09, developed [121] and implemented [122] by Vydrov and Van Voorhis;
- VV10 by Vydrov and Van Voorhis [123].

All these functionals are implemented self-consistently and analytic gradients with respect to nuclear displacements are available [119, 122, 123]. The nonlocal 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. Note that 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:

```
CORRELATION        PW92
NL_CORRELATION     vdW-DF-10
```

VV10 is used in combination with PBE correlation, which must be added explicitly. In addition, the values of two parameters, C and b 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:

```
CORRELATION        PBE
NL_CORRELATION     VV10
NL_VV_C            93
NL_VV_B            590
```

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.

As opposed to local (LSDA) and semilocal (GGA and meta-GGA) functionals, evaluated as a single 3D integral over space [see Eq. (4.37)], non-local functionals require double integration over the spatial variables:

$$E_{\rm c}^{\rm nl} = \int f(\mathbf{r}, \mathbf{r}')\, d\mathbf{r} d\mathbf{r}'. \qquad (4.53)$$

In practice, this double integration is performed numerically on a quadrature grid [119, 122, 123]. By default, the SG-1 quadrature (described in Section 4.3.13 below) is used to evaluate $E_{\rm c}^{\rm nl}$, 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, so that SG-1 or even SG-0 grids can be used in most cases. However, a finer grid may be required for the (semi)local parts of the functional, as controlled by the XC_GRID variable.

Example 4.13 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            75000302
NL_CORRELATION     VV10
NL_GRID            1
NL_VV_C            93
NL_VV_B            590
$end
```

In the above example, an EML-(75,302) 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.

4.3.6 DFT-D Methods

4.3.6.1 Empirical dispersion correction from Grimme

Thanks to the efforts of the Sherrill group, the popular empirical dispersion corrections due to Grimme [67] are now available in Q-CHEM. Energies, analytic gradients, and analytic second derivatives are available. Grimme's empirical dispersion corrections can be added to any of the density functionals available in Q-CHEM.

DFT-D methods add an extra term,

$$E_{\text{disp}} = -s_6 \sum_A \sum_{B<A} \frac{C_6^{AB}}{R_{AB}^6} f_{dmp}(R_{AB}) \tag{4.54}$$

$$C_6^{AB} = \sqrt{C_6^A C_6^B}, \tag{4.55}$$

$$f_{\text{dmp}}(R_{AB}) = \frac{1}{1+e^{-d(R_{AB}/R_{AB}^0-1)}} \tag{4.56}$$

where s_6 is a global scaling parameter (near unity), f_{dmp} is a damping parameter meant to help avoid double-counting correlation effects at short range, d is a global scaling parameter for the damping function, and R_{AB}^0 is the sum of the van der Waals radii of atoms A and B.

DFT-D using Grimme's parameters may be turned on using

```
DFT_D      EMPIRICAL_GRIMME
```

Grimme has suggested scaling factors s_6 of 0.75 for PBE, 1.2 for BLYP, 1.05 for BP86, and 1.05 for B3LYP; these are the default values of s_6 when those functionals are used. Otherwise, the default value of s_6 is 1.0.

It is possible to specify different values of s_6, d, the atomic C_6 coefficients, or the van der Waals radii by using the *$empirical_dispersion* keyword; 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
```

Any values not specified explicitly will default to the values in Grimme's model.

4.3.6.2 Empirical dispersion correction from Chai and Head-Gordon

The empirical dispersion correction from Chai and Head-Gordon [106] employs a different damping function and can be activated by using

```
DFT_D      EMPIRICAL_CHG
```

It uses another keyword DFT_D_A to control the strength of dispersion corrections.

DFT_D
Controls the application of DFT-D or DFT-D3 scheme.
TYPE:
LOGICAL
DEFAULT:
None
OPTIONS:

FALSE	(or 0) Do not apply the DFT-D or DFT-D3 scheme
EMPIRICAL_GRIMME	dispersion correction from Grimme
EMPIRICAL_CHG	dispersion correction from Chai and Head-Gordon
EMPIRICAL_GRIMME3	dispersion correction from Grimme's DFT-D3 method (see Section 4.3.8)

RECOMMENDATION:
NONE

DFT_D_A

Controls the strength of dispersion corrections in the Chai-Head-Gordon DFT-D scheme in Eq.(3) of Ref. 106.

TYPE:

INTEGER

DEFAULT:

600

OPTIONS:

n Corresponding to $a = n/100$.

RECOMMENDATION:

Use default, *i.e.*, $a = 6.0$

4.3.7 XDM DFT Model of Dispersion

While standard DFT functionals describe chemical bonds relatively well, one major deficiency is their inability to cope with dispersion interactions, *i.e.*, van der Waals (vdW) interactions. Becke and Johnson have proposed a conceptually simple yet accurate dispersion model called the exchange-dipole model (XDM) [32, 124]. In this model the dispersion attraction emerges from the interaction between the instant dipole moment of the exchange hole in one molecule and the induced dipole moment in another. It is a conceptually simple but 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. 125.

Fundamental to the XDM model is the calculation of the norm of the dipole moment of the exchange hole at a given point:

$$d_\sigma(r) = -\int h_\sigma(r, r')r'd^3r' - r \tag{4.57}$$

where σ labels the spin and $h_\sigma(r, r')$ is the exchange-hole function. The XDM version that is implemented in Q-CHEM employs the Becke-Roussel (BR) model exchange-hole function. It was not given in an analytical form and one had to determine its value at each grid point numerically. Q-CHEM has developed for the first time an analytical expression for this function based on non-linear interpolation and spline techniques, which greatly improves efficiency as well as the numerical stability [27].

There are two different damping functions used in the XDM model of Becke and Johnson. One of them uses only the intermolecular C6 dispersion coefficient. In its Q-CHEM implementation it is denoted as "XDM6". In this version the dispersion energy is computed as

$$E_{vdW} = \sum E_{vdW},_{ij} = -\sum_{i>j} \frac{C_{6,ij}}{R_{ij}^6 + kC_{6,ij}/(E_i^C + E_j^C)} \tag{4.58}$$

where k is a universal parameter, R_{ij} is the distance between atoms i and j, and E_{ij}^C is the sum of the absolute values of the correlation energy of free atoms i and j. The dispersion coefficients $C_{6,ij}$ is computed as

$$C_{6,ij} = \frac{\langle d_X^2 \rangle_i \langle d_X^2 \rangle_j \alpha_i \alpha_j}{\langle d_X^2 \rangle_i \alpha_j + \langle d_X^2 \rangle_j \alpha_i} \tag{4.59}$$

where $\langle d_X^2 \rangle_i$ is the exchange hole dipole moment of the atom, and α_i is the effective polarizability of the atom i in the molecule.

The XDM6 scheme is further generalized to include higher-order dispersion coefficients, which leads to the "XDM10" model in Q-CHEM implementation. The dispersion energy damping function used in XDM10 is

$$E_{vdW} = -\sum_{i>j}\left(\frac{C_{6,ij}}{R^6_{vdW,ij}+R^6_{ij}}+\frac{C_{8,ij}}{R^8_{vdW,ij}+R^8_{ij}}+\frac{C_{10,ij}}{R^{10}_{vdW,ij}+R^{10}_{ij}}\right) \tag{4.60}$$

where $C_{6,ij}$, $C_{8,ij}$ and $C_{10,ij}$ are dispersion coefficients computed at higher-order multipole (including dipole, quadrupole and octopole) moments of the exchange hole [126]. In above, $R_{vdW,ij}$ is the sum of the effective vdW radii of atoms i and j, which is a linear function of the so called critical distance $R_{C,ij}$ between atoms i and j:

$$R_{vdW,ij} = a_1 R_{C,ij} + a_2 \tag{4.61}$$

The critical distance, $R_{C,ij}$, is computed by averaging these three distances:

$$R_{C,ij} = \frac{1}{3}\left[\left(\frac{C_{8,ij}}{C_{6,ij}}\right)^{1/2}+\left(\frac{C_{10,ij}}{C_{6,ij}}\right)^{1/4}+\left(\frac{C_{10,ij}}{C_{8,ij}}\right)^{1/2}\right] \tag{4.62}$$

In the XDM10 scheme there are two universal parameters, a_1 and a_2. Their default values of 0.83 and 1.35, respectively, are due to Johnson and Becke [124], determined by least square fitting to the binding energies of a set of intermolecular complexes. Please keep in mind that these values are not the only possible optimal set to use with XDM. Becke's group has suggested later on several other XC functional combinations with XDM that employ different a_1 and a_2 values. The user is advised to consult their recent papers for more details (*e.g.*, Refs. 127, 128).

The computed vdW energy is added as a post-SCF correction. In addition, Q-CHEM also has implemented the first and second nuclear derivatives of vdW energy correction in both the XDM6 and XDM10 schemes.

Listed below are a number of useful options to customize the vdW calculation based on the XDM DFT approach.

DFTVDW_JOBNUMBER
Basic vdW job control
TYPE:
INTEGER
DEFAULT:
0
OPTIONS:
0 Do not apply the XDM scheme.
1 add vdW as energy/gradient correction to SCF.
2 add VDW as a DFT functional and do full SCF (this option only works with C6 XDM formula).
RECOMMENDATION:
none

DFTVDW_METHOD

Choose the damping function used in XDM

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 use Becke's damping function including C6 term only.

2 use Becke's damping function with higher-order (C8,C10) terms.

RECOMMENDATION:

none

DFTVDW_MOL1NATOMS

The number of atoms in the first monomer in dimer calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0-NATOMS default 0

RECOMMENDATION:

none

DFTVDW_KAI

Damping factor K for C6 only damping function

TYPE:

INTEGER

DEFAULT:

800

OPTIONS:

10-1000 default 800

RECOMMENDATION:

none

DFTVDW_ALPHA1

Parameter in XDM calculation with higher-order terms

TYPE:

INTEGER

DEFAULT:

83

OPTIONS:

10-1000

RECOMMENDATION:

none

DFTVDW_ALPHA2

Parameter in XDM calculation with higher-order terms.

TYPE:

INTEGER

DEFAULT:

155

OPTIONS:

10-1000

RECOMMENDATION:

none

DFTVDW_USE_ELE_DRV

Specify whether to add the gradient correction to the XDM energy. only valid with Becke's C6 damping function using the interpolated BR89 model.

TYPE:

LOGICAL

DEFAULT:

1

OPTIONS:

1 use density correction when applicable (default).

0 do not use this correction (for debugging purpose)

RECOMMENDATION:

none

DFTVDW_PRINT

Printing control for VDW code

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 no printing.

1 minimum printing (default)

2 debug printing

RECOMMENDATION:

none

Example 4.14 Below is a sample input illustrating a frequency calculation of a vdW complex consisted of He atom and N2 molecule.

```
$molecule

0 1
He .0 .0 3.8
N .000000 .000000 0.546986
N .000000 .000000 -0.546986
$end
```

```
    $rem
    JOBTYPE       FREQ
    IDERIV         2
    EXCHANGE      B3LYP
    !default SCF setting
    INCDFT         0
    SCF_CONVERGENCE   8
    BASIS       6-31G*
    XC_GRID           1
    SCF_GUESS       SAD
    !vdw parameters setting
    DFTVDW_JOBNUMBER    1
    DFTVDW_METHOD       1
    DFTVDW_PRINT        0
    DFTVDW_KAI         800
    DFTVDW_USE_ELE_DRV  0
    $end
```

One should note that the XDM option can be used in conjunction with different GGA, meta-GGA pure or hybrid functionals, even though the original implementation of Becke and Johnson was in combination with Hartree-Fock exchange, or with a specific meta-GGA exchange and correlation (the BR89 exchange and the BR94 correlation described in previous sections above). For example, encouraging results were obtained using the XDM option with the popular B3LYP [125]. Becke has found more recently that this model can be efficiently combined with the old GGA exchange of Perdew 86 (the P86 exchange option in Q-CHEM), and with his hyper-GGA functional B05. Using XDM together with PBE exchange plus LYP correlation, or PBE exchange plus BR94 correlation has been also found fruitful.

4.3.8 DFT-D3 Methods

Recently, Grimme proposed DFT-D3 method [129] to improve his previous DFT-D method [67] (see Section 4.3.6). Energies and analytic gradients of DFT-D3 methods are available in Q-CHEM. Grimme's DFT-D3 method can be combined with any of the density functionals available in Q-CHEM.

The total DFT-D3 energy is given by

$$E_{\text{DFT-D3}} = E_{\text{KS-DFT}} + E_{\text{disp}} \tag{4.63}$$

where $E_{\text{KS-DFT}}$ is the total energy from KS-DFT and E_{disp} is the dispersion correction as a sum of two- and three-body energies,

$$E_{\text{disp}} = E^{(2)} + E^{(3)} \tag{4.64}$$

DFT-D3 method can be turned on by five keywords, DFT_D, DFT_D3_S6, DFT_D3_RS6, DFT_D3_S8 and DFT_D3_3BODY.

DFT_D

Controls the application of DFT-D3 or DFT-D scheme.

TYPE:

LOGICAL

DEFAULT:

None

OPTIONS:

FALSE	(or 0) Do not apply the DFT-D3 or DFT-D scheme
EMPIRICAL_GRIMME3	dispersion correction from Grimme's DFT-D3 method
EMPIRICAL_GRIMME	dispersion correction from Grimme (see Section 4.3.6)
EMPIRICAL_CHG	dispersion correction from Chai and Head-Gordon (see Section 4.3.6)

RECOMMENDATION:

NONE

Grimme suggested three scaling factors s_6, $s_{r,6}$ and s_8 that were optimized for several functionals (see Table IV in Ref. 129). For example, $s_{r,6}$ of 1.217 and s_8 of 0.722 for PBE, 1.094 and 1.682 for BLYP, 1.261 and 1.703 for B3LYP, 1.532 and 0.862 for PW6B95, 0.892 and 0.909 for BECKE97, and 1.287 and 0.928 for PBE0; these are the Q-CHEM default values of $s_{r,6}$ and s_8. Otherwise, the default values of s_6, $s_{r,6}$ and s_8 are 1.0.

DFT_D3_S6

Controls the strength of dispersion corrections, s_6, in Grimme's DFT-D3 method (see Table IV in Ref. 129).

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n Corresponding to $s_6 = n/1000$.

RECOMMENDATION:

NONE

DFT_D3_RS6

Controls the strength of dispersion corrections, s$_{r6}$, in the Grimme's DFT-D3 method (see Table IV in Ref. 129).

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n Corresponding to $s_{r6} = n/1000$.

RECOMMENDATION:

NONE

DFT_D3_S8

Controls the strength of dispersion corrections, s_8, in Grimme's DFT-D3 method (see Table IV in Ref. 129).

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n Corresponding to $s_8 = n/1000$.

RECOMMENDATION:

NONE

The three-body interaction term, mentioned in Ref. 129, can also be turned on, if needed.

DFT_D3_3BODY

Controls whether the three-body interaction in Grimme's DFT-D3 method should be applied (see Eq. (14) in Ref. 129).

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

Example 4.15 Applications of B3LYP-D3 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           EMPIRICAL_GRIMME3
DFT_D3_S6       1000
DFT_D3_RS6      1261
DFT_D3_S8       1703
DFT_D3_3BODY    FALSE
$end

@@@

$molecule
READ
$end

$rem
jobtype         sp
exchange        B3LYP
basis           6-311++G**
DFT_D           EMPIRICAL_GRIMME3
DFT_D3_S6       1000
DFT_D3_RS6      1261
DFT_D3_S8       1703
DFT_D3_3BODY    FALSE
$end
```

4.3.9 Double-Hybrid Density Functional Theory

The recent advance in double-hybrid density functional theory (DH-DFT) [38, 93, 113–115], has demonstrated its great 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) [116]. This MP2 correlation correction includes a a same-spin (ss) component, E_c^{ss}, as well as an opposite-spin (os) component, E_c^{os}, which are added to the total energy obtained from the KS-DFT calculation. Two scaling parameters, c_{ss} and c_{os}, are introduced in order to avoid double-counting correlation:

$$E_{\text{DH-DFT}} = E_{\text{KS-DFT}} + c_{ss}E_c^{ss} + c_{os}E_c^{os} \tag{4.65}$$

Among DH functionals, ωB97X-2 [39], a long-range corrected DH functional, is described in Section 4.3.4.3.

There are three keywords for turning on DH-DFT as below.

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

SSS_FACTOR

Controls the strength of the same-spin component of PT2 correlation energy.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $c_{ss} = n/1000000$ in Eq. (4.65).

RECOMMENDATION:

NONE

SOS_FACTOR

Controls the strength of the opposite-spin component of PT2 correlation energy.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $c_{os} = n/1000000$ in Eq. (4.65).

RECOMMENDATION:

NONE

For example, B2PLYP [67], which involves 53% Hartree-Fock exchange, 47% Becke 88 GGA exchange, 73% LYP GGA correlation and 27% PT2 orbital correlation, can be called with the following input file. The PT2 correlation energy can also be computed with the efficient resolution-of-identity (RI) methods (see Section 5.5).

Example 4.16 Applications of B2PLYP functional to LiH molecule.

```
$comment
Geometry optimization and frequency calculation on LiH, followed by
single-point calculations with non-RI and RI approaches.
$end

$molecule
0 1
H
```

```
Li H 1.6
$end

$rem
jobtype          opt
exchange         general
correlation      mp2
basis            cc-pvtz
DH               1
SSS_FACTOR       270000        !0.27 = 270000/1000000
SOS_FACTOR       270000        !0.27 = 270000/1000000
$end

$XC_Functional
X HF   0.53
X B    0.47
C LYP  0.73
$end

@@@

$molecule
READ
$end

$rem
jobtype          freq
exchange         general
correlation      mp2
basis            cc-pvtz
DH               1
SSS_FACTOR       270000
SOS_FACTOR       270000
$end

$XC_Functional
X HF   0.53
X B    0.47
C LYP  0.73
$end

@@@

$molecule
READ
$end

$rem
jobtype          sp
exchange         general
correlation      mp2
basis            cc-pvtz
DH               1
SSS_FACTOR       270000
SOS_FACTOR       270000
$end

$XC_Functional
```

```
X HF   0.53
X B    0.47
C LYP  0.73
$end

@@@

$molecule
READ
$end

$rem
jobtype          sp
exchange         general
correlation      rimp2
basis            cc-pvtz
aux_basis        rimp2-cc-pvtz
DH               1
SSS_FACTOR       270000
SOS_FACTOR       270000
$end

$XC_Functional
X HF   0.53
X B    0.47
C LYP  0.73
$end
```

A more detailed gist of one particular class of DH functionals, the XYG3 & XYGJ-OS functionals follows below thanks to Dr Yousung Jung who implemented these functionals in Q-CHEM.

A starting point of these DH functionals is the adiabatic connection formula which provides a rigorous way to define them. One considers an adiabatic path between the fictitious noninteracting Kohn-Sham system (λ= 0) and the real physical system (λ= 1) while holding the electron density fixed at its physical state for all λ:

$$E_{XC}\left[\rho\right] = \int_0^1 U_{XC,\lambda}\left[\rho\right] d\lambda\,, \tag{4.66}$$

where $U_{XC,\lambda}$ is the exchange correlation potential energy at a coupling strength λ. If one assumes a linear model of the latter:

$$U_{XC,\lambda} = a + b\lambda\,, \tag{4.67}$$

one obtains the popular hybrid functional that includes the Hartree-Fock exchange (or occupied orbitals) such as B3LYP. 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 (see Eq. 4.65) that includes MP2 type perturbative terms (PT2) involving virtual Kohn-Sham orbitals:

$$U_{XC,\lambda} = \left.\frac{\partial U_{XC,\lambda}}{\lambda}\right|_{\lambda=0} = 2E_C^{GL2}\,. \tag{4.68}$$

In the DH functional XYG3, as implemented in Q-CHEM, the B3LYP orbitals are used to generate the density and evaluate the PT2 terms. This is different from P2PLYP, an earlier doubly hybrid functional by Grimme. P2PLYP uses truncated Kohn-Sham orbitals while XYG3 uses converged KS orbitals to evaluate the PT2 terms. The performance of XYG3 is not only comparable to that of the G3 or G2 theory for thermochemistry, but barrier heights and non-covalent interactions, including stacking interactions, are also very well described by XYG3 [93].

The recommended basis set for XYG3 is 6-311+G(3df,2p).

Due to the inclusion of PT2 terms, XYG3 or all other forms of doubly hybrid functionals formally scale as the 5th power of system size as in conventional MP2, thereby not applicable to large systems and partly losing DFT's cost advantages. With the success of SOS-MP2 and local SOS-MP2 algorithms developed in Q-CHEM, the natural extension of XYG3 is to include only opposite-spin correlation contributions, ignoring the same-spin parts. The resulting DH functional is XYGJ-OS also implemented in Q-CHEM. It has 4 parameters that are optimized using thermochemistry data. This new functional is both accurate (comparable or even slightly better than XYG3) and faster. If the local algorithm is applied, the formal scaling of XYGJ-OS is cubic, without the locality, it has still 4th order scaling. Recently, XYGJ-OS becomes the only DH functional with analytical gradient [130].

Example 1: XYG3 calculation of N2. XYG3 invokes automatically the B3LYP calculation first, and use the resulting orbitals for evaluating the MP2-type correction terms. One can also use XYG3 combined with RI approximation for the PT2 terms; use EXCHANGE = XYG3RI to do so, along with an appropriate choice of auxiliary basis set.

Example 4.17 XYG3 calculation of N2

```
$molecule
0 1
 N     0.00000000    0.00000000    0.54777500
 N     0.00000000    0.00000000   -0.54777500
$end

$rem
 exchange   xyg3
 basis   6-311+G(3df,2p)
$end
```

Example 2: XYGJ-OS calculation of N2. Since it uses the RI approximation by default, one must define the auxiliary basis.

Example 4.18 XYGJ-OS calculation of N2

```
$molecule
0 1
 N      0.00000000    0.00000000    0.54777500
 N      0.00000000    0.00000000   -0.54777500
$end

$rem
 exchange   xygjos
 basis   6-311+G(3df,2p)
 aux_basis   rimp2-cc-pVtZ
 purecart   1111
 time_mp2   true
$end
```

Example 3: Local XYGJ-OS calculation of N2. The same as XYGJ-OS, except for the use of the attenuated

Coulomb metric to solve the RI coefficients. Omega determines the locality of the metric.

Example 4.19 Local XYGJ-OS calculation of N2

```
$molecule
0 1
 N    0.000    0.000   0.54777500
 N    0.000    0.000  -0.54777500
$end

$rem
 exchange   lxygjos
 omega   200
 basis   6-311+G(3df,2p)
 aux_basis   rimp2-cc-pVtZ
 purecart   1111
$end
```

4.3.10 Asymptotically Corrected Exchange-Correlation Potentials

It is well known that no gradient-corrected exchange functional can simultaneously produce the correct contribution to the exchange energy density and exchange potential in the asymptotic region of molecular systems [131]. Existing GGA exchange-correlation (xc) potentials decay much faster than the correct $-1/r$ xc potential in the asymptotic region [132]. High-lying occupied orbitals and low-lying virtual orbitals are therefore too loosely bounded from these GGA functionals, and the minus HOMO energy becomes much less than the exact ionization potential (as required by the exact DFT) [133, 134]. Moreover, response properties could be poorly predicted from TDDFT calculations with GGA functionals [134]. Long-range corrected hybrid DFT (LRC-DFT), described in Section 4.3.4, has greatly remedied this situation. However, due to the use of long-range HF exchange, LRC-DFT is computationally more expensive than KS-DFT with GGA functionals.

To circumvent this, van Leeuwen and Baerends proposed an asymptotically corrected (AC) exchange potential [131]:

$$v_x^{\mathrm{LB}} = -\beta \frac{x^2}{1 + 3\beta \mathrm{sinh}^{-1}(x)} \tag{4.69}$$

that will reduce to $-1/r$, for an exponentially decaying density, in the asymptotic region of molecular systems, where $x = \frac{|\nabla\rho|}{\rho^{4/3}}$ is the reduced density gradient. The LB94 xc potential is formed by a linear combination of LDA xc potential and the LB exchange potential [131]:

$$v_{xc}^{\mathrm{LB94}} = v_{xc}^{\mathrm{LDA}} + v_x^{\mathrm{LB}} \tag{4.70}$$

The parameter β was determined by fitting the LB94 xc potential to the beryllium atom. As mentioned in Ref. 135, 136, for TDDFT and TDDFT/TDA 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 LB94 xc potential in Q-CHEM thus follows this; using LB94 xc potential for ground-state calculations, followed by TDDFT calculations with an adiabatic LDA xc kernel. This TDLDA/LB94 approach has been widely applied to study excited-state properties of large molecules in literature.

Since the LB exchange potential does not come from the functional derivative of some exchange functional, we use the Levy-Perdew virial relation [137] (implemented in Q-CHEM) to obtain its exchange energy:

$$E_x^{\mathrm{LB}} = -\int v_x^{\mathrm{LB}}([\rho], \mathbf{r})[3\rho(\mathbf{r}) + \mathbf{r}\nabla\rho(\mathbf{r})]d\mathbf{r} \tag{4.71}$$

LB94_BETA
Set the β parameter of LB94 xc potential
TYPE:
INTEGER
DEFAULT:
500
OPTIONS:
n Corresponding to $\beta = n/10000$.
RECOMMENDATION:
Use default, *i.e.*, $\beta = 0.05$

Example 4.20 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
jobtype = sp
exchange = lb94
basis = 6-311(2+,2+)G**
cis_n_roots = 30
rpa = true
$end
```

4.3.11 DFT Numerical Quadrature

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 (*i.e.*, no fitting of the XC potential in an auxiliary basis is done). Q-CHEM provides a standard quadrature grid by default which is sufficient for most purposes.

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 [138], though without the atomic size adjustments". The atomic integrals are then evaluated through standard one-center numerical techniques.

Thus, the exchange-correlation energy E_{XC} is obtained as

$$E_{\mathrm{XC}} = \sum_{A} \sum_{i} w_{Ai} f\left(\mathbf{r}_{Ai}\right) \tag{4.72}$$

where the first summation is over the atoms and the second is over the numerical quadrature grid points for the current atom. The f function is the exchange-correlation functional. The w_{Ai} are the quadrature weights, and the grid points $\mathbf{r}_{Ai}$ are given by

$$\mathbf{r}_{Ai} = \mathbf{R}_A + \mathbf{r}_i \tag{4.73}$$

where $\mathbf{R}_A$ is the position of nucleus A, with the $\mathbf{r}_i$ defining a suitable one-center integration grid, which is independent of the nuclear configuration.

The single-center integrations are further separated into radial and angular integrations. Within Q-CHEM, the radial part is usually treated by the Euler-Maclaurin scheme proposed by Murry *et al.* [139]. This scheme maps the semi-infinite domain $[0, \infty) \rightarrow [0, 1)$ and applies the extended trapezoidal rule to the transformed integrand. Recently Gill and Chien [140] proposed a radial scheme based on a Gaussian quadrature on the interval $[0, 1]$ with weight function $\ln^2 x$. This scheme 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. The authors refer to this scheme as MultiExp.

4.3.12 Angular Grids

Angular quadrature rules may be characterized by their degree, which is the highest degree of spherical harmonics for which the formula is exact, and their efficiency, which is the number of spherical harmonics exactly integrated per degree of freedom in the formula. Q-CHEM supports the following types of angular grids:

Lebedev These are specially constructed grids for quadrature on the surface of a sphere [141–143] based on the octahedral group. Lebedev grids of the following degrees are available:

Degree	3rd	5th	7th	9th	11th	15th	17th	19th	23rd	29th
Points	6	18	26	38	50	86	110	146	194	302

Additional grids with the following number of points are also available: 74, 170, 230, 266, 350, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, 5294. Lebedev grids typically have efficiencies near one, with efficiencies greater than one in some cases.

Gauss-Legendre These are spherical product rules separating the two angular dimensions θ and ϕ. Integration in the θ dimension is carried out with a Gaussian quadrature rule derived from the Legendre polynomials (orthogonal on $[-1, 1]$ with weight function unity), while the ϕ integration is done with equally spaced points.

A Gauss-Legendre grid is selected by specifying the total number of points, $2N^2$, to be used for the integration. This gives a grid with $2N_\phi$ ϕ-points, N_θ θ-points, and a degree of $2N - 1$.

In contrast with Lebedev grids, Gauss-Legendre grids have efficiency of only 2/3 (hence more Gauss-Legendre points are required to attain the same accuracy as Lebedev). However, since Gauss-Legendre grids of general degree are available, this is a convenient mechanism for achieving arbitrary accuracy in the angular integration if desired.

Combining these radial and angular schemes yields an intimidating selection of three-dimensional quadratures. In practice, is it useful to standardize the grids used in order to facilitate the comparison of calculations at different levels of theory.

4.3.13 Standard Quadrature Grids

Both the SG-0 [144] and SG-1 [145] standard quadrature grids were designed to yield the performance of a large, accurate quadrature grid, but with as few points as possible for the sake of computational efficiency. This is accomplished by reducing 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, while retaining large numbers of angular points in the valence region where angular accuracy is critical.

The SG-0 grid was derived in this fashion from a MultiExp-Lebedev-(23,170), (*i.e.*, 23 radial points and 170 angular points per radial point). This grid was pruned whilst ensuring the error in the computed exchange energies for the atoms and a selection of small molecules was not larger than the corresponding error associated with SG-1. In our evaluation, the RMS error associated with the atomization energies for the molecules in the G1 data set is 72 microhartrees. While relative energies are expected to be reproduced well by this scheme, if absolute energies are being sought, a larger grid is recommended.

The SG-0 grid is implemented in Q-CHEM from H to micro Hartrees, excepted He and Na; in this scheme, each atom has around 1400-point, and SG-1 is used for those their SG-0 grids have not been defined. It should be noted that, since the SG-0 grid used for H has been re-optimized in this version of Q-CHEM (version 3.0), quantities calculated in this scheme may not reproduce those generated by the last version (version 2.1).

The SG-1 grid is derived from a Euler-Maclaurin-Lebedev-(50,194) grid (*i.e.*, 50 radial points, and 194 angular points per radial point). This grid has been found to give numerical integration errors of the order of 0.2 kcal/mol for medium-sized molecules, including particularly demanding test cases such as isomerization energies of alkanes. This error is deemed acceptable since it is significantly smaller than the accuracy typically achieved by quantum chemical methods. In SG-1 the total number of points is reduced to approximately 1/4 of that of the original EML-(50,194) grid, with SG-1 generally giving the same total energies as EML-(50,194) to within a few microhartrees (0.01 kcal/mol). Therefore, the SG-1 grid is relatively efficient while still maintaining the numerical accuracy necessary for chemical reliability in the majority of applications.

Both the SG-0 and SG-1 grids were optimized so that the error in the energy when using the grid did not exceed a target threshold. For single point calculations this criterion is appropriate. However, derivatives of the energy can be more sensitive to the quality of the integration grid, and it is recommended that a larger grid be used when calculating these. Special care is required when performing DFT vibrational calculations as imaginary frequencies can be reported if the grid is inadequate. This is more of a problem with low-frequency vibrations. 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 calculation again with a larger grid and check for convergence of the frequencies. Of course the geometry must be re-optimized, but if the existing geometry is used as an initial guess, the geometry optimization should converge in only a few cycles.

4.3.14 Consistency Check and Cutoffs for Numerical Integration

Whenever Q-CHEM calculates numerical density functional integrals, the electron density itself is also integrated numerically as a test on the quality of the quadrature formula used. The deviation of the numerical result from the number of electrons in the system is an indication of the accuracy of the other numerical integrals. If the relative error in the numerical electron count reaches 0.01%, a warning is printed; this is an indication that the numerical XC results may not be reliable. If the warning appears at the first SCF cycle, it is probably not serious, because the initial-guess density matrix is sometimes not idempotent, as is the case with the SAD guess and the density matrix taken from a different geometry in a geometry optimization. If that is the case, the problem will be corrected as the idempotency is restored in later cycles. On the other hand, if the warning is persistent to the end of SCF iterations, then either a finer grid is needed, or choose an alternative method for generating the initial guess.

Users should be aware, however, of the potential flaws that have been discovered in some of the grids currently in use. Jarecki and Davidson [146], for example, have recently shown that correctly integrating the density is a necessary, but not sufficient, test of grid quality.

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, where N is the size of the system. 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 very 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. As mentioned above, when an inaccurate electron count is obtained, it maybe possible to remedy the problem by increasing the size of the quadrature grid.

Finally we note that early implementations of quadrature-based Kohn-Sham DFT employing standard basis sets were plagued by lack of rotational invariance. That is, rotation of the system yielded a significantly energy change. Clearly, such behavior is highly undesirable. Johnson *et al.* rectified the problem of rotational invariance by completing the specification of the grid procedure [147] to ensure that the computed XC energy is the same for any orientation of the molecule in any Cartesian coordinate system.

4.3.15 Basic DFT Job Control

For most DFT jobs two *$rem* variables are required: METHOD and BASIS. To request a specific pair of exchange and correlation functionals, the EXCHANGE and CORRELATION keywords can be used instead of METHOD. In addition, all of the basic input options discussed for Hartree-Fock calculations in Section 4.2.3, and the extended options discussed in Section 4.2.4 are all valid for DFT calculations. Below we list only the basic DFT-specific options (keywords).

METHOD

Specifies the exchange-correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use METHOD = *NAME*, where *NAME* is one of the DFT methods listed in Table 4.3.

RECOMMENDATION:

Consult the literature to guide your selection.

METHOD =	Exchange	Correlation
HF	Hartree–Fock	
LDA	Slater	Vosko-Wilk-Nusair parameterization #5 (VWN)
BLYP	Becke 1988 (B)	Lee-Yang-Parr (LYP)
BP86	B	Perdew 1986 (P86)
BP86-VWN	B	P86-VWN
PBE	Perdew-Burke-Ernzerhof 1996 (PBE)	
PBE0	PBE hybrid with 25% HF exchange	
PBE50	PBE hybrid with 50% HF exchange	
B3LYP	B3LYP hybrid	
BHHLYP	Modified BLYP functional with 50% HF exchange	
B5050LYP	Modified B3LYP functional with 50% HF exchange	
B97-D	Grimme's modified B97 with empirical dispersion	
CAM-B3LYP	Coulomb-attenuated B3LYP	
X3LYP	X3LYP from Xu and Goddard	
EDF1	EDF1	
EDF2	EDF2	
LRC-wPBE	Long-range-corrected pure PBE	
LRC-wPBEH	Long-range-corrected hybrid PBE	
M05	M05 hybrid	
M05-2X	M05-2X hybrid	
M06	M06 hybrid	
M06-L	M06-L hybrid	
M06-HF	M06-HF hybrid	
M06-2X	M06-2X hybrid	
M08-HX	M08-HX hybrid	
M08-SO	M08-SO hybrid	
M11	M11 long-range-corrected hybrid	
M11-L	M11-L hybrid	
wB97	ωB97 long-range-corrected hybrid	
wB97X	ωB97X long-range-corrected hybrid	
wB97X-D	ωB97X-D long-range-corrected hybrid with dispersion corrections	

Table 4.2: Combinations of exchange and correlation density functionals and hybrid functionals accessible through the METHOD keyword.

EXCHANGE

Specifies the exchange functional or exchange-correlation functional for hybrid.

TYPE:

STRING

DEFAULT:

No default exchange functional

OPTIONS:

NAME	Use EXCHANGE = *NAME*, where *NAME* is one of the exchange functionals listed in Table 4.3.

RECOMMENDATION:

Consult the literature to guide your selection.

CORRELATION

Specifies the correlation functional.

TYPE:

STRING

DEFAULT:

None No correlation.

OPTIONS:

None	No correlation
VWN	Vosko-Wilk-Nusair parameterization #5
LYP	Lee-Yang-Parr (LYP)
PW91, PW	GGA91 (Perdew-Wang)
PW92	LSDA 92 (Perdew and Wang) [43]
PK09	LSDA (Proynov-Kong) [44]
LYP(EDF1)	LYP(EDF1) parameterization
Perdew86, P86	Perdew 1986
PZ81, PZ	Perdew-Zunger 1981
PBE	Perdew-Burke-Ernzerhof 1996
TPSS	The correlation component of the TPSS functional
B94	Becke 1994 correlation in fully analytic form
B95	Becke 1995 correlation
B94hyb	Becke 1994 correlation as above, but re-adjusted for use only within the hybrid scheme BR89B94hyb
PK06	Proynov-Kong 2006 correlation (known also as “tLap”
(B88)OP	OP correlation [73], optimized for use with B88 exchange
(PBE)OP	OP correlation [73], optimized for use with PBE exchange
Wigner	Wigner

RECOMMENDATION:

Consult the literature to guide your selection.

EXCHANGE =	Description
HF	Fock exchange
Slater, S	Slater (Dirac 1930)
Becke86, B86	Becke 1986
Becke, B, B88	Becke 1988

muB88	Short-range Becke exchange, as formulated by Song *et al.* [104]
Gill96, Gill	Gill 1996
GG99	Gilbert and Gill, 1999
Becke(EDF1), B(EDF1)	Becke (uses EDF1 parameters)
PW86,	Perdew-Wang 1986
rPW86,	Re-fitted PW86 [48]
PW91, PW	Perdew-Wang 1991
mPW1PW91	modified PW91
mPW1LYP	modified PW91 exchange + LYP correlation
mPW1PBE	modified PW91 exchange + PBE correlation
mPW1B95	modified PW91 exchange + B97 correlation
mPWB1K	mPWB1K
PBE	Perdew-Burke-Ernzerhof 1996
TPSS	The nonempirical exchange-correlation scheme of Tao, Perdew, Staroverov, and Scuseria (requires also that the user specify “TPSS” for correlation)
TPSSH	The hybrid version of TPSS (with no input line for correlation)
PBE0, PBE1PBE	PBE hybrid with 25% HF exchange
PBE50	PBE hybrid with 50% HF exchange
revPBE	revised PBE exchange [57]
PBEOP	PBE exchange + one-parameter progressive correlation
wPBE	Short-range ωPBE exchange, as formulated by Henderson *et al.* [105]
muPBE	Short-range μPBE exchange, as formulated by Song *et al.* [104]
LRC-wPBEPBE	long-range corrected pure PBE
LRC-wPBEhPBE	long-range corrected hybrid PBE
B1B95	Becke hybrid functional with B97 correlation
B97	Becke97 XC hybrid
B97-1	Becke97 re-optimized by Hamprecht *et al.* (1998)
B97-2	Becke97-1 optimized further by Wilson *et al.* (2001)
B97-D	Grimme’s modified B97 with empirical dispersion
B3PW91, Becke3PW91, B3P	B3PW91 hybrid
B3LYP, Becke3LYP	B3LYP hybrid
B3LYP5	B3LYP based on correlation functional #5 of Vosko, Wilk, and Nusair (rather than their functional #3)
CAM-B3LYP	Coulomb-attenuated B3LYP
HC-O3LYP	O3LYP from Handy
X3LYP	X3LYP from Xu and Goddard
B5050LYP	modified B3LYP functional with 50% Hartree-Fock exchange
BHHLYP	modified BLYP functional with 50% Hartree-Fock exchange
B3P86	B3 exchange and Perdew 86 correlation
B3PW91	B3 exchange and GGA91 correlation
B3tLAP	Hybrid Becke exchange and PK06 correlation
HCTH	HCTH hybrid
HCTH-120	HCTH-120 hybrid
HCTH-147	HCTH-147 hybrid
HCTH-407	HCTH-407 hybrid
BOP	B88 exchange + one-parameter progressive correlation
EDF1	EDF1

EDF2	EDF2
VSXC	VSXC meta-GGA, not a hybrid
BMK	BMK hybrid
M05	M05 hybrid
M052X	M05-2X hybrid
M06L	M06-L hybrid
M06HF	M06-HF hybrid
M06	M06 hybrid
M062X	M06-2X hybrid
M08HX	M08-HX hybrid
M08SO	M08-SO hybrid
M11L	M11-L hybrid
M11	M11 long-range corrected hybrid
SOGGA	SOGGA hybrid
SOGGA11	SOGGA11 hybrid
SOGGA11X	SOGGA11-X hybrid
BR89	Becke-Roussel 1989 represented in analytic form
BR89B94h	Hybrid BR89 exchange and B94hyb correlation
omegaB97	ωB97 long-range corrected hybrid
omegaB97X	ωB97X long-range corrected hybrid
omegaB97X-D	ωB97X-D long-range corrected hybrid with dispersion corrections
omegaB97X-2(LP)	ωB97X-2(LP) long-range corrected double-hybrid
omegaB97X-2(TQZ)	ωB97X-2(TQZ) long-range corrected double-hybrid
MCY2	The MCY2 hyper-GGA exchange-correlation (with no input line for correlation)
B05	Full exact-exchange hyper-GGA functional of Becke 05 with RI approximation for the exact-exchange energy density
BM05	Modified B05 hyper-GGA scheme with RI approximation for the exact-exchange energy density used as a variable.
XYG3	XYG3 double-hybrid functional
XYGJOS	XYGJ-OS double-hybrid functional
LXYGJOS	LXYGJ-OS double-hybrid functional with localized MP2
General, Gen	User defined combination of K, X and C (refer to the next section)

Table 4.3: DFT exchange functionals available within Q-CHEM.

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 nonlocal part of vdW-DF-10 (aka vdW-DF2)
VV09	the nonlocal part of VV09
VV10	the nonlocal 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 semilocal exchange functional is used. $C = 0.0089$ is recommended when a long-range corrected (LRC) hybrid functional is used. See further details in Ref. [123].

NL_VV_B

Sets the parameter b in VV10. This parameter controls the short range behavior of the nonlocal 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. See further details in Ref. [123].

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:

1 SG-1 hybrid

OPTIONS:

0 Use SG-0 for H, C, N, and O, SG-1 for all other atoms.

1 Use SG-1 for all atoms.

2 Low Quality.

mn The first six integers correspond to m radial points and the second six integers correspond to n angular points where possible numbers of Lebedev angular points are listed in section 4.3.11.

$-mn$ The first six integers correspond to m radial points and the second six integers correspond to n angular points where the number of Gauss-Legendre angular points $n = 2N^2$.

RECOMMENDATION:

Use 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) grid specified by XC_GRID (which defaults to SG-1, if not otherwise specified) for final cycles of the SCF.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

The use of the smart grid can save some time on initial SCF cycles.

NL_GRID

Specifies the grid to use for non-local correlation.

TYPE:

INTEGER

DEFAULT:

1 SG-1 grid

OPTIONS:

Same as for XC_GRID

RECOMMENDATION:

Use default unless computational cost becomes prohibitive, in which case SG-0 may be used. XC_GRID should generally be finer than NL_GRID.

4.3.16 Example

Example 4.21 Q-CHEM input for a DFT single point energy calculation on water.

```
$comment
   B-LYP/STO-3G water single point calculation
$end

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

   oh  =   1.2
   hoh = 120.0
$end.

$rem
   METHOD         BLYP     Becke88 exchange + LYP correlation
   BASIS          sto-3g   Basis set
$end
```

4.3.17 User-Defined Density Functionals

The format for entering user-defined exchange-correlation density functionals is one line for each component of the functional. Each line requires three variables: the first defines whether the component is an exchange or correlation functional by declaring an X or C, respectively. The second variable is the symbolic representation of the functional as used for the EXCHANGE and CORRELATION *$rem* variables. The final variable is a real number corresponding to the contribution of the component to the functional. Hartree-Fock exchange contributions (required for hybrid density functionals) can be entered using only two variables (K, for HF exchange) followed by a real number.

```
$xc_functional
   X   exchange_symbol   coefficient
   X   exchange_symbol   coefficient
   ...
   C   correlation_symbol   coefficient
   C   correlation_symbol   coefficient
   ...
   K   coefficient
$end
```

Note: (1) Coefficients are real.
(2) A user-defined functional does not require all X, C and K components.

Examples of user-defined XCs: these are XC options that for the time being can only be invoked via the user defined XC input section:

Example 4.22 Q-CHEM input of water with B3tLap.

```
$comment
water with B3tLap
$end

$molecule
  0  1
  O
  H1  O  oh
  H2  O  oh  H1  hoh
  oh  =   0.97
  hoh = 120.0
$end

$rem
   EXCHANGE    gen
   CORRELATION  none
   XC_GRID   000120000194  ! recommended for high accuracy
   BASIS     G3LARGE       ! recommended for high accuracy
   THRESH   14             ! recommended for high accuracy and better convergence
$end

$xc_functional
X    Becke     0.726
X    S         0.0966
C    PK06      1.0
K    0.1713
$end
```

Example 4.23 Q-CHEM input of water with BR89B94hyb.

```
$comment
water with BR89B94hyb
$end

$molecule
   0  1
   O
   H1  O  oh
   H2  O  oh  H1  hoh
   oh  =   0.97
   hoh = 120.0
$end

$rem
   EXCHANGE    gen
   CORRELATION  none
   XC_GRID   000120000194  ! recommended for high accuracy
   BASIS     G3LARGE       ! recommended for high accuracy
   THRESH    14            ! recommended for high accuracy and better convergence
$end

$xc_functional
 X    BR89      0.846
 C    B94hyb    1.0
 K              0.154
$end
```

More specific is the use of the RI-B05 and RI-PSTS functionals. In this release we offer only single-point SCF calculations with these functionals. Both options require a converged LSD or HF solution as initial guess, which greatly facilitates the convergence. It also requires specifying a particular auxiliary basis set:

Example 4.24 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. The input lines
purecar 222
SCF_GUESS   CORE
are obligatory for the time being here.
$end

$molecule
0 1
H   0.  0.   0.0
H   0.  0.   0.7414
$end

$rem
JOBTYPE     SP
SCF_GUESS  CORE
METHOD LDA
BASIS    G3LARGE
purcar 222
THRESH    14
MAX_SCF_CYCLES   80
PRINT_INPUT     TRUE
INCDFT    FALSE
XC_GRID 000128000302
SYM_IGNORE    TRUE
SYMMETRY      FALSE
SCF_CONVERGENCE   9
$end
```

```
@@@@

$comment
For the time being the following input lines are obligatory:
purcar 22222
AUX_BASIS riB05-cc-pvtz
dft_cutoffs 0
1415 1
MAX_SCF_CYCLES   0
JOBTYPE  SP
$end

$molecule
READ
$end

$rem
JOBTYPE    SP
SCF_GUESS  READ
EXCHANGE B05
! EXCHANGE PSTS     ! use this line for RI-PSTS
purcar 22222
BASIS G3LARGE
AUX_BASIS riB05-cc-pvtz  ! The aux basis for RI-B05 and RI-PSTS
THRESH    14
PRINT_INPUT     TRUE
INCDFT    FALSE
XC_GRID 000128000302
SYM_IGNORE    TRUE
SYMMETRY      FALSE
MAX_SCF_CYCLES   0
dft_cutoffs 0
1415 1
$end
```

Besides post-LSD, the RI-B05 option can be used as post-Hartree-Fock method as well, in which case one first does a well-converged HF calculation and uses it as a guess read in the consecutive RI-B05 run.

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 wavefunction space, depending upon which part of that space the initial guess places the system in.

- 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 five options for the initial guess:

- Superposition of Atomic Density (SAD)
- Core Hamiltonian (CORE)
- Generalized Wolfsberg-Helmholtz (GWH)
- Reading previously obtained MOs from disk. (READ)
- Basis set projection (BASIS2)

The first three 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 Simple Initial Guesses

There are three simple initial guesses available in Q-CHEM. While they are all simple, they are by no means equal in quality, as we discuss below.

1. **Superposition of Atomic Densities (SAD):** The SAD guess is almost trivially constructed by summing together atomic densities that have been spherically averaged to yield a trial density matrix. The SAD guess is far superior to the other two options below, particularly when large basis sets and/or large molecules are employed. There are three issues associated with the SAD guess to be aware of:

   (a) 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, and,

   (b) The SAD guess is not available for general (read-in) basis sets. All internal basis sets support the SAD guess.

   (c) The SAD guess is not idempotent and thus requires *at least* two SCF iterations to ensure proper SCF convergence (idempotency of the density).

2. **Generalized Wolfsberg-Helmholtz (GWH):** The GWH guess procedure [148] uses a combination of the overlap matrix elements in Eq. (4.12), and the diagonal elements of the Core Hamiltonian matrix in Eq. (4.18). This initial guess is most satisfactory in small basis sets for small molecules. It is constructed according to the relation given below, where c_x is a constant.

$$H_{\mu v} = c_x S_{\mu v}(H_{\mu\mu} + H_{vv})/(H_{\mu\mu} + H_{vv})22 \tag{4.74}$$

3. **Core Hamiltonian:** The core Hamiltonian guess simply obtains the guess MO coefficients by diagonalizing the core Hamiltonian matrix in Eq. (4.18). This approach works best with small basis sets, and degrades as both the molecule size and the basis set size are increased.

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 density (available only with standard basis sets)
GWH	For ROHF where a set of orbitals are required.
FRAGMO	For a fragment MO calculation

OPTIONS:

CORE	Diagonalize core Hamiltonian
SAD	Superposition of atomic density
GWH	Apply generalized Wolfsberg-Helmholtz approximation
READ	Read previous MOs from disk
FRAGMO	Superimposing converged fragment MOs

RECOMMENDATION:
SAD guess for standard basis sets. For general basis sets, it is best to use the BASIS2 *$rem*. Alternatively, try the GWH or core Hamiltonian guess. 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 default unless SCF convergence issues arise

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 *qchem* 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 qchem 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.

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 alpha orbitals to be occupied on the first line and beta on the second. In the former, only pair of orbitals that needs to be swapped is specified.

Note: To prevent Q-CHEM to change orbital occupation during SCF procedure, MOMSTART option is often used in combination with *$occupied* or *$swap_occupied_virtual* keywords.

Note: The need for orbitals renders these options incompatible with the SAD guess. Most often, they are used with SCF_GUESS=READ.

Example 4.25 Format for modifying occupied guess orbitals.

```
$occupied
   1  2  3  4 ...  nalpha
   1  2  3  4 ...  nbeta
$end
```

Example 4.26 Alternative format for modifying occupied guess orbitals.

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

Example 4.27 Example of swapping guess orbitals.

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

This is identical to:

Example 4.28 Example of specifying occupied guess orbitals.

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

or

Example 4.29 Example of specifying occupied guess orbitals.

```
$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*.

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 7.

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:
FOPPROJECTION (when DUAL_BASIS_ENERGY=false)
OVPROJECTION (when DUAL_BASIS_ENERGY=true)
OPTIONS:

FOPPROJECTION	Construct the Fock matrix in the second basis
OVPROJECTION	Projects MO's from BASIS2 to BASIS.

RECOMMENDATION:
None

Note: BASIS2 sometimes messes up 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.

4.4.6 Examples

Example 4.30 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
```

Example 4.31 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 1 iteration 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
   MAX_SCF_CYCLES  =  1
   THRESH          =  10
$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
```

Example 4.32 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
   UNRESTRICTED     =  true
   METHOD           =  hf
   BASIS            =  6-31g**
   SCF_ALGORITHM    =  diis_gdm
   MAX_DIIS_CYCLES  =  1
   SCF_GUESS        =  gwh
   SCF_GUESS_MIX    =  2
$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 alternative SCF optimization algorithms, which are discussed in the following sections, along with the *$rem* variables that are used to control the calculations. The main options are discussed in sections which follow and are, in brief:

- The highly successful DIIS procedures, which are the default, except for restricted open-shell SCF calculations.
- The new geometric direct minimization (GDM) method, which is highly robust, and the recommended fall-back when DIIS fails. It can also be invoked after a few initial iterations with DIIS to improve the initial guess. GDM is the default algorithm for restricted open-shell SCF calculations.
- The older and less robust direct minimization method (DM). As for GDM, it can also be invoked after a few DIIS iterations (except for RO jobs).
- The maximum overlap method (MOM) which ensures that DIIS always occupies a continuous set of orbitals and does not oscillate between different occupancies.
- The relaxed constraint algorithm (RCA) which guarantees that the energy goes down at every step.

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:
User-defined.
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.
DM	Direct minimizer.
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).
GDM	Geometric Direct Minimization.
RCA	Relaxed constraint algorithm
RCA_DIIS	Use RCA initially, switching to DIIS for later iterations (see THRESH_RCA_SWITCH and MAX_RCA_CYCLES described later in this chapter)
ROOTHAAN	Roothaan repeated diagonalization.

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 is the recommended fallback option. If DIIS approaches the correct solution but fails to finally converge, DIIS_GDM is the recommended fallback.

SCF_CONVERGENCE

SCF is considered converged when the wavefunction error is less that $10^{-\text{SCF\_CONVERGENCE}}$. Adjust the value of THRESH at the same time. Note that in 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.
7	For geometry optimizations and vibrational analysis.
8	For SSG calculations, see Chapter 5.

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 in the input: SCF_PRINT 1 .

4.5.3 Direct Inversion in the Iterative Subspace (DIIS)

The SCF implementation of the Direct Inversion in the Iterative Subspace (DIIS) method [149, 150] uses the property of an SCF solution that requires the density matrix to commute with the Fock matrix:

$$\mathbf{SPF} - \mathbf{FPS} = \mathbf{0} \tag{4.75}$$

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:

$$\mathbf{SP}_i\mathbf{F}_i - \mathbf{F}_i\mathbf{P}_i\mathbf{S} = \mathbf{e}_i \tag{4.76}$$

Here, $\mathbf{P}_i$ is obtained from diagonalization of $\hat{\mathbf{F}}_i$, and

$$\hat{\mathbf{F}}_k = \sum_{j=1}^{k-1} c_j\mathbf{F}_j \tag{4.77}$$

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

$$Z = \left(\sum_k c_k\mathbf{e}_k\right) \cdot \left(\sum_k c_k\mathbf{e}_k\right) \tag{4.78}$$

where the constraint

$$\sum_k c_k = 1 \tag{4.79}$$

is imposed to yield a set of linear equations, of dimension $N + 1$:

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_1 \cdot \mathbf{e}_N & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{e}_N \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_N \cdot \mathbf{e}_N & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_N \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{4.80}$$

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

The rate of convergence may be improved by restricting the number of previous Fock matrices (size of the DIIS subspace, *$rem* variable DIIS_SUBSPACE_SIZE) used for determining the DIIS coefficients:

$$\hat{\mathbf{F}}_k = \sum_{j=k-(L+1)}^{k-1} c_j\mathbf{F}_j \tag{4.81}$$

where L is the size of the DIIS subspace. As the Fock matrix nears self-consistency, the linear matrix equations in Eq. (4.80) 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 wavefunction 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 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 default, the maximum error provides a more reliable criterion.

4.5.4 Geometric Direct Minimization (GDM)

Troy Van Voorhis, working at Berkeley with Martin Head-Gordon, has developed a novel direct minimization method that is extremely robust, and at the same time is only slightly less efficient than DIIS. This method is called geometric direct minimization (GDM) because it takes steps in an orbital rotation space that correspond properly to the hyper-spherical geometry of that space. In other words, rotations are variables that describe a space which 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, we refer the reader to Ref. 151. GDM is a good alternative to DIIS for SCF jobs that exhibit convergence difficulties with DIIS.

Recently, Barry Dunietz, also working at Berkeley with Martin Head-Gordon, has extended the GDM approach to restricted open-shell SCF calculations. Their results indicate that GDM is much more efficient than the older direct minimization method (DM).

In section 4.5.3, we 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

4.5.5 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 hyperspherical space of orbital rotations, and therefore converges less rapidly and less robustly than GDM, in general. It is retained for legacy purposes, and because it is at present the only method available for restricted open shell (RO) SCF calculations in Q-CHEM. 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.

4.5.6 Maximum Overlap Method (MOM)

In general, the DIIS procedure 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) [152], developed by Andrew Gilbert and Peter Gill now at the Australian National University.

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.

4.5.7 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, $E(\mathbf{P})$ [153, 154]. 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 [155]. Here, the current density matrix is written as a linear combination of the previous density matrices:

$$\mathbf{P}(x) = \sum_i x_i \mathbf{P}_i \tag{4.82}$$

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

$$\mathbf{E}(x) = \sum_i E_i x_i + \frac{1}{2} \sum_i x_i (\mathbf{P}_i - \mathbf{P}_j) \cdot (\mathbf{F}_i - \mathbf{F}_j) x_j \tag{4.83}$$

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,

$$\mathbf{F}(x) = \sum_i x_i \mathbf{F}_i + \delta \mathbf{F}_{xc}(x) \tag{4.84}$$

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 INCFOCK; if RCA fails to converge, disabling INCFOCK may improve convergence in some cases.

RCA options are:

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

Please see next section for an example using RCA.

4.5.8 Small-Gap Systems

SCF calculations for systems with zero or small HOMO-LUMO gap (such as metals) can exhibit very slow convergence or may even fail to converge. This problem arises because the energetic ordering of orbitals and states can switch during the SCF optimization leading to discontinuities in the optimization. Using fractional MO occupation numbers can improve the convergence for small-gap systems. In this approach, the occupation numbers of MOs around the Fermi level are allowed to assume non-integer values. This "occupation smearing" allows one to include multiple electron configurations in the same optimization, which improves the stability of the optimization.

We follow the pseudo-Fractional Occupation Number (pFON) method of Rabuck and Scuseria [156] that scales the MO occupation used to construct the AO density:

$$P_{\mu\nu} = \sum_{p=1}^{N} n_p C_{\mu p} C_{\nu p}. \tag{4.85}$$

For a conventional (integer occupation number) SCF run, the occupation number n_p is either one (occupied) or zero (virtual). In pFON, the occupation numbers are following a Fermi-Dirac distribution,

$$n_p = \frac{1}{1 + e^{(\epsilon_p - \epsilon_F)/kT}}, \tag{4.86}$$

where ϵ_p is the respective orbital energy and kT the Boltzmann constant and temperature, respectively. The Fermi energy ϵ_F is set to $(\epsilon_{HOMO} + \epsilon_{LUMO})/2$ in our implementation. To ensure conservation of the total number of electrons, the pFON approach rescales the occupation numbers so that $\sum_p n_p = N_{el}$.

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_{init} = T_{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.

4.5.9 Examples

Example 4.33 Input for a UHF calculation using geometric direct minimization (GDM) on the phenyl radical, after initial iterations with DIIS. This example fails to converge if DIIS is employed directly.

```
$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
   INTSBUFFERSIZE  = 15000000
   SCF_ALGORITHM   = diis_gdm
   SCF_CONVERGENCE = 7
   THRESH          = 10
$end
```

Example 4.34 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.

```
$molecule
  +1 1
   C       0.000000     0.000000     0.990770
   H       0.000000     0.000000     2.081970
   C      -1.233954     0.000000     0.290926
   C      -2.444677     0.000000     1.001437
   H      -2.464545     0.000000     2.089088
   H      -3.400657     0.000000     0.486785
   C      -1.175344     0.000000    -1.151599
   H      -2.151707     0.000000    -1.649364
   C       0.000000     0.000000    -1.928130
   C       1.175344     0.000000    -1.151599
   H       2.151707     0.000000    -1.649364
   C       1.233954     0.000000     0.290926
   C       2.444677     0.000000     1.001437
   H       2.464545     0.000000     2.089088
   H       3.400657     0.000000     0.486785
$end

$rem
   UNRESTRICTED       false
   METHOD             hf
   BASIS              6-31+G*
   SCF_GUESS          core
$end

@@@

$molecule
   read
$end

$rem
   UNRESTRICTED       false
   METHOD             hf
   BASIS              6-31+G*
   SCF_GUESS          read
   MOM_START          1
$end

$occupied
1:26 28
1:26 28
$end

@@@

$molecule
-1 3
   ... <as above> ...
$end
```

```
$rem
   UNRESTRICTED      false
   METHOD            hf
   BASIS             6-31+G*
   SCF_GUESS         read
$end
```

Example 4.35 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
   DIIS_SUBSPACE_SIZE      15
   THRESH                  9
$end
```

Example 4.36 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
  jobtyp sp
  method PBE
  ecp lanl2dz
  BASIS      lanl2dz
  Maxscf 2000
  symmetry 0
  occupations 2 ! pseudo-fractional occupation numbers
  FON_NORB 10
  FON_T_START 300  ! electronic temperature: 300 K
  FON_T_END   300
  FON_E_THRESH 5   ! freeze occupation numbers once DIIS error is 10^-5
$end
```

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 B). 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 4.3.11.

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 (FMM) [157] 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 [158] the original Fast Multipole Method before generalizing it to the CFMM [159]. 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 co-workers further improved the underlying CFMM algorithm [160, 161] then implemented it efficiently [162], 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 [163]. 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 [164] and forces [165] and gives substantial speedups relative to the use of two-electron integrals without any approximation (roughly a factor of 10 (energies) and 30 (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 co-workers [166].

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 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 [167]. 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 [168] to evaluate both exchange energies and their gradients [169] 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 [168, 169]. 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.

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

$$tmp\_thresh = varthresh \times DIIS\_error \tag{4.87}$$

where the $DIIS\_error$ is that calculated from the previous cycle, $varthresh$ is the variable threshold set by the program (by default) and tmp_thresh 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 [170], **F** is computed recursively as

$$\mathbf{F^m} = \mathbf{F^{m-1}} + \Delta\mathbf{J^{m-1}} - \frac{1}{2}\Delta\mathbf{K^{m-1}} \tag{4.88}$$

where m is the SCF cycle, and $\Delta\mathbf{J}^m$ and $\Delta\mathbf{K}^m$ are computed using the difference density

$$\Delta\mathbf{P^m} = \mathbf{P^m} - \mathbf{P^{m-1}} \tag{4.89}$$

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. $tmp\_thresh = 10^{-\mathrm{VARTHRESH}} \times DIIS\_error$

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.

4.6.5 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.

4.6.6 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 [171, 172] 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 [173], 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 [174].

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.

4.6.7 Multiresolution Exchange-Correlation (mrXC) Method

MrXC (multiresolution exchange-correlation) [175–177] is a new method developed by the Q-CHEM development team for the accelerating the computation of exchange-correlation (XC) energy and matrix originated from the XC functional. As explained in 4.6.6, the XC functional is so complicated that the integration of it is usually done on a numerical quadrature. There are two basic types of quadrature. One is the atom-centered grid (ACG), a superposition of atomic quadrature described in 4.6.6. ACG has high density of points near the nucleus to handle the compact core density and low density of points in the valence and nonbonding region where the electron density is smooth. The other type is even-spaced cubic grid (ESCG), which is typically used together with pseudopotentials and planewave basis functions where only the e electron density is assumed smooth. In quantum chemistry, 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 speed-up of several times for the calculations of electron-density and XC matrix is achieved. The smooth part of the calculation with mrXC can also be combined with FTC (see section 4.6.6) to achieve further gain of 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*10.0^{-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*10.0^{-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

4.6.8 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 approximation (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

$$(\mu\nu|\rho) = \sum_{\lambda\sigma}(\mu\nu|\lambda\sigma)\mathbf{P}_{\lambda\sigma} \tag{4.90}$$

is much less complicated than an individual $\lambda\sigma$ function pair. The same principle applies to the FTC method in subsection 4.6.6, 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:

$$\sum_{\lambda\sigma}\int \mathrm{d}^3\mathbf{r}_1 P_{\lambda\sigma}\lambda\sigma(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1-\mathbf{r}|} \approx \sum_K \int \mathrm{d}^3\mathbf{r}_1 P_K K(\mathbf{r}_1)\frac{1}{|\mathbf{r}_1-\mathbf{r}|} \tag{4.91}$$

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 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 [178] and exchange (K) matrix evaluation [179]. 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.

4.6.9 PARI-K fast exchange algorithm

PARI-K 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.* [180]. 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 [181] for constructing the coulomb matrix. The exchange fitting basis sets of Weigend (cc-pVTZ-JK and cc-pVQZ-JK) [181] 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 [180].

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.

4.6.10 Examples

Example 4.37 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 dna.inp
$end

$rem
   METHOD     HF      Hartree-Fock
   BASIS      3-21G   Basis set
   LIN_K      TRUE    Calculate K using LinK
$end
```

Example 4.38 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 dna.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
$end
```

4.7 Dual-Basis Self-Consistent Field Calculations

The dual-basis approximation [182–187] 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:

$$E_{\text{total}} = E_{\text{small basis}} + \text{Tr}[(\Delta\mathbf{P}) \cdot \mathbf{F}]_{\text{large basis}} \tag{4.92}$$

where **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 [188–190] 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.

4.7.1 Dual-Basis MP2

The dual-basis approximation can also be used for the reference energy of a correlated second-order Møller-Plesset (MP2) calculation [183, 187]. 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 5.5), in which the cost of the underlying SCF calculation often dominates.

Furthermore, efficient analytic gradients of the DB-RI-MP2 energy have been developed [185] 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.

4.7.2 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 4.7.2; 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 [186]. 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. 186 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 [187], which can often be problematic for large, augmented (aug-cc-pVTZ, *etc.*.) basis set calculations. The same standard keyword (LINDEPTHRESH) is utilized for linear dependence in the projection code. Because of the scheme utilized 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 7) 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.

4.7.3 Job Control

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).

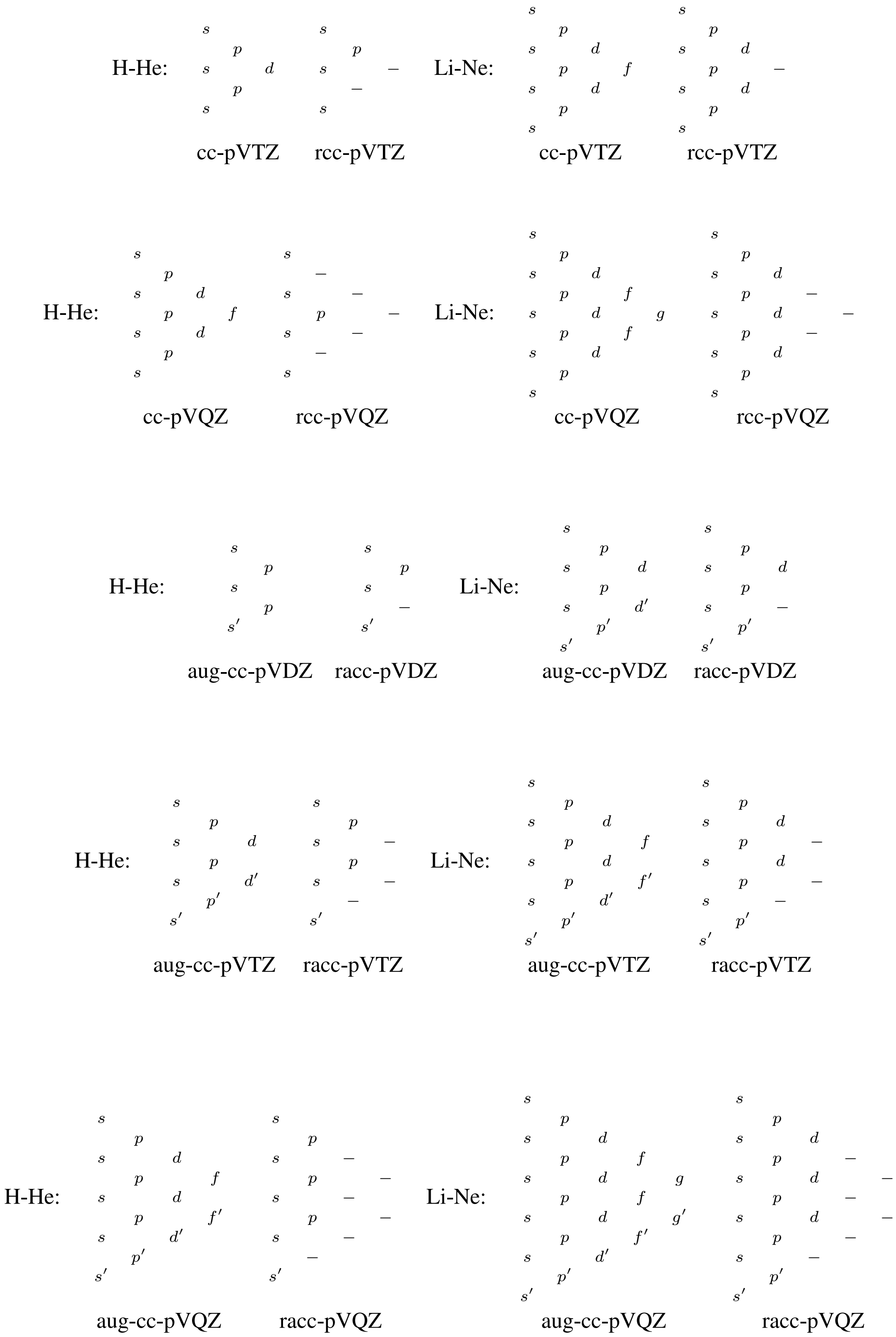


Figure 4.1: Structure of the truncated basis set pairings for cc-pV(T,Q)Z and aug-cc-pV(D,T,Q)Z. The most compact functions are listed at the top. Primed functions depict *aug* (diffuse) functions. Dashes indicate eliminated functions, relative to the paired standard basis set. In each case, the truncations for hydrogen and heavy atoms are shown, along with the nomenclature used in Q-CHEM.

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 4.4: Summary and nomenclature of recommended dual-basis pairings

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.

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.

4.7.4 Examples

Example 4.39 Input for a Dual-Basis B3LYP single-point calculation.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   JOBTYPE             sp
   METHOD              b3lyp
   BASIS               6-311++G(3df,3pd)
   BASIS2              6-311G*
   DUAL_BASIS_ENERGY   true
$end
```

Example 4.40 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             sp
   METHOD              b3lyp
   BASIS               6-31G*
   BASIS2              r64G
   DUAL_BASIS_ENERGY   true
$end
```

Example 4.41 Input for a Dual-Basis RI-MP2 single-point calculation.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   JOBTYPE             sp
   METHOD              rimp2
   AUX_BASIS           rimp2-cc-pVQZ
   BASIS               cc-pVQZ
   BASIS2              rcc-pVQZ
   DUAL_BASIS_ENERGY   true
$end
```

Example 4.42 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 4.43 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
```

4.7.5 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. Dual-basis BOMD has demonstrated [191] 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 [192].

Two algorithmic factors make modest but worthwhile improvements to dual-basis dynamics. First, the iterative, small-basis calculation can benefit from Fock matrix extrapolation [192]. Second, extrapolation of the response equations (the so-called "Z-vector" equations) for nuclear forces further increases efficiency [193] . Both sets of keywords are described in Section 9.9, and the code automatically adjusts to extrapolate in the proper basis set when DUAL_BASIS_ENERGY is activated.

4.8 Hartree-Fock and Density-Functional Perturbative Corrections

4.8.1 Hartree-Fock Perturbative Correction

An HFPC [194, 195] 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. 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

$$F_{\mu\nu} = h_{\mu\nu} + \sum_{\lambda\sigma}^{n} P_{\lambda\sigma}\left[(\mu\nu|\lambda\sigma) - \frac{1}{2}(\mu\lambda|\nu\sigma)\right] \tag{4.93}$$

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

$$F_{ab}^{[1]} = h_{ab} + \sum_{\lambda\sigma}^{n} P_{\lambda\sigma}\left[(ab|\lambda\sigma) - \frac{1}{2}(a\lambda|b\sigma)\right] \tag{4.94}$$

where λ, σ indicate primary basis functions and a, b represent secondary basis functions. Diagonalization of $\mathbf{F}^{[1]}$ yields improved molecular orbitals and an associated density matrix $\mathbf{P}^{[1]}$. The HFPC energy is given by

$$E^{\text{HFPC}} = \sum_{ab}^{N} P_{ab}^{[1]} h_{ab} + \frac{1}{2}\sum_{abcd}^{N} P_{ab}^{[1]} P_{cd}^{[1]}\left[2(ab|cd) - (ac|bd)\right] \tag{4.95}$$

where a, b, c and d represent secondary basis functions. This differs from the DBHF energy evaluation where $\mathbf{P}\mathbf{P}^{[1]}$, rather than $\mathbf{P}^{[1]}\mathbf{P}^{[1]}$, is used. The inclusion of contributions that are quadratic in $\mathbf{P}^{[1]}$ is the key reason for the fact that HFPC is more accurate than DBHF.

Unlike DBHF, HFPC does not require proper subset/superset basis set combinations 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 show 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.

4.8.2 Density Functional Perturbative Correction (Density Functional "Triple Jumping")

Density Functional Perturbation Theory (DFPC) [196] seeks to combine the low cost of pure calculations using small bases and grids with the high accuracy of hybrid calculations using large bases and grids. Our method is motivated by the dual functional method of Nakajima and Hirao [197] and the dual grid scheme of Tozer *et al.* [198] We combine these with dual basis ideas to obtain a triple perturbation in the functional, grid and basis directions.

4.8.3 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.

4.8.4 Examples

Example 4.44 Input for a HFPC single-point calculation.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   JOBTYPE             sp
   EXCHANGE            hf
   BASIS               cc-pVDZ !primary basis
   HFPT_BASIS             cc-pVQZ     !secondary basis
   PURECART    1111    ! set to purecart of the target basis
   HFPT   true
$end
```

Example 4.45 Input for a DFPC single-point calculation.

```
$molecule
   0 1
   H
   H   1   0.75
$end

$rem
   JOBTYPE                      sp
   EXCHANGE                 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
$end
```

4.9 Constrained Density Functional Theory (CDFT)

Under certain circumstances, it is desirable to apply constraints to the electron density during a self-consistent 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 may 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 charge transfer configuration. In these situations, one is interested in controlling the average value of some density observable, $O(\mathbf{r})$, to take on a given value, N:

$$\int \rho(\mathbf{r})O(\mathbf{r})d^3\mathbf{r} = N \tag{4.96}$$

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

$$V[\rho, V] = E[\rho] - V(\int \rho(\mathbf{r})O(\mathbf{r})d^3\mathbf{r} - N) \tag{4.97}$$

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. 199.

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:

$$O(\mathbf{r}) = \sum_{A,\sigma} C_A^\sigma w_A(\mathbf{r}) \tag{4.98}$$

Here the summation runs over the atoms in the system (A) and over the electron spin ($\sigma = \alpha, \beta$). Note that each weight function is designed to be nearly 1 near the nucleus of atom A and rapidly fall to zero near the nucleus of any other atom in the system. The specification of the C_A^σ coefficients is accomplished using

```
$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 (C_α) 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: $\mathrm{X}, \mathrm{Y}, \ldots$. Q-CHEM can handle an arbitrary number of constraints and will minimize the energy subject to all of these constraints simultaneously.

In addition to the *$cdft* input section of the input file, a constrained DFT calculation must also set the CDFT flag to TRUE for the calculation to run. 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: Charge 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 weights populations. As a result, the printed populations will not generally add up to the specified constrained values, even though the constraint is exactly satisfied. You can print Becke populations to confirm that the computed states have the desired charge/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, diradical-like 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. Please see the section on SCF convergence for ideas on how to improve convergence for constrained calculations. Also, CDFT is more sensitive to grid size than ground-state DFT, so sometimes upping the integration grid to (50,194) or (75, 302) improves the convergence.

Note: 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.

Note: 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 [200]. Second derivatives are only available by finite difference of gradients. For details on how to apply constrained DFT to compute magnetic exchange couplings, see Ref. 201. For details on using constrained DFT to compute electron transfer parameters, see Ref. 202.

CDFT options are:

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 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 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:

Use default unless problems occur.

CDFT_CRASHONFAIL

Whether the calculation should crash or not if the constraint iterations do not converge.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE	Crash if constraint iterations do not converge.
FALSE	Do not crash.

RECOMMENDATION:

Use default.

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 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.

Example 4.46 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
```

Example 4.47 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
JOBTYPE          SP
METHOD           B3LYP
```

```
BASIS          6-31G*
SCF_PRINT      TRUE
CDFT           TRUE
$end

$cdft
2
 1 1   3  s
-1 17 19  s
$end
```

4.10 Configuration Interaction with Constrained Density Functional Theory (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 wavefunction methods, is a multireference 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 wavefunctions) 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 Density-Functional Theory only gives converged densities, not actual wavefunctions, computing the off-diagonal coupling elements H_{12} is not completely straightforward, as the physical meaning of the Kohn-Sham wavefunction is not entirely clear. We can, however, perform the following manipulation [203]:

$$\begin{aligned} H_{12} &= \frac{1}{2}\left[\langle 1|H + V_{C1}\omega_{C1} - V_{C1}\omega_{C1}|2\rangle + \langle 1|H + V_{C2}\omega_{C2} - V_{C2}\omega_{C2}|2\rangle\right] \\ &= \frac{1}{2}\left[\left(E_1 + V_{C1}N_{C1} + E_2 + V_{C2}N_{C2}\right)\langle 1|2\rangle - V_{C1}\langle 1|\omega_{C1}|2\rangle - V_{C2}\langle 1|\omega_{C2}|2\rangle\right] \end{aligned}$$

(where the converged states $|i\rangle$ are assumed to be the ground state of $H + V_{Ci}\omega_{Ci}$ with eigenvalue $E_i + V_{Ci}N_{Ci}$). 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 $\langle 1|\omega_{Ci}|2\rangle$.

We 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 [204]. 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; this is done in a very similar fashion to the specification for 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.

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 [204]. (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. For details on using CDFT-CI to calculate reaction barrier heights, see Ref. 205.

CDFT-CI options are:

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 $\vec{S}^{-1/2}$, 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_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.

Many of the CDFT-related rem variables are also applicable to CDFT-CI calculations.

4.10.1 C-DFT/CI Examples

Example 4.48 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 Chro-(Ms=0)....Tyr(Ms=0) and Chro(Ms=1/2)....Tyr-(Ms=1/2).

```
$molecule
-1 1
C        -1.453000       -1.953000       -0.264000
N        -0.278000       -1.402000       -0.440000
N        -1.804000       -2.052000        1.091000
C        -0.687000       -1.548000        1.806000
O        -0.688000       -1.514000        3.031000
C         0.291000       -1.140000        0.799000
C         1.500000       -0.563000        1.254000
H         1.585000       -0.660000        2.346000
```

```
C         2.608000        0.030000        0.605000
C         2.763000        0.182000       -0.865000
H         1.926000       -0.073000       -1.543000
C         3.733000        0.548000        1.313000
H         3.682000        0.571000        2.326000
C         3.821000        0.875000       -1.473000
H         3.844000        1.102000       -2.575000
C         4.938000        1.111000        0.700000
H         5.734000        1.441000        1.308000
C         5.037000        1.228000       -0.739000
O         6.011000        1.818000       -1.261000
C        -3.000000       -2.533000        1.832000
H        -2.859000       -2.250000        2.892000
H        -3.829000       -2.121000        1.354000
C        -2.373000       -2.282000       -1.448000
H        -1.790000       -3.026000       -2.045000
H        -2.626000       -1.300000       -1.865000
H        -3.054000       -3.631000        1.855000
H        -3.308000       -2.854000       -1.357000
C         7.648000       -5.429000        0.303000
H         8.028000       -4.514000        0.845000
H         7.274000       -5.098000       -0.671000
C         6.499001       -5.986000        1.016000
C         6.462999       -6.032001        2.390000
H         7.284000       -5.579000        2.957000
C         5.243000       -6.435000        3.018000
H         5.190001       -6.315001        4.035000
C         4.242001       -7.048000        2.189000
O         3.095000       -7.615000        2.715000
H         2.500999       -7.869000        1.979000
C         5.454000       -6.469000        0.200000
H         5.565001       -6.363000       -0.835000
C         4.294001       -7.003000        0.803000
H         3.469000       -7.324000        0.139000
H         8.511000       -6.108000        0.245000
$end

$rem
symmetry off
sym_ignore = true
method = b3lyp
basis = cc-pvdz
unrestricted = true
scf_convergence = 8
max_scf_cycles = 200
cdftci = true
cdftci_print = 2
cdft_thresh = 7
$end

$cdft
1.0
1.0   1 27
0.0
1.0   1 27 s
--------------
0.0
1.0  1  27
```

```
-1.0
1.0  1  27 s
$end
```

4.11 Unconventional SCF Calculations

4.11.1 CASE Approximation

The Coulomb Attenuated Schrödinger Equation (CASE) [206] approximation follows from the KWIK [207] algorithm in which the Coulomb operator is separated into two pieces using the error function, Eq. (4.45). Whereas in Section 4.3.4 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. (4.45). The parameter ω in Eq. (4.45) 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, wavefunctions, 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 default unless the CASE operator is desired.

4.11.2 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 [208]. 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 [209]. 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 PAO's 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 EPAO's after a conventional SCF.

RECOMMENDATION:

None

4.12 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 [210] 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 $d_{wx}^2 = \langle {}^w\Psi | {}^w\hat{\rho} - {}^x\hat{\rho} | {}^w\Psi \rangle$, 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 $d_{wx}^2 = N - {}^wP^{\mu\nu} S_{\nu\sigma} \cdot {}^xP^{\sigma\tau} S_{\tau\mu}$. and is easily calculated.

d_{wx}^2 is bounded by 0 and the number of electrons, and can be taken as the distance between two solutions. As an example, any singly excited determinant from an SCF determinant (which will not in general be another SCF solution), would be a distance 1 away from it.

In a manner analogous to classical metadynamics, to bias against the set of previously located solutions, x, we create a new Lagrangian,

$$\tilde{E} = E + \sum_x N_x e^{-\lambda_x d_{0x}^2} \tag{4.99}$$

where 0 represents the present density. From this we may derive a new effective Fock matrix,

$$\tilde{F}_{\mu\nu} = F_{\mu\nu} + \sum_x^x P_{\mu\nu} N_x \lambda_x e^{-\lambda_x d_{0x}^2} \tag{4.100}$$

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.

The solutions found through metadynamics also 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 - Configuration Interaction. As they are distinct solutions of the SCF equations, these states are non-orthogonal (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 is a Non-Orthogonal CI. For more information see the NOCI section in Chapter 6

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.

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).

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.

4.13 Ground State Method Summary

To summarize the main features of Q-CHEM's ground state self-consistent field capabilities, the user needs to consider:

- Input a molecular geometry (*$molecule* keyword)
  - Cartesian

  - *Z*-matrix
  - Read from prior calculations

- Declare the job specification (*$rem*keyword)
  - JOBTYPE
    - * Single point
    - * Optimization
    - * Frequency
    - * See Table 4.1 for further options
  - BASIS
    - * Refer to Chapter 7 (note: *$basis* keyword for user defined basis sets)
    - * Effective core potentials, as described in Chapter 8
  - EXCHANGE
    - * Linear scaling algorithms for all methods
    - * Arsenal of exchange density functionals
    - * User definable functionals and hybrids
  - CORRELATION
    - * DFT or wavefunction-based methods
    - * Linear scaling (CPU and memory) incorporation of correlation with DFT
    - * Arsenal of correlation density functionals
    - * User definable functionals and hybrids
    - * See Chapter 5 for wavefunction-based correlation methods.

- Exploit Q-CHEM's special features
  - CFMM, LinK large molecule options
  - SCF rate of convergence increased through improved guesses and alternative minimization algorithms
  - Explore novel methods if desired: CASE approximation, PAOs.

References and Further Reading

[1] Basis Sets (Chapter 7) and Effective Core Potentials (Chapter 8).

[2] Molecular Geometry and Critical Points (Chapter 9).

[3] Molecular Properties Analysis (Chapter 10).

[4] AOINTS (Appendix B).

[5] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory*, Wiley, New York, 1986.

[6] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry*, Dover, 1996.

[7] F. Jensen, *Introduction to Computational Chemistry*, Wiley, New York, 1994.

[8] E. Schrödinger, *Ann. Physik* **79**, 361 (1926).

[9] W. Heisenberg, *Z. Phys.* **39**, 499 (1926).

[10] M. Born and J. R. Oppenheimer, *Ann. Phys.* **84**, 457 (1927).

[11] J. C. Slater, *Phys. Rev.* **34**, 1293 (1929).

[12] J. C. Slater, *Phys. Rev.* **35**, 509 (1930).

[13] J. A. Pople and R. K. Nesbet, *J. Chem. Phys.* **22**, 571 (1954).

[14] R. Seeger and J. A. Pople, *J. Chem. Phys.* **66**, 3045 (1977).

[15] T. Takada, M. Dupuis, and H. F. King, *J. Chem. Phys.* **75**, 332 (1981).

[16] M. Dupuis and H. F. King, *Int. J. Quantum Chem.* **11**, 613 (1977).

[17] M. Dupuis and H. F. King, *J. Chem. Phys.* **68**, 3998 (1978).

[18] R. G. Parr and W. Yang, *Density-Functional Theory of Atoms and Molecules*, Oxford University Press, New York, 1989.

[19] W. Kohn, A. D. Becke, and R. G. Parr, *J. Phys. Chem.* **100**, 12974 (1996).

[20] B. B. Laird, R. B. Ross, and T. Ziegler, editors, volume 629 of *ACS Symposium Series*, American Chemical Society, Washington, D.C., 1996.

[21] T. Ziegler, *Chem. Rev.* **91**, 651 (1991).

[22] P. Hohenberg and W. Kohn, *Phys. Rev. B* **136**, 864 (1964).

[23] W. Kohn and L. J. Sham, *Phys. Rev. A* **140**, 1133 (1965).

[24] P. A. M. Dirac, *P. Camb. Philos. Soc.* **26**, 376 (1930).

[25] J. A. Pople, P. M. W. Gill, and B. G. Johnson, *Chem. Phys. Lett.* **199**, 557 (1992).

[26] T. Van Voorhis and G. E. Scuseria, *J. Chem. Phys.* **109**, 400 (1998).

[27] A. D. Becke and M. R. Roussel, *Phys. Rev. A* **39**, 3761 (1989).

[28] A. D. Becke, *Int. J. Quantum Chem. Symp.* **28**, 625 (1994).

[29] E. Proynov, Z. Gan, and J. Kong, *Chem. Phys. Lett.* **455**, 103 (2008).

[30] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria, *Phys. Rev. Lett.* **91**, 146401 (2003).

[31] F. Liu, E. Proynov, J.-G. Yu, T. R. Furlani, and J. Kong, *J. Chem. Phys.* **137**, 114104 (2012).

[32] A. D. Becke and E. R. Johnson, *J. Chem. Phys.* **122**, 154104 (2005).

[33] P. Mori-Sánchez, A. J. Cohen, and W. Yang, *J. Chem. Phys.* **124**, 091102 (2006).

[34] A. J. Cohen, P. Mori-Sánchez, and W. Yang, *J. Chem. Phys.* **127**, 034101 (2007).

[35] J. P. Perdew, V. N. Staroverov, J. Tao, and G. E. Scuseria, *Phys. Rev. A* **78**, 052513 (2008).

[36] E. Proynov, Y. Shao, and J. Kong, *Chem. Phys. Lett.* **493**, 381 (2010).

[37] E. Proynov, F. Liu, Y. Shao, and J. Kong, *J. Chem. Phys.* **136**, 034102 (2012).

[38] S. Grimme, *J. Chem. Phys.* **124**, 034108 (2006).

[39] J.-D. Chai and M. Head-Gordon, *J. Chem. Phys.* **131**, 174105 (2009).

[40] S. H. Vosko, L. Wilk, and M. Nusair, *Can. J. Phys.* **58**, 1200 (1980).

[41] J. P. Perdew and A. Zunger, *Phys. Rev. B* **23**, 5048 (1981).

[42] E. P. Wigner, *Trans. Faraday Soc.* **34**, 678 (1938).

[43] J. P. Perdew and Y. Wang, *Phys. Rev. B* **45**, 13244 (1992).

[44] E. Proynov and J. Kong, *Phys. Rev. A* **79**, 014103 (2009).

[45] A. D. Becke, *J. Chem. Phys.* **84**, 4524 (1986).

[46] A. D. Becke, *Phys. Rev. A* **38**, 3098 (1988).

[47] J. P. Perdew and Y. Wang, *Phys. Rev. B* **33**, 8800 (1986).

[48] E. D. Murray, K. Lee, and D. C. Langreth, *J. Chem. Theory Comput.* **5**, 2754 (2009).

[49] P. M. W. Gill, *Mol. Phys.* **89**, 443 (1996).

[50] A. T. B. Gilbert and P. M. W. Gill, *Chem. Phys. Lett.* **312**, 511 (1999).

[51] C. Lee, W. Yang, and R. G. Parr, *Phys. Rev. B* **37**, 785 (1988).

[52] J. P. Perdew, *Phys. Rev. B* **33**, 8822 (1986).

[53] J. P. Perdew et al., *Phys. Rev. B* **46**, 6671 (1992).

[54] C. Adamo and V. Barone, *J. Chem. Phys.* **108**, 664 (1998).

[55] J. P. Perdew, K. Burke, and M. Ernzerhof, *Phys. Rev. Lett.* **77**, 3865 (1996).

[56] The PBE exchange and correlation functionals were obtained from the Density Functional Respository, as developed and distributed by the Quantum Chemistry Group, CCLRC Daresbury Laboratory, Cheshire, WA4 4AD United Kingdom.

[57] Y. Zhang and W. Yang, *Phys. Rev. Lett.* **80**, 890 (1998).

[58] C. Adamo, G. E. Scuseria, and V. Barone, *J. Chem. Phys.* **111**, 2889 (1999).

[59] A. D. Becke, *J. Chem. Phys.* **98**, 1372 (1993).

[60] Y. Shao, M. Head-Gordon, and A. I. Krylov, *J. Chem. Phys.* **188**, 4807 (2003).

[61] W.-M. Hoe, A. J. Cohen, and N. H. Handy, *Chem. Phys. Lett.* **341**, 319 (2001).

[62] X. Xu and W. A. Goddard III, *Proc. Natl. Acad. Sci. USA* **101**, 2673 (2004).

[63] T. Yanai, D. Tew, and N. Handy, *Chem. Phys. Lett.* **393**, 51 (2004).

[64] A. D. Becke, *J. Chem. Phys.* **107**, 8554 (1997).

[65] F. A. Hamprecht, A. J. Cohen, D. J. Tozer, and N. C. Handy, *J. Chem. Phys.* **109**, 6264 (1998).

[66] P. J. Wilson, T. J. Bradley, and D. J. Tozer, *J. Chem. Phys.* **115**, 9233 (2001).

[67] S. Grimme, *J. Comput. Chem.* **27**, 1787 (2006).

[68] A. D. Boese, N. L. Doltsinis, N. C. Handy, and M. Sprik, *J. Chem. Phys.* **112**, 1670 (2000).

[69] A. D. Boese and N. C. Handy, *J. Chem. Phys.* **114**, 5497 (2001).

[70] J.-D. Chai and M. Head-Gordon, *J. Chem. Phys.* **128**, 084106 (2008).

[71] R. Baer and D. Neuhauser, *Phys. Rev. Lett.* **94**, 043002 (2005).

[72] E. Livshits and R. Baer, *Phys. Chem. Chem. Phys.* **9**, 2932 (2007).

[73] T. Tsuneda, T. Suzumura, and K. Hirao, *J. Chem. Phys.* **110**, 10664 (1999).

[74] Y. Zhao and D. G. Truhlar, *J. Chem. Phys.* **128**, 184109 (2006).

[75] R. Peverati, Y. Zhao, and D. G. Truhlar, *J. Phys. Chem. Lett.* **2**, 1991 (2011).

[76] R. Peverati and D. G. Truhlar, *J. Chem. Phys.* **135**, 191102 (2011).

[77] M. A. Rohrdanz and J. M. Herbert, *J. Chem. Phys.* **129**, 034107 (2008).

[78] M. A. Rohrdanz, K. M. Martins, and J. M. Herbert, *J. Chem. Phys.* **130**, 054112 (2009).

[79] V. N. Staroverov, G. E. Scuseria, J. Tao, and J. P. Perdew, *J. Chem. Phys.* **119**, 12129 (2003).

[80] A. D. Boese and J. M. L. Martin, *J. Chem. Phys.* **121**, 3405 (2004).

[81] Y. Zhao, N. E. Schultz, and D. G. Truhlar, *J. Chem. Phys.* **123**, 161103 (2005).

[82] Among M05- and M06-series functionals, Zhao and Truhlar recommend M06-2X and M05-2X for main-group thermochemistry and kinetics; M06-L, M06, and M05 for organometallic and inorganic thermochemistry; M06-2X, M05-2X, M06-HF, and M06 for non-covalent interactions, and M06-HF for long-range charge transfer via a TDDFT approach. See Ref. 211 for a review of the Minnesota density functionals.

[83] Y. Zhao, N. E. Schultz, and D. G. Truhlar, *J. Chem. Theory Comput.* **2**, 364 (2006).

[84] Y. Zhao and D. G. Truhlar, *J. Chem. Phys.* **125**, 194101 (2006).

[85] Y. Zhao and D. G. Truhlar, *J. Phys. Chem. A* **110**, 13126 (2006).

[86] Y. Zhao and D. G. Truhlar, *Theor. Chem. Acc.* **120**, 215 (2008).

[87] Y. Zhao and D. G. Truhlar, *J. Chem. Theory Comput.* **4**, 1849 (2007).

[88] R. Peverati and D. G. Truhlar, *J. Phys. Chem. Lett.* **3**, 117 (2012).

[89] R. Peverati and D. G. Truhlar, *J. Phys. Chem. Lett.* **2**, 2810 (2011).

[90] A. D. Becke, *J. Chem. Phys.* **104**, 1040 (1996).

[91] E. Proynov and J. Kong, *J. Chem. Theory Comput.* **3**, 746 (2007).

[92] E. Proynov, F. Liu, and J. Kong, *Chem. Phys. Lett.* **525**, 150 (2012).

[93] Y. Zhang, X. Xu, and W. A. Goddard III, *Proc. Natl. Acad. Sci. USA* **106**, 4963 (2009).

[94] R. D. Adamson, P. M. W. Gill, and J. A. Pople, *Chem. Phys. Lett.* **284**, 6 (1998).

[95] C. Y. Lin, M. W. George, and P. M. W. Gill, *Aust. J. Chem.* **57**, 365 (2004).

[96] P. J. Stephens, F. J. Devlin, C. F. Chabolowski, and M. J. Frisch, *J. Phys. Chem.* **98**, 11623 (1994).

[97] E. Proynov and J. Kong, in *Theoretical Aspects of Catalysis*, edited by G. Vaysilov and T. Mineva, Heron Press, Birmingham, UK, 2008.

[98] A. Dreuw, J. L. Weisman, and M. Head-Gordon, *J. Chem. Phys.* **119**, 2943 (2003).

[99] A. Lange and J. M. Herbert, *J. Chem. Theory Comput.* **3**, 1680 (2007).

[100] A. W. Lange and J. M. Herbert, *J. Am. Chem. Soc.* **131**, 124115 (2009).

[101] A. W. Lange, M. A. Rohrdanz, and J. M. Herbert, *J. Phys. Chem. B* **112**, 6304 (2008).

[102] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill, *J. Comput. Chem.* **20**, 921 (1999).

[103] H. Iikura, T. Tsuneda, T. Yanai, and K. Hirao, *J. Chem. Phys.* **115**, 3540 (2001).

[104] J. W. Song, T. Hirosawa, T. Tsuneda, and K. Hirao, *J. Chem. Phys.* **126**, 154105 (2007).

[105] T. M. Henderson, B. G. Janesko, and G. E. Scuseria, *J. Chem. Phys.* **128**, 194105 (2008).

[106] J.-D. Chai and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **10**, 6615 (2008).

[107] J.-D. Chai and M. Head-Gordon, *Chem. Phys. Lett.* **467**, 176 (2008).

[108] J. Heyd, G. E. Scuseria, and M. Ernzerhof, *J. Chem. Phys.* **118**, 8207 (2003).

[109] T. Leininger, H. Stoll, H.-J. Werner, and A. Savin, *Chem. Phys. Lett.* **275**, 151 (1997).

[110] R. Baer, E. Livshits, and U. Salzner, *Annu. Rev. Phys. Chem.* **61**, 85 (2010).

[111] T. Stein, L. Kronik, and R. Baer, *J. Am. Chem. Soc.* **131**, 2818 (2009).

[112] U. Salzner and R. Baer, *J. Chem. Phys.* **131**, 231101 (2009).

[113] T. Schwabe and S. Grimme, *Phys. Chem. Chem. Phys.* **9**, 3397 (2007).

[114] A. Tarnopolsky, A. Karton, R. Sertchook, D. Vuzman, and J. M. L. Martin, *J. Phys. Chem. A* **112**, 3 (2008).

[115] T. Benighaus, R. A. DiStasio, Jr., R. C. Lochan, J.-D. Chai, and M. Head-Gordon, *J. Phys. Chem. A* **112**, 2702 (2008).

[116] C. Møller and M. S. Plesset, *Phys. Rev.* **46**, 618 (1934).

[117] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist, *Phys. Rev. Lett.* **92**, 24601 (2005).

[118] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist, *Phys. Rev. Lett.* **95**, 109902 (2005).

[119] O. A. Vydrov, Q. Wu, and T. Van Voorhis, *J. Chem. Phys.* **129**, 014106 (2008).

[120] K. Lee, É. D. Murray, L. Kong, B. I. Lundqvist, and D. C. Langreth, *Phys. Rev. B* **82**, 081101 (2010).

[121] O. A. Vydrov and T. Van Voorhis, *Phys. Rev. Lett.* **103**, 063004 (2009).

[122] O. A. Vydrov and T. Van Voorhis, *J. Chem. Phys.* **132**, 164113 (2010).

[123] O. A. Vydrov and T. Van Voorhis, *J. Chem. Phys.* **133**, 244103 (2010).

[124] E. R. Johnson and A. D. Becke, *J. Chem. Phys.* **123**, 024101 (2005).

[125] J. Kong, Z. Gan, E. Proynov, M. Freindorf, and T. Furlani, *Phys. Rev. A* **79**, 042510 (2009).

[126] E. R. johnson and A. D. Becke, *J. Chem. Phys.* **124**, 174104 (2006).

[127] A. D. Becke and F. O. Kannemann, *Can. J. Chem.* **88**, 1057 (2010).

[128] F. O. Kannemann and A. D. Becke, *J. Chem. Theory Comput.* **6**, 1081 (2010).

[129] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, *J. Chem. Phys.* **132**, 154104 (2010).

[130] H. Ji, Y. Shao, W. A. Goddard, and Y. Jung, *J. Chem. Theory Comput.* **9**, 1971 (2013).

[131] R. van Leeuwen and E. J. Baerends, *Phys. Rev. A* **49**, 2421 (1994).

[132] M. E. Casida and D. R. Salahub, *J. Chem. Phys.* **113**, 8918 (2000).

[133] Q. Wu, P. W. Ayers, and W. Yang, *J. Chem. Phys.* **119**, 2978 (2003).

[134] S. J. A. van Gisbergen et al., *J. Chem. Phys.* **105**, 3142 (1996).

[135] M. E. Casida, C. Jamorski, K. C. Casida, and D. R. Salahub, *J. Chem. Phys.* **108**, 4439 (1998).

[136] S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.* **314**, 291 (1999).

[137] M. Levy and J. P. Perdew, *Phys. Rev. A* **32**, 2010 (1985).

[138] A. D. Becke, *J. Chem. Phys.* **88**, 2547 (1988).

[139] C. W. Murray, N. C. Handy, and G. J. Laming, *Mol. Phys.* **78**, 997 (1993).

[140] S.-H. Chien and P. M. W. Gill, *J. Comput. Chem.* **24**, 732 (2003).

[141] V. I. Lebedev, *Sibirsk. Mat. Zh.* **18**, 132 (1977).

[142] V. I. Lebedev, *Zh. Vychisl. Mat. Mat. Fix.* **15**, 48 (1975).

[143] V. I. Lebedev, *Zh. Vychisl. Mat. Mat. Fix.* **16**, 293 (1976).

[144] S.-H. Chien and P. M. W. Gill, *J. Comput. Chem.* **27**, 730 (2006).

[145] P. M. W. Gill, B. G. Johnson, and J. A. Pople, *Chem. Phys. Lett.* **209**, 506 (1993).

[146] A. A. Jarecki and E. R. Davidson, *Chem. Phys. Lett.* **300**, 44 (1999).

[147] B. G. Johnson, P. M. W. Gill, and J. A. Pople, *Chem. Phys. Lett.* **220**, 377 (1994).

[148] M. Wolfsberg and L. Helmholtz, *J. Chem. Phys.* **20**, 837 (1952).

[149] P. Pulay, *Chem. Phys. Lett.* **73**, 393 (1980).

[150] P. Pulay, *J. Comput. Chem.* **3**, 556 (1982).

[151] T. Van Voorhis and M. Head-Gordon, *Mol. Phys.* **100**, 1713 (2002).

[152] A. T. B. Gilbert, N. A. Besley, and P. M. W. Gill, *J. Phys. Chem. A* **112**, 13164 (2008).

[153] E. Cancès and C. Le Bris, *Int. J. Quantum Chem.* **79**, 82 (2000).

[154] E. Cancès, *J. Chem. Phys.* **114**, 10616 (2001).

[155] K. N. Kudin, G. E. Scuseria, and E. Cancès, *J. Chem. Phys.* **116**, 8255 (2002).

[156] A. D. Rabuck and G. E. Scuseria, *J. Chem. Phys.* **110**, 695 (1999).

[157] L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems*, MIT Press, London, 1987.

[158] C. A. White and M. Head-Gordon, *J. Chem. Phys.* **101**, 6593 (1994).

[159] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *Chem. Phys. Lett.* **230**, 8 (1994).

[160] C. A. White and M. Head-Gordon, *J. Chem. Phys.* **105**, 5061 (1996).

[161] C. A. White and M. Head-Gordon, *Chem. Phys. Lett.* **257**, 647 (1996).

[162] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *Chem. Phys. Lett.* **253**, 268 (1996).

[163] C. A. White and M. Head-Gordon, *J. Chem. Phys.* **104**, 2620 (1996).

[164] Y. Shao and M. Head-Gordon, *Chem. Phys. Lett.* **323**, 425 (2000).

[165] Y. Shao and M. Head-Gordon, *J. Chem. Phys.* **114**, 6572 (2001).

[166] T. R. Adams, R. D. Adamson, and P. M. W. Gill, *J. Chem. Phys.* **107**, 124 (1997).

[167] E. Schwegler, M. Challacombe, and M. Head-Gordon, *J. Chem. Phys.* **106**, 9708 (1997).

[168] C. Ochsenfeld, C. A. White, and M. Head-Gordon, *J. Chem. Phys.* **109**, 1663 (1998).

[169] C. Ochsenfeld, *Chem. Phys. Lett.* **327**, 216 (2000).

[170] E. Schwegler and M. Challacombe, *J. Chem. Phys.* **106**, 9708 (1996).

[171] L. Fusti-Molnar and P. Pulay, *J. Chem. Phys.* **116**, 7795 (2002).

[172] L. Fusti-Molnar and P. Pulay, *J. Chem. Phys.* **117**, 7827 (2002).

[173] L. Fusti-Molnar, *J. Chem. Phys.* **119**, 11080 (2003).

[174] L. Fusti-Molnar and J. Kong, *J. Chem. Phys.* **122**, 074108 (2005).

[175] J. Kong, S. T. Brown, and L. Fusti-Molnar, *J. Chem. Phys.* **124**, 094109 (2006).

[176] N. J. Russ, C.-M. Chang, and J. Kong, *Can. J. Chem.* **89**, 657 (2011).

[177] C.-M. Chang, N. J. Russ, and J. Kong, *Phys. Rev. A* **84**, 022504 (2011).

[178] A. Sodt and M. Head-Gordon, *J. Chem. Phys.* **125**, 074116 (2006).

[179] A. Sodt and M. Head-Gordon, *J. Chem. Phys.* **128**, 104106 (2008).

[180] P. Merlot et al., *J. Comput. Chem.* **34**, 1486 (2013).

[181] F. Weigend, *Phys. Chem. Chem. Phys.* **4**, 4285 (2002).

[182] W. Z. Liang and M. Head-Gordon, *J. Phys. Chem. A* **108**, 3206 (2004).

[183] R. P. Steele, R. A. DiStasio, Jr., Y. Shao, J. Kong, and M. Head-Gordon, *J. Chem. Phys.* **125**, 074108 (2006).

[184] R. P. Steele, Y. Shao, R. A. DiStasio, Jr., and M. Head-Gordon, *J. Phys. Chem. A* **110**, 13915 (2006).

[185] R. A. DiStasio, Jr., R. P. Steele, and M. Head-Gordon, *Mol. Phys.* **105**, 27331 (2007).

[186] R. P. Steele and M. Head-Gordon, *Mol. Phys.* **105**, 2455 (2007).

[187] R. P. Steele, R. A. DiStasio, Jr., and M. Head-Gordon, *J. Chem. Theory Comput.* **5**, 1560 (2009).

[188] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople, *J. Chem. Phys.* **94**, 7221 (1991).

[189] L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov, and J. A. Pople, *J. Chem. Phys.* **109**, 7764 (1998).

[190] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople, *J. Chem. Phys.* **112**, 7374 (2000).

[191] R. P. Steele, M. Head-Gordon, and J. C. Tully, *J. Phys. Chem. A* **114**, 11853 (2010).

[192] J. M. Herbert and M. Head-Gordon, *J. Chem. Phys.* **121**, 11542 (2004).

[193] R. P. Steele and J. C. Tully, *Chem. Phys. Lett.* **500**, 167 (2010).

[194] J. Deng, A. T. B. Gilbert, and P. M. W. Gill, *J. Chem. Phys.* **130**, 231101 (2009).

[195] J. Deng, A. T. B. Gilbert, and P. M. W. Gill, *J. Chem. Phys.* **133**, 044116 (2009).

[196] J. Deng, A. T. B. Gilbert, and P. M. W. Gill, *Phys. Chem. Chem. Phys.* **12**, 10759 (2010).

[197] T. Nakajima and K. Hirao, *J. Chem. Phys.* **124**, 184108 (2006).

[198] D. J. Tozer, M. E. Mura, R. D. Amos, and N. C. Handy, in *Computational Chemistry*, AIP Conference Proceedings, page 3, 1994.

[199] Q. Wu and T. Van Voorhis, *Phys. Rev. A* **72**, 024502 (2005).

[200] Q. Wu and T. Van Voorhis, *J. Phys. Chem. A* **110**, 9212 (2006).

[201] Q. Wu and T. Van Voorhis, *J. Chem. Theory Comput.* **2**, 765 (2006).

[202] Q. Wu and T. Van Voorhis, *J. Chem. Phys.* **125**, 164105 (2006).

[203] Q. Wu and T. Van Voorhis, *J. Chem. Phys.* **125**, 164105 (2006).

[204] Q. Wu, C. L. Cheng, and T. Van Voorhis, *J. Chem. Phys.* **127**, 164119 (2007).

[205] Q. Wu, B. Kaduk, and T. Van Voorhis, *J. Chem. Phys.* **130**, 034109 (2009).

[206] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill, *Chem. Phys. Lett.* **254**, 329 (1996).

[207] J. P. Dombroski, S. W. Taylor, and P. M. W. Gill, *J. Phys. Chem.* **100**, 6272 (1996).

[208] M. S. Lee and M. Head-Gordon, *J. Chem. Phys.* **107**, 9085 (1997).

[209] M. S. Lee and M. Head-Gordon, *Comp. Chem.* **24**, 295 (2000).

[210] A. J. W. Thom and M. Head-Gordon, *Phys. Rev. Lett.* **101**, 193001 (2008).

[211] Y. Zhao and D. G. Truhlar, *Chem. Phys. Lett.* **502**, 1 (2011).

Chapter 5

Wavefunction-Based Correlation Methods

5.1 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 wavefunction-based methods available in Q-CHEM to describe electron correlation.

Wavefunction-based electron correlation methods concentrate on the design of corrections to the wavefunction 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. Wavefunction-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 [6] and G3 methods [7].

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 5.9. 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 wavefunction-based electron correlation calculation using Q-CHEM, three *$rem* variables need to be set:

- BASIS to specify the basis set (see Chapter 7)
- METHOD for treating correlation
- N_FROZEN_CORE frozen core electrons (0 default, optionally FC, or n)

For wavefunction-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 5.11).

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 6.7.9). 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 wavefunction-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 wavefunction-based.

TYPE:

STRING

DEFAULT:

HF No correlation, Hartree-Fock exchange

OPTIONS:

MP2	Sections 5.2 and 5.3
RI-MP2	Section 5.5
Local_MP2	Section 5.4
RILMP2	Section 5.5.1
ATTMP2	Section 5.6.1
ATTRIMP2	Section 5.6.1
ZAPT2	A more efficient restricted open-shell MP2 method [8].
MP3	Section 5.2
MP4SDQ	Section 5.2
MP4	Section 5.2
CCD	Section 5.7
CCD(2)	Section 5.8
CCSD	Section 5.7
CCSD(T)	Section 5.8
CCSD(2)	Section 5.8
CCSD(fT)	Section 5.8.3
CCSD(dT)	Section 5.8.3
QCISD	Section 5.7
QCISD(T)	Section 5.8
OD	Section 5.7
OD(T)	Section 5.8
OD(2)	Section 5.8
VOD	Section 5.9
VOD(2)	Section 5.9
QCCD	Section 5.7
QCCD(T)	
QCCD(2)	
VQCCD	Section 5.9

RECOMMENDATION:

Consult the literature for guidance.

5.2 Møller-Plesset Perturbation Theory

5.2.1 Introduction

Møller-Plesset Perturbation Theory [9] 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 5.3 and 5.4 respectively. The MP3 method is still occasionally used, while MP4 calculations are quite commonly employed as part of the G2 and G3 thermochemical methods [6, 7]. In the remainder of this section, the theoretical basis of Møller-Plesset theory is reviewed.

5.2.2 Theoretical Background

The Hartree-Fock wavefunction 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 wavefunction and energy are, however, exact solutions for the Hartree-Fock Hamiltonian H_0 eigenvalue problem. If we assume that the Hartree-Fock wavefunction 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

$$H = H_0 + \lambda V \tag{5.1}$$

where V is the small perturbation and λ is a dimensionless parameter. Expanding the exact wavefunction and energy in terms of the HF wavefunction and energy yields

$$E = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \lambda^3 E^{(3)} + \ldots \tag{5.2}$$

and

$$\Psi = \Psi_0 + \lambda \Psi^{(1)} + \lambda^2 \Psi^{(2)} + \lambda^3 \Psi^{(3)} + \ldots \tag{5.3}$$

Substituting these expansions into the Schrödinger equation and collecting terms according to powers of λ yields

$$H_0 \Psi_0 = E^{(0)} \Psi_0 \tag{5.4}$$

$$H_0 \Psi^{(1)} + V \Psi_0 = E^{(0)} \Psi^{(1)} + E^{(1)} \Psi_0 \tag{5.5}$$

$$H_0 \Psi^{(2)} + V \Psi^{(1)} = E^{(0)} \Psi^{(2)} + E^{(1)} \Psi^{(1)} + E^{(2)} \Psi_0 \tag{5.6}$$

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:

$$E^{(0)} = \langle \Psi_0 | H_0 | \Psi_0 \rangle \tag{5.7}$$

$$E^{(1)} = \langle \Psi_0 | V | \Psi_0 \rangle \tag{5.8}$$

$$E^{(2)} = \left\langle \Psi_0 | V | \Psi^{(1)} \right\rangle \tag{5.9}$$

Thus, the Hartree-Fock energy

$$E_0 = \langle \Psi_0 | \, H_0 + V \, | \Psi_0 \rangle \tag{5.10}$$

is simply the sum of the zeroth- and first- order energies

$$E_0 = E^{(0)} + E^{(1)} \tag{5.11}$$

The correlation energy can then be written

$$E_{\text{corr}} = E_0^{(2)} + E_0^{(3)} + E_0^{(4)} + \ldots \tag{5.12}$$

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

$$E_0^{(2)} = -\frac{1}{4}\sum_{ab}^{\text{virt}}\sum_{ij}^{\text{occ}} \frac{|\langle ab| \,|ij\rangle|^2}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} \tag{5.13}$$

where

$$\langle ab \,||ij\,\rangle = \langle ab|abijij\rangle - \langle ab|abjiji\rangle \tag{5.14}$$

and

$$\langle ab|abcdcd\rangle = \int \psi_a(\mathbf{r}_1)\psi_c(\mathbf{r}_1)\left[\frac{1}{r_{12}}\right]\psi_b(\mathbf{r}_2)\psi_d(\mathbf{r}_2)d\mathbf{r}_1 d\mathbf{r}_2 \tag{5.15}$$

which can be written in terms of the two-electron repulsion integrals

$$\langle ab|abcdcd\rangle = \sum_{\mu}\sum_{\nu}\sum_{\lambda}\sum_{\sigma} C_{\mu a}C_{\nu c}C_{\lambda b}C_{\sigma d}\,(\mu\nu|\lambda\sigma) \tag{5.16}$$

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 5.7 while the computational cost of MP4 is similar to the “(T)” corrections discussed in Section 5.8.

5.3 Exact MP2 Methods

5.3.1 Algorithm

Second order Møller-Plesset theory (MP2) [9] probably the simplest useful wavefunction-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 [10]. Indeed, in a recent comparative study of small closed-shell molecules [11], 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 [12]. 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 [13, 14], 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 sortless 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 sortless 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. The AO2MO_DISK option is also ignored in this algorithm, which requires up to O^2VN megabytes of disk space.

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 *$rem* variable must be sufficient to permit efficient integral evaluation (10-80Mb) and to hold a large temporary array whose size is OVN, the product of the number of occupied, virtual and total numbers of orbitals.
- AO2MO_DISK: The value specified for this *$rem* variable should be as large as possible (*i.e.*, perhaps 80% of the free space on your *$QCSCRATCH* partition where temporary job files are held). The value of this variable will determine how many times the two-electron integrals in the atomic orbital basis must be re-evaluated, which is a major computational step in MP2 calculations.
- 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.

5.3.2 The Definition of Core Electron

The number of core electrons in an atom is relatively well defined, and consists of certain atomic shells, (note that ECPs are available in 'small-core' and 'large-core' varieties, see Chapter 8 for further details). 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 [15].

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 5.3.2.

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 5.1: A summary of the effects of different core definitions

5.3.3 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:

64 corresponding to 64 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 (2 Gb)

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Use default, or set to the physical memory of your machine. Note that if more than 1GB is specified for a CCMAN job, the memory is allocated as follows

12%	MEM_STATIC
50%	CC_MEMORY
35%	Other memory requirements:

AO2MO_DISK

Sets the amount of disk space (in megabytes) available for MP2 calculations.

TYPE:

INTEGER

DEFAULT:

2000 Corresponding to 2000 Mb.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Should be set as large as possible, discussed in Section 5.3.1.

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 5.3.1).

RECOMMENDATION:

Semi-direct is usually most efficient, and will normally be chosen by default.

N_FROZEN_CORE

Sets the number of frozen core orbitals in a post-Hartree–Fock calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

FC	Frozen Core approximation (all core orbitals frozen).
n	Freeze n core orbitals.

RECOMMENDATION:

While the default is not to freeze orbitals, MP2 calculations are more efficient with frozen core orbitals. Use FC 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

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 5.3.2 for details).

RECOMMENDATION:

Use 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 default, unless you are uncertain about what the core character is.

5.3.4 Example

Example 5.1 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
```

5.4 Local MP2 Methods

5.4.1 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 reformulations 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 [16, 17] 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) [18]. 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. (5.13). First, for the DIM model:

$$E_{\mathrm{DIM\,MP2}} = -\frac{1}{2}\sum_{\bar{P},\bar{Q}} \frac{(\bar{P}|\bar{Q})(\bar{P}||\bar{Q})}{\Delta_{\bar{P}} + \Delta_{\bar{Q}}} \tag{5.17}$$

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:

$$E_{\mathrm{TRIM\,MP2}} = -\sum_{\bar{P},jb} \frac{(\bar{P}|jb)(\bar{P}||jb)}{\Delta_{\bar{P}} + \varepsilon_b - \varepsilon_j} - E_{\mathrm{DIM\,MP2}} \tag{5.18}$$

where the sum is now mixed between atomic substitutions $\bar{P}$, and nonlocal occupied j to virtual b substitutions. See Refs. 16, 17 for a full derivation and discussion.

The accuracy of the local TRIM and DIM models has been tested in a series of calculations [16, 17]. 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 [19]. 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. 16 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 [20], 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.

5.4.2 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.

- *Uniterated 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 nonorthogonal minimal basis is symmetrically orthogonalized, and then modified as discussed in Ref. 18 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 [16], 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. 18. 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 uniterated EPAOs based on atomic blocks of SPS.

OPTIONS:

n Optimize the EPAOs for up to n iterations.

RECOMMENDATION:

Use 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 default, unless convergence failure is encountered.

5.4.3 Algorithm Control and Customization

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 [21]. 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 AO2MO_DISK keyword is ignored for LOCAL_MP2 calculations.

The evaluation of the local MP2 energy does not require any further customization. An adequate amount of MEM_STATIC (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.

5.4.4 Examples

Example 5.2 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
   CORRELATION   local_mp2
   BASIS         6-311g**
$end
```

5.5 Auxiliary Basis Set (Resolution-of-Identity) MP2 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 [22–24]. 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:

$$|\mu\nu\rangle \approx |\widetilde{\mu\nu}\rangle = \sum_K |K\rangle C_{\mu\nu}^K \tag{5.19}$$

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, **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:

$$\sum_L \langle K\,|L\rangle C_{\mu\nu}^L = \langle K\,|\mu\nu\rangle \tag{5.20}$$

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:

$$\langle \mu\nu|\lambda\sigma\rangle \approx \langle \widetilde{\mu\nu}|\widetilde{\lambda\sigma}\rangle = \sum K, LC^{L}_{\mu}\langle L|K\rangle C^{K}_{\lambda\sigma} \tag{5.21}$$

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 [25, 26] of the correlation energy. With these basis sets, small absolute errors (*e.g.*, below 60 μHartree per atom in MP2) and even smaller relative errors in computed energies are found, while the speed-up can be 3–30 fold. 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 5.14 (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 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.

5.5.1 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 [22] and gradients [27, 28] 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).

5.5.2 Example

Example 5.3 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
   SYMMETRY      false
$end
```

For the size of required memory, the followings need to be considered.

MEM_STATIC

Sets the memory for AO-integral evaluations and their transformations.

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 Mb.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

For RI-MP2 calculations, $150(ON + V)$ 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 default, or set to the physical memory of your machine. The minimum requirement is $3X^2$.

5.5.3 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 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.

Example 5.4 Example of OpenMP-parallel RI-MP2 job.

```
$molecule
0 1
 C1
 H1  C1    1.0772600000
 H2  C1    1.0772600000  H1  131.6082400000
$end

$rem
 jobtype              SP
 exchange             HF
 correlation          pRIMP2
 basis                cc-pVTZ
 aux_basis            rimp2-cc-pVTZ
 purecart             11111
 symmetry             false
 thresh               12
 scf_convergence      8
 max_sub_file_num     128
 !time_mp2            true
$end
```

5.5.4 GPU Implementation of RI-MP2

5.5.4.1 Requirements

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
- 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
- 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.

5.5.4.2 Options

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:

INTEGER

DEFAULT:

0

OPTIONS:

0	MGEMM disabled
1	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-defined 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.

5.5.4.3 Input examples

Example 5.5 RI-MP2 double-precision calculation

```
$comment
RI-MP2 double-precision example
$end
$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
jobtype sp
exchange hf
method rimp2
basis cc-pvdz
aux_basis rimp2-cc-pvdz
cuda_rimp2 1
$end
```

Example 5.6 RI-MP2 calculation with MGEMM

```
$comment
MGEMM example
$end
$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
jobtype sp
exchange hf
method rimp2
basis cc-pvdz
aux_basis rimp2-cc-pvdz
cuda_rimp2 1
USE_MGEMM 1
mgemm_thresh 10000
$end
```

5.5.5 Spin-biased MP2 (SCS-MP2, SOS-MP2, MOS-MP2, and O2) Energies and Gradients

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 [29]. 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)[30].

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 [31]. 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 [31] and the analytical gradient [32] of this method are available in Q-CHEM 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 [33]. 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 [33]:

$$g_\omega(r_{12}) = \frac{1}{r_{12}} + c_{\mathrm{MOS}} \frac{\mathrm{erf}\,(\omega r_{12})}{r_{12}} \tag{5.22}$$

The recommended OMEGA value is $\omega = 0.6$ *a.u.* [33]. 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.

More recently, Brueckner orbitals (BO) 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. So 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" abbreviated as O2 method [34] 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.

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 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 $n = 600$ ($\omega = 0.6$ *a.u.*)
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)

5.5.6 Examples

Example 5.7 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
exchange           hf
correlation        rimp2
basis              aug-cc-pvdz
aux_basis          rimp2-aug-cc-pvdz
basis2             racc-pvdz            Optional Secondary basis
thresh             12
scf_convergence    8
max_sub_file_num   128
SCS                1                    Turn on spin-component scaling
dual_basis_energy  true                 Optional dual-basis approximation
n_frozen_core      fc
```

```
symmetry          false
sym_ignore        true
$end
```

Example 5.8 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
jobtype           sp
exchange          hf
correlation       rimp2
basis             aug-cc-pvtz
aux_basis         rimp2-aug-cc-pvtz
basis2            racc-pvtz           Optional Secondary basis
thresh            12
scf_convergence   8
max_sub_file_num  128
SCS               3                   Spin-component scale arbitrarily
SOS_FACTOR        0400000             Specify OS parameter
SSS_FACTOR        1290000             Specfiy SS parameter
dual_basis_energy true                Optional dual-basis approximation
n_frozen_core     fc
symmetry          false
sym_ignore        true
$end
```

Example 5.9 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
   SYMMETRY       false
$end
```

Example 5.10 Example of MOS-MP2 energy evaluation with frozen core approximation

```
$molecule
   0 1
   Cl
   Cl 1 2.05
$end

$rem
   JOBTYPE           sp
   METHOD            mosmp2
   OMEGA             600
   BASIS             cc-pVTZ
   AUX_BASIS         rimp2-cc-pVTZ
   N_FROZEN_CORE     fc
   THRESH            12
   SCF_CONVERGENCE   8
$end
```

Example 5.11 Example of O2 methodology applied to $\mathcal{O}(N^4)$ SOSMP2

```
   $molecule
   1 2
   F
   H 1 1.001
$end

$rem
   UNRESTRICTED      TRUE
   JOBTYPE           FORCE            Options are SP/FORCE/OPT
   EXCHANGE          HF
   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
   BASIS             sto-3g
   AUX_BASIS         rimp2-vdz
   SCF_CONVERGENCE   8
   THRESH            14
   SYMMETRY          FALSE
   PURECART          1111
$end
```

Example 5.12 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
   SYMMETRY         FALSE
   PURECART         1111
$end
```

5.5.7 RI-TRIM MP2 Energies

The triatomics in molecules (TRIM) local correlation approximation to MP2 theory [16] was described in detail in Section 5.4.1 which also discussed our implementation of this approach based on conventional four-center two-electron integrals. Q-CHEM 3.0 also includes an auxiliary basis implementation of the TRIM model. The new RI-TRIM MP2 energy algorithm [35] 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 [35]. 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 5.13 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
   SYMMETRY       false
$end
```

5.5.8 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 [36, 37] and analytic first derivatives [38].

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.

5.6 Short-Range Correlation Methods

5.6.1 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[39, 40], 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 noncovalent interactions. Attenuated MP2 improves upon MP2 for inter- and intramolecular interactions, with significantly better performance for relative and binding energies of noncovalent complexes, frequently outperforming complete basis set estimates of MP2[41, 42].

Attenuated MP2, denoted MP2(attenuator, basis) is implemented in Q-CHEM based on the complementary terf function, below.

$$\mathrm{s(r)} = \mathrm{terfc(r, r_0)} = \frac{1}{2}\left\{\mathrm{erfc}\left[\omega(\mathrm{r} - \mathrm{r_0})\right] + \mathrm{erfc}\left[\omega(\mathrm{r} + \mathrm{r_0})\right]\right\} \tag{5.23}$$

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[41] and aug-cc-pVTZ[42]. 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.

5.6.2 Examples

Example 5.14 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
jobtype sp
method attrimp2
basis aug-cc-pvdz
aux_basis rimp2-aug-cc-pvdz
n_frozen_core fc
$end
```

Example 5.15 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
n_frozen_core fc
$end
```

5.7 Coupled-Cluster Methods

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 Profs. 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 Prof. 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 began. 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
CCSD(T)	CCMAN and CCMAN2
CCSD(2)	CCMAN
CCSD(fT)	CCMAN
CCSD(dT)	CCMAN
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.

5.7.1 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 [43], 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 wavefunction is:

$$|\Psi_{\mathrm{CCSD}}\rangle = \exp\left(\hat{T}_1 + \hat{T}_2\right)|\Phi_0\rangle \tag{5.24}$$

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):

$$\hat{T}_1 |\Phi_0\rangle = \sum_i^{\text{occ}} \sum_a^{\text{virt}} t_i^a |\Phi_i^a\rangle \tag{5.25}$$

$$\hat{T}_2 |\Phi_0\rangle = \frac{1}{4} \sum_{ij}^{\text{occ}} \sum_{ab}^{\text{virt}} t_{ij}^{ab} |\Phi_{ij}^{ab}\rangle \tag{5.26}$$

It is not feasible to determine the CCSD energy by variational minimization of $\langle E\rangle_{\text{CCSD}}$ 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:

$$\begin{aligned}
E_{\text{CCSD}} &= \langle \Phi_0 | \hat{H} | \Psi_{\text{CCSD}} \rangle \\
&= \left\langle \Phi_0 \left| \hat{H} \right| \left(1 + \hat{T}_1 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2 \right) \Phi_0 \right\rangle_C && (5.27) \\
0 &= \left\langle \Phi_i^a \left| \hat{H} - E_{\text{CCSD}} \right| \Psi_{\text{CCSD}} \right\rangle \\
&= \left\langle \Phi_i^a \left| \hat{H} \right| \left(1 + \hat{T}_1 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2 + \hat{T}_1\hat{T}_2 + \frac{1}{3!}\hat{T}_1^3 \right) \Phi_0 \right\rangle_C && (5.28) \\
0 &= \left\langle \Phi_{ij}^{ab} \left| \hat{H} - E_{\text{CCSD}} \right| \Psi_{\text{CCSD}} \right\rangle \\
&= \left\langle \Phi_{ij}^{ab} \left| \hat{H} \right| \left(1 + \hat{T}_1 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2 + \hat{T}_1\hat{T}_2 + \frac{1}{3!}\hat{T}_1^3 \right. \right. \\
&\qquad \left. \left. + \frac{1}{2}\hat{T}_2^2 + \frac{1}{2}\hat{T}_1^2\hat{T}_2 + \frac{1}{4!}\hat{T}_1^4 \right) \Phi_0 \right\rangle_C && (5.29)
\end{aligned}$$

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.

5.7.2 Quadratic Configuration Interaction (QCISD)

Quadratic configuration interaction with singles and doubles (QCISD) [44] 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:

$$E_{QCISD} = \left\langle \Phi_0 \left| \hat{H} \right| \left(1 + \hat{T}_2 \right) \Phi_0 \right\rangle_C \tag{5.30}$$

$$0 = \left\langle \Phi_i^a \left| \hat{H} \right| \left(\hat{T}_1 + \hat{T}_2 + \hat{T}_1 \hat{T}_2 \right) \Phi_0 \right\rangle_C \tag{5.31}$$

$$0 = \left\langle \Phi_{ij}^{ab} \left| \hat{H} \right| \left(1 + \hat{T}_1 + \hat{T}_2 + \frac{1}{2} \hat{T}_2^2 \right) \Phi_0 \right\rangle_C \tag{5.32}$$

QCISD energies are available in Q-CHEM, and are requested with the QCISD keyword. As discussed in Section 5.8, the non iterative QCISD(T) correction to the QCISD solution is also available to approximately incorporate the effect of higher substitutions.

5.7.3 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, by the following equations:

$$E_{\mathrm{CCD}} = \left\langle \Phi_0 \left| \hat{H} \right| \left(1 + \hat{T}_2 \right) \Phi_0 \right\rangle_C \tag{5.33}$$

$$0 = \left\langle \Phi_{ij}^{ab} \left| \hat{H} \right| \left(1 + \hat{T}_2 + \frac{1}{2} \hat{T}_2^2 \right) \Phi_0 \right\rangle_C \tag{5.34}$$

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) [45]. 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

$$\frac{\partial E_{\mathrm{CCD}}}{\partial \theta_i^a} = 0 \tag{5.35}$$

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 5.9.

5.7.4 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 [46]. The Quadratic Coupled Cluster Doubles (QCCD) method [47] 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

$$E_{\mathrm{QCCD}} = \left\langle \Phi_0 \left(1 + \hat{\Lambda}_2 + \frac{1}{2}\hat{\Lambda}_2^2 \right) \left| \hat{H} \right| \exp\left(\hat{T}_2 \right) \Phi_0 \right\rangle_C \tag{5.36}$$

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

$$E_{\mathrm{CCD}} = \left\langle \Phi_0 \left(1 + \hat{\Lambda}_2 \right) \left| \hat{H} \right| \exp\left(\hat{T}_2 \right) \Phi_0 \right\rangle_C \tag{5.37}$$

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. (5.36) and (5.37), 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 wavefunction and the ket wavefunction in the energy expression are treated more equivalently in QCCD than in CCD. This makes QCCD closer to a true variational treatment [46] where the bra and ket wavefunctions 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 [47], and the follow-up paper describing the full implementation [48]. 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.

5.7.5 Resolution-of-the-identity with CC (RI-CC)

The RI approximation (see Section 5.5) 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 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. In addition, while RI is available for jobs that need analytical gradients, only energies and properties are computed using RI. Energy derivatives are calculated using regular electron repulsion integral derivatives. Full RI implementation (with integrals used in decomposed form) is only available for CCMAN2. For maximum computational efficiency, combine with FNO (see Sections 5.10 and 6.7.7) when appropriate.

5.7.6 Cholesky decomposition with CC (CD-CC)

Two-electron integrals can be decomposed using Cholesky decomposition [49] giving rise to the same representation as in RI and substantially reducing the cost of integral transformation, disk storage requirements, and improving parallel performance:

$$(\mu\nu|\lambda\sigma) \approx \sum_{P=1}^{M} B^P_{\mu\nu} B^P_{\lambda\sigma}, \tag{5.38}$$

The rank of Cholesky decomposition, M, is typically 3-10 times larger than the number of basis functions N [50]; it depends on the decomposition threshold δ and is considerably smaller than the full rank of the matrix, $N(N+1)/2$ [50–52]. Cholesky decomposition removes linear dependencies in product densities [50] $(\mu\nu|$ 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 DIRECT_RI keyword.

Note: Cholesky decomposition is available for all CCMAN2 methods. Analytic gradients are not yet available; only energies and properties are computed using CD. For maximum computational efficiency, combine with FNO (see Sections 5.10 and 6.7.7) when appropriate.

5.7.7 Job Control Options

There are a large number of options for the coupled-cluster singles and doubles methods. They are documented in Appendix C, 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 5.9. The more common options relating to convergence control are discussed there, in Section 5.9.5. Below we list the options that one should be aware of for routine calculations.

For memory options and parallel execution, see Section 5.13.

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 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 $abc \times 10^{-de}$, *e.g.*, 2502 corresponds to 0.25
RECOMMENDATION:
Increase to 0.25, 0.5 or 0.75 for non convergent coupled-cluster calculations.

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_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:

$0 \rightarrow 7$ 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 to define tolerance of 10^{-n}

RECOMMENDATION:

2 - qualitative calculations, 3 - appropriate for most cases, 4 - quantitative (error in total energy typically less than 1e-6 hartree)

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.

5.7.8 Examples

Example 5.16 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, 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          od
   BASIS           6-31g*
   N_FROZEN_CORE   fc
$end

@@@

$molecule
   read
$end

$rem
   METHOD          qccd
   BASIS            6-31g*
   N_FROZEN_CORE    fc
$end
```

Example 5.17 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 5.18 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 5.19 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
```

5.8 Non-iterative Corrections to Coupled Cluster Energies

5.8.1 (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) [53] and QCISD(T) methods [44] The accuracy of these methods is well-documented for many cases [54], 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.

5.8.2 (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 [55]. 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) [55, 56]. 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 wavefunction 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 [57], 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 [55]. 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.

5.8.3 (dT) and (fT) corrections

Alternative inclusion of non-iterative N^7 triples corrections is described in Section 6.7.19. 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 co-workers.

5.8.4 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).

There is only one additional job control option. 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.

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 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).

5.8.5 Example

Example 5.20 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 5.21 Water: Ground state CCSD(dT) calculation using RI

```
$molecule
0 1
O
H1 O OH
H2 O OH H1 HOH

OH  = 0.957
```

```
HOH = 104.5
$end

$rem
JOBTYPE       SP
BASIS         cc-pvtz
AUX_BASIS     rimp2-cc-pvtz
METHOD        CCSD(dT)
$end
```

5.9 Coupled Cluster Active Space Methods

5.9.1 Introduction

Electron correlation effects can be qualitatively divided into two classes. The first class is static or nondynamical 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 nondynamical 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 nondynamical correlation energy is conveniently defined as the solution of the Schrödinger equation within a small basis set composed of valence bonding, antibonding 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) [58], or equivalently, the fully optimized reaction space (FORS) method [59].

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 wavefunction with optimized orbitals [60]. They are active space generalizations of the OD and QCCD methods discussed previously in Sections 5.7.3 and 5.7.4, 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 antibonding) 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 the 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

$$\frac{\partial E_{\mathrm{CCD}}}{\partial \theta_i^{j'}} = 0 \tag{5.39}$$

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:

$$\frac{\partial E_{\mathrm{CCD}}}{\partial \theta_a^{b'}} = 0 \tag{5.40}$$

5.9.2 VOD and VOD(2) Methods

The VOD method is the active space version of the OD method described earlier in Section 5.7.3. 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 permits more accurate calculations of relative energies by accounting for dynamical correlation.

5.9.3 VQCCD

The VQCCD method is the active space version of the QCCD method described earlier in Section 5.7.3. 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.

5.9.4 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:

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:

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:

MGC_LOCALINTS
Pair filter on an integrals.
TYPE:
BOOL
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:

MGC_LOCALINTER
Pair filter on an intermediate.
TYPE:
BOOL
DEFAULT:
FALSE
OPTIONS:
Any nonzero value enforces the pair constraint on intermediates,
significantly reducing computational cost. Not recommended for $\leq$ 2 pair locality
RECOMMENDATION:

5.9.5 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 C. 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

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.

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 preconvergence 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.

5.9.6 Examples

Example 5.22 Two jobs that compare the correlation energy of the water molecule with partially stretched bonds, calculated via the two coupled-cluster active space methods, VOD, and VQCCD. These are relatively “easy” jobs 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  a

   r =   1.5
   a = 104.5
$end
```

```
$rem
   METHOD        vod
   BASIS         6-31G
$end

@@@

$molecule
  read
$end

$rem
   METHOD        vqccd
   BASIS         6-31G
$end
```

Example 5.23 The water molecule with highly stretched bonds, calculated via the two coupled-cluster active space methods, VOD, and VQCCD. These are "difficult" jobs 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. In this case, the VQCCD method yields far better results than VOD!.

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

@@@

$molecule
   read
$end

$rem
   METHOD                vqccd
   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
```

5.10 Frozen Natural Orbitals in CCD, CCSD, OD, QCCD, and QCISD Calculations

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 co-workers [61–63]. Extension of the FNO approach to ionized states within EOM-CC formalism was recently introduced and benchmarked [64] (see Section 6.7.7).

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 wavefunctions.

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. 62, 63. 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 [64].

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 [64]. 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 [64]. 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. 64).

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 [64].

5.10.1 Job Control Options

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

5.10.2 Example

Example 5.24 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
```

5.11 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 [65]. This can be achieved by first doing DFT calculation and then reading the orbitals and turning the Hartree-Fock procedure off. A more convenient way is just to specify EXCHANGE, *e.g.*, EXCHANGE=B3LYP means that B3LYP orbitals will be computed and used in the CCMAN/CCMAN2 module.

5.11.1 Example

Example 5.25 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
jobtype            SP            single point
exchange           b3lyp
correlation        ccsd
BASIS              cc-pVDZ
N_FROZEN_CORE      1
$end
```

5.12 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. In addition, gradients for selected GVB models are available.

For the CCSD and OO-CCD wavefunctions, Q-CHEM can also calculate dipole moments, $\langle R^2 \rangle$ (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 $\langle S^2 \rangle$ 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 6.7.12). 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.

5.12.1 Job Control Options

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 electron density, and the total $\langle R^2\rangle = \langle X^2\rangle + \langle Y^2\rangle + \langle Z^2\rangle$ (in atomic units). Incompatible with JOBTYPE=FORCE, OPT, FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations need to be solved (lambda CCSD equations) for properties with the cost approximately the same as CCSD equations. Use 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 $\langle S^2\rangle$. 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 $\langle S^2\rangle$. 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).

5.12.2 Examples

Example 5.26 CCSD geometry optimization of HHeF followed up by properties calculations

```
$molecule
0 1
H    .000000     .000000   -1.886789
He   .000000     .000000   -1.093834
F    .000000     .000000     .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
```

Example 5.27 CCSD on 1,2-dichloroethane gauche conformation using SCRF solvent model

```
$molecule
0 1
C     0.6541334418569877  -0.3817051480045552   0.8808840579322241
C    -0.6541334418569877   0.3817051480045552   0.8808840579322241
Cl    1.7322599856434779   0.0877596094659600  -0.4630557359272908
H     1.1862455146007043  -0.1665749506296433   1.7960750032785453
H     0.4889356972641761  -1.4444403797631731   0.8058465784063975
Cl   -1.7322599856434779  -0.0877596094659600  -0.4630557359272908
H    -1.1862455146007043   0.1665749506296433   1.7960750032785453
H    -0.4889356972641761   1.4444403797631731   0.8058465784063975
$end

$rem
JOBTYPE            SP
METHOD             CCSD
BASIS              6-31g**
N_FROZEN_CORE      FC
CC_SAVEAMPL        1         Save CC amplitudes on disk
SOLVENT_METHOD     SCRF
SOL_ORDER          15        L=15 Multipole moment order
SOLUTE_RADIUS      36500     3.65 Angstrom Solute Radius
```

```
SOLVENT_DIELECTRIC 89300     8.93 Dielectric (Methylene Chloride)
$end
```

5.13 Memory Options and Parallelization of Coupled-Cluster Calculations

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 the multi-core architecture. The code is parallelized at the level of the tensor library such that the most time consuming operation, tensor contraction, is performed on different processors (or different cores of the same processor) using shared memory and shared scratch disk space[66].

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 the 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 serial mode using single thread only.

Setting the memory limit correctly is also very important for high performance when running larger jobs. To estimate the amount of memory required for coupled-clusters and related calculations, one can use the following formula:

$$\text{Memory} = \frac{(\text{Number of basis set functions})^4}{131072}\ \text{Mb} \tag{5.41}$$

If the new code (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. In addition, if gradients are calculated, the amount should be multiplied by 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 older CCMAN and newer 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 rarely exceed 1000–2000 Mb even for relatively large jobs.

The memory limit for coupled-clusters calculations is set by CC_MEMORY. When running older CCMAN, its 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 of approximately 1.5 Gb is not 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 will affect other jobs that you may wish to run on the same node.

In addition, the user should verify that the disk and RAM together have enough space by using the above formula. In cases when CC_MEMORY set up is in conflict with the available space on a particular platform, the CC job may segfault at run time. In such cases readjusting the CC_MEMORY value in the input is necessary so as to eliminate the segfaulting.

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.)

5.14 Simplified Coupled-Cluster Methods Based on a Perfect-Pairing Active Space

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 wavefunction, for example) and nominally empty orbitals. Examples include so-called diradicaloid molecules [67], 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 wavefunction 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 antibonding in character). This is usually called the perfect pairing active space, and it clearly is well-suited to represent the bonding-antibonding 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 [60], which scales as $\mathcal{O}(N^6)$ (see Section 5.9.2). 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) [68, 69], imperfect pairing (IP) [70] and restricted coupled cluster (RCC) [71] 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 wavefunctions [72], 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 [73–75], is a valence bond approach that goes well beyond the power of GVB-PP and related methods, as discussed below in Sec. 5.14.2.

5.14.1 Perfect pairing (PP)

To be more specific, the coupled cluster PP wavefunction is written as

$$|\Psi\rangle = \exp\left(\sum_{i=1}^{n_{\text{active}}} t_i \hat{a}_{i*}^{\dagger} \hat{a}_{\bar{i}*}^{\dagger} \hat{a}_{\bar{i}} \hat{a}_i\right) |\Phi\rangle \tag{5.42}$$

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 antibonding 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 4.5.4. Despite the simplicity of the PP wavefunction, 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) [76] 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 2x2 Given's 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 (it's default value)

GVB_OLD_UPP

Which unrestricted algorithm to use for GVB.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use Unrestricted-in-Active Pairs

1 Use Unrestricted Implementation described in Ref. 69

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 projectively, it is also possible, as described in Sec. 5.14.2 below, to solve variationally for the PP energy. This variational PP solution is the reference wavefunction 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.

5.14.2 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) [73, 74] is an appropriate method. 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 wavefunction is the perfect pairing wavefunction, which is shown in eqn. (5.42). One important difference is that CCVB uses the PP wavefunction as a reference in the same way that other GVBMAN methods use a reference determinant.

The PP wavefunction is a product of simple, strongly orthogonal singlet geminals. Ignoring normalizations, two equivalent ways of displaying these geminals are

$$\begin{aligned} (\phi_i\phi_i + t_i\phi_i^*\phi_i^*)(\alpha\beta - \beta\alpha) &\quad \text{(Natural-orbital form)} \\ \chi_i\chi_i'(\alpha\beta - \beta\alpha) &\quad \text{(Valence-bond form)}, \end{aligned} \tag{5.43}$$

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 recoupling 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 wavefunction:

$$
\begin{aligned}
|\Psi\rangle = |\Phi_0\rangle + \sum_{k<l} t_{kl} |\Phi_{(kl)}\rangle + \sum_{k<l<m<n} \Big[& t_{kl} t_{mn} |\Phi_{(kl)(mn)}\rangle \\
& + t_{km} t_{ln} |\Phi_{(km)(ln)}\rangle + t_{kn} t_{lm} |\Phi_{(kn)(lm)}\rangle \Big] + \cdots . \qquad (5.44)
\end{aligned}
$$

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 wavefunction 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 [70] and RCC [71] wavefunctions. 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 wavefunction 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 nonlocal 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 5.28 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
  jobtype             = sp
  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
  symmetry            = false
  sym_ignore          = true
  print_orbitals      = true
$end
```

5.14.3 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) [77], 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 Sec. 5.14.1 and Sec. 5.14.4 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 5.29 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)
%   PURECART        11111
   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
```

5.14.4 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 [78, 79]. 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 5.5 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 5.30 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
%   PURECART       11111
$end
```

If further improvement in the orbitals are needed, the GVB_SIP, GVB_DIP, OP, NP and 2P models are also included [76]. 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 wavefunction 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)$(e^{t^p_{ij}} - 1)/(e^1 - 1)$

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 [80]. 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 wavefunction.

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_{\text{active}} = n_\beta$). Or alternatively one could make only formal bonding electron pairs active ($n_{\text{active}} = N_{\text{STO}-3\text{G}} - 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 (ie 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 (ie 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 $\geq$ 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 $\geq$ 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 [81, 82], 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 wavefunctions (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° [70]. Likewise the allyl radical breaks symmetry in the unrestricted PP model [69], 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 hypervalent 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.

5.15 Geminal Models

5.15.1 Reference wavefunction

Computational models that use single reference wavefunction 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 wavefunctions. Requirements of computational efficiency and size consistency constrain geminals to have $S_z = 0$ [83], with each geminal spanning its own subspace of molecular orbitals [84]. Geminal wavefunctions were introduced into computational chemistry by Hurley, Lennard-Jones, and Pople [85]. An excellent review of the history and properties of geminal wavefunctions is given by Surjan [86].

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 wavefunction 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 wavefunction has the form

$$\begin{aligned}
\Psi_{\mathrm{SSG}} &= \hat{A}\left[\psi_1(\mathbf{r}_1,\mathbf{r}_2)\ldots\psi_{n_\beta}(\mathbf{r}_{2n_\beta-1},\mathbf{r}_{2n_\beta})\phi_i(\mathbf{r}_{2n_\beta+1})\ldots\phi_j(\mathbf{r}_{n_\beta+n_\alpha})\right] \\
\psi_a(\mathbf{r}_1,\mathbf{r}_2) &= \sum_{k\in A}\frac{D_i^A}{\sqrt{2}}[\phi_k(\mathbf{r}_1)\bar{\phi_k}(\mathbf{r}_2)-\phi_k(\mathbf{r}_2)\bar{\phi_k}(\mathbf{r}_1)] \qquad (5.45)\\
\phi_k(\mathbf{r}_1) &= \sum_\lambda C_\lambda^k\chi_\lambda(\mathbf{r}_1) \\
\bar{\phi_k}(\mathbf{r}_1) &= \sum_\lambda \bar{C}_\lambda^k\chi_\lambda(\mathbf{r}_1)
\end{aligned}$$

with the coefficients C, D, and subspaces A chosen to minimize the energy

$$E_{\mathrm{SSG}} = \frac{\langle\Psi_{\mathrm{SSG}}|\hat{H}|\Psi_{\mathrm{SSG}}\rangle}{\langle\Psi_{\mathrm{SSG}}|\Psi_{\mathrm{SSG}}\rangle} \qquad (5.46)$$

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 wavefunctions, and in former case makes preliminary assignment of individual orbital pairs into geminals. The verification of orbital assignments is performed every ten wavefunction 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. The direct minimization steps [87] seeks a minimum along the gradient direction, rescaled by the quantity analogous to the orbital energy differences in SCF theory [83]. 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 [88, 89]. 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 [90].

5.15.2 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 [91, 92] 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 wavefunction to be an eigenfunction of the reference Hamiltonian [93].

All perturbative contributions to the SSG(EN2) energy (second-order Epstein-Nesbet perturbation theory of SSG wavefunction) 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 wavefunction. Formally, the SSG(EN2) correction can be applied both to RSSG and USSG wavefunctions. 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 wavefunction rarely exceeding 0.1 in magnitude.

SSG

Controls the calculation of the SSG wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not compute the SSG wavefunction

1 Do compute the SSG wavefunction

RECOMMENDATION:

See also the UNRESTRICTED and DIIS_SUBSPACE_SIZE *$rem* variables.

References and Further Reading

[1] Self-Consistent Field Methods (Chapter 4).

[2] Excited-State Calculations (Chapter 6).

[3] Basis Sets (Chapter 7) and Effective Core Potentials (Chapter 8).

[4] For a tutorial introduction to electron correlation methods based on wavefunctions, see Ref. 94.

[5] For a general textbook introduction to electron correlation methods and their respective strengths and weaknesses, see Ref. 95.

[6] L. A. Curtiss, K. Raghavachari, G. W. Trucks, and J. A. Pople, *J. Chem. Phys.* **94**, 7221 (1991).

[7] L. A. Curtiss, K. Raghavachari, P. C. Redfern, V. Rassolov, and J. A. Pople, *J. Chem. Phys.* **109**, 7764 (1998).

[8] D. Jayatilaka and T. J. Lee, *Chem. Phys. Lett.* **199**, 211 (1992).

[9] C. Møller and M. S. Plesset, *Phys. Rev.* **46**, 618 (1934).

[10] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory*, Wiley, New York, 1986.

[11] T. Helgaker, J. Gauss, P. Jorgensen, and J. Olsen, *J. Chem. Phys.* **106**, 6430 (1997).

[12] M. Head-Gordon, *Mol. Phys.* **96**, 673 (1999).

[13] M. Head-Gordon, J. A. Pople, and M. J. Frisch, *Chem. Phys. Lett.* **153**, 503 (1988).

[14] M. J. Frisch, M. Head-Gordon, and J. A. Pople, *Chem. Phys. Lett.* **166**, 275 (1990).

[15] V. A. Rassolov, J. A. Pople, P. C. Redfern, and L. A. Curtiss, *Chem. Phys. Lett.* **350**, 573 (2001).

[16] M. S. Lee, P. E. Maslen, and M. Head-Gordon, *J. Chem. Phys.* **112**, 3592 (2000).

[17] M. Head-Gordon, M. S. Lee, and P. E. Maslen, Simulation and theory of electrostatic interactions in solution, volume 492 of *AIP Conference Proceedings*, page 301, American Institute of Physics, New York, 1999.

[18] M. S. Lee and M. Head-Gordon, *Int. J. Quantum Chem.* **76**, 169 (2000).

[19] S. Saebo and P. Pulay, *Annu. Rev. Phys. Chem.* **44**, 213 (1993).

[20] L. A. Curtiss, K. Raghavachari, P. C. Redfern, and J. A. Pople, *J. Chem. Phys.* **112**, 7374 (2000).

[21] M. S. Lee, PhD thesis, University of California, Berkeley, CA, 2000.

[22] M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.* **208**, 359 (1993).

[23] B. I. Dunlap, *Phys. Chem. Chem. Phys.* **2**, 2113 (2000).

[24] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon, *Proc. Natl. Acad. Sci. USA* **102**, 6692 (2005).

[25] F. Weigend, M. Haser, H. Patzelt, and R. Ahlrichs, *Chem. Phys. Lett.* **294**, 143 (1998).

[26] F. Weigend, A. Kohn, and C. Hättig, *J. Chem. Phys.* **116**, 3175 (2002).

[27] F. Weigend and M. Haser, *Theor. Chem. Acc.* **97**, 331 (1997).

[28] R. A. DiStasio, Jr., R. P. Steele, Y. M. Rhee, Y. Shao, and M. Head-Gordon, *J. Comput. Chem.* **28**, 839 (2007).

[29] S. Grimme, *J. Chem. Phys.* **118**, 9095 (2003).

[30] R. A. DiStasio, Jr. and M. Head-Gordon, *Mol. Phys.* **105**, 1073 (2007).

[31] Y. Jung, R. C. Lochan, A. D. Dutoi, and M. Head-Gordon, *J. Chem. Phys.* **121**, 9793 (2004).

[32] R. C. Lochan, Y. Shao, and M. Head-Gordon, *J. Chem. Theory Comput.* **3**, 988 (2007).

[33] R. C. Lochan, Y. Jung, and M. Head-Gordon, *J. Phys. Chem. A* **109**, 7598 (2005).

[34] R. C. Lochan and M. Head-Gordon, *J. Chem. Phys.* **126**, 164101 (2007).

[35] R. A. DiStasio, Jr., Y. Jung, and M. Head-Gordon, *J. Chem. Theory Comput.* **1**, 862 (2005).

[36] R. P. Steele, R. A. DiStasio, Jr., Y. Shao, J. Kong, and M. Head-Gordon, *J. Chem. Phys.* **125**, 074108 (2006).

[37] R. P. Steele, R. A. DiStasio, Jr., and M. Head-Gordon, *J. Chem. Theory Comput.* **5**, 1560 (2009).

[38] R. A. DiStasio, Jr., R. P. Steele, and M. Head-Gordon, *Mol. Phys.* **105**, 27331 (2007).

[39] J. P. Dombroski, S. W. Taylor, and P. M. W. Gill, *J. Phys. Chem.* **100**, 6272 (1996).

[40] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill, *Chem. Phys. Lett.* **254**, 329 (1996).

[41] M. Goldey and M. Head-Gordon, *J. Phys. Chem. Lett.* **3**, 3592 (2012).

[42] M. Goldey, A. Dutoi, and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **15**, 15869 (2013).

[43] G. D. Purvis and R. J. Bartlett, *J. Chem. Phys.* **76**, 1910 (1982).

[44] J. A. Pople, M. Head-Gordon, and K. Raghavachari, *J. Chem. Phys.* **87**, 5968 (1987).

[45] C. D. Sherrill, A. I. Krylov, E. F. C. Byrd, and M. Head-Gordon, *J. Chem. Phys.* **109**, 4171 (1998).

[46] T. Van Voorhis and M. Head-Gordon, *J. Chem. Phys.* **113**, 8873 (2000).

[47] T. Van Voorhis and M. Head-Gordon, *Chem. Phys. Lett.* **330**, 585 (2000).

[48] E. F. C. Byrd, T. Van Voorhis, and M. Head-Gordon, *J. Phys. Chem. B* **106**, 8070 (2002).

[49] E. Epifanovsky et al., *J. Chem. Phys.* **139**, 134105 (2013).

[50] F. Aquilante, T. B. Pedersen, and R. Lindh, *Theor. Chem. Acc.* **124**, 1 (2009).

[51] N. Beebe and J. Linderberg, *Int. J. Quantum Chem.* **12**, 683 (1977).

[52] S. Wilson, *Comput. Phys. Commun.* **58**, 71 (1990).

[53] K. Raghavachari, G. W. Trucks, J. A. Pople, and M. Head-Gordon, *Chem. Phys. Lett.* **157**, 479 (1989).

[54] T. J. Lee and G. E. Scuseria, in *Quantum Mechanical Calculations with Chemical Accuracy*, edited by S. R. Langhoff, page 47, Kluwer, Dordrecht, 1995.

[55] S. R. Gwaltney and M. Head-Gordon, *Chem. Phys. Lett.* **323**, 21 (2000).

[56] S. R. Gwaltney and M. Head-Gordon, *J. Chem. Phys.* **115**, 5033 (2001).

[57] S. A. Kucharski and R. J. Bartlett, *J. Chem. Phys.* **108**, 5243 (1998).

[58] B. O. Roos, *Adv. Chem. Phys.* **69**, 399 (1987).

[59] K. Ruedenberg, M. W. Schmidt, M. M. Gilbert, and S. T. Elbert, *Chem. Phys.* **71**, 49 (1982).

[60] A. I. Krylov, C. D. Sherrill, E. F. C. Byrd, and M. Head-Gordon, *J. Chem. Phys.* **109**, 10669 (1998).

[61] C. Sosa, J. Geertsen, G. W. Trucks, and R. J. Bartlett, *Chem. Phys. Lett.* **159**, 148 (1989).

[62] A. G. Taube and R. J. Bartlett, *Collect. Czech. Chem. Commun.* **70**, 837 (2005).

[63] A. G. Taube and R. J. Bartlett, *J. Chem. Phys.* **128**, 164101 (2008).

[64] A. Landau, K. Khistyaev, S. Dolgikh, and A. I. Krylov, *J. Chem. Phys.* **132**, 014109 (2010).

[65] G. J. O. Beran, S. R. Gwaltney, and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **5**, 2488 (2003).

[66] E. Epifanovsky et al., *J. Comput. Chem.* **34**, 2293 (2013).

[67] Y. Jung and M. Head-Gordon, *ChemPhysChem* **4**, 522 (2003).

[68] J. Cullen, *Chem. Phys.* **202**, 217 (1996).

[69] G. J. O. Beran, B. Austin, A. Sodt, and M. Head-Gordon, *J. Phys. Chem. A* **109**, 9183 (2005).

[70] T. Van Voorhis and M. Head-Gordon, *Chem. Phys. Lett.* **317**, 575 (2000).

[71] T. Van Voorhis and M. Head-Gordon, *J. Chem. Phys.* **115**, 7814 (2001).

[72] W. A. Goddard III and L. B. Harding, *Annu. Rev. Phys. Chem.* **29**, 363 (1978).

[73] D. W. Small and M. Head-Gordon, *J. Chem. Phys.* **130**, 084103 (2009).

[74] D. W. Small and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **13**, 19285 (2011).

[75] D. W. Small and M. Head-Gordon, *J. Chem. Phys.* **137**, 114103 (2012).

[76] K. V. Lawler, D. W. Small, and M. Head-Gordon, *J. Phys. Chem. A* **114**, 2930 (2010).

[77] G. J. O. Beran, M. Head-Gordon, and S. R. Gwaltney, *J. Chem. Phys.* **124**, 114107 (2006).

[78] T. Van Voorhis and M. Head-Gordon, *J. Chem. Phys.* **117**, 9190 (2002).

[79] A. Sodt, G. J. O. Beran, Y. Jung, B. Austin, and M. Head-Gordon, *J. Chem. Theory Comput.* **2**, 300 (2006).

[80] K. V. Lawler, J. A. Parkhill, and M. Head-Gordon, *J. Chem. Phys.* **130**, 184113 (2009).

[81] K. V. Lawler, G. J. O. Beran, and M. Head-Gordon, *J. Chem. Phys.* **128**, 024107 (2008).

[82] K. V. Lawler, J. A. Parkhill, and M. Head-Gordon, *Mol. Phys.* **106**, 2309 (2008).

[83] V. A. Rassolov, *J. Chem. Phys.* **117**, 5978 (2002).

[84] T. Arai, *J. Chem. Phys.* **33**, 95 (1960).

[85] A. C. Hurley, J. E. Lennard-Jones, and J. A. Pople, *Proc. Roy. Soc. London A* **220**, 446 (1953).

[86] P. R. Surján, *Topics Curr. Chem.* (1999).

[87] R. Seeger and J. A. Pople, *J. Chem. Phys.* **65**, 265 (1976).

[88] P. Pulay, *Chem. Phys. Lett.* **73**, 393 (1980).

[89] P. Pulay, *J. Comput. Chem.* **3**, 556 (1982).

[90] F. W. Bobrowicz and W. A. Goddard III, in *Mehods of Electronic Structure Theory*, edited by H. F. Schaefer III, volume 3, page 79, Plenum, New York, 1977.

[91] P. S. Epstein, *Phys. Rev.* **28**, 695 (1926).

[92] R. K. Nesbet, *Proc. Roy. Soc. Ser. A* **230**, 312 (1955).

[93] V. A. Rassolov, F. Xu, and S. Garaschchuk, *J. Chem. Phys.* **120**, 10385 (2004).

[94] R. J. Bartlett and J. F. Stanton, in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz and D. B. Boyd, volume 5, chapter 2, page 65, Wiley-VCH, New York, 1994.

[95] F. Jensen, *Introduction to Computational Chemistry*, Wiley, New York, 1994.

Chapter 6

Open-Shell and Excited-State Methods

6.1 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 wavefunctions 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 wavefunction. 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 wavefunctions, *e.g.* diradicals, ionized and electron attachment states and more.

In excited-state calculations, as for ground state calculations, the user must strike a compromise between cost and accuracy. The few sections of this Chapter summarize Q-CHEM's capabilities in four general classes of excited state methods:

- Single-electron wavefunction-based methods (Section 6.2). 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 6.3). TDDFT is the most useful extension of density functional theory to excited states that has been developed so far. For a cost that is little greater than the simple wavefunction methods such as CIS, a significantly more accurate method results. TDDFT can be extended to treat di- and tri-radicals and bond-breaking by adopting the spin-flip approach (see Section 6.3.1 for details).

- The Maximum Overlap Method (MOM) for excited SCF states (Section 6.4). This method overcomes some of the deficiencies of TDDFT and, in particular, can be used for modeling charge-transfer and Rydberg transitions.

- Restricted open-shell Kohn-Sham (ROKS) method is another Δ-SCF approach for excited states (Section 6.5).
- Wavefunction-based electron correlation treatments (Sections 6.6, 6.8, 6.9 and 6.7). Roughly speaking, these are excited state analogs of the ground state wavefunction-based electron correlation methods discussed in Chapter 5. They are more accurate than the methods of Section 6.2, but also significantly more computationally expensive. These methods can also describe certain multi-configurational wavefunctions, for example, problematic doublet radicals, diradicals, triradicals, and more.

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 7). 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.
- **Visualization**
  It is possible to visualize the excited states either by attachment/detachment density analysis (available for CIS, RPA, TDDFT, and ADC only) 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 6.11.1 of this Chapter. In addition Dyson orbitals can be calculated and plotted for the ionization from the ground and electronically excited states for the CCSD and EOM-CCSD wavefunctions. For the RAS-SF method (Section 6.9), 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, EOM-OD excited state optimizations may also be performed using finite difference methods, however, these can be very time-consuming to compute.
- **Optimization of the crossings between potential energy surfaces**
  Seams between potential energy surfaces can be located and optimized by using analytic gradients within CCSD and EOM-CCSD formalisms.
- **Properties**
  Properties such as transition dipoles, dipole moments, spatial extent of electron densities and $\langle S^2\rangle$ values can be computed for ADC, EOM-CCSD, EOM-OD, RAS-SF and CIS wavefunctions.

- **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.

EOM-CC, and most of the CI codes are part of CCMAN and CCMAN2.

METHOD
Specifies the level of theory.
TYPE:
STRING
DEFAULT:
None No Correlation
OPTIONS:

CIS	Section 6.2.1
CIS(D)	Section 6.6.1
RI-CIS(D)	Section 6.6.2
SOS-CIS(D)	Section 6.6.3
SOS-CIS(D0)	Section 6.6.4
CISD	Section 6.7.2
CISDT	Section 6.7.2
EOM-OD	Section 6.7.2
EOM-CCSD	Section 6.7.2
EOM-CCSD(dT)	Section 6.7.19
EOM-CCSD(fT)	Section 6.7.19
EOM-CC(2,3)	Section 6.7.15
ADC(0)	Section 6.8
ADC(1)	Section 6.8
ADC(2)	Section 6.8
ADC(2)X	Section 6.8
ADC(3)	Section 6.8
RAS-CI	Section 6.9
RAS-CI-2	Section 6.9

RECOMMENDATION:
Consult the literature for guidance.

6.2 Non-Correlated Wavefunction Methods

Q-CHEM includes several excited state methods which do not incorporate correlation: CIS, XCIS and RPA. 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.

6.2.1 Single Excitation Configuration Interaction (CIS)

The derivation of the CI-singles [3, 4] energy and wave function begins by selecting the HF single-determinant wavefunction as reference for the ground state of the system:

$$\Psi_{\mathrm{HF}} = \frac{1}{\sqrt{n!}} \det \{\chi_1 \chi_2 \cdots \chi_i \chi_j \cdots \chi_n\} \tag{6.1}$$

where n is the number of electrons, and the spin orbitals

$$\chi_i = \sum_{\mu}^{N} c_{\mu i} \phi_\mu \tag{6.2}$$

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

$$\mathbf{FC} = \varepsilon \mathbf{SC} \tag{6.3}$$

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

$$F_{\mu\upsilon} = H_{\mu\upsilon} + \sum_{\lambda\sigma} \sum_{i} c_{\mu i} c_{\upsilon i} \left(\mu\lambda \mid\mid \upsilon\sigma\right) \tag{6.4}$$

involving the core Hamiltonian and the anti-symmetrized two-electron integrals

$$(\mu\nu||\lambda\sigma) = \int \int \phi_\mu(\mathbf{r}_1)\phi_\nu(\mathbf{r}_2)\,(1/1r_{12}r_{12})\,[\phi_\lambda(\mathbf{r}_1)\phi_\sigma(\mathbf{r_2}) - \phi_\lambda(\mathbf{r_2})\phi_\sigma(\mathbf{r}_1)]\,d\mathbf{r_1}d\mathbf{r_2} \tag{6.5}$$

On solving Eq. (6.3), the total energy of the ground state single determinant can be expressed as

$$E_{\mathrm{HF}} = \sum_{\mu\upsilon} P_{\mu\upsilon}^{\mathrm{HF}} H_{\mu\upsilon} + \frac{1}{2} \sum_{\mu\upsilon\lambda\sigma} P_{\mu\upsilon}^{\mathrm{HF}} P_{\lambda\sigma}^{\mathrm{HF}} \left(\mu\lambda \mid\mid \upsilon\sigma\right) + V_{\mathrm{nuc}} \tag{6.6}$$

where P^{HF} is the HF density matrix and V_{nuc} is the nuclear repulsion energy.

Equation (6.1) 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 ($i, j, k, \ldots$) with an orbital unoccupied in the ground state ($a, b, c, \ldots$). Such wavefunctions and energies can be written

$$\Psi_i^a = \frac{1}{\sqrt{n!}} \det \{\chi_1 \chi_2 \cdots \chi_a \chi_j \cdots \chi_n\} \tag{6.7}$$

$$E_{ia} = E_{\mathrm{HF}} + \varepsilon_a - \varepsilon_i - (ia \mid\mid ia) \tag{6.8}$$

where we have introduced the anti-symmetrized two-electron integrals in the molecular orbital basis

$$(pq \mid\mid rs) = \sum_{\mu\upsilon\lambda\sigma} c_{\mu p} c_{\upsilon q} c_{\lambda r} c_{\sigma s} \left(\mu\lambda \mid\mid \upsilon\sigma\right) \tag{6.9}$$

These singly excited wavefunctions and energies could be considered crude approximations to the excited states of the system. However, determinants of the form Eq. (6.7) 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 wavefunction as a linear combination of *all* possible singly excited determinants,

$$\Psi_{\mathrm{CIS}} = \sum_{ia} a_i^a \Psi_i^a \tag{6.10}$$

where the coefficients $\{a_{ia}\}$ can be obtained by diagonalizing the many-electron Hamiltonian, **A**, in the space of all single substitutions. The appropriate matrix elements are:

$$A_{ia,jb} = \langle \Psi_i^a | \, H \left| \Psi_j^b \right\rangle = \left[E_{\mathrm{HF}} + \varepsilon_a - \varepsilon_j \right] \delta_{ij} \delta_{ab} - (ja \, || \, ib) \tag{6.11}$$

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 **A** correspond to the amplitudes of the single-electron promotions.

CIS calculations can be performed in Q-CHEM using restricted (RCIS) [3, 4], unrestricted (UCIS), or restricted open shell (ROCIS) [5] spin orbitals.

6.2.2 Random Phase Approximation (RPA)

The Random Phase Approximation (RPA) [6, 7] also known as time-dependent Hartree-Fock (TD-HF) is an alternative to CIS for uncorrelated calculations of excited states. It offers some advantages for computing oscillator strengths, and is roughly comparable in accuracy to CIS for excitation energies to singlet states, but is inferior for triplet states. RPA energies are non-variational.

6.2.3 Extended CIS (XCIS)

The motivation for the extended CIS procedure (XCIS) [8] stems from the fact that ROCIS and UCIS are less effective for radicals that CIS is for closed shell molecules. Using the attachment/detachment density analysis procedure [9], 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

$$\left| \tilde{\Psi}_i^a(1) \right\rangle = \frac{1}{\sqrt{6}} \left(|\Psi_{\bar{i}}^{\bar{a}}\rangle - |\Psi_i^a\rangle \right) + \frac{2}{\sqrt{6}} |\Psi_{p\bar{i}}^{a\bar{p}}\rangle \tag{6.12}$$

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. (6.12). 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 wavefunction is

$$|\Psi_{\mathrm{ROCIS}}\rangle = \frac{1}{\sqrt{2}} \sum_{ia}^{dv} a_i^a \left(|\Psi_i^a\rangle + |\Psi_{\bar{i}}^{\bar{a}}\rangle \right) + \sum_{pa}^{sv} a_p^a |\Psi_p^a\rangle + \sum_{ip}^{ds} a_{\bar{i}}^{\bar{p}} |\Psi_{\bar{i}}^{\bar{p}}\rangle \tag{6.13}$$

The extension of CIS theory to incorporate higher excitations maintains the ROHF as the ground state reference and adds terms to the ROCIS wavefunction similar to that of Eq. (6.13), as well as those where the double excitation occurs through different orbitals in the α and β space:

$$\begin{aligned} |\Psi_{\mathrm{XCIS}}\rangle = {} & \tfrac{1}{\sqrt{2}} \sum_{ia}^{dv} a_i^a \left(|\Psi_i^a\rangle + |\Psi_{\bar{i}}^{\bar{a}}\rangle \right) + \sum_{pa}^{sv} a_p^a |\Psi_p^a\rangle + \sum_{ip}^{ds} a_{\bar{i}}^{\bar{p}} |\Psi_{\bar{i}}^{\bar{p}}\rangle \\ & + \sum_{iap}^{dvs} \tilde{a}_i^a(p) |\tilde{\Psi}_i^a(p)\rangle + \sum_{ia,p \neq q}^{dv,ss} a_{p\bar{i}}^{a\bar{q}} |\Psi_{p\bar{i}}^{a\bar{q}}\rangle \end{aligned} \tag{6.14}$$

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 wavefunction. 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 wavefunction 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.

6.2.4 Spin-Flip Extended CIS (SF-XCIS)

Spin-flip extended CIS (SF-XCIS) [10] is a spin-complete extension of the spin-flip single excitation configuration interaction (SF-CIS) method [11]. 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 wavefunctions 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 prefactor 13 times larger.

6.2.5 Basic Job Control Options

See also JOBTYPE, BASIS, EXCHANGE and CORRELATION. EXCHANGE must be HF and CORRELATION must be NONE. The minimum input required above a ground state HF calculation is to specify a nonzero value for CIS_N_ROOTS.

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

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:

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

SPIN_FLIP_XCIS

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 customer 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 customer 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

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

6.2.6 Customization

N_FROZEN_CORE

Controls the number of frozen core orbitals

TYPE:

INTEGER

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 Sec. 6.6).

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 $10^{-\text{CIS\_CONVERGENCE}}$

TYPE:

INTEGER

DEFAULT:

6 CIS convergence threshold 10^{-6}

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

None

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) to hold a temporary array whose minimum size is OV × 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

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 the CIS and TDDFT calculations (including SF-CIS and SF-DFT).

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.

Note: This option is not available for SF-XCIS.

CIS_MOMENTS

Controls calculation of excited-state (CIS or TDDFT) multipole moments

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE (or 0)

OPTIONS:

FALSE (or 0) Do not calculate excited-state moments.

TRUE (or 1) 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.

6.2.7 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 [12–14]. 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 [15, 16], to yield analytical gradients, excited state vibrational frequencies, force constants, polarizabilities, and infrared intensities. Their evaluation is controlled by two *$rem* variables, listed below. Analytical gradients can be evaluated for any job where the CIS excitation energy calculation itself is feasible.

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:

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 no change order during an optimization

The semi-direct method [8] used to evaluate the frequencies is generally similar to the semi-direct method used to evaluate Hartree-Fock frequencies for the ground state. Memory and disk requirements (see below) are similar, and the computer time scales approximately as the cube of the system size for large molecules.

The main complication associated with running analytical CIS second derivatives is ensuring Q-CHEM has sufficient memory to perform the calculations. For most purposes, the defaults will be adequate, but if a large calculation fails due to a memory error, then the following additional information may be useful in fine tuning the input, and understanding why the job failed. Note that the analytical CIS second derivative code does not currently support frozen core or virtual orbitals (unlike Q-CHEM's MP2 code). Unlike MP2 calculations, applying frozen core/virtual orbital approximations does not lead to large computational savings in CIS calculations as all computationally expensive steps are performed in the atomic basis.

The memory requirements for CIS (and HF) analytical frequencies are primarily extracted from "C" memory, which is defined as

"C" memory = MEM_TOTAL - MEM_STATIC

"C" memory must be large enough to contain a number of arrays whose size is $3 \times N_{\rm Atoms} \times N^2_{\rm Basis}$ ($N_{\rm Atoms}$ is the number of atoms and $N_{\rm Basis}$ refers to the number of basis functions). The value of the *$rem* variable MEM_STATIC should be set sufficiently large to permit efficient integral evaluation. If too large, it reduces the amount of "C" memory available. If too small, the job may fail due to insufficient scratch space. For most purposes, a value of about 80 Mb is sufficient, and by default MEM_TOTAL is set to a very large number (larger than physical memory on most computers) and thus *malloc* (memory allocation) errors may occur on jobs where the memory demands exceeds physical memory.

6.2.8 Examples

Example 6.1 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
```

Example 6.2 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
```

Example 6.3 A SF-XCIS calculation of ground and excited states of trimethylenemethane (TMM) diradical, for which double substitutions make considerable contributions to low-lying exc ited states.

```
$comment
   TMM ground and excited states
$end

$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
```

Example 6.4 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.

```
$comment
   singlet n -> pi* state optimization and frequencies for formaldehyde
$end

$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
```

6.2.9 Non-Orthogonal Configuration Interaction

Some systems such as transition metals, open-shell species, and dissociating molecules feature multiple closely lying solutions to the SCF equations. By using SCF Metadynamics (see Chapter 4), these can be successfully located, however, there is no clear physical reason to choose one SCF solution over another, rather, it is appropriate to have a method that treats these solutions on an equal footing. In particular, these SCF solutions are not subject to non-crossing rules, and do in fact often cross each other as geometry is changed, so the lowest energy state may switch abruptly with consequent discontinuities in the energy gradients. To achieve a smoother, more qualitatively correct surface, these SCF solutions can be used as a basis for a Configuration Interaction calculation, where the resultant wavefunction will either smoothly interpolate between these states. As the SCF states are not orthogonal to each other (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 denoted Non-Orthogonal Configuration Interaction (NOCI) [17].

This can be viewed as an alternative to CASSCF within an "active space" of SCF states of interest, and has the advantage that the SCF states, and thus the NOCI wavefunctions 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), and conical intersections (where through symmetry or 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 purification. 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 Hamiltonian, its true eigenfunctions will preserve these symmetries. In the case of spin, this manifests itself as spin-contamination, and for spatial symmetries, the orbitals will usually adopt a more localized structure. To recover a (yet lower energy) wavefunction retaining the correct symmetries, one can include these symmetry broken states (with all relevant symmetry permutations) in a NOCI calculation, and the resultant eigenfunction will have the true symmetries restored as a linear combination of these broken symmetry states. A common example would be for a UHF state which has an indefinite spin (value of S not M_s). By including a UHF solution along with its spin-flipped version (where all alpha and beta orbitals have been switched) in NOCI, the resultant wavefunction will be a more pure spin state (though there is still no guarantee of finding an eigenfunction of $\hat{S}^2$), reducing spin contamination in the same way as the Half-Projected Hartree-Fock method [18]. As an example using an $M_s = 0$ UHF wavefunction and its spin-flipped version will produce two new NOCI eigenfunctions, one with even S (a mixture of $S = 0$, $S = 2$, $\ldots$), and one with odd (mixing $S = 1$, $S = 3$,$\ldots$), which may be use as approximations to singlet and triplet wavefunctions.

NOCI can be enabled by specifying CORRELATION NOCI, and will automatically use all of the states located with SCF metadynamics. To merely include the two spin-flipped versions of a UHF wavefunction, this can be specified without turning metadynamics on. For more customization, a *$noci* section can be included in the input file to specify the states to include:

Example 6.5 *$noci* section example

```
$noci
   1 2 -2 4
   2
$end
```

This section specifies (first line) that states 1,2, and 4 are to be included as well as the spin-flipped version of state 2 (the -2 indicates this). The second line (optional) indicates which (zero-based) eigenvalue is to be returned to Q-CHEM (the third in this case). Analytic gradients are not available for NOCI, but finite difference geometry optimizers are available.

NOCI_PRINT
Specify the debug print level of NOCI
TYPE:
INTEGER
DEFAULT:
1
OPTIONS:

RECOMMENDATION:
Increase this for more debug information

6.3 Time-Dependent Density Functional Theory (TDDFT)

6.3.1 Brief Introduction to TDDFT

Excited states may be obtained from density functional theory by time-dependent density functional theory [19, 20], which calculates poles in the response of the ground state density to a time-varying applied electric field. These poles are Bohr frequencies or excitation energies, and are available in Q-CHEM [21], together with the CIS-like Tamm-Dancoff approximation [22]. TDDFT is becoming very popular as a method for studying excited states because the computational cost is roughly similar to the simple CIS method (scaling as roughly the square of molecular size), but a description of differential electron correlation effects is implicit in the method. The excitation energies for low-lying valence excited states of molecules (below the ionization threshold, or more conservatively, below the first Rydberg threshold) are often remarkably improved relative to CIS, with an accuracy of roughly 0.1–0.3 eV being observed with either gradient corrected or local density functionals.

However, standard density functionals do not yield a potential with the correct long-range Coulomb tail, owing to the so-called self-interaction problem, and 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 [23–25]. The extent to which a particular excited state is characterized by charge transfer can be assessed using an a spatial overlap metric proposed by Peach, Benfield, Helgaker, and Tozer (PBHT) [26]. (However, see Ref. 27 for a cautionary note regarding this metric.)

It is advisable to only employ TDDFT for low-lying valence excited states that are below the first ionization potential of the molecule. This makes radical cations a particularly favorable choice of system, as exploited in Ref. 28. TDDFT for low-lying valence excited states of radicals is in general a remarkable improvement relative to CIS, including some states, that, when treated by wavefunction-based methods can involve a significant fraction of double excitation character [21]. The calculation of the nuclear gradients of full TDDFT and within the Tamm-Dancoff approximation is also implemented [29].

Standard TDDFT also does not yield a good description of static correlation effects (see Section 5.9), because it is based on a single reference configuration of Kohn-Sham orbitals. Recently, a new variation of TDDFT called spin-flip density functional theory (SFDFT) was developed by Yihan Shao, Martin Head-Gordon and Anna Krylov to address this issue [30]. SFDFT 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-DFT 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. Recently, we also implemented[31] a SFDFT method with a non-collinear exchange-correlation potential from Tom Ziegler *et al.* [32, 33], which is in many case an improvement over collinear SFDFT [30]. Recommended functionals for SF-DFT calculations are 5050 and PBE50 (see Ref. [31] for extensive benchmarks). See also Section 6.7.3 for details on wavefunction-based spin-flip models.

6.3.2 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 [34]. 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. 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 [35], CO on the Pt(111) surface [36], and the tryptophan chromophore within proteins [37].

6.3.3 Job Control for TDDFT

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 6.2.5). 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.
- If RPA is set to TRUE, a full TDDFT calculation will be performed. This is not the default. The default is RPA = FALSE, which leads to a calculation employing the Tamm-Dancoff approximation (TDA), which is usually a good approximation to full TDDFT.
- If SPIN_FLIP is set to TRUE when performing a TDDFT calculation, a SFDFT calculation will also be performed. At present, SFDFT is only implemented within TDDFT/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).

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

N_SOL

Specifies number of atoms or orbitals in $\$solute$

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

User defined

RECOMMENDATION:

None

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: CUTOFF=CUTOCC/100

DEFAULT:

50

OPTIONS:

0-200

RECOMMENDATION:

None

CUTVIR

Specifies virtual orbital cutoff

TYPE:

INTEGER: CUTOFF=CUTVIR/100

DEFAULT:

0 No truncation

OPTIONS:

0-100

RECOMMENDATION:

None

PBHT_ANALYSIS

Controls whether overlap analysis of electronic excitations is performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Do not perform overlap analysis
TRUE	Perform overlap analysis

RECOMMENDATION:

None

PBHT_FINE

Increases accuracy of overlap analysis

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	
TRUE	Increase accuracy of overlap analysis

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

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

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}=∞.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to HF_LR = $n/1000$

RECOMMENDATION:

None

WANG_ZIEGLER_KERNEL

Controls whether to use the Wang-Ziegler non-collinear exchange-correlation kernel in a SFDFT calculation.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Do not use non-collinear kernel
TRUE	Use non-collinear kernel

RECOMMENDATION:

None

6.3.4 TDDFT coupled with C-PCM for excitation energies and properties calculations

As described in Section 11.2 (and especially Section 11.2.2), continuum solvent models such as C-PCM allow one to include solvent effect in the calculations. TDDFT/C-PCM allows excited-state modeling in solution. Q-CHEM also features TDDFT coupled with C-PCM which extends TDDFT to calculations of properties of electronically-excited molecules in solution. In particular, TDDFT/C-PCM allows one to perform geometry optimization and vibrational analysis [38].

When TDDFT/C-PCM is applied to calculate vertical excitation energies, the solvent around vertically excited solute is out of equilibrium. While the solvent electron density equilibrates fast to the density of the solute (electronic response), the relaxation of nuclear degrees of freedom (e.g., orientational polarization) takes place on a slower timescale. To describe this situation, an optical dielectric constant is employed. To distinguish between equilibrium and non-equilibrium calculations, two dielectric constants are used in these calculations: a static constant (ε_0), equal to the equilibrium bulk value, and a fast constant (ε_{fast}) related to the response of the medium to high frequency perturbations. For vertical excitation energy calculations (corresponding to the unrelaxed solvent nuclear degrees of freedom), it is recommended to use the optical dielectric constant for ε_{fast}), whereas for the geometry optimization and vibrational frequency calculations, the static dielectric constant should be used [38].

The example below illustrates TDDFT/C-PCM calculations of vertical excitation energies. More information concerning the C-PCM and the various PCM job control options can be found in Section 11.2.

Example 6.6 TDDFT/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*
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
```

6.3.5 Analytical Excited-State Hessian in TDDFT

To carry out vibrational frequency analysis of an excited state with TDDFT [39, 40], 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 analytical 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 analytical 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.

Analytical 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. At present, only a few exchange correlation functionals, including Slater+VWN, BLYP, B3LYP, are available for the analytical Hessian calculation.

Example 6.7 A 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
RPA 0
xc_grid 000075000302
$end
```

Example 6.8 The optimization and Hessian calculation for low-lying excited state with TDDFT/C-PCM

```
$comment
9-Fluorenone + 2 methanol in methanol solution
$end

$molecule
0 1
          6   -1.987249    0.699711    0.080583
          6   -1.987187   -0.699537   -0.080519
          6   -0.598049   -1.148932   -0.131299
          6    0.282546    0.000160    0.000137
          6   -0.598139    1.149219    0.131479
          6   -0.319285   -2.505397   -0.285378
          6   -1.386049   -3.395376   -0.388447
          6   -2.743097   -2.962480   -0.339290
          6   -3.049918   -1.628487   -0.186285
          6   -3.050098    1.628566    0.186246
          6   -2.743409    2.962563    0.339341
          6   -1.386397    3.395575    0.388596
          6   -0.319531    2.505713    0.285633
          8    1.560568    0.000159    0.000209
          1    0.703016   -2.862338   -0.324093
          1   -1.184909   -4.453877   -0.510447
          1   -3.533126   -3.698795   -0.423022
          1   -4.079363   -1.292006   -0.147755
          1    0.702729    2.862769    0.324437
          1   -1.185378    4.454097    0.510608
          1   -3.533492    3.698831    0.422983
          1   -4.079503    1.291985    0.147594
          8    3.323150    2.119222    0.125454
          1    2.669309    1.389642    0.084386
          6    3.666902    2.489396   -1.208239
          1    4.397551    3.298444   -1.151310
          1    4.116282    1.654650   -1.759486
          1    2.795088    2.849337   -1.768206
          1    2.669205   -1.389382   -0.084343
          8    3.322989   -2.119006   -0.125620
          6    3.666412   -2.489898    1.207974
          1    4.396966   -3.299023    1.150789
          1    4.115800   -1.655485    1.759730
          1    2.794432   -2.850001    1.767593
$end

$rem
jobtype              OPT
EXCHANGE             B3lyp
CIS_N_ROOTS          10
cis_singlets         true
cis_triplets         true
cis_state_deriv      1   Lowest TDDFT state
RPA                  TRUE
BASIS                6-311G**
XC_GRID              000075000302
solvent_method       pcm
$end

$pcm
Theory      CPCM
Method      SWIG
Solver      Inversion
Radii       Bondi
$end

$solvent
Dielectric 32.613
$end

@@@

$molecule
read
$end

$rem
jobtype              freq
EXCHANGE             B3lyp
CIS_N_ROOTS          10
cis_singlets         true
cis_triplets         true
RPA                  TRUE
cis_state_deriv      1   Lowest TDDFT state
BASIS                6-311G**
XC_GRID              000075000302
solvent_method       pcm
mem_static           4000
mem_total            24000
cpscf_nseg    3
$end

$pcm
Theory      CPCM
Method      SWIG
Solver      Inversion
Radii       Bondi
$end
```

```
$solvent
Dielectric 32.613
$end
```

6.3.6 Various TDDFT-Based Examples

Example 6.9 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
   RPA               true
   BASIS             6-31+G*
   SCF_CONVERGENCE   7
$end
```

Example 6.10 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
$end

$solute
1
2
3
4
5
6
$end
```

Example 6.11 This example shows a calculation of the core-excited states at the oxygen K-edge of CO with a short-range corrected functional.

```
$comment
   TDDFT with short-range corrected (SRC1) functional for the
   oxygen K-edge of CO
$end

$molecule
0 1
C           0.000000    0.000000   -0.648906
O           0.000000    0.000000    0.486357
$end

$rem
exchange gen
basis 6-311(2+,2+)G**
cis_n_roots 6
cis_triplets false
trnss true
trtype 3
```

```
n_sol 1
src_dft 1
omega  560
omega2 2450
HF_SR  500
HF_LR  170
$end

$solute
1
$end

$XC_Functional
X HF    1.00
X B     1.00
C LYP   0.81
C VWN   0.19
$end
```

Example 6.12 This example shows a calculation of the core-excited states at the phosphorus K-edge with a short-range corrected functional.

```
$comment
   TDDFT with short-range corrected (SRC2) functional for the
   phosphorus K-edge of PH3
$end

$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 gen
basis 6-311(2+,2+)G**
cis_n_roots 6
cis_triplets false
trnss true
trtype 3
n_sol 1
src_dft 2
omega  2200
omega2 1800
HF_SR  910
HF_LR  280
$end

$solute
1
$end

$XC_Functional
X HF    1.00
X B     1.00
```

```
C LYP   0.81
C VWN   0.19
$end
```

Example 6.13 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
jobtype            SP
EXCHANGE           GENERAL            Exact exchange
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 0.5
 X S  0.08
 X B  0.42
 C VWN 0.19
 C LYP 0.81
$end
```

Example 6.14 SFDFT with non-collinear exchange-correlation functional for low-lying states of CH_2

```
$comment
non-collinear SFDFT 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
JOBTYPE             SP
UNRESTRICTED        TRUE
EXCHANGE            PBE0
BASIS               cc-pVTZ
SPIN_FLIP           1
WANG_ZIEGLER_KERNEL TRUE
SCF_CONVERGENCE     10
CIS_N_ROOTS         6
CIS_CONVERGENCE     10
$end
```

6.4 Maximum Overlap Method (MOM) for SCF Excited States

The Maximum Overlap Method (MOM) [41] is a useful alternative to CIS and TDDFT for obtaining low-cost excited states. 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, excited SCF determinants are able to be obtained. The MOM has several advantages over existing low-cost excited state methods. Current implementations of TDDFT usually struggle to accurately model charge-transfer and Rydberg transitions, both of which can be well-modeled using the MOM. The MOM also allows the user to target very high energy states, such as those involving excitation of core electrons [42], which are hard to capture using other excited state methods.

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.

Example 6.15 Calculation of the lowest singlet state of CO.

```
$comment
   CO spin-purified calculation
$end

$molecule
0 1
C
O C 1.05
$end

$rem
   JOBTYPE            SP
   METHOD             B3LYP
   BASIS              6-31G*
$end

@@@

$molecule
read
$end

$rem
   JOBTYPE            SP
   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
```

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 using SCF_GUESS=true and MOM_START set to 1.

Example 6.16 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 2 123.107874
N  1 1.359042 2 111.982794 3 -180.000000 0
H  4 0.996369 1 121.060099 2   -0.000000 0
H  4 0.998965 1 119.252752 2 -180.000000 0
$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 2 123.107874
N  1 1.359042 2 111.982794 3 -180.000000 0
H  4 0.996369 1 121.060099 2   -0.000000 0
H  4 0.998965 1 119.252752 2 -180.000000 0
$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
```

6.5 Restricted Open-Shell Kohn-Sham Method for Δ-SCF Calculations of Excited States

Q-CHEM provides access to certain singlet excited states – namely, those well-described by a single-electron HOMO-LUMO transition – via restricted open-shell Kohn-Sham (ROKS) theory. In contrast to the MOM approach (see Section 6.4), which requires separate SCF calculations of the non-Aufbau and triplet energies, the ROKS approach attempts to combine the properties of both determinants at the level of the Fock matrix in one SCF calculation. ROKS thus presents as a single SCF loop, but the structure of the

Fock matrix differs from the ground-state case. Note that this excited-state method is distinct from ROKS theory for open-shell ground states.

The implementation of ROKS excited states in Q-CHEM largely follows the theoretical framework established by Filatov and Shaik[43] and is described in detail in Ref. [44]. Singlet excited state energies and gradients are available, enabling single-point, geometry optimization and molecular dynamics.

To perform an ROKS excited state calculation, simply set the keywords ROKS and UNRESTRICTED both to TRUE (the latter sounds as a contradiction to the method's name, but yes, this is what you should do). An additional keyword ROKS_LEVEL_SHIFT is included to assist in cases of convergence difficulties with a standard level-shift technique. 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. You should also set UNRESTRICTED=TRUE

ROKS_LEVEL_SHIFT
Introduce a level shift of N/100 Hartree to aid convergence.
TYPE:
INTEGER
DEFAULT:
0
OPTIONS:

0	No shift
N	level shift of N/100 Hartree.

RECOMMENDATION:
Use in cases of problematic convergence.

Example 6.17 RO-PBE0/6-311+G* excited state gradient of formaldehyde, using the ground state orbitals as an initial guess.

```
$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
  sym_ignore             true
$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
  sym_ignore              true
  scf_guess               read
$end
```

6.6 Correlated Excited State Methods: the CIS(D) Family

CIS(D) [45, 46] 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. 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 5.5) [47].

6.6.1 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 wavefunction, truncated at single and double excitations, is the exponential of the single and double substitution operators acting on the Hartree-Fock determinant:

$$|\Psi\rangle = \exp\left(T_1 + T_2\right)|\Psi_0\rangle \tag{6.15}$$

Determination of the singles and doubles amplitudes requires solving the two equations

$$\left\langle \Psi_i^a \right| H - E \left| \left(1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1 T_2 + \frac{1}{3!}T_1^3\right)\Psi_0 \right\rangle = 0 \tag{6.16}$$

and

$$\left\langle \Psi_{ij}^{ab} \right| H - E \left| \left(1 + T_1 + T_2 + \frac{1}{2}T_1^2 + T_1 T_2 + \frac{1}{3!}T_1^3 + \frac{1}{2}T_2^2 + \frac{1}{2}T_1^2 T_2 + \frac{1}{4!}T_1^4\right)\Psi_0 \right\rangle = 0 \tag{6.17}$$

which lead to the CCSD excited state equations. These can be written

$$\left\langle \Psi_i^a \right| H - E \left| \left(U_1 + U_2 + T_1 U_1 + T_1 U_2 + U_1 T_2 + \frac{1}{2} T_1^2 U_1 \right) \Psi_0 \right\rangle = \omega b_i^a \tag{6.18}$$

and

$$\begin{aligned} \langle \Psi_i^a | \, H - E \, \big| \big(U_1 + U_2 + T_1 U_1 + T_1 U_2 + U_1 T_2 + \tfrac{1}{2} T_1^2 U_1 \; + T_2 U_2 \\ + \tfrac{1}{2} T_1^2 U_2 + T_1 T_2 U_1 + \tfrac{1}{3!} T_1^3 U_1 \big| \; \Psi_0 \rangle = \omega b_{ij}^{ab} \end{aligned} \tag{6.19}$$

This is an eigenvalue equation $\mathbf{A}\mathbf{b} = \omega\mathbf{b}$ 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

$$\omega^{(2)} = \mathbf{b}^{(0)^{\mathbf{t}}} \mathbf{A}^{(1)} \mathbf{b}^{(1)} + \mathbf{b}^{(0)^{\mathbf{t}}} \mathbf{A}^{(2)} \mathbf{b}^{(0)} \tag{6.20}$$

or in the alternative form

$$\omega^{(2)} = \omega^{\mathrm{CIS(D)}} = E^{\mathrm{CIS(D)}} - E^{\mathrm{MP2}} \tag{6.21}$$

where

$$E^{\mathrm{CIS(D)}} = \left\langle \Psi^{\mathrm{CIS}} \right| V \left| U_2 \Psi^{\mathrm{HF}} \right\rangle + \left\langle \Psi^{\mathrm{CIS}} \right| V \left| T_2 U_1 \Psi^{\mathrm{HF}} \right\rangle \tag{6.22}$$

and

$$E^{\mathrm{MP2}} = \left\langle \Psi^{\mathrm{HF}} \right| V \left| T_2 \Psi^{\mathrm{HF}} \right\rangle \tag{6.23}$$

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" [48]. 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.

6.6.2 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).

6.6.3 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:

$$E^{\mathrm{SOS-CIS(D)}} = c_U \left\langle \Psi^{\mathrm{CIS}} \right| V \left| U_2^{\mathrm{OS}} \Psi^{\mathrm{HF}} \right\rangle + c_T \left\langle \Psi^{\mathrm{CIS}} \right| V \left| T_2^{\mathrm{OS}} U_1 \Psi^{\mathrm{HF}} \right\rangle \tag{6.24}$$

with the corresponding ground state energy

$$E^{\mathrm{SOS-MP2}} = c_T \left\langle \Psi^{\mathrm{HF}} \right| V \left| T_2^{\mathrm{OS}} \Psi^{\mathrm{HF}} \right\rangle \tag{6.25}$$

More importantly, this SOS-CIS(D) energy can be evaluated with the 4th power of the molecular size by adopting Laplace transform technique [47]. Accordingly, SOS-CIS(D) can be applied to the calculations of excitation energies for relatively large molecules.

6.6.4 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 6.6.1 [49].

First, starting from Eq. (6.20), one can be explicitly write the CIS(D) energy as [49, 50]

$$\omega^{\mathrm{CIS}} + \omega^{(2)} = \mathbf{b^{(0)^t}} \mathbf{A}_{\mathrm{SS}}^{(0)} \mathbf{b^{(0)}} + \mathbf{b^{(0)^t}} \mathbf{A}_{\mathrm{SS}}^{(2)} \mathbf{b^{(0)}} - \mathbf{b^{(0)^t}} \mathbf{A}_{\mathrm{SD}}^{(1)} \left(\mathbf{D}_{\mathrm{DD}}^{(0)} - \omega^{\mathrm{CIS}} \right)^{-1} \mathbf{A}_{\mathrm{DS}}^{(1)} \mathbf{b^{(0)}} \tag{6.26}$$

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 [49]:

$$\left(\mathbf{D}_{\mathrm{DD}}^{(0)} - \omega^{\mathrm{CIS}} \right)^{-1} = \left(\mathbf{D}_{\mathrm{DD}}^{(0)} \right)^{-1} \left(1 + \omega \left(\mathbf{D}_{\mathrm{DD}}^{(0)} \right)^{-1} + \omega^2 \left(\mathbf{D}_{\mathrm{DD}}^{(0)} \right)^{-2} + ... \right) \tag{6.27}$$

The CIS(D_0) energy ω is defined as the eigensolution of the response matrix with the zero-th order expansion of this equation. Namely,

$$\left(\mathbf{A}_{\mathrm{SS}}^{(0)} + \mathbf{A}_{\mathrm{SS}}^{(2)} - \mathbf{A}_{\mathrm{SD}}^{(1)} (\mathbf{D}_{\mathrm{DD}}^{(0)})^{-1} \mathbf{A}_{\mathrm{DS}}^{(1)} \right) \mathbf{b} = \omega \mathbf{b} \tag{6.28}$$

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 [50]:

$$\left(\mathbf{A}_{\mathrm{SS}}^{(0)} + c_T \mathbf{A}_{\mathrm{SS}}^{\mathrm{OS}(2)} - c_U \mathbf{A}_{\mathrm{SD}}^{\mathrm{OS}(1)} (\mathbf{D}_{\mathrm{DD}}^{(0)})^{-1} \mathbf{A}_{\mathrm{DS}}^{\mathrm{OS}(1)} \right) \mathbf{b} = \omega \mathbf{b} \tag{6.29}$$

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.

6.6.5 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 6.6.6.

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 5.7.7).

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* 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.

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 6.18 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
JOBTYPE            SP
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 two lowest triplets in each irrep.
$end
```

Example 6.19 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 (first state in first irrep)
$end
```

Example 6.20 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
jobtype            SP
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
$end
```

6.6.6 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 6.2.1. As these methods rely on the RI approximation, AUX_BASIS needs to be set by following the same guide as in RI-MP2 (Section 5.5).

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

SET_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:

64 corresponding to 64 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.

AO2MO_DISK

Sets the scratch space size for individual program modules

TYPE:

INTEGER

DEFAULT:

2000 2 Gb

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

The minimum disk requirement of RI-CIS(D) is approximately $3SOVXD$. Again, the batching scheme will become more efficient with more available disk space. There is no simple formula for SOS-CIS(D) and SOS-CIS(D_0) disk requirement. However, because the disk space is abundant in modern computers, this should not pose any problem. Just put the available disk space size in this case. The actual disk usage information will also be printed in the output file.

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

6.6.7 Examples

Example 6.21 Q-CHEM 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
   jobtype                 sp
   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 6.22 Q-CHEM 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
   jobtype                 sp
   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 6.23 Q-CHEM 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
```

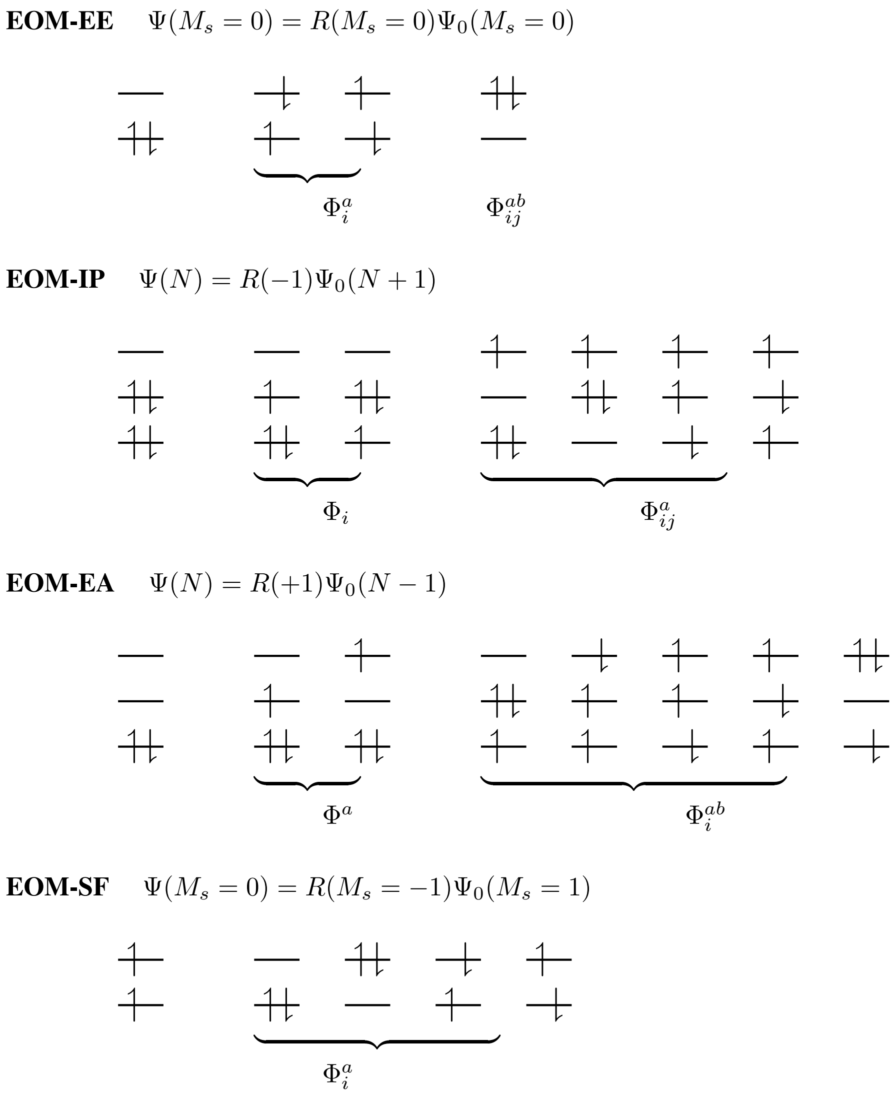
6.7 Coupled-Cluster Excited-State and Open-Shell Methods

6.7.1 Excited States via EOM-EE-CCSD and EOM-EE-OD

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 [51] or equation-of-motion methods [52]. 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 the EOM(2,3) or EOM-CCSD(dT)/(fT) methods.

Q-CHEM includes coupled-cluster methods for excited states based on the optimized orbital coupled-cluster doubles (OD), and the coupled cluster singles and doubles (CCSD) methods, described earlier. OD excitation energies have been shown to be essentially identical in numerical performance to CCSD excited states [53].

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 tri-radicals.



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 6.1: 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 hight-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.

6.7.2 EOM-XX-CCSD and CI Suite of Methods

Q-CHEM features the most complete set of EOM-CCSD models [54] that enables accurate, robust, and efficient calculations of electronically excited states (EOM-EE-CCSD or EOM-EE-OD) [52, 53, 55–57]; ground and excited states of diradicals and triradicals (EOM-SF-CCSD and EOM-SF-OD [57, 58]); ionization potentials and electron attachment energies as well as problematic doublet radicals, cation or anion radicals, (EOM-IP/EA-CCSD) [59–61], as well as EOM-DIP-CCSD and EOM-2SF-CCSD. Conceptually, EOM is very similar to configuration interaction (CI): target EOM states are found by diagonalizing the similarity transformed Hamiltonian $\bar{H} = e^{-T}He^{T}$,

$$\bar{H}R = ER, \tag{6.30}$$

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 $|\Phi_0\rangle$:

$$\langle\Phi_i^a|\bar{H}|\Phi_0\rangle = 0 \tag{6.31}$$

$$\langle\Phi_{ij}^{ab}|\bar{H}|\Phi_0\rangle = 0 \tag{6.32}$$

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. 6.1. For example, electronically excited states can be described when the reference $|\Phi_0\rangle$ corresponds to the ground state wave function, and operators R conserve the number of electrons and a total spin [52]. In the ionized/electron attached EOM models [60, 61], 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 [57, 58] in which the excitation operators include spin-flip allows one to access diradicals, triradicals, and bond-breaking.

Q-CHEM features EOM-EE/SF/IP/EA-CCSD methods for both closed and open-shell references (RHF/UHF/ROHF), including frozen core/virtual options. 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 [62]. Properties calculations (permanent and transition dipole moments, $\langle S^2\rangle$, $\langle R^2\rangle$, *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 5.7.5 or Cholesky decomposition 5.7.6 helps to reduce integral transformation time and disk usage enabling calculations on much larger systems.

Q-CHEM includes two implementations of EOM-IP-CCSD. The proper implementation [63] 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. Our current implementation of EOM-EA-CCSD also uses this trick. Fake IP/EA calculations are only recommended for Dyson orbital calculations and debug purposes.

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 6.7.7).

Finally, a more economical CI variant of EOM-IP-CCSD, IP-CISD is also available. This is an N^5 approximation of IP-CCSD, and is recommended for geometry optimizations of problematic doublet states [64].

EOM and CI methods are handled by the CCMAN/CCMAN2 modules.

6.7.3 Spin-Flip Methods for Di- and Triradicals

The spin-flip method [11, 58, 65] addresses the bond-breaking problem associated with a single-determinant description of the wavefunction. Both closed and open shell singlet states are described within a single reference as spin-flipping, (*e.g.*, $\alpha \to \beta$ 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 wavefunction 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 6.2.4). An 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 ($\langle S^2 \rangle$), and densities. Analytic gradients are also for SF-CIS and EOM-SF-CCSD methods. To invoke a spin-flip calculation the EOM_SF_STATES *$rem* should be used, along with the associated *$rem* settings for the chosen level of correlation (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 [66]. To invoke these methods, use
EOM_DSF_STATES.

6.7.4 EOM-DIP-CCSD

Double-ionization potential (DIP) is another non-electron-conserving variant of EOM-CCSD [67–69]. In DIP, target states are reached by detaching two electrons from the reference state:

$$\Psi_k = R_{N-2}\Psi_0(N+2), \tag{6.33}$$

and the excitation operator R has the following form:

$$\begin{aligned}
R &= R_1 + R_2, && (6.34)\\
R_1 &= 1/2\sum_{ij} r_{ij} j i, && (6.35)\\
R_2 &= 1/6\sum_{ijka} r_{ijk}^{a} a^{\dagger} k j i. && (6.36)
\end{aligned}$$

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 wavefunctions.

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 EOM_DIP_STATES, or EOM_DIP_SINGLETS and EOM_DIP_TRIPLETS.

6.7.5 EOM-CC Calculations of Metastable States: Super-Excited Electronic States, Temporary Anions, and Core-Ionized 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 [54], metastable 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 autoionizing states, such as transient anions, highly excited electronic states, and core-ionized species [70, 71]. In addition, users can employ stabilization techniques using charged sphere and scaled atomic charges options [69]. 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 will use spherical CAP)
$end
```

CS_THETA is specified in radian× 10^{-3}. CS_ALPHA, CAP_X/Y/Z are specified in a.u.× 10^{-3}, i.e., CS_THETA=10 means θ=0.01; CAP_ETA is specified in a.u.× 10^{-5}. When CAP calculations are performed, CC_EOM_PROP=TRUE by default; this is necessary for calculating first-order deperturbative correction.

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. 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. Finally, only RHF references are supported in complex-scaled and CAP-augmented calculations.

6.7.6 Charge Stabilization for EOM-DIP and Other Methods

Unfortunately, the performance of EOM-DIP deteriorates when the reference state is unstable with respect to electron-detachment [68, 69], which is usually the case for dianion references employed to describe neutral diradicals by EOM-DIP. 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, as described in Refs. [68, 69]. In this approach (which can also be used with any other electronic structure method implemented in Q-CHEM), an additional Coulomb potential is introduced to stabilize unstable wave functions. The following keywords invoke stabilization potentials: SCALE_NUCLEAR_CHARGE and ADD_CHARGED_CAGE. In the former case, the potential is generated by increasing nuclear charges by a specified amount. In the latter, the potential is generated by a cage built out of point charges comprising the molecule. There are two cages available: dodecahedral and spherical. The shape, radius, number of points, and the total charge of the cage are set by the user.

Note: A perturbative correction estimating the effect of the external Coulomb potential on EOM energy will be computed when target state densities are calculated, e.g., when CC_EOM_PROP is set to TRUE.

Note: Charge stabilization techniques can be used with other methods such as EOM-EE, CIS, and TDDFT to improve the description of resonances. It can also be employed to describe metastable ground states.

6.7.7 Frozen Natural Orbitals in CC and IP-CC Calculations

Large computational savings are possible if the virtual space is truncated using the frozen natural orbital (FNO) approach (see Section 5.10). Extension of the FNO approach to ionized states within EOM-CC formalism was recently introduced and benchmarked [72]. 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 5.10.

6.7.8 EOM-CC Jobs: Controlling Guess Formation and Iterative Diagonalizers

An EOM-CC eigen problem is solved by an iterative diagonalization procedure that avoids full diagonalization and only looks for several eigen-states, as specified by the XX_STATES keywords.

The default procedure is based on the modified Davidson diagonalization algorithm, as explained in Ref. [57]. 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 the 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), include doubly excited guess vectors (EOM_NGUESS_DOUBLES), and even preconverge them (EOM_PRECONV_SINGLES and EOM_PRECONV_DOUBLES).

If a state of a particular character is desired (e.g., HOMO→LUMO+10 excitation or 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. Note that the point group symmetry of user defined guess vectors should be consistent with the symmetry of the transition specified by XX_STATES. The multiplicity of the state is determined as in the regular calculations, by using the EOM_XX_SINGLETS and EOM_EE_TRIPLETS keywords. This option is useful for looking for high-lying states such as core-ionized states.

The examples below illustrate how to use user-specified guess in EOM calculations:

```
$eom_user_guess
4  Calculate excited state corresponding to 4(OCC)->5(VIRT) transition.
5
$end
```

In IP/EA calculations, only one orbital is specified:

```
$eom_user_guess
4
$end
```

If IP_STATES is specified, this will invoke calculation of the EOM-IP state corresponding to the ionization from 4th occupied MO. If EA_STATES is requested, then EOM-EA equations will be solved for a root corresponding to electron-attachment to the 4th virtual MO.

The symmetries of the MOs should be consistent with the EE_STATES value, which should only request one state in the correct irrep. The orbitals are ordered by energy as printed at the beginning of the CCMAN2 output.

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 the Koopmans theorem. This option is useful when highly excited states (i.e., interior eigenstates) are desired.

Finally, a new diagonalization technique, so-called GPLMR, is available[73]. This solver is engaged by EOM_GPLMR keyword and is only available in CCMAN2. The GPLMR method usually converges in fewer iterations relative to Davidson and can use less memory, but it performs more floating point operations. Similarly to the Davidson procedure, GPLMR can be applied to look for either the lowest eigenstates, or for an interior set, as specified by EOM_SHIFT. The convergence is controlled by the same keywords as in the Davidson algorithm. One additional keyword controlling the maximum subspace size in the GPLMR solver is EOM_GPLMR_MSUBSIZE.

6.7.9 Equation-of-Motion Coupled-Cluster 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 5.13).

Equation-of-motion methods require a coupled-cluster reference state, which is computed when METHOD is set to EOMCCSD or EOMOD. 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: In finite-difference calculations, the symmetry is turned off automatically, and the user must ensure that XX_STATES is adjusted accordingly.

Note: Mixing different EOM models in a single calculation is only allowed in Dyson orbitals calculations.

By default, the level of correlation of the EOM part of the wavefunction (*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.

The table below shows the correct combinations of CORRELATION and EOM_CORR for standard EOM and CI models.

The most relevant EOM-CC input options follow.

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(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 6.1: 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.

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_SNGLETS,EE_TRIPLETS
SF-CIS	CI	CIS	SF_STATES
CIS(D)	CI	CIS(D)	EE_STATES
			EE_SNGLETS,EE_TRIPLETS
SF-CIS(D)	CI	CIS(D)	SF_STATES
CISD	CI	CISD	EE_STATES
			EE_SNGLETS,EE_TRIPLETS
SF-CISD	CI	CISD	SF_STATES
IP-CISD	CI	CISD	IP_STATES
CISDT	CI	SDT	EE_STATES
			EE_SNGLETS,EE_TRIPLETS
SF-CISDT	CI	SDT or DT	SF_STATES
EOM-EE-CCSD	CCSD		EE_STATES
			EE_SNGLETS,EE_TRIPLETS
EOM-SF-CCSD	CCSD		SF_STATES
EOM-IP-CCSD	CCSD		IP_STATES
EOM-EA-CCSD	CCSD		EA_STATES
EOM-DIP-CCSD	CCSD		DIP_STATES
			DIP_SNGLETS,DIP_TRIPLETS
EOM-2SF-CCSD	CCSD	SDT or DT	DSF_STATES
EOM-EE-(2,3)	CCSD	SDT	EE_STATES
			EE_SNGLETS,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 6.2: 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

EOM_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

EOM_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, α electron will be attached (see EOM_EA_ALPHA).

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

EOM_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

EOM_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

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: 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.

TYPE:

LOGICAL

DEFAULT:

FALSE (use proper EOM-IP code)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None

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_GPLMR

Specifies whether to engage GPLMR solver in EOM calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use GPLMR.

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 (see also EOM_PRECONV_DOUBLES), 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.

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

N perform N Davidson iterations pre-converging singles.

RECOMMENDATION:

Sometimes helps with problematic convergence.

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 calculations instead of EOM_PRECONV_SINGLES

EOM_PRECONV_SD

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

N perform N Davidson iterations pre-converging singles and doubles.

RECOMMENDATION:

Occasionally necessary to ensure a doubly excited state is found. Also, very useful in EOM(2,3) calculations.

None

EOM_DAVIDSON_CONVERGENCE

Convergence criterion for the RMS residuals of excited state vectors

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Use default. Should normally be set to the same value 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:

00105 Corresponding to 0.00001

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-de}$

RECOMMENDATION:

Use 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 less 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_GPLMR_MSUBSIZE

Specifies the number of Krylov-space residuals in GPLMR.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Generate n residuals at each iteration.

RECOMMENDATION:

Use default. The convergence is faster for larger n, but the memory usage and the overall cost will increase.

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

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 n point charges are used.

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

6.7.10 Examples

Example 6.24 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 6.25 EOM-EE-CCSD calculations of the singlet excited states of PYP using Cholesky decomposition

```
$molecule
0 1
...too long to enter...
$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]
$end
```

Example 6.26 EOM-SF-CCSD calculations for methylene from high-spin $^3B_2$ reference

```
$molecule
0 3
C
H  1 rCH
H  1 rCH 2 aHCH

rCH    = 1.1167
aHCH   = 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 6.27 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 6.28 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       [1,0,1,1]
cc_fno_thresh   9900        99% of the total natural population recovered
$end
```

Example 6.29 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
$end
```

Example 6.30 EOM-EA-CCSD job for cyano radical. 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 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
jobtype       sp
method        hf
basis         6-311+G*
purecart      111
scf_convergence 8
eom_fake_ipea true
$end

@@@
$molecule
0 2
C
N 1 bond

bond   1.1718
$end

$rem
jobtype        sp
basis          6-311+G*
purecart       111
scf_guess      read
max_scf_cycles 0
method         eom-ccsd
cc_dov_thresh  2501   use threshold for CC iterations with problematic convergence
```

```
ea_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 6.31 EOM-DIP-CCSD calculation of electronic states in methylene using charged cage stabilization method.

```
$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
jobtype = sp
basis = 6-311g(d,p)
scf_algorithm = diis_gdm
symmetry = false
method = eom-ccsd
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
$end
```

Example 6.32 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
jobtype = sp
input_bohr = true
basis = 6-31g
symmetry = false
cc_symmetry = false
method = eom-ccsd
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
$end
```

Example 6.33 EOM-EE-CCSD calculation for phenol with user-specified guess requesting the EE transition from the occupied orbital number 24 (3 A") to the virtual orbital number 2 (23 A')

```
$molecule
0 1
C         0.935445   -0.023376    0.000000
C         0.262495    1.197399    0.000000
C        -1.130915    1.215736    0.000000
C        -1.854154    0.026814    0.000000
C        -1.168805   -1.188579    0.000000
C         0.220600   -1.220808    0.000000
O         2.298632   -0.108788    0.000000
H         2.681798    0.773704    0.000000
H         0.823779    2.130309    0.000000
H        -1.650336    2.170478    0.000000
H        -2.939976    0.044987    0.000000
H        -1.722580   -2.123864    0.000000
H         0.768011   -2.158602    0.000000
$end

$rem
jobtype             SP           single point
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_DAVIDSON_THRESHOLD      5    Threshold for inclusion of new vectors to the subspace
eom_user_guess      true         Use user guess from $eom_user_guess section
$end

$eom_user_guess
24   Transition from the occupied orbital number 24(3 A")
2    to the virtual orbital number 2 (23 A')
$end
```

Example 6.34 Complex-scaled EOM-EE calculation for He. All roots of Ag symmetry are computed (full diagonalization)

```
$molecule
0 1
HE  0    0  0.0
$END

$rem
complex_ccman       1             engage complex_ccman
jobtype             SP            single point
METHOD              EOM-CCSD
BASIS               gen           use general basis
```

```
PURECART           1111
EE_SINGLETS        [2000,0,0,0,0,0,0,0]  compute all excitation energies of Ag symmetry
EOM_DAVIDSON_CONV      5
EOM_DAVIDSON_THRESH    5
EOM_NGUESS_SINGLES    2000       Number of guess singles
EOM_NGUESS_DOUBLES    2000       Number of guess doubles
cc_memory 5000
mem_total 3000
$end

$complex_ccman
CS_HF 1                         Use complex HF
CS_ALPHA 1000                   Set alpha equal 1
CS_THETA 300                    Set theta (angle) equals 0.3 (radian)
$end

$basis
He   0
S    4    1.000000
   2.34000000E+02    2.58700000E-03
   3.51600000E+01    1.95330000E-02
   7.98900000E+00    9.09980000E-02
   2.21200000E+00    2.72050000E-01
S    1    1.000000
   6.66900000E-01    1.00000000E+00
S    1    1.000000
   2.08900000E-01    1.00000000E+00
P    1    1.000000
   3.04400000E+00    1.00000000E+00
P    1    1.000000
   7.58000000E-01    1.00000000E+00
D    1    1.000000
   1.96500000E+00    1.00000000E+00
S    1    1.000000
   5.13800000E-02    1.00000000E+00
P    1    1.000000
   1.99300000E-01    1.00000000E+00
D    1    1.000000
   4.59200000E-01    1.00000000E+00
S    1    1.000000
   2.44564000E-02    1.00000000E+00
S    1    1.000000
   1.2282000E-02    1.00000000E+00
S    1    1.000000
   6.1141000E-03    1.00000000E+00
P   1   1.0
    8.130000e-02    1.0
P   1   1.0
    4.065000e-02    1.0
P   1   1.0
    2.032500e-02    1.0
D   1   1.0
    2.34375e-01    1.0
D   1   1.0
    1.17187e-01    1.0
D   1   1.0
    5.85937e-02    1.0
****
```

```
$end
```

Example 6.35 CAP-augmented EOM-EA-CCSD calculation for N2-. aug-cc-pVTZ basis augmented by the 3s3p3d diffuse functions placed in the COM. 2 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
jobtype            SP             single point
METHOD             EOM-CCSD
BASIS              gen            use general basis
EA_STATES          [0,0,2,0,0,0,0,0]  compute electron attachment energies
cc_memory          5000           ccman2 memory
mem_total          2000
CC_EOM_PROP        true           compute excited state properties
$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
```

6.7.11 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. A more convenient way is just to specify EXCHANGE, *e.g.*, if EXCHANGE=B3LYP, B3LYP orbitals will be computed and used in the CCMAN/CCMAN2 module.

6.7.12 Analytic Gradients and Properties for the CCSD and EOM-XX-CCSD Methods

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 [62] (see also Section 5.12).

Application limit: same as for the single-point CCSD or EOM-CCSD calculations.

Limitations: Gradients for ROHF and non-HF (*e.g.*, B3LYP) orbitals are not yet available.

For the CCSD and EOM-CCSD wavefunctions, Q-CHEM currently can calculate permanent and transition dipole moments, oscillator strengths, $\langle R^2 \rangle$ (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 $\langle S^2 \rangle$ values. Interface of the CCSD and EOM-CCSD codes with the NBO 5.0 package is also available. Similar functionality is available for some EOM-OD and CI models.

The coupled-cluster package in Q-CHEM can calculate properties of target EOM states including transition dipoles and geometry optimizations. The target state of interest is selected by CC_STATE_TO_OPT *$rem*, which specifies the symmetry and the number of the EOM state.

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.

6.7.13 EOM-CC Optimization and Properties Job Control

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: 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. The properties currently include permanent dipole moment, the second moments $\langle X^2\rangle$, $\langle Y^2\rangle$, and $\langle Z^2\rangle$ of electron density, and the total $\langle R^2\rangle = \langle X^2\rangle + \langle Y^2\rangle + \langle Z^2\rangle$ (in atomic units). Incompatible with JOBTYPE=FORCE, OPT, 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. Sometimes the equations for left and right eigenvectors converge to different sets of target states. In this case, the simultaneous iterations of left and right vectors will diverge, and the properties for several or all the target states may be incorrect! The problem can be solved by varying the number of requested states, specified with XX_STATES, or the number of guess vectors (EOM_NGUESS_SINGLES). 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 package 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 for the EOM-CCSD target states will be calculated. By default, the transition dipole moment is calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition dipole moment 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 (no transition dipole and oscillator strength will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (for the left EOM-CCSD eigenvectors plus lambda CCSD equations in case if 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.

EOM_REF_PROP_TE

Request for calculation of non-relaxed two-particle EOM-CC properties. The two-particle properties currently include $\langle S^2 \rangle$. 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 $\langle S^2 \rangle$. The variable CC_EOM_PROP must be also set to TRUE. Alternatively, CC_CALC_SSQ can be used to request $\langle S^2 \rangle$ 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 (including orbital relaxation terms) will be computed. The variable CC_EOM_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. 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.

6.7.14 Examples

Example 6.36 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
jobtype            SP
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 6.37 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 6.38 Geometry optimization with tight convergence for the ^{2}A$_1$ excited state of CH$_2$Cl, followed by calculation of non-relaxed and fully relaxed permanent dipole moment and $\langle S^2\rangle$.

```
$molecule
0 2
H
C 1 CH
CL 2 CCL 1 CCLH
H 2 CH 3 CCLH 1 DIH

CH=1.096247
CCL=2.158212
CCLH=122.0
DIH=180.0
$end

$rem
JOBTYPE            OPT
METHOD             EOM-CCSD
BASIS              6-31G*  Basis Set
SCF_GUESS          SAD
EOM_DAVIDSON_CONVERGENCE    9    EOM amplitude convergence
CC_T_CONV          9    CCSD amplitudes convergence
EE_STATES          [0,0,0,1]
cc_state_to_opt    [4,1]
EOM_NGUESS_SINGLES  2
GEOM_OPT_TOL_GRADIENT 2
GEOM_OPT_TOL_DISPLACEMENT 2
GEOM_OPT_TOL_ENERGY 2
$end

@@@
$molecule
READ
$end

$rem
JOBTYPE            SP
METHOD             EOM-CCSD
BASIS              6-31G*  Basis Set
SCF_GUESS          READ
EE_STATES          [0,0,0,1]
CC_NGUESS_SINGLES  2
CC_EOM_PROP        1   calculate one-electron properties
CC_EOM_PROP_TE     1   and two-electron properties (S^2)
$end

@@@
$molecule
READ
$end

$rem
```

```
JOBTYPE            SP
METHOD             EOM-CCSD
BASIS              6-31G*  Basis Set
SCF_GUESS          READ
EE_STATES          [0,0,0,1]
EOM_NGUESS_SINGLES  2
CC_EOM_PROP         1  calculate one-electron properties
CC_EOM_PROP_TE      1  and two-electron properties (S^2)CC_EXSTATES_PROP 1
CC_FULLRESPONSE     1  same as above, but do fully relaxed properties
$end
```

Example 6.39 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  3  1.0  1  120
$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 6.40 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
```

6.7.15 EOM(2,3) Methods for Higher-Accuracy and Problematic Situations

In the EOM-CC(2,3) approach [74], 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 [74], 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 6.7.15.

System	**EOM-CCSD**	**EOM-CC(2,3)**
Singly-excited electronic states	0.1–0.2 eV	0.01 eV
Doubly-excited electronic states	≥ 1 eV	0.1–0.2 eV
Severe spin-contamination of the reference	~ 0.5 eV	≤ 0.1 eV
Breaking single bond (EOM-SF)	0.1–0.2 eV	0.01 eV
Breaking double bond (EOM-2SF)	~ 1 eV	0.1–0.2 eV

Table 6.3: 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.

6.7.16 Active-Space EOM-CC(2,3): Tricks of the Trade

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.

6.7.17 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:
INTEGER
DEFAULT:
1
OPTIONS:
0 Do apply restrictions
1 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.

Example 6.41 Example *$reorder_mo*section with 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
```

6.7.18 Examples

Example 6.42 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 6.43 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
   jobtype            sp
   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 6.44 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
jobtype      sp
method       eom-cc(2,3)
basis        6-311G
```

```
ip_states     [1,0,1,1]
$end
```

6.7.19 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 N^7 computational scheme. Using EOM-CCSD wavefunctions as zero-order wavefunctions, the second order triples correction to the μth EOM-EE or SF state is:

$$\Delta E_{\mu}^{(2)} = -\frac{1}{36}\sum_{i,j,k}\sum_{a,b,c}\frac{\tilde{\sigma}_{ijk}^{abc}(\mu)\sigma_{ijk}^{abc}(\mu)}{D_{ijk}^{abc}-\omega_{\mu}} \tag{6.37}$$

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:

$$\begin{aligned}\tilde{\sigma}_{ijk}^{abc}(\mu) &= \langle\Phi_0|(L_{1\mu}+L_{2\mu})(He^{(T_1+T_2)})_c|\Phi_{ijk}^{abc}\rangle \\ \sigma_{ijk}^{abc}(\mu) &= \langle\Phi_{ijk}^{abc}|[He^{(T_1+T_2)}(R_{0\mu}+R_{1\mu}+R_{2\mu})]_c|\Phi_0\rangle\end{aligned} \tag{6.38}$$

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 wavefunction), the (fT) and (dT) corrections are identical to the CCSD(2)$_T$ and CR-CCSD(T)$_L$ corrections of Piecuch and co-workers [75].

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. 76.

Similar corrections are available for EOM-IP-CCSD [77], where triples correspond to $3h2p$ excitations.

6.7.20 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, and IP
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, and IP
SD(ST)	EOM-CCSD(sT), available for IP

RECOMMENDATION:

None

6.7.21 Examples

Example 6.45 EOM-EE-CCSD(fT) calculation of CH^+

```
$molecule
1 1
C
H  C CH

CH  = 2.137130
$end

$rem
input_bohr         true
jobtype            sp
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 6.46 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
$end
```

Example 6.47 EOM-IP-CCSD(dT) calculations of Mg

```
$molecule
0 1
          Mg       0.000000     0.000000    0.000000
$end

$rem
jobtype       sp
method        eom-ccsd(dt)
basis         6-31g
ip_states     [1,0,0,0,0,1,1,1]
$end
```

6.7.22 Potential Energy Surface Crossing Minimization

Potential energy surface crossing optimization procedure finds energy minima on crossing seams. On the seam, the potential surfaces are degenerated in the subspace perpendicular to the plane defined by two vectors: the gradient difference

$$\mathbf{g} = \frac{\partial}{\partial \mathbf{q}}(E_1 - E_2) \tag{6.39}$$

and the derivative coupling

$$\mathbf{h} = \left\langle \Psi_1 \left| \frac{\partial \mathbf{H}}{\partial \mathbf{q}} \right| \Psi_2 \right\rangle \tag{6.40}$$

At this time Q-CHEM is unable to locate crossing minima for states which have non-zero derivative coupling. Fortunately, often this is not the case. 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.

6.7.22.1 Job Control Options

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.
spin = 1	Addresses states with high spin, see also EE_TRIPLETS.
irrep	Specifies the irreducible representation to which the state belongs, for C_{2v} point group symmetry irrep = 1 for A_1, irrep = 2 for A_2, irrep = 3 for B_1, irrep = 4 for B_2.
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.

XOPT_SEAM_ONLY

Orders an intersection seam search only, no minimization is to perform.

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 setting this option to TRUE and use that geometry as a starting point for the minimization.

6.7.22.2 Examples

Example 6.48 Minimize the intersection of $\tilde{A}^2A_1$ and $\tilde{B}^2B_1$ states of the NO_2 molecule using EOM-IP-CCSD method

```
$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
   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
$END
```

Example 6.49 Minimize the intersection of $\tilde{A}^1B_2$ and $\tilde{B}^1A_2$ states of the N_3^+ ion using EOM-CCSD method

```
$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_SINGLES             [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
$end
```

6.7.23 Dyson Orbitals for Ionization from Ground and Excited States within 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 wavefunction and the N-1/N+1 electron wavefunction of the corresponding cation/anion:

$$\phi^d(1) = \frac{1}{N-1}\int \Psi^N(1,\ldots,n)\Psi^{N-1}(2,\ldots,n)d2\ldots dn \tag{6.41}$$

$$\phi^d(1) = \frac{1}{N+1}\int \Psi^N(2,\ldots,n+1)\Psi^{N+1}(1,\ldots,n+1)d2\ldots d(n+1) \tag{6.42}$$

For the Hartree-Fock wavefunctions and within Koopmans' approximation, these are just the canonical HF orbitals. For correlated wavefunctions, Dyson orbitals are linear combinations of the reference molecular orbitals:

$$\phi^d = \sum_p \gamma_p \phi_p \tag{6.43}$$

$$\gamma_p = \langle \Psi^N | p^+ | \Psi^{N-1} \rangle \tag{6.44}$$

$$\gamma_p = \langle \Psi^N | p | \Psi^{N+1} \rangle \tag{6.45}$$

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-right and right-left):

$$\gamma_p^R = \langle \Phi_0 e^{T_1+T_2} L^{EE} | p^+ | R^{IP} e^{T_1+T_2} \Phi_0 \rangle \tag{6.46}$$

$$\gamma_p^L = \langle \Phi_0 e^{T_1+T_2} L^{IP} | p | R^{EE} e^{T_1+T_2} \Phi_0 \rangle \tag{6.47}$$

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 wavefunction of the leaving electron. For additional details, see Refs. 78, 79.

6.7.23.1 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. In CCMAN2, EOM_FAKE_IPEA will be automatically set to TRUE.

CC_DO_DYSON

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

For calculating Dyson orbitals between excited states from the reference configuration and IP/EA states, CC_TRANS_PROP=TRUE and CC_DO_DYSON_EE = TRUE have to be added to the usual EOM-IP/EA-CCSD calculation. 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-IP/EA pair of states in the order: EE1 - IP/EA1, EE1 - IP/EA2,. . . , EE2 - IP/EA1, EE2 - IP/EA2, . . ., and so on.

CC_DO_DYSON_EE

Whether excited-state or spin-flip 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

Dyson orbitals can be also 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 xyz Cartesian grid will be written in the resulting plot.mo file. For RHF(UHF) reference, the columns order in plot.mo is: $\phi_1^{lr}\alpha$ ($\phi_1^{lr}\beta$) $\phi_1^{rl}\alpha$ ($\phi_1^{rl}\beta$) $\phi_2^{lr}\alpha$ ($\phi_2^{lr}\beta$) $\ldots$

In addition, setting the MAKE_CUBE_FILES keyword to TRUE will create cube files for Dyson orbitals which can be viewed with VMD or other programs (see Section 10.5.4 for details). Other means of visualization (*e.g.*, with MOLDEN_FORMAT=TRUE or GUI=2) are currently not available.

6.7.23.2 Examples

Example 6.50 Plotting grd-ex and ex-grd state Dyson orbitals for ionization of the oxygen molecule. The target states of the cation are ^{2}A$_g$ and ^{2}B$_{2u}$.

```
$molecule
0 3
O 0.000 0.000 0.000
O 1.222 0.000 0.000
$end

$rem
jobtype        sp
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 6.51 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^-

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

$rem
jobtype            SP
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 true          calculate transition properties
CC_DO_DYSON_EE 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
```

6.7.24 Interpretation of EOM/CI Wavefunction and Orbital Numbering

Analysis of the leading wavefunction amplitudes is always necessary for determining the character of the state (*e.g.*, HOMO-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 wavefunction 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).

6.8 Correlated Excited State Methods: ADC(n) Family

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 and ADC(2)-x are implemented into Q-CHEM. 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.

6.8.1 The Algebraic Diagrammatic Construction (ADC) Scheme

The Algebraic Diagrammatic Construction (ADC) scheme 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 [80]. An alternative derivation is available in terms of the intermediate state representation (ISR) [81] 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 $\hat{C}_J$.

$$\left|\bar{\Psi}_J^N\right\rangle = \hat{C}_J \left|\Psi_0^N\right\rangle \tag{6.48}$$

with

$$\left\{\hat{C}_J\right\} = \left\{c_a^\dagger c_i; c_a^\dagger c_b^\dagger c_i c_j, i < j, a < b; \ldots\right\} \tag{6.49}$$

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

$$M_{IJ} = \left\langle \tilde{\Psi}_I^N \middle| \hat{H} - E_0^N \middle| \tilde{\Psi}_J^N \right\rangle \tag{6.50}$$

Here, the Hamiltonian of the system is shifted by the exact ground state energy E_0^N. The solution of the secular ISR equation

$$\mathbf{M}\mathbf{X} = \mathbf{X}\boldsymbol{\Omega}, \quad \text{with} \quad \mathbf{X}^\dagger\mathbf{X} = \mathbf{1} \tag{6.51}$$

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

$$\left|\Psi_n^N\right\rangle = \sum_J X_{nJ} \left|\tilde{\Psi}_J^N\right\rangle \tag{6.52}$$

This also allows for the calculation of dipole transition moments via

$$T_n = \left\langle \Psi_n^N \right| \hat{\mu} \left| \Psi_0^N \right\rangle = \sum_J X_{nJ}^\dagger \left\langle \tilde{\Psi}_J^N \middle| \hat{\mu} \middle| \Psi_0^N \right\rangle, \tag{6.53}$$

as well as excited state properties via

$$O_n = \left\langle \Psi_n^N \right| \hat{o} \left| \Psi_n^N \right\rangle = \sum_{I,J} X_{nI}^\dagger X_{nJ} \left\langle \tilde{\Psi}_I^N \middle| \hat{o} \middle| \Psi_J^N \right\rangle, \tag{6.54}$$

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 wavefunctions. 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 in Refs. 82–84. 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 wavefunction methods in Section 6.2 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 6.6) 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 [85].

6.8.2 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.

6.8.3 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. [86–88]

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.[87]

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.[88]

The spin-opposite scaling models are invoked, if ADC_SOS is set to TRUE. 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.

6.8.4 Properties and Visualization

The calculation of excited states using the ADCMAN module yields by default the usual excitation energies and the excitation amplitudes, as well as the transition dipole moments, oscillator strengths, and the norm of the doubles part of the amplitudes (if applicable). In addition, the calculation of excited state properties, like dipole moments, and transition properties between excited states can be requested by setting the *$rem* variables ADC_PROP_ES and ADC_PROP_ES2ES, respectively.

Furthermore, the ADCMAN module allows for the export of transition density and/or the attachment and detachment densities of the excited states [89] as CUBE files for visualization using external programs like VMD. Therefore, MAKE_CUBE_FILES has to be set to TRUE and a *$plots* section must be given in the input (for details see). To export attachment and detachment densities, currently, also ADC_PROP_ES has to be set to TRUE.

6.8.5 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.

All *$rem* variables to set the number of excited 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 to FALSE.

METHOD
Controls the order in perturbation theory of ADC.
TYPE:
STRING
DEFAULT:
None
OPTIONS:

ADC(0)	Perform ADC(0) calculation.
ADC(1)	Perform ADC(1) calculation.
ADC(2)	Perform ADC(2)-s calculation.
ADC(2)-x	Perform ADC(2)-x calculation.
ADC(3)	Perform ADC(3) calculation.

RECOMMENDATION:
None

ADC_ORDER

Deprecated! Controls the order in perturbation theory of ADC. Only read, if METHOD is not set.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

0	Activate ADC(0).
1	Activate ADC(1).
2	Activate ADC(2)-s or ADC(2)-x.
3	Activate ADC(3).

RECOMMENDATION:

Use METHOD instead.

ADC_EXTENDED

Deprecated! Activates the ADC(2)-x variant. This option is ignored unless ADC_ORDER is set to 2 and METHOD is not set.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	Activate ADC(2)-x.
FALSE	Do an ADC(2)-s calculation.

RECOMMENDATION:

Use METHOD instead.

EE_STATES

Controls the number of excited states to calculate.

TYPE:

INTEGER

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

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.

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_PROP_ES

Controls the calculation of excited state properties (currently only dipole moments).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate excited state properties.

FALSE Do not compute state properties.

RECOMMENDATION:

Set to TRUE, if properties are required or attachment/detachment densities of the excited states should be plotted.

ADC_PROP_ES2ES

Controls the calculation of transition properties between excited states (currently only transition dipole moments and oscillator strengths).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate state-to-state transition properties.

FALSE Do not compute transition properties between excited states.

RECOMMENDATION:

Set to TRUE, if state-to-state properties are required.

ADC_SOS

Activates the SOS models in second order ADC calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE Perform a standard ADC calculation

OPTIONS:

TRUE Employ the SOS model, if METHOD is set to ADC(2) or ADC(2)-x.

FALSE Do not use the SOS model.

RECOMMENDATION:

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 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 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 default

ADC_NGUESS_SINGLES

Controls the number of excited state guess vectors which are single excitations. 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:

ADC_NGUESS_DOUBLES

Controls the number of excited state guess vectors which are double excitations.

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 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 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 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	Debug1: more status information, extended information on timings.
...	

RECOMMENDATION:

6.8.6 Examples

Example 6.52 Q-CHEM 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
jobtype             sp
method              adc(2)
basis               6-31g(d,p)
mem_total           4000
mem_static          100
ee_singlets         [0,4,2,0]
adc_nguess_singles  3
adc_nguess_doubles  3
adc_davidson_conv   6
$end
```

Example 6.53 Q-CHEM input for an unrestricted 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
jobtype             sp
method              adc(2)
basis               aug-cc-pVDZ
mem_total           4000
mem_static          100
cc_symmetry         false
ee_states           6
adc_nguess_singles  6
adc_do_diis         true
adc_prop_es         true
adc_prop_es2es      true
$end
```

Example 6.54 Q-CHEM 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.

```
$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
jobtype             sp
method              adc(2)-x
basis               6-31g(d,p)
threads             2
mem_total           3000
mem_static          100
adc_print           1
ee_singlets         [0,4]
ee_triplets         [2,0]
adc_nguess_singles  4
adc_nguess_doubles  6
adc_davidson_conv   8
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
```

6.9 Restricted active space spin-flip (RAS-SF) and configuration interaction (RAS-CI) methods

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 [90–93], including strongly correlated systems. The RAS-SF approach is essentially a much lower computational cost alternative to Complete Active Space SCF (CAS-SCF) 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 CAS-SCF 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 [90–92, 94–96].

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 6.7), other target RAS-CI wavefunctions 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 [93].

Q-CHEM features two versions of RAS-CI code with different, complementary, functionalities. One code (invoked by specifying CORR=RASCI) has been written by David Casanova [93]; below we will refer to this code as RASCI1. The second implementation (invoked by specifying CORR=RASCI2) is primarily by Paul Zimmerman [91]; we will refer to it as RASCI2 below.

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 only allows for the calculation of systems with an even number of electrons, with the multiplicity 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 $\langle S^2 \rangle$ 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 2.7.1. A RASCI2 input requires the specification of an auxiliary basis set analogous to RI-MP2 computations (see Section 5.5.1). 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; however, the energy differences between different states should closely match each other.

6.9.1 The Restricted Active Space (RAS) Scheme

In the RAS formalism, we divide the orbital space into three subspaces called RAS1, RAS2 and RAS3 (Fig. 6.2). The RAS-CI states are defined by the number of orbitals and the restrictions in each subspace.

The single reference RAS-CI electronic wavefunctions are obtained by applying a spin-flipping or excitation operator $\hat{R}$ on the reference determinant $\phi^{(0)}$.

$$\left|\Psi^{\text{RAS}}\right\rangle = \hat{R}\left|\phi^{(0)}\right\rangle \tag{6.55}$$

The $\hat{R}$ operator must obey the restrictions imposed in the subspaces RAS1, RAS2 and RAS3, and can be decomposed as:

$$\hat{R} = \hat{r}^{\text{RAS2}} + \hat{r}^{h} + \hat{r}^{p} + \hat{r}^{hp} + \hat{r}^{2h} + \hat{r}^{2p} + ... \tag{6.56}$$

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 superindex) in RAS1 and electrons in RAS3 (p superindex). 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. (6.56).

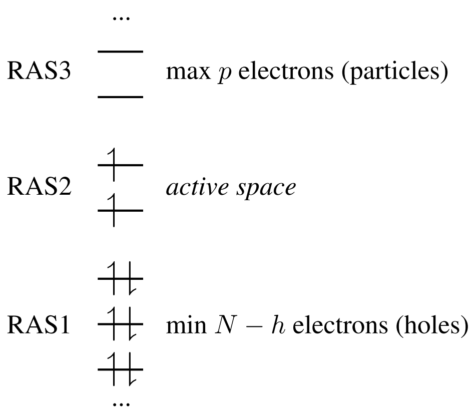


Figure 6.2: Orbital subspaces in RAS-CI employing a ROHF triplet reference.

6.9.2 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 (even electron systems, spin adapted algorithm using exact integrals).

The use of the RAS-CI1 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 even numbers of electrons and 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 RASCI.

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 RASCI.

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 RASCI.

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 RASCI.

RAS_DO_HOLE

Controls the presence of hole excitations in the RAS-CI wavefunction.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE	Include hole configurations (RAS1 to RAS2 excitations)
FALSE	Do not include hole configurations

RECOMMENDATION:

None. Only works with RASCI.

RAS_DO_PART

Controls the presence of particle excitations in the RAS-CI wavefunction.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE	Include particle configurations (RAS2 to RAS3 excitations)
FALSE	Do not include particle configurations

RECOMMENDATION:

None. Only works with RASCI.

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 RASCI.

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 RASCI.

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 RASCI.

RAS_NATORB_STATE

Allows to save the natural orbitals of a RAS-CI computed state.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

i Saves the natural orbitals for the i-th state

RECOMMENDATION:

None. Only works with RASCI.

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 RASCI.

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:

Only for RASCI, which at present only allows for the computation of systems with an even number of electrons. Thus, RAS_SPIN_MULT only can take odd values.

6.9.3 Job control options for the RASCI2 implementation

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 alpha vs. beta electrons

TYPE:

Integer

DEFAULT:

None

OPTIONS:

n user defined integer

RECOMMENDATION:

Set to 1 for an odd number of electrons or a cation, -1 for an anion. Only works with RASCI2.

Other *$rem* variables that can be used to control the evaluation of RASCI2 calculations are SET_ITER 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.

6.9.4 Examples

Example 6.55 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
  jobtype       sp
  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
$end
```

Example 6.56 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  rFF

rFF = 1.4136
$end

$rem
  jobtype       sp
  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 6.57 Input for a FCI/6-31G calculation of water molecule expanding the RAS2 space to the entire molecular orbital set. RASCI 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
  jobtype        sp
  exchange       hf
  correlation    rasci
  basis          6-31G
  mem_total      4000
  mem_static     100
  ras_roots      1
  ras_act        13
  ras_elec       10
  ras_occ        0
  ras_spin_mult  1
  ras_do_hole    false
  ras_do_part    false
$end
```

Example 6.58 Methylene single spin-flip calculation using RASCI2

```
$rem
jobtype sp
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
set_iter       25   ! number of iterations in RASCI
$end


$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
```

Example 6.59 Two methylene separated by 10 Å; double spin-flip calculation using RASCI2. Note that the $\langle S^2 \rangle$ values for this case will not be uniquely defined at the triply-degenerate ground state.

```
$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

jobtype sp

ras_n_roots    8
set_iter       25
$end

$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
```

Example 6.60 RASCI2 calculation of the nitrogen cation using double spin-flip.

```
$rem
exchange HF
correlation RASCI2
jobtype sp

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
set_iter      25
```

```
$end


$molecule
1 6
N 0.0000000 0.0000000 0.0000000
N 0.0000000 0.0000000 4.5
$end
```

6.10 How to Compute Ionization Energies of Core Electrons and Excited States Involving Excitations of Core Electrons

In experiments using high-energy radiation (such as X-ray spectroscopy) core electrons can be ionized or excited to low-lying virtual orbitals. There are two ways to compute ionization energies of core electrons in Q-CHEM. The first approach is a simple energy difference calculation in which core ionization is computed from energy differences computed for the neutral and core-ionized state. It is illustrated by this example:

Example 6.61 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
JOB_TYPE SP
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
JOB_TYPE SP
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: $E_{\rm SCF} = -40.1949062375$ and $E_{\rm CCSD} = -40.35748087$ (HF orbital energy of the neutral gives 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 $E_{\rm SCF} = -29.4656758483$ ($\langle S^2 \rangle$ = 0.7730) and $E_{\rm CCSD} = -29.64793957$. Thus, $\Delta E_{\rm CCSD} = (40.357481 - 29.647940) = 10.709$ hartree = 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.

The second approach is illustrated by the following input:

Example 6.62 Q-CHEM input for calculating chemical shift for $1s$-level of methane (CH_4) 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
JOB_TYPE SP
exchange HF
basis 6-31G*
MAX_CIS_CYCLES  = 100
CORRELATION CCSD
CCMAN2 = false
N_FROZEN_CORE 4 Freeze all valence orbitals
EOM_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.

The accuracy of both calculations can be improved using triples corrections, *e.g.*, CCSD(T) and EOM-IP-CCSD(dT). It is also recommend using a better basis that has more core functions.

6.10.1 Calculations of States Involving Core Electron Excitation/Ionization with DFT and TDDFT

TDDFT is not suited to describe the extended X-ray absorption fine structure (EXAFS) region, wherein the core electron is ejected and scattered by the neighboring atoms. Core-excitation energies computed with TDDFT with standard hybrid functionals are many electron volts too low compared with experiment. Exchange-correlation functionals specifically designed to treat core excitations are available in Q-CHEM. These short-range corrected (SRC) functionals are a modification of the more familiar long-range corrected functionals (discussed in Section 4.3.4). However, in SRC-DFT the short-range component of the Coulomb operator utilizes predominantly Hartree-Fock exchange, while the mid to long-range component is primarily treated with standard DFT exchange. These functionals can be invoked by using the SRC_DFT rem. In addition, a number of parameters (OMEGA, OMEGA2, HF_LR, HF_SR) that control the shape of the short and long-range Hartree-Fock components need to be specified. Full details of these functionals and appropriate values for the parameters can be found in Refs. 97, 98. An example of how to use these functionals is given below. For the K-shell of heavy elements (2nd row of the periodic table) relativistic effects become increasing important and a further correction for these effects is required. Also calculations for L-shell excitations are complicated by core-hole spin orbit coupling.

6.11 Visualization of Excited States

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. Recently, several new methods of analyzing molecular excited states have been proposed, including attachment/detachment density analysis [9] and natural transition orbitals [99]. This section describes the theoretical background behind these methods, while details of the input for creating data suitable for plotting these quantities is described separately in Chapter 10.

6.11.1 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

$$\Delta = \mathbf{P}_1 - \mathbf{P}_2 \tag{6.57}$$

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

$$\mathbf{U}^{\dagger}\Delta\mathbf{U} = \delta \tag{6.58}$$

The sum of the occupation numbers δ_p of these orbitals is then

$$tr(\Delta) = \sum_{p=1}^{N} \delta_p = n \tag{6.59}$$

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

$$\mathbf{D} = \mathbf{U}\mathbf{d}\mathbf{U}^\dagger \tag{6.60}$$

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

$$d_p = -\min(\delta_p, 0) \tag{6.61}$$

The detachment density corresponds to the electron density associated with single particle levels vacated in an electronic transition or hole density.

The attachment density

$$\mathbf{A} = \mathbf{U}\mathbf{a}\mathbf{U}^\dagger \tag{6.62}$$

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

$$a_p = \max(\delta_p, 0) \tag{6.63}$$

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

$$\Delta = \mathbf{A} - \mathbf{D} \tag{6.64}$$

6.11.2 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 [100]. 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.

Analysis of these and other excited states is greatly simplified by constructing Natural Transition Orbitals (NTOs) for the excited states. (The basic idea behind NTOs is rather old [101], and has been rediscovered several times [99, 102]; the term "natural transition orbitals" was coined in Ref. 99.) Let $\mathbf{T}$ denote the transition density matrix from a CIS, RPA, or TDDFT 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.* [102]

$$\mathbf{U}\mathbf{T}\mathbf{V}^\dagger = \Lambda \tag{6.65}$$

The matrices $\mathbf{U}$ and $\mathbf{V}$ are unitary and Λ 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 $\mathbf{TT}^{\dagger}$ and the particles are eigenvectors of the $V \times V$ matrix $\mathbf{T}^{\dagger}\mathbf{T}$ [99].) These "hole" and "particle" NTOs come in pairs, and their relative importance in describing the excitation is governed by the diagonal elements of Λ, which are excitation amplitudes in the NTO basis. By virtue of the SVD in Eq. (6.65), 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. (6.65) 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. Even when there is more than one significant NTO amplitude, as in systems of electronically-coupled chromophores [100], 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, and TDDFT excited states. 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] Ground-State Methods (Chapters 4 and 5).

[2] Basis Sets (Chapter 7) and Effective Core Potentials (Chapter 8).

[3] J. E. Del Bene, R. Ditchfield, and J. A. Pople, *J. Chem. Phys.* **55**, 2236 (1971).

[4] J. B. Foresman, M. Head-Gordon, J. A. Pople, and M. J. Frisch, *J. Phys. Chem.* **96**, 135 (1992).

[5] D. Maurice and M. Head-Gordon, *Int. J. Quantum Chem.* **29**, 361 (1995).

[6] T. D. Bouman and A. E. Hansen, *Int. J. Quantum Chem. Symp.* **23**, 381 (1989).

[7] A. E. Hansen, B. Voight, and S. Rettrup, *Int. J. Quantum Chem.* **23**, 595 (1983).

[8] D. Maurice and M. Head-Gordon, *J. Phys. Chem.* **100**, 6131 (1996).

[9] M. Head-Gordon, A. M. G. na, D. Maurice, and C. A. White, *J. Phys. Chem.* **99**, 14261 (1995).

[10] D. Casanova and M. Head-Gordon, *J. Chem. Phys.* **129**, 064104 (2008).

[11] A. I. Krylov, *Chem. Phys. Lett.* **350**, 522 (2002).

[12] J. F. Stanton, J. Gauss, N. Ishikawa, and M. Head-Gordon, *J. Chem. Phys.* **103**, 4160 (1995).

[13] S. Zilberg and Y. Haas, *J. Chem. Phys.* **103**, 20 (1995).

[14] C. M. Gittins, E. A. Rohlfing, and C. M. Rohlfing, *J. Chem. Phys.* **105**, 7323 (1996).

[15] D. Maurice and M. Head-Gordon, *Mol. Phys.* **96**, 1533 (1999).

[16] D. Maurice, *Single Electron Theories of Excited States*, PhD thesis, University of California, Berkeley, CA, 1998.

[17] A. J. W. Thom, E. J. Sundstrom, and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **11**, 11297 (2009).

[18] I. Mayer and P.-O. Löwdin, *Chem. Phys. Lett.* **202**, 1 (1993).

[19] E. Runge and E. K. U. Gross, *Phys. Rev. Lett.* **52**, 997 (1984).

[20] M. E. Casida, in *Recent Advances in Density Functional Methods, Part I*, edited by D. P. Chong, page 155, World Scientific, Singapore, 1995.

[21] S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.* **302**, 375 (1999).

[22] S. Hirata and M. Head-Gordon, *Chem. Phys. Lett.* **314**, 291 (1999).

[23] M. E. Casida, C. Jamorski, K. C. Casida, and D. R. Salahub, *J. Chem. Phys.* **108**, 4439 (1998).

[24] D. J. Tozer and N. C. Handy, *J. Chem. Phys.* **109**, 10180 (1998).

[25] A. Lange and J. M. Herbert, *J. Chem. Theory Comput.* **3**, 1680 (2007).

[26] M. J. G. Peach, P. Benfield, T. Helgaker, and D. J. Tozer, *J. Chem. Phys.* **128**, 044118 (2008).

[27] R. M. Richard and J. M. Herbert, *J. Chem. Theory Comput.* **7**, 1296 (2011).

[28] S. Hirata, T. J. Lee, and M. Head-Gordon, *J. Chem. Phys.* **111**, 8904 (1999).

[29] F. Liu et al., *Mol. Phys.* **108**, 2791 (2010).

[30] Y. Shao, M. Head-Gordon, and A. I. Krylov, *J. Chem. Phys.* **188**, 4807 (2003).

[31] Y. A. Bernard, Y. Shao, and A. I. Krylov, *J. Chem. Phys.* **136**, 204103 (2012).

[32] F. Wang and T. Ziegler, *J. Chem. Phys.* **121**, 12191 (2004).

[33] M. Seth, G. Mazur, and T. Ziegler, *Theor. Chem. Acc.* **129**, 331 (2011).

[34] N. A. Besley, *Chem. Phys. Lett.* **390**, 124 (2004).

[35] N. A. Besley, M. T. Oakley, A. J. Cowan, and J. D. Hirst, *J. Am. Chem. Soc.* **126**, 13502 (2004).

[36] N. A. Besley, *J. Chem. Phys.* **122**, 184706 (2005).

[37] D. M. Rogers, N. A. Besley, P. O'Shea, and J. D. Hirst, *J. Phys. Chem. B* **109**, 23061 (2005).

[38] J. Liu and W. Liang, *J. Chem. Phys.* **138**, 024101 (2013).

[39] J. Liu and W. Liang, *J. Chem. Phys.* **135**, 014113 (2011).

[40] J. Liu and W. Liang, *J. Chem. Phys.* **135**, 184111 (2011).

[41] A. T. B. Gilbert, N. A. Besley, and P. M. W. Gill, *J. Phys. Chem. A* **112**, 13164 (2008).

[42] N. A. Besley, A. T. B. Gilbert, and P. M. W. Gill, *J. Chem. Phys.* **130**, 124308 (2009).

[43] M. Filatov and S. Shaik, *Chem. Phys. Lett.* **304**, 429 (1999).

[44] T. Kowalczyk, T. Tsuchimochi, L. Top, P.-T. Chen, and T. Van Voorhis, *J. Chem. Phys.* **138**, 164101 (2013).

[45] M. Head-Gordon, R. J. Rico, M. Oumi, and T. J. Lee, *Chem. Phys. Lett.* **219**, 21 (1994).

[46] M. Head-Gordon, D. Maurice, and M. Oumi, *Chem. Phys. Lett.* **246**, 114 (1995).

[47] Y. M. Rhee and M. Head-Gordon, *J. Phys. Chem. A* **111**, 5314 (2007).

[48] M. Oumi, D. Maurice, T. J. Lee, and M. Head-Gordon, *Chem. Phys. Lett.* **279**, 151 (1997).

[49] M. Head-Gordon, M. Oumi, and D. Maurice, *Mol. Phys.* **96**, 593 (1999).

[50] D. Casanova, Y. M. Rhee, and M. Head-Gordon, *J. Chem. Phys.* **128**, 164106 (2008).

[51] H. Koch and P. Jørgensen, *J. Chem. Phys.* **93**, 3333 (1990).

[52] J. F. Stanton and R. J. Bartlett, *J. Chem. Phys.* **98**, 7029 (1993).

[53] A. I. Krylov, C. D. Sherrill, and M. Head-Gordon, *J. Chem. Phys.* **113**, 6509 (2000).

[54] A. I. Krylov, *Annu. Rev. Phys. Chem.* **59**, 433 (2008).

[55] H. Sekino and R. J. Bartlett, *Int. J. Quantum Chem. Symp.* **18**, 255 (1984).

[56] H. Koch, H. J. A. Jensen, P. Jørgensen, and T. Helgaker, *J. Chem. Phys.* **93**, 3345 (1990).

[57] S. V. Levchenko and A. I. Krylov, *J. Chem. Phys.* **120**, 175 (2004).

[58] A. I. Krylov, *Chem. Phys. Lett.* **338**, 375 (2001).

[59] D. Sinha, D. Mukhopadhya, R. Chaudhuri, and D. Mukherjee, *Chem. Phys. Lett.* **154**, 544 (1989).

[60] J. F. Stanton and J. Gauss, *J. Chem. Phys.* **101**, 8938 (1994).

[61] M. Nooijen and R. J. Bartlett, *J. Chem. Phys.* **102**, 3629 (1995).

[62] S. V. Levchenko, T. Wang, and A. I. Krylov, *J. Chem. Phys.* **122**, 224106 (2005).

[63] P. A. Pieniazek, S. E. Bradforth, and A. I. Krylov, *J. Chem. Phys.* **129**, 074104 (2008).

[64] A. A. Golubeva, P. A. Pieniazek, and A. I. Krylov, *J. Chem. Phys.* **130**, 124113 (2009).

[65] A. I. Krylov, *Acc. Chem. Res.* **39**, 83 (2006).

[66] D. Casanova, L. V. Slipchenko, A. I. Krylov, and M. Head-Gordon, *J. Chem. Phys.* **130**, 044103 (2009).

[67] M. Wladyslawski and M. Nooijen, volume 828 of *ACS Symposium Series*, page 65, American Chemical Society, Washington, D. C., 2002.

[68] T. Kuś and A. I. Krylov, *J. Chem. Phys.* **135**, 084109 (2011).

[69] T. Kuś and A. I. Krylov, *J. Chem. Phys.* **136**, 244109 (2012).

[70] K. B. Bravaya, D. Zuev, E. Epifanovsky, and A. I. Krylov, *J. Chem. Phys.* **138**, 124106 (2013).

[71] T.-C. Jagau, D. Zuev, K. B. Bravaya, E. Epifanovsky, and A. I. Krylov, *J. Phys. Chem. Lett.* **5**, 310 (2014).

[72] A. Landau, K. Khistyaev, S. Dolgikh, and A. I. Krylov, *J. Chem. Phys.* **132**, 014109 (2010).

[73] D. Zuev, N. Orms, A. Krylov, and E. Vecharynsky, *J. Comput. Chem.* (2014), submitted.

[74] S. Hirata, M. Nooijen, and R. J. Bartlett, *Chem. Phys. Lett.* **326**, 255 (2000).

[75] P. Piecuch and M. Włoch, *J. Chem. Phys.* **123**, 224105 (2005).

[76] P. U. Manohar and A. I. Krylov, *J. Chem. Phys.* **129**, 194105 (2008).

[77] P. U. Manohar, J. F. Stanton, and A. I. Krylov, *J. Chem. Phys.* **131**, 114112 (2009).

[78] C. M. Oana and A. I. Krylov, *J. Chem. Phys.* **127**, 234106 (2007).

[79] C. M. Oana and A. I. Krylov, *J. Chem. Phys.* **131**, 124114 (2009).

[80] J. Schirmer, *Phys. Rev. A* **26**, 2395 (1982).

[81] J. Schirmer and A. B. Trofimov, *J. Chem. Phys.* **120**, 11449 (2004).

[82] A. B. Trofimov and J. Schirmer, *J. Phys. B* **28**, 2299 (1995).

[83] A. B. Trofimov, G. Stelter, and J. Schirmer, *J. Chem. Phys.* **111**, 9982 (1999).

[84] A. B. Trofimov, G. Stelter, and J. Schirmer, *J. Chem. Phys.* **117**, 6402 (2002).

[85] P. H. P. Harbach, M. Wormit, and A. Dreuw, *J. Chem. Phys.* (2014), accepted.

[86] A. Hellweg, S. A. Grün, and C. Hättig, *Phys. Chem. Chem. Phys.* **10**, 4119 (2008).

[87] N. O. Winter and C. Hättig, *J. Chem. Phys.* **134**, 184101 (2011).

[88] C. M. Krauter, M. Pernpointner, and A. Dreuw, *J. Chem. Phys.* **138**, 044107 (2013).

[89] M. Wormit et al., *Mol. Phys.* **112**, 774 (2014).

[90] D. Casanova and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **11**, 9779 (2009).

[91] P. M. Zimmerman, F. Bell, M. Goldey, A. T. Bell, and M. Head-Gordon, *J. Chem. Phys.* **137**, 164110 (2012).

[92] F. Bell, P. M. Zimmerman, D. Casanova, M. Goldey, and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **15**, 358 (2013).

[93] D. Casanova, *J. Comput. Chem.* **34**, 720 (2013).

[94] F. Bell, D. Casanova, and M. Head-Gordon, *J. Am. Chem. Soc.* **132**, 11314 (2010).

[95] P. M. Zimmerman, F. Bell, D. Casanova, and M. Head-Gordon, *J. Am. Chem. Soc.* **133** (2011).

[96] D. Casanova, *J. Chem. Phys.* **137** (2012).

[97] N. A. Besley, M. J. G. Peach, and D. J. Tozer, *Phys. Chem. Chem. Phys.* **11**, 10350 (2009).

[98] N. A. Besley and F. A. Asmuruf, *Phys. Chem. Chem. Phys.* **12**, 12024 (2010).

[99] R. L. Martin, *J. Chem. Phys.* **118**, 4775 (2003).

[100] A. W. Lange and J. M. Herbert, *J. Am. Chem. Soc.* **131**, 124115 (2009).

[101] A. V. Luzanov, A. A. Sukhorukov, and V. E. Umanskii, *Theor. Exp. Chem.* **10**, 354 (1976).

[102] I. Mayer, *Chem. Phys. Lett.* **437**, 284 (2007).

Chapter 7

Basis Sets

7.1 Introduction

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" [3], 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 B). 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 and review articles [3–7] to aid your selection and see the section "Further Reading" at the end of this chapter.

7.2 Built-In Basis Sets

Q-CHEM is equipped with many standard basis sets [8], and allows the user to specify the required basis set by its standard symbolic representation. The available built-in basis sets are of four types:

- Pople basis sets
- Dunning basis sets
- Correlation consistent Dunning basis sets
- Ahlrichs basis sets
- Jensen polarization consistent basis sets

In addition, Q-CHEM supports the following features:

- Extra diffuse functions available for high quality excited state calculations.
- Standard polarization functions.
- Basis sets are requested by symbolic representation.
- s, p, sp, d, f and g angular momentum types of basis functions.
- Maximum number of shells per atom is 100.
- Pure and Cartesian basis functions.
- Mixed basis sets (see section 7.5).
- Basis set superposition error (BSSE) corrections.

The following *$rem* keyword controls the basis set:

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.

7.3 Basis Set Symbolic Representation

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. (*i.e.*, *heavy*, *light*). Use a period (.) as a place-holder (see examples).

	j	k	l	m	n
STO$-j(k+, l+)$G(m, n)	2,3,6	^{a}	^{b}	d	p
$j-21(k+, l+)$G(m, n)	3	^{a}	^{b}	$2d$	$2p$
$j-31(k+, l+)$G(m, n)	4,6	^{a}	^{b}	$3d$	$3p$
$j-311(k+, l+)$G(m, n)	6	^{a}	^{b}	df,$2df$,$3df$	pd,$2pd$,$3pd$

Table 7.1: Summary of Pople type basis sets available in Q-CHEM. m and $n$refer to the polarization functions on heavy and light atoms respectively. $^a k$ is the number of sets of diffuse functions on heavy $^b l$ is the number of sets of diffuse functions on light atoms.

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→Sb
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→Zn
6-311G	H, He, Li→Ne, Na→Ar, Ga→Kr
G3LARGE	H, He, Li→Ne, Na→Ar, K→Kr
G3MP2LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr

Table 7.2: Atoms supported for Pople basis sets available in Q-CHEM (see the Table below for specific examples).

7.3.1 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.

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.

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*	H, He, Na→Cl
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→Kr
6-311+G	H, He, Li→Ne, Na→Ar
6-311G*	H, He, Li→Ne, Na→Ar, Ga→Kr
6-311G(d,p)	H, He, Li→Ne, Na→Ar, Ga→Kr
G3LARGE	H, He, Li→Ne, Na→Ar, K→Kr
G3MP2LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr

Table 7.3: Examples of extended Pople basis sets.

	SV$(k+, l+)(md, np)$, DZ$(k+, l+)(md, np)$, TZ$(k+, l+)(md, np)$
k	# sets of heavy atom diffuse functions
l	# sets of light atom diffuse functions
m	# sets of d functions on heavy atoms
n	# sets of p functions on light atoms

Table 7.4: Summary of Dunning-type basis sets available in Q-CHEM.

Symbolic Name	**Atoms Supported**
SV	H, Li→Ne
DZ	H, Li→Ne, Al→Cl
TZ	H, Li→Ne

Table 7.5: Atoms supported for old Dunning basis sets available in Q-CHEM.

Symbolic Name	Atoms Supported
SV	H, Li→Ne
SV*	H, B→Ne
SV(d,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
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 7.6: Examples of extended Dunning basis sets.

Symbolic Name	Atoms Supported
cc-pVDZ	H, He, B→Ne, Al→Ar, Ga→Kr
cc-pVTZ	H, He, B→Ne, Al→Ar, Ga→Kr
cc-pVQZ	H, He, B→Ne, Al→Ar, Ga→Kr
cc-pCVDZ	B→Ne
cc-pCVTZ	B→Ne
cc-pCVQZ	B→Ne
aug-cc-pVDZ	H, He, B→Ne, Al→Ar, Ga→Kr
aug-cc-pVTZ	H, He, B→Ne, Al→Ar, Ga→Kr
aug-cc-pVQZ	H, He, B→Ne, Al→Ar, Ga→Kr
aug-cc-pCVDZ	B→F
aug-cc-pCVTZ	B→Ne
aug-cc-pCVQZ	B→Ne

Table 7.7: Atoms supported Dunning correlation-consistent basis sets available in Q-CHEM.

Symbolic Name	Atoms Supported
TZV	Li→Kr
VDZ	H→Kr
VTZ	H→Kr

Table 7.8: Atoms supported for Ahlrichs basis sets available in Q-CHEM.

Symbolic Name	Atoms Supported
pc-0, pc-1, pc-2, pc-3, pc-4	H→Ar
pcJ-0, pcJ-1, pcJ-2, pcJ-3, pcJ-4	H→Ar, except Li, Be, Na, Mg
pcS-0, pcS-1, pcS-2, pcS-3, pcS-4	H→Ar

Table 7.9: Atoms supported for Jensen polarization consistent basis sets available in Q-CHEM.

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, if you have a poorly behaved SCF, and you suspect there maybe linear dependence in you basis, the threshold should be increased.

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.

7.4 User-Defined Basis Sets (*$basis*)

7.4.1 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 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 the section in Chapter 4 on initial guesses for more information.

7.4.2 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 *$rem* variable PURECART needs to be initiated. This *$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, 111 would specify all g, f and d basis functions as being in the pure form. 121 would indicate g- and d- functions are pure and f-functions Cartesian.

PURECART
INTEGER
TYPE:
Controls the use of pure (spherical harmonic) or Cartesian angular forms
DEFAULT:
2111 Cartesian h-functions and pure 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.

7.4.3 Format for User-Defined Basis Sets

The format for the user-defined basis section is as follows:

```
$basis
```

$$\begin{array}{lllll} X & 0 & & & \\ L & K & scale & & \\ \alpha_1 & C_1^{L_{min}} & C_1^{L_{min}+1} & \dots & C_1^{L_{max}} \\ \alpha_2 & C_2^{L_{min}} & C_2^{L_{min}+1} & \dots & C_2^{L_{max}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \alpha_K & C_K^{L_{min}} & C_K^{L_{min}+1} & \dots & C_K^{L_{max}} \end{array}$$

```
****
$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.

7.4.4 Example

Example 7.1 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    0.0494720   0.511541
      0.493363   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
```

7.5 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.

7.5.1 Examples

Example 7.2 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 7.3 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. Note also that the dummy atom is not counted.

```
$molecule
   0  1
   C
   O   C  rco
   H1  C  rch1  O  h1co
   x   C  1.0   O  xcol  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
   exchange   hf
   basis      mixed   user-defined mixed basis
$end

$basis
   C  1
   3-21G
   ****
   O  2
   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  3
   6-31(+,+)G(d,p)
   ****
   H  4
   sto-3g
   ****
   H  5
   sto-3g
   ****
   H  6
   sto-3g
   ****
$end
```

7.6 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. Typically 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 4.8.2.
- Dual basis MP2 calculations. See discussion in Section 5.5.1.

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 4.7.2.

In addition to built-in basis sets for BASIS2, it is also possible to enter user-defined second basis sets using an additional *$basis*2 input section, whose syntax generally follows the *$basis* input section documented above in Section 7.4.

7.7 Auxiliary basis sets for RI / density fitting

Whilst 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 [9–11]. 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:

$$|\mu\nu\rangle \approx |\widetilde{\mu\nu}\rangle = \sum_K |K\rangle C^K_{\mu\nu} \tag{7.1}$$

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:

$$\sum_L \langle K\,|L\rangle C^L_{\mu\nu} = \langle K\,|\mu\nu\,\rangle \tag{7.2}$$

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:

$$\langle\mu\nu|\lambda\sigma\rangle \approx \langle\widetilde{\mu\nu}|\widetilde{\lambda\sigma}\rangle = \sum K, LC^L_\mu \langle L|K\rangle C^K_{\lambda\sigma} \tag{7.3}$$

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 5.5 on RI-MP2 and related methods, Section 5.14 on pairing methods, Section 5.7.5 on coupled cluster methods, Section 4.6.8 on DFT methods, and

Section 6.9 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.

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 EMSL Basis Set Exchange to aid your selection.

Symbolic Name	Atoms Supported
RIMP2-VDZ	H, He, Li→Ne, Na→Ar, 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-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

Table 7.10: Built-in auxiliary basis sets available in Q-CHEM for electron correlation.

In addition to built-in auxiliary basis sets, it is also possible to enter user-defined auxiliary basis sets using an *$aux_basis* input section, whose syntax generally follows the *$basis* input section documented above in Section 7.4.

7.8 Basis Set Superposition Error (BSSE)

When calculating binding energies, the energies of the fragments are usually higher than they should be due to the smaller effective basis set used for the individual species. This leads to an overestimate of the binding energy called the basis set superposition error. The effects of this can be corrected for by performing the calculations on the individual species in the presence of the basis set associated with the other species. This requires basis functions to be placed at arbitrary points in space, not just those defined by the nuclear centers. This can be done within Q-CHEM by using ghost atoms. These atoms have zero nuclear charge, but can support a user defined basis set. Ghost atom locations are specified in the *$molecule* section, as for any other atom, and the basis must be specified in a *$basis* section in the same manner as for a mixed basis.

Example 7.4 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
```

Ghosts atoms can also be specified by placing @ in front of the corresponding atomic symbol in the *$molecule* section of the input file. If @ is used to designate the ghost atoms in the system then it is not necessary to use MIXED basis set and include the *$basis* section in the input.

Example 7.5 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
JOBTYPE         SP
METHOD          B3LYP
BASIS           6-31G(d,p)
PURECART        1112
$end
```

Finally, there is an alternative approach to counterpoise corrections that is also available in Q-CHEM. The powerful Absolutely Localized Molecular Orbital (ALMO) methods can be very conveniently used for the fully automated evaluation of BSSE corrections with associated computational advantages also. This is described in detail, including examples, in Section 12.4.3.

References and Further Reading

[1] Ground-State Methods (Chapters 4 and 5).

[2] Effective Core Potentials (Chapter 8).

[3] W. J. Hehre, L. Radom, P. v. R. Schleyer, and J. A. Pople, *Ab Initio Molecular Orbital Theory*, Wiley, New York, 1986.

[4] S. Huzinaga, *Comp. Phys. Rep.* **2**, 279 (1985).

[5] E. R. Davidson and D. Feller, *Chem. Rev.* **86**, 681 (1986).

[6] D. Feller and E. R. Davidson, in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz and D. B. Boyd, volume 1, page 1, Wiley-VCH, New York, 1990.

[7] F. Jensen, *Introduction to Computational Chemistry*, Wiley, New York, 1994.

[8] 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 multi-program 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.

[9] M. Feyereisen, G. Fitzgerald, and A. Komornicki, *Chem. Phys. Lett.* **208**, 359 (1993).

[10] B. I. Dunlap, *Phys. Chem. Chem. Phys.* **2**, 2113 (2000).

[11] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon, *Proc. Natl. Acad. Sci. USA* **102**, 6692 (2005).

Chapter 8

Effective Core Potentials

8.1 Introduction

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. For reviews of relativistic effects in chemistry, see for example Refs. 3–8.

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.

The most widely used ECPs today are of the form first proposed by Kahn *et al.* [9] in the 1970s. 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

$$U(r) = U_L(r) + \sum_{l=0}^{L-1} \sum_{m=-l}^{+l} |Y_{lm}\rangle \left[U_l(r) - U_L(r)\right] \langle Y_{lm}| \tag{8.1}$$

where the $|Y_{lm}\rangle$ are spherical harmonic projectors and the $U_l(r)$ are linear combinations of Gaussians, multiplied by r^{-2}, r^{-1} or r^0. In addition, $U_L(r)$ contains a Coulombic term N_c/r, where N_c is the number of core electrons.

One of the key issues in the development of pseudopotentials 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 either on 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.

8.2 Built-In Pseudopotentials

8.2.1 Overview

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 (HWMB and LANL2DZ)
- The Stevens-Basch-Krauss-Jansien-Cundari set (SBKJC)
- The Christiansen-Ross-Ermler-Nash-Bursten sets (CRENBS and CRENBL)
- The Stuttgart-Bonn sets (SRLC and SRSC)

References and information about the definition and characteristics of most of these sets can be found at the EMSL site of the Pacific Northwest National Laboratory [10]:

http://www.emsl.pnl.gov/forms/basisform.html

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 pseudopotential
OPTIONS:

General, Gen	User defined. (*$ecp* keyword required)
Symbol	Use standard pseudopotentials discussed above.

RECOMMENDATION:
Pseudopotentials are recommended for first row transition metals and heavier elements.
Consul the reviews for more details.

8.2.2 Combining Pseudopotentials

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 “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 a sequence of four asterisks. There is also a built-in combination that can be invoked specifying “ECP=LACVP”. It assigns automatically 6-31G* or other suitable type basis sets for atoms H-Ar, while uses LANL2DZ for heavier atoms.

8.2.3 Examples

Example 8.1 Computing the HF/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        lanl2dz   Using the Hay-Wadt ECP
   BASIS      lanl2dz   And the matching basis set
$end
```

Example 8.2 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

   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
```

8.3 User-Defined Pseudopotentials

Many users will find that the library of built-in pseudopotentials 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 7).

8.3.1 Job Control for User-Defined ECPs

To apply a user-defined pseudopotential, 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 *$basis* block is described in Chapter 7. The syntax for each record within the *$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., ****)
$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.

8.3.2 Example

Example 8.3 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
   H3  Al  r  H1  120  H2  180

   r = 1.6
$end

$rem
   JOBTYPE    opt   Geometry optimization
   METHOD     hf    Hartree-Fock theory
   ECP        gen   User-defined ECP
   BASIS      gen   User-defined basis
$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
```

8.4 Pseudopotentials and Density Functional Theory

Q-CHEM's pseudopotential package and DFT package are tightly integrated and facilitate the application of advanced density functionals to molecules containing heavy elements. Any of the local, gradient-corrected and hybrid functionals discussed in Chapter 4 may be used and you may also perform ECP calculations with user-defined hybrid functionals.

In a DFT calculation with pseudopotentials, the exchange-correlation energy is obtained entirely from the non-core electrons. This will be satisfactory if there are no chemically important core-valence effects but may introduce significant errors, particularly if you are using a "large-core" ECP.

Q-CHEM's default quadrature grid is SG-1 (see section 4.3.11) which was originally defined only for the elements up to argon. In Q-CHEM 2.0 and above, the SG-1 grid has been extended and it is now defined for all atoms up to, and including, the actinides.

8.4.1 Example

Example 8.4 Optimization of the structure of XeF_5^+ using B3LYP theory and the ECPs of Stevens and collaborators. Note that the BASIS keyword has been omitted and, therefore, the matching SBKJC orbital basis set will be used.

```
$molecule
   1  1
   Xe
   F1  Xe  r1
   F2  Xe  r2  F1  a
   F3  Xe  r2  F1  a  F2  90
   F4  Xe  r2  F1  a  F3  90
   F5  Xe  r2  F1  a  F4  90

   r1  =  2.07
   r2  =  2.05
   a   =  80.0
$end

$rem
   JOBTYP     opt
   METHOD     b3lyp
   ECP        sbkjc
$end
```

8.5 Pseudopotentials and Electron Correlation

The pseudopotential 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.

8.5.1 Example

Example 8.5 Optimization of the structure of Se$_8$ using HF/LANL2DZ, followed by a single-point energy calculation at the MP2/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
   JOBTYP     opt
   METHOD     hf
   ECP        lanl2dz
$end

@@@

$molecule
   read
$end

$rem
   JOBTYP        sp         Single-point energy
   METHOD        mp2        MP2 correlation energy
   ECP           lanl2dz    Hay-Wadt ECP and basis
   SCF_GUESS     read       Read in the MOs
$end
```

8.6 Pseudopotentials, Forces and Vibrational Frequencies

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 pseudopotentials. However, as documented in Section 8.6.2, these capabilities are more limited than those for undifferentiated matrix elements. To avoid this limitation, Q-CHEM will switch seamlessly to numerical derivatives of the ECP matrix elements when needed, which are combined with analytical evaluation of the remainder of the force contributions (where available, as documented in Table 9.1.

The pseudopotential 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 pseudopotentials.

Q-CHEM cannot calculate analytic second derivatives of the nuclear potential-energy term when ECP's are used, and must therefore resort to finite difference methods. However, for HF and DFT calculations, it can compute analytic second derivatives for all other terms in the Hamiltonian. The program takes full advantage of this by only computing the potential-energy derivatives numerically, and adding these to the analytically calculated second derivatives of the remaining energy terms.

There is a significant speed advantage associated with this approach as, at each finite-difference step, only the potential-energy term needs to be calculated. This term requires only three-center integrals, which are far fewer in number and much cheaper to evaluate than the four-center, two-electron integrals associated with the electron-electron interaction terms. Readers are referred to Table 9.1 for a full list of the analytic derivative capabilities of Q-CHEM.

8.6.1 Example

Example 8.6 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. Note also that only the second derivatives of the potential energy term will be calculated by finite difference, all other terms will be calculated analytically.

```
$molecule
   0  1
   Te
   O1  Te  r
   O2  Te  r  O1  a

   r = 1.8
   a = 108
$end

$rem
   JOBTYP     opt
   METHOD     hf
   ECP        srlc
$end

@@@

$molecule
   read
$end

$rem
   JOBTYP      freq
   METHOD      hf
   ECP         srlc
   SCF_GUESS   read
$end
```

8.6.2 A Brief Guide to Q-CHEM's Built-In ECPs

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.

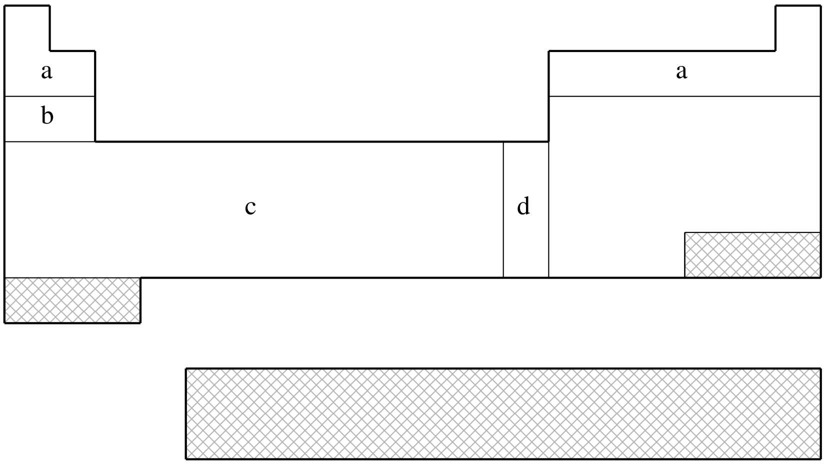
The following information is provided for each ECP:

- 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 pseudopotentials—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 LANL2DZ pseudopotential 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 LANL2DZ pseudopotential 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 LANL2DZ pseudopotential 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.

Finally, we note the limitations of the current ECP implementation within Q-CHEM:

- Energies can be calculated only for s, p, d and f basis functions with G projectors. Consequently, Q-CHEM cannot perform energy calculations on actinides using SRLC.
- Analytical ECP gradients can be calculated only for s, p and d basis functions with F projectors and only for s and p basis functions with G projectors. This limitation does not affect evaluation of forces and frequencies as discussed in Section 8.6.

8.6.3 The HWMB Pseudopotential at a Glance

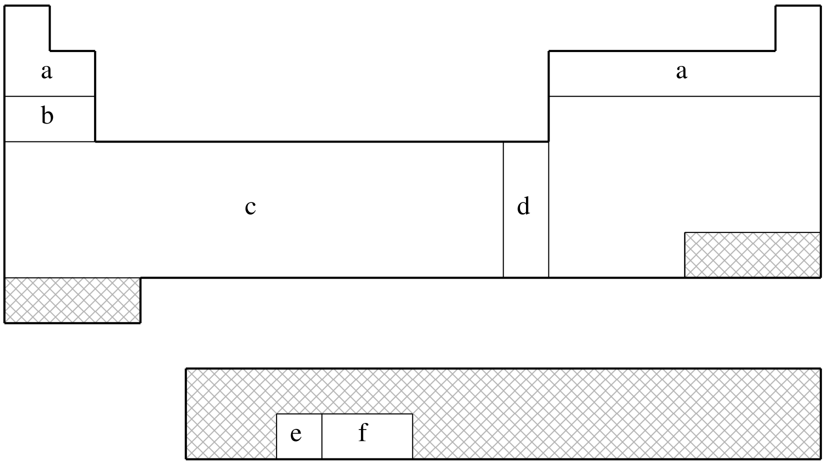


*HWMB is **not** available for shaded elements*

(a) No pseudopotential; Pople STO-3G basis used
(b) Wadt & Hay, J. Chem. Phys. 82 (1985) 285
(c) Hay & Wadt, J. Chem. Phys. 82 (1985) 299
(d) Hay & Wadt, J. Chem. Phys. 82 (1985) 270

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)

8.6.4 The LANL2DZ Pseudopotential at a Glance



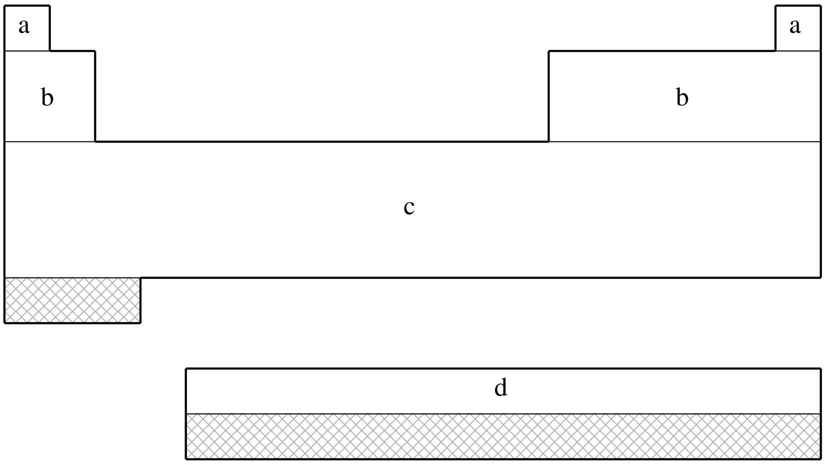
*LANL2DZ is **not** available for shaded elements*

(a) No pseudopotential; Pople 6-31G basis used
(b) Wadt & Hay, J. Chem. Phys. 82 (1985) 285
(c) Hay & Wadt, J. Chem. Phys. 82 (1985) 299
(d) Hay & Wadt, J. Chem. Phys. 82 (1985) 270
(e) Hay, J. Chem. Phys. 79 (1983) 5469
(f) Wadt, to be published

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)

Note that Q-Chem 4.2.2 and later versions also support LANL2DZ-SV basis, which employs SV basis functions (instead of 6-31G) on H, Li-He elements (like some other quantum chemistry packages).

8.6.5 The SBKJC Pseudopotential at a Glance



*SBKJC is **not** available for shaded elements*

(a) No pseudopotential; Pople 3-21G basis used

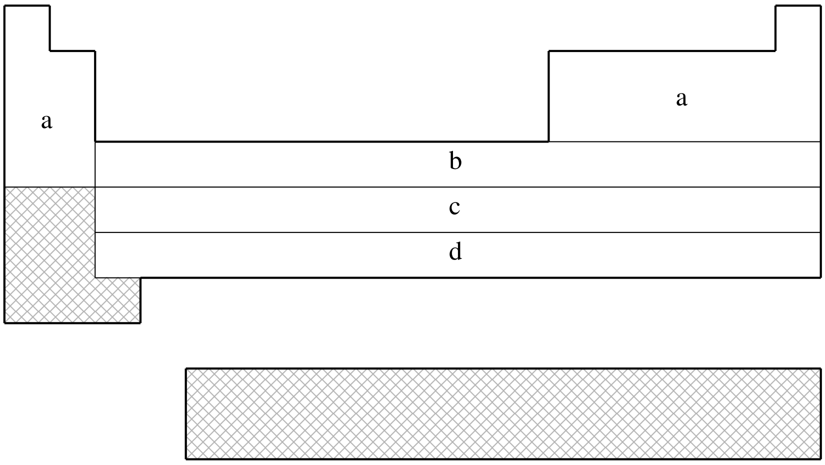
(b) W.J. Stevens, H. Basch & M. Krauss, J. Chem. Phys. 81 (1984) 6026

(c) W.J. Stevens, M. Krauss, H. Basch & P.G. Jasien, Can. J. Chem 70 (1992) 612

(d) T.R. Cundari & W.J. Stevens, J. Chem. Phys. 98 (1993) 5555

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)

8.6.6 The CRENBS Pseudopotential at a Glance



*CRENBS is **not** available for shaded elements*

(a) No pseudopotential; Pople STO-3G basis used

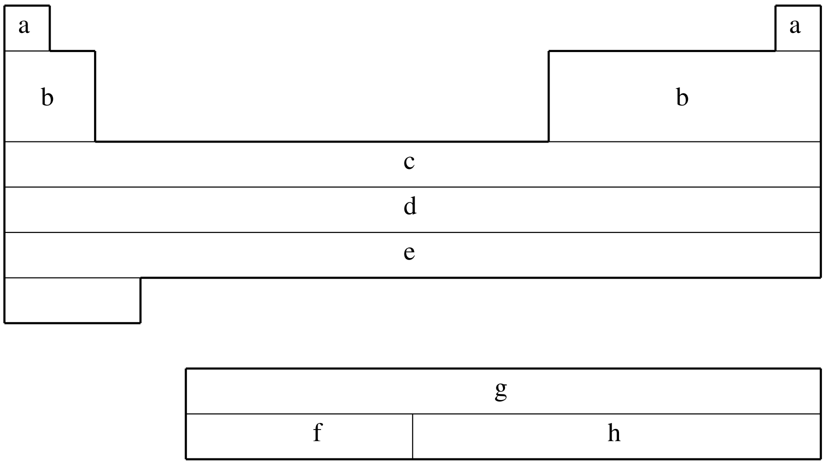
(b) Hurley, Pacios, Christiansen, Ross & Ermler, J. Chem. Phys. 84 (1986) 6840

(c) LaJohn, Christiansen, Ross, Atashroo & Ermler, J. Chem. Phys. 87 (1987) 2812

(d) Ross, Powers, Atashroo, Ermler, LaJohn & Christiansen, J. Chem. Phys. 93 (1990) 6654

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)

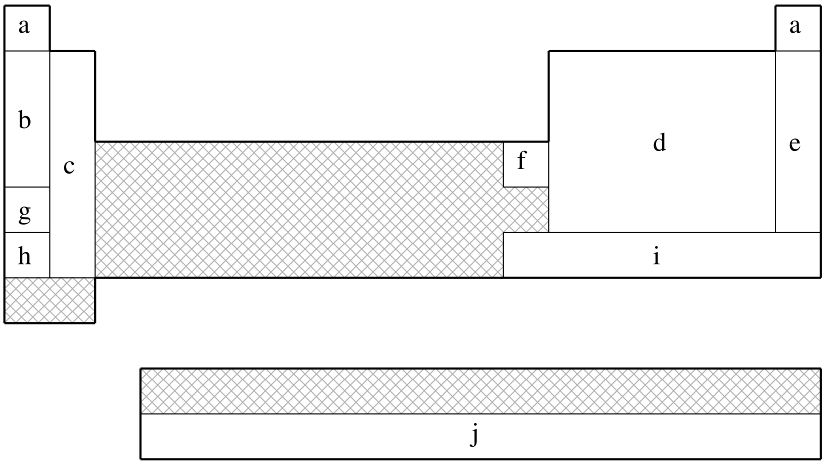
8.6.7 The CRENBL Pseudopotential at a Glance



(a) No pseudopotential; Pople 6-311G* basis used
(b) Pacios & Christiansen, J. Chem. Phys. 82 (1985) 2664
(c) Hurley, Pacios, Christiansen, Ross & Ermler, J. Chem. Phys. 84 (1986) 6840
(d) LaJohn, Christiansen, Ross, Atashroo & Ermler, J. Chem. Phys. 87 (1987) 2812
(e) Ross, Powers, Atashroo, Ermler, LaJohn & Christiansen, J. Chem. Phys. 93 (1990) 6654
(f) Ermler, Ross & Christiansen, Int. J. Quantum Chem. 40 (1991) 829
(g) Ross, Gayen & Ermler, J. Chem. Phys. 100 (1994) 8145
(h) Nash, Bursten & Ermler, J. Chem. Phys. 106 (1997) 5133

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)

8.6.8 The SRLC Pseudopotential at a Glance



*SRLC is **not** available for shaded elements*

(a) No pseudopotential; Pople 6-31G basis used

(b) Fuentealba, Preuss, Stoll & Szentpaly, Chem. Phys. Lett. 89 (1982) 418

(c) Fuentealba, Szentpály, Preuss & Stoll, J. Phys. B 18 (1985) 1287

(d) Bergner, Dolg, Küchle, Stoll & Preuss, Mol. Phys. 80 (1993) 1431

(e) Nicklass, Dolg, Stoll & Preuss, J. Chem. Phys. 102 (1995) 8942

(f) Schautz, Flad & Dolg, Theor. Chem. Acc. 99 (1998) 231

(g) Fuentealba, Stoll, Szentpaly, Schwerdtfeger & Preuss, J. Phys. B 16 (1983) L323

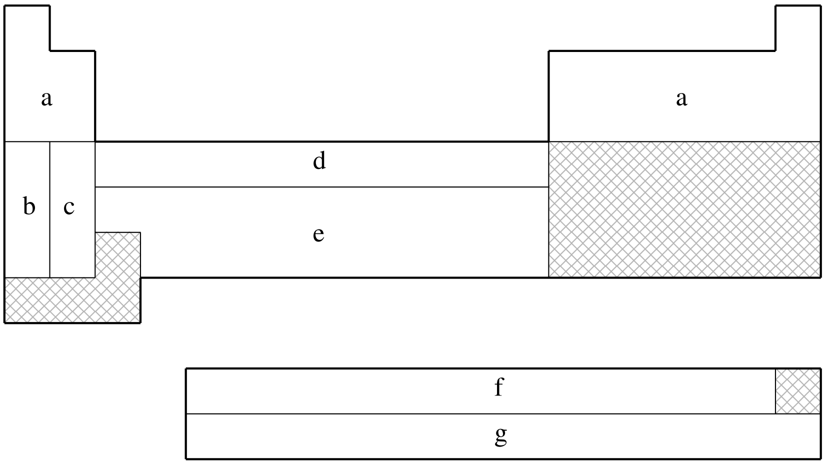
(h) Szentpaly, Fuentealba, Preuss & Stoll, Chem. Phys. Lett. 93 (1982) 555

(i) Küchle, Dolg, Stoll & Preuss, Mol. Phys. 74 (1991) 1245

(j) Küchle, to be published

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)

8.6.9 The SRSC Pseudopotential at a Glance



*SRSC is **not** available for shaded elements*

(a) No pseudopotential; Pople 6-311G* basis used

(b) Leininger, Nicklass, Küchle, Stoll, Dolg & Bergner, Chem. Phys. Lett. 255 (1996) 274

(c) Kaupp, Schleyer, Stoll & Preuss, J. Chem. Phys. 94 (1991) 1360

(d) Dolg, Wedig, Stoll & Preuss, J. Chem. Phys. 86 (1987) 866

(e) Andrae, Haeussermann, Dolg, Stoll & Preuss, Theor. Chim. Acta 77 (1990) 123

(f) Dolg, Stoll & Preuss, J. Chem. Phys. 90 (1989) 1730

(g) Küchle, Dolg, Stoll & Preuss, J. Chem. Phys. 100 (1994) 7535

Element	**Core**	**Max Projector**	**Valence**
H–Ar	*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)

References and Further Reading

[1] Ground-State Methods (Chapters 4 and 5).

[2] Basis Sets (Chapter 7).

[3] P. A. Christiansen, W. C. Ermler, and K. S. Pitzer, *Annu. Rev. Phys. Chem.* **36**, 407 (1985).

[4] P. Pyykko, *Chem. Rev.* **88**, 563 (1988).

[5] M. S. Gordon and T. R. Cundari, *Coord. Chem. Rev.* **147**, 87 (1996).

[6] G. Frenking et al., in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz and D. B. Boyd, volume 8, page 63, Wiley-VCH, New York, 1996.

[7] T. R. Cundari, M. T. Benson, M. L. Lutz, and S. O. Sommerer, in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz and D. B. Boyd, volume 8, page 145, Wiley-VCH, New York, 1996.

[8] J. Almlöf and O. Gropen, in *Reviews in Computational Chemistry*, edited by K. B. Lipkowitz and D. B. Boyd, volume 8, page 203, Wiley-VCH, New York, 1996.

[9] L. R. Kahn and W. A. Goddard III, *J. Chem. Phys.* **56**, 2685 (1972).

[10] 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 multi-program 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.

Chapter 9

Molecular Geometry Critical Points and *ab Initio* Molecular Dynamics

9.1 Equilibrium Geometries and Transition Structures

Molecular potential energy surfaces rely on the Born-Oppenheimer separation of nuclear and electronic motion. Minima on such energy surfaces correspond to the classical picture of equilibrium geometries and first-order saddle points for transition structures. Both equilibrium and transition structures are stationary points and therefore the energy gradients will vanish. Characterization of the critical point requires consideration of the eigenvalues of the Hessian (second derivative matrix). Equilibrium geometries have Hessians whose eigenvalues are all positive. Transition structures, on the other hand, have Hessians with exactly one negative eigenvalue. That is, a transition structure is a maximum along a reaction path between two local minima, but a minimum in all directions perpendicular to the 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. Thus, Q-CHEM incorporates the most advanced geometry optimization features currently available through Jon Baker's OPTIMIZE package (see Appendix A), a product of over ten years of research and development.

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
- optimization algorithm
- quality of the Hessian (and gradient)
- coordinate system

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 optimizing algorithm and the coordinate system are generally made by the OPTIMIZE package to maximize the rate of convergence. Users are able to override these decisions, but in general, this is not recommended.

Another consideration when trying to minimize the optimization time concerns the quality of the gradient and Hessian. A higher quality Hessian (*i.e.*, analytical *vs.* approximate) will in many cases lead to faster convergence and hence, fewer optimization steps. 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 9.1.

Level of Theory (Algorithm)	**Analytical Gradients**	**Maximum Angular Momentum Type**	**Analytical Hessian**	**Maximum Angular Momentum Type**
DFT	✓	h	✓	f
HF	✓	h	✓	f
ROHF	✓	h	✗	
MP2	✓	h	✗	
(V)OD	✓	h	✗	
(V)QCCD	✓	h	✗	
CIS (except RO)	✓	h	✓	f
CFMM	✓	h	✗	

Table 9.1: Gradients and Hessians currently available for geometry optimizations with maximum angular momentum types for analytical derivative calculations (for higher angular momentum, derivatives are computed numerically). Analytical Hessian is not yet available to tau-dependent functionals, such as BMK, M05 and M06 series.

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
- Initial Hessian and Hessian update options
- Reaction pathways using intrinsic reaction coordinates (IRC)
- Optimization of minimum-energy crossing points (MECPs) along conical seams

9.1.1 Job Control

Note: Users input starting geometry through the *$molecule* keyword.

Users must first define what level of theory is required. Refer back to previous sections regarding enhancements and customization of these features. METHOD (or EXCHANGE and CORRELATION, if required) and BASIS *$rem* variables must be set.

The remaining *$rem* variables are those specifically relating to the OPTIMIZE package.

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_HESSIAN

Determines the initial Hessian status.

TYPE:

STRING

DEFAULT:

DIAGONAL

OPTIONS:

DIAGONAL	Set up diagonal Hessian.
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 is expensive to compute.

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; delocalized internals are more efficient.

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 gradient component.

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.

9.1.2 Customization

GEOM_OPT_SYMFLAG

Controls the use of symmetry in OPTIMIZE.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 Make use of point group symmetry.
0 Do not make use of point group symmetry.

RECOMMENDATION:

Use 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 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 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 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 (GDISS default).

RECOMMENDATION:

Use default.

GEOM_OPT_LINEAR_ANGLE

Threshold for near linear bond angles (degrees).

TYPE:

INTEGER

DEFAULT:

165 degrees.

OPTIONS:

n User-defined level.

RECOMMENDATION:

Use 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 default, unless on a very flat potential, in which case a larger value should be used. See FDIFF_STEPSIZE_QFF for third and fourth derivatives.

9.1.3 Example

Example 9.1 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

```
$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
```

9.2 Constrained Optimization

9.2.1 Introduction

Constrained optimization refers to the optimization of molecular structures (transition or equilibrium) in which certain parameters (*e.g.*, bond lengths, bond angles or dihedral angles) are fixed. Jon Baker's OPTIMIZE package implemented in the Q-CHEM program has been modified to handle constraints directly in delocalized internal coordinates using the method of Lagrange multipliers (see Appendix A). Constraints are imposed in an *$opt* keyword section of the input file.

Features of constrained optimizations in Q-CHEM are:

- Starting geometries do not have to satisfy imposed constraints.
- Delocalized internal coordinates are the most efficient system for large molecules.
- Q-CHEM's free format *$opt* section allows the user to apply constraints with ease.

Note: The *$opt* input section is case-insensitive and free-format, except that there should be no space at the start of each line.

9.2.2 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 angstroms; *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
```

9.2.3 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 coordinates will be Cartesian.

9.2.4 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.

9.2.5 Dummy Atom Placement in Dihedral Constraints

Bond and dihedral angles cannot be constrained in Cartesian optimizations to exactly 0° or ±180°. 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 9.2.5. Note that the order of atoms is important to obtain the correct signature on the dihedral angles. For a 0° dihedral constraint, J and K should be switched in the definition of the second torsion constraint in Cartesian coordinates.

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 9.2: Comparison of dihedral angle constraint method for adopted 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.

9.2.6 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.

9.2.7 Example

Example 9.2 Methanol geometry optimization with constraints.

```
$comment
   Methanol geom opt with constraints in bond length and bond angles.
$end

$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
   GEOM_OPT_PRINT    6
   JOBTYPE           opt
   METHOD            hf
   BASIS             3-21g
$end

$opt
CONSTRAINT
stre  1  6  1.8
bend  2  1  4  110.0
bend  2  1  5  110.0
ENDCONSTRAINT
$end
```

9.2.8 Summary

```
$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
```

9.3 Potential Energy Scans

It is often useful to scan potential energy surfaces (PES). In a S_N1 chemical reaction, for example, such a scan can give an idea about how the potential energy changes upon bond breaking. In more complicated reactions involving multiple bond breaking/formation, a multi-dimensional PES reveals one (or more) reaction pathway connecting the reactant to the transition state and finally to the product. In force-field development, 1-dimensional torsional scan is essential for the optimization of accurate dihedral parameters. Finally, Ramachandran plots, which are essentially 2-dimensional torsional scans, are key tools for studying conformational changes of peptides and proteins.

Q-CHEM supports 1-dimensional and 2-dimensional PES scans in which one or two coordinates (e.g., stretching, bending, torsion) are being scanned whereas all other degrees of freedom are being optimized. For these calculations, JOBTYPE needs to be set to PES_SCAN, and the following input section (with one or two motions) should be specified:

```
$scan
stre  atom1  atom2  value1 value2 incr
...
bend  atom1  atom2  atom3  value1 value2 incr
...
tors  atom1  atom2  atom3  atom4  value1 value2 incr
...
$end
```

The example below allows us 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, $\cdots$, 165, 180 degrees.

Example 9.3 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
```

A 2-dimensional scan of butane can be performed using the following input:

Example 9.4 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
```

Here the first dimension is the the scan of the C1-C2-C3-C4 dihedral angle from -180 to 180 degree at 30-degree intervals. For the second dimension, we scan the C2-C3 bond length from 1.5 Å to 1.6 Å at 0.05 Å increments.

9.4 Minimum-Energy Crossing Points

Conical intersections are the regions of the potential energy surface characterized by degeneracy between two or more electronic states with the same symmetry. For a two-state intersection, the intersection consists of an $(N-2)$-dimensional hypersurface (the *seam space*) within which the two states are degenerate, where N is the number of internal coordinates of the molecule. (The remaining two dimensions for a *branching space* in which the degeneracy is lifted by any infinitesimal displacement.) Radiationless transitions between the two electronic states are likely to occur in an around a conical seam. The first step in any study of nonadiabatic excited-state dynamics is often an exploration of the geometries and energies of the lowest-energy point within the seam space, which is the so-called *minimum-energy crossing point* (MECP). In some sense, the MECP is to internal conversion and photochemical processes what the transition state is to single-surface chemical reactions.

The two-dimensional branching space between electronic states I and J is spanned by a pair of vectors that are usually denoted $\mathbf{g}$ and $\mathbf{h}$ [2]:

$$\mathbf{g}^{IJ} = \hat{\boldsymbol{\nabla}}_{\mathbf{R}}\big(E_I(\mathbf{R}) - E_J(\mathbf{R})\big) \tag{9.1a}$$

$$\mathbf{h}^{IJ} = \langle \Psi_I|\hat{\boldsymbol{\nabla}}_{\mathbf{R}}|\Psi_J\rangle \ . \tag{9.1b}$$

While $\mathbf{g}^{IJ}$ is available analytically for any electronic structure method that has analytic excited-state gradients, analytic implementations of the nonadiabatic coupling vector $\mathbf{h}^{IJ}$ are not routinely available. For this reason, several algorithms have been developed to optimize MECPs without the need to evaluate $\mathbf{h}^{IJ}$, and three such algorithms are available in Q-CHEM. The first of these is a penalty-constrained optimization algorithm developed by Levine *et al.* [3], in which the objective function that is optimized to locate the MECP is

$$F_\sigma(\mathbf{R}) = \tfrac{1}{2}\big(E_I(\mathbf{R}) + E_J(\mathbf{R})\big) + \sigma\frac{\big(E_I(\mathbf{R}) - E_J(\mathbf{R})\big)^2}{\big(E_I(\mathbf{R}) - E_J(\mathbf{R})\big) + \alpha} \ , \tag{9.2}$$

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. Optimization of F_σ proceeds iteratively for increasingly large values of the parameter σ. The second MECP optimization algorithm is a simplification of the first one that we call the "direct" method. Here, the gradient of the objective function is

$$\mathbf{G} = \mathbf{P}\mathbf{G}_{\text{mean}} + 2(E_J - E_I)\mathbf{G}_{\text{diff}} \ , \tag{9.3}$$

where

$$\mathbf{G}_{\text{mean}} = \tfrac{1}{2}(\mathbf{G}_I + \mathbf{G}_J) \tag{9.4}$$

is the mean energy gradient (with $\mathbf{G}_i = \hat{\boldsymbol{\nabla}}_{\mathbf{R}}E_i(\mathbf{R})$ being the nuclear gradient for state i), and

$$\mathbf{G}_{\text{diff}} = \frac{\mathbf{G}_J - \mathbf{G}_I}{||\mathbf{G}_J - \mathbf{G}_I||} \tag{9.5}$$

is the normalized difference gradient. Finally,

$$\mathbf{P} = \mathbf{1} - \mathbf{G}_{\text{diff}}^\top\mathbf{G}_{\text{diff}} \tag{9.6}$$

projects the gradient difference direction out of the mean energy gradient in Eq. (9.3). The third and final MECP optimization algorithm that is available in Q-CHEM is the branching-plane updating method developed by Maeda *et al.* [4]. This algorithm uses a gradient that is similar to that in Eq. (9.3) but projects out not just $\mathbf{G}_{\text{diff}}$ in Eq. (9.6) but also a second vector that is orthogonal to it.

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. Note, however, that all linear-response methods (a category that includes CIS, TDDFT, and EOM-XX-CCSD) incorrectly describe the topology of any conical intersection that involves the reference (usually, ground) state. Specifically, it can be shown in such cases that only a one-dimensional branching space is obtained [5]. If conical intersections involving the ground state are to be described correctly, one must use a different reference state, which can be accomplished with spin-flip (SF) methods [6–8]. In particular, using a high-spin triplet reference state, the ground-state singlet (S_0) appears as an excitation (possibly with a negative excitation energy) alongside the excited-state singlets, $S_1, S_2, \ldots$, so there is no topology problem in describing an S_0/S_1 conical intersection. Accurate MECP geometries, as compared to multireference configuration-interaction benchmarks, have been reported [7]. It should be noted, however, that the $M_S = 0$ component of the triplet state also shows up as an excitation in SF methods, and while Q-CHEM attempts to identify this triplet state automatically (based on a threshold for $\langle \hat{S}^2 \rangle$), severe spin contamination can sometimes hamper one's ability to distinguish this state from the singlet excited states [7].

9.4.1 Job Control

MECP_OPT

Determines whether we are doing MECP optimizations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Do MECP optimizations.

FALSE Don't do MECP optimizations.

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 symmetries, but the branching-plane updating method is more efficient for larger molecules. However, the latter does not work if the two states have different symmetries. If using branching plane updating method, GEOM_OPT_COORDS must be set to 0 in the *$rem* section (*i.e.*, this algorithm is available in Cartesian coordinates only). The penalty-constrained method converges slowly and is suggested only when the other methods do not work.

MECP_STATE1

Determines the first state for crossing.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

$[i,j]$ find the jth excited state with the total spin of 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

Determines the second state for crossing.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

$[i,j]$ find the jth excited state with the total spin of 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 singlet or triplet in unrestricted calculations.

TYPE:

INTEGER

DEFAULT:

120

OPTIONS:

None

RECOMMENDATION:

If set to 120, the states with $\langle \hat{S}^2 \rangle > 1.20$ are treated as triplet states, with 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 Default.

9.4.2 Examples

The MECP between the S_2 and S_3 states of NO_2^- is optimized using the direct method:

Example 9.5 MECP optimization for NO_2^- using SOS-CIS(D0)

```
$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
```

The MECP between the S_0 and S_1 states of ethylidene is optimized using the branching-plane update method:

Example 9.6 MECP optimization for ethylidene using SF-TDDFT

```
$molecule
0 3
C  0.0446266041 -0.2419241370 0.3571573801
C  0.0089051507 0.6727548956 1.4605006396
H  0.9284257388 -0.1459163900 -0.2720952334
H -0.8310326564 -0.1926895078 -0.2885298629
H -0.0092388670 0.9611331703 2.4799363398
H  0.0683140308 -1.2533580302 0.7788470826
$end

$rem
jobtype       opt
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

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
$end
```

The MECP between S_0 and S_1 states of twisted-pyramidalized ethylene is optimized using the penalty-constrained method:

Example 9.7 MECP optimization for ethylene using SF-TDDFT

```
$molecule
0 3
C -0.0158897609 0.0735325545 -0.0595597308
C  0.0124274563 -0.0024687284 1.3156941918
H  0.8578762360 0.1470146857 -0.7105293671
H -0.9364708648 -0.0116961121 -0.6267613144
H  0.7645577838 0.6633816890 1.7625731128
H  0.7407739370 -0.8697640880 1.3285831079
$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
$end
```

The MECP between $\tilde{B}^1A_2$ and $\tilde{A}^1B_2$ states of the N_3^+ ion using the direct method:

Example 9.8 MECP optimization for N_3^+ using EOM-EE-CCSD

```
$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
$END
```

9.5 Intrinsic Reaction Coordinates

The concept of a reaction path, although seemingly well-defined chemically (*i.e.*, how the atoms in the system move to get from reactants to products), is somewhat ambiguous mathematically because, using the usual definitions, it depends on the coordinate system. Stationary points on a potential energy surface are *independent* of coordinates, but the path connecting them is not, and so different coordinate systems will produce different reaction paths. There are even different definitions of what constitutes a "reaction path"; the one used in Q-CHEM is based on the intrinsic reaction coordinate (IRC), first defined in this context by Fukui [9]. This is essentially a series of 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" path. A series of small steepest descent steps will zig-zag along the actual reaction path (this is known as "stitching"). Ishida *et al.* [10] 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; this was subsequently improved by Schmidt *et al.* [11], and is the method we have adopted. For the first step downhill from the transition state this approach cannot be used (as the gradient is zero); instead a step is taken along the Hessian mode corresponding to the imaginary frequency.

The reaction path can be defined and followed in Z-matrix coordinates, Cartesian coordinates or mass-weighted Cartesians. The latter represents the "true" IRC as defined by Fukui [9]. However, if the main reason for following the reaction path is simply to determine which minima a given transition state connects (perhaps the major use), then it doesn't matter which coordinates are used. In order to use the IRC code the transition state geometry and the exact Hessian must be available. These must be computed via transition state (JOBTYPE = TS) and frequency calculation (JOBTYPE = FREQ) respectively.

9.5.1 Job Control

An IRC calculation is invoked by setting the JOBTYPE *$rem* to RPATH.

RPATH_COORDS
Determines which coordinate system to use in the IRC search.
TYPE:
INTEGER
DEFAULT:
0
OPTIONS:
0 Use mass-weighted coordinates.
1 Use Cartesian coordinates.
2 Use Z-matrix coordinates.
RECOMMENDATION:
Use default.

RPATH_DIRECTION
Determines the direction of the eigen mode 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 eigen mode.
-1 Descend in the negative direction of the eigen mode.
RECOMMENDATION:
It is usually not possible to determine in which direction to go *a priori*, and therefore both directions will need to be considered.

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 thousandths of *a.u.*).

TYPE:

INTEGER

DEFAULT:

150 corresponding to a step size of 0.15 *a.u.*.

OPTIONS:

n Step size = n/1000.

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.

RECOMMENDATION:

Use 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 default, 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.

9.5.2 Example

Example 9.9

```
$molecule
   0  1
   C
   H  1  1.20191
   N  1  1.22178  2  72.76337
$end

$rem
   JOBTYPE    freq
   BASIS      sto-3g
   METHOD     hf
$end

@@@

$molecule
   read
$end

$rem
   JOBTYPE            rpath
   BASIS              sto-3g
   METHOD             hf
   SCF_GUESS          read
   RPATH_MAX_CYCLES   30
$end
```

9.6 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, where the search space dimensionality is high. Interpolation algorithms are promising methods 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 [12, 13] and the string method [14] 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 an interpolation method heavily relies on the choice of the initial reaction pathway, and a poorly chosen initial pathway can cause slow convergence, or convergence to an incorrect pathway. The freezing string [15, 16] and growing string methods [17] offer elegant solutions to this problem, in which two string fragments (one from the reactant and the other from the product state) are grown until the two fragments join. The freezing string method offers a choice between Cartesian and Linear Synchronous Transit (LST) interpolation methods. It also allows users to choose between conjugate gradient and quasi-Newton optimization techniques. It can be invoked by (JOBTYPE = FSM) using the following *$rem* keyword:

FSM_NNODE
Specifies the number of nodes along the string
TYPE:
INTEGER
DEFAULT:
Undefined
OPTIONS:
N number of nodes in FSM calculation
RECOMMENDATION:
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 9.7).

FSM_NGRAD
Specifies the number of perpendicular gradient steps used to optimize each node
TYPE:
INTEGER
DEFAULT:
Undefined
OPTIONS:
N number of perpendicular gradients per node
RECOMMENDATION:
4. 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:

2. In most cases, LST is superior to Cartesian interpolation.

FSM_OPT_MODE

Specifies the method of optimization

TYPE:

INTEGER

DEFAULT:

Undefined

OPTIONS:

1 Conjugate gradients

2 Quasi-Newton method with BFGS Hessian update

RECOMMENDATION:

2. The quasi-Newton method is more efficient when the number of nodes is high.

References [15] and [16] provide a guide to a typical use of this method. The following example input will be helpful for setting up the job:

Example 9.10

```
$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   -0.8556026   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
```

The *$molecule* section should include geometries for two optimized intermediates separated by **** symbols. The order of the atoms is important, as Q-CHEM will assume atom X in the reaction complex moves to atom X in the product complex. The FSM string is printed out in the file 'stringfile.txt', which is an XYZ file containing the structures connecting reactant to product. Each node along the path is labeled with its energy. The highest energy node can be taken from this file and used to run a TS search, as detailed in section 9.1. If the string returns a pathway that is unreasonable, double check whether the atoms in the two input geometries are in the correct order.

9.7 Hessian-Free Transition State Search

Once a guess structure to the transition state is obtained, standard eigenvector-following methods such as the Partitioned-Rational Function Optimization (P-RFO) [18] 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. 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 [19]. 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 [20] [21] 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.

An example of this 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 is shown below:

Example 9.11

```
    $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   -0.8556026   0.079077   1.213023
    $end

    $rem
```

```
jobtype         fsm
fsm_ngrad   3
fsm_nnode   18
fsm_mode     2
fsm_opt_mode 2
method      b3lyp
basis       6-31g
symmetry false
sym_ignore true
$end

@@@

$rem
jobtype ts
scf_guess read
geom_opt_hessian read
max_scf_cycles 250
geom_opt_dmax 50
geom_opt_max_cycles 100
method b3lyp
basis 6-31g
symmetry false
sym_ignore true
$end

$molecule
read
$end
```

9.8 Improved Dimer Method

Once a good approximation to the minimum energy pathway is obtained, *e.g.*, with the help of an interpolation algorithm such as the growing string method, local surface walking algorithms can be used to determine the exact location of the saddle point. Baker's partitioned rational function optimization (P-RFO) method, which utilizes an approximate or exact Hessian, has proven to be a very powerful method for this purpose.

The dimer method [22] on the other hand, is a mode following algorithm that utilizes only the curvature along one direction in configuration space (rather than the full Hessian) and requires only gradient evaluations. It is therefore especially applicable for large systems where a full Hessian calculation is very time consuming, or for saddle point searches where the eigenvector of the lowest Hessian eigenvalue of the starting configuration does not correspond to the reaction coordinate. A recent modification of this method has been developed [23, 24] to significantly reduce the influence of numerical noise, as it is common in quantum chemical methods, on the performance of the dimer algorithm, and to significantly reduce its computational cost. This improved dimer method has recently been implemented within Q-CHEM.

9.9 *Ab initio* Molecular Dynamics

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*). This procedure, in which the forces on the nuclei are evaluated on-the-fly, is known variously as "direct dynamics", "*ab initio* molecular dynamics", 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 gradients; however, methods lacking analytic gradients will be prohibitively expensive, except for very small systems.

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 *a.u.* 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 (QCT-MD, see below).

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 2.7.) 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 Hartrees).
- 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 angstroms.
- 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: An xyz-formatted version of NucCarts for viewing trajectories in an external visualization program (new in v.4.0).

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 [25].
- DeltaNorm: Records the norm and largest magnitude element of the curvy-steps rotation angle matrix Δ defined in Ref. 25. 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.

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.

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_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 [26], based on a suggestion by Pulay and Fogarasi [27], is available for this purpose.

The Fock matrix elements $\mathbf{F}_{\mu\nu}$ 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:

$$\mathbf{F}_{\mu\nu}(t) = \sum_{n=0}^{N} c_n\, t^n \tag{9.7}$$

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. 26, 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 [28]. While these equations are linear, their dimensionality necessitates an iterative solution [29, 30], which, in practice, looks much like the SCF equations. Extrapolation is again useful here [31], 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 (4,2)=(M,N) 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 [25], 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 $\tau_{\rm nuc}$ is set by the period of the fastest vibrational frequency, and electronic oscillations, whose time scale $\tau_{\rm elec}$ may be estimated according to [25]

$$\tau_{\rm elec} \geq \sqrt{\mu/(\varepsilon_{\rm LUMO} - \varepsilon_{\rm HOMO})} \tag{9.8}$$

A conservative estimate, suggested in Ref. 25, is that essentially exact classical dynamics is attained when $\tau_{\rm nuc} > 10\ \tau_{\rm elec}$. 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 $\tau_{\rm elec}$, 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 [26, 27].

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 [25] for examples and discussion.

Additional job control variables for *ab initio* molecular dynamics.

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.

RECOMMENDATION:

This variable need only be specified in the event that velocities are not specified explicitly in a *$velocity* section.

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.

9.9.1 Examples

Example 9.12 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 9.13 Propagating $F-(H_2O)_4$ on its first excited-state potential energy surface, calculated at the CIS level.

```
$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         827      (500 fs)
$end
```

Example 9.14 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        sbkjc
$end

@@@

$molecule
   read
$end
```

```
$rem
   JOBTYPE           aimd
   METHOD            b3lyp
   ECP               sbkjc
   TIME_STEP         14
   AIMD_STEPS        500
   AIMD_METHOD       curvy
   AIMD_FICT_MASS    360
   AIMD_INIT_VELOC   zpe
$end
```

9.9.2 AIMD with Correlated Wavefunctions

While the number of time steps required in most AIMD trajectories dictates that economical (typically SCF-based) underlying electronic structure methods are required, other methods are also now possible. Any method with available analytic forces can be utilized for BOMD. Currently, Q-CHEM can perform AIMD simulations with HF, DFT, MP2, RI-MP2, CCSD, and CCSD(T). The RI-MP2 method, especially when combined with Fock matrix and response equation extrapolation, is particularly effective as an alternative to DFT-based dynamics.

9.9.3 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 autocorrelation functions. Q-CHEM can provide both the vibrational spectral density from the velocity autocorrelation function

$$D(\omega) \propto \int_{-\infty}^{\infty} dt\, e^{-i\omega t} \langle \vec{v}(0)\cdot\vec{v}(t)\rangle \tag{9.9}$$

and infrared absorption intensity from the dipole autocorrelation function

$$I(\omega) \propto \frac{\omega}{2\pi} \int_{-\infty}^{\infty} dt\, e^{-i\omega t} \langle \vec{\mu}(0)\cdot\vec{\mu}(t)\rangle \tag{9.10}$$

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 autocorrelation functions can exhibit contaminating information if the molecule is allowed to rotate/translate. 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. The trans/rot correction in Q-CHEM is activated by the PROJ_TRANSROT keyword.

AIMD_NUCL_VACF_POINTS

Number of time points to utilize in the velocity autocorrelation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute velocity autocorrelation function.
$1 \leq n \leq$ AIMD_STEPS	Compute velocity autocorrelation function for last n time steps of the trajectory.

RECOMMENDATION:

If the VACF is desired, set equal to AIMD_STEPS.

AIMD_NUCL_DACF_POINTS

Number of time points to utilize in the dipole autocorrelation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute dipole autocorrelation function.
$1 \leq n \leq$ AIMD_STEPS	Compute dipole autocorrelation function for last n timesteps of the trajectory.

RECOMMENDATION:

If the DACF is desired, set equal to AIMD_STEPS.

AIMD_NUCL_SAMPLE_RATE

The rate at which sampling is performed for the velocity and/or dipole autocorrelation 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 autocorrelation 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/DACF are 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 utilized to remove artificial, contaminating peaks stemming from translational and/or rotational motion. Recommend setting to TRUE for all dynamics-based spectral simulations.

9.9.4 Quasi-Classical Molecular Dynamics

Molecular dynamics simulations based on quasi classical trajectories (QCT-MD) [32–34] put vibrational energy into each mode in the initial velocity setup step. We (as well as others [35]) have found that this can improve on the results of purely classical simulations, for example in the calculation of photoelectron [36] or infrared spectra [37]. Improvements include better agreement of spectral linewidths with experiment at lower temperatures or 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. 36.

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 9.15 Simulating the IR spectrum of water using QCT-MD.

```
$comment
  Don’t forget to run a frequency calculation prior to this job.
$end

$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     aimd
  input_bohr  true
  method      hf
  basis       3-21g
  scf_convergence 6
  ! AIMD input
  time_step                   20      (in atomic units)
  aimd_steps               12500      6 ps 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
  ! IR spectral sampling
  aimd_moments                 1
  aimd_nucl_sample_rate        5
  aimd_nucl_vacf_points     1000
  ! QCT-specific settings
  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 [36]. 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 wavefunction 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 wavefunction 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 wavefunction.
$-n$	Generates n random geometries sampled from the harmonic vibrational wavefunction.

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 quasiclassical MD). We have included another script in the aimdman/tools directory to help with the calculation of vibrationally averaged properties (monte_geom).

Example 9.16 MC sampling of the vibrational wavefunction for HCl.

```
$comment
  Generates 1000 random geometries for HCl based on the harmonic vibrational
  wavefunction at 1 Kelvin. The wavefunction is averaged over 1000
  sets of random vibrational quantum numbers (\ie{}, the ground state in
  this case due to the low temperature).
$end

$molecule
  0 1
  H           0.000000    0.000000   -1.216166
  Cl          0.000000    0.000000    0.071539
$end

$rem
  jobtype                               aimd
  method                              B3LYP
  basis                          6-311++G**
  scf_convergence                         1
  SKIP_SCFMAN                             1
  maxscf                                  0
  xc_grid                                 1
  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)
  ! set aimd_qct_which_trajectory to the desired
  ! trajectory number
  aimd_qct_which_trajectory           -1000
  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 keyword. 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 9.17 User control over the QCT variables.

```
$comment
  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.
$end

...

$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" (see, *e.g.*, Ref. 38), 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 [36].

We have implemented a routine that monitors the kinetic energy within each normal mode along the trajectory and that is automatically switched on for quasiclassical 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 (see Ref. 38 for a different approach to the zero-point leakage problem). Furthermore, this monitoring routine prints the squares of the (harmonic) vibrational wavefunction 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.

9.10 *Ab initio* Path Integrals

Even in cases where the Born-Oppenheimer separation is valid, solving the electronic Schroedinger equation—Q-CHEM’s main purpose—is still only half the battle. The remainder involves the solution of the *nuclear* Schroedinger equation for its resulting eigenvalues/functions. This half is typically treated by the harmonic approximation at critical points, but anharmonicity, tunneling, and low-frequency “floppy” motion can lead to extremely delocalized nuclear distributions, particularly for protons and non-covalently bonded systems.

While the Born-Oppenheimer separation allows for a local solution of the electronic problem (in nuclear space), the nuclear half of the Schroedinger 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 thermal, equilibrium distributions, the path integral (PI) formalism of Feynman provides both an elegant and computationally feasible alternative. The equilibrium partition function, for example, may be written as a trace of the thermal, configuration-space density matrix:

$$\begin{aligned} Z &= Tr\left[e^{-\beta\hat{H}}\right] \\ &= \int dx\langle x|e^{-\beta\hat{H}}|x\rangle \\ &= \int dx\rho(x,x;\beta) \end{aligned} \tag{9.11}$$

Solving for the partition function directly in this form is equally difficult, as it still requires the eigenvalues/ eigenstates of $\hat{H}$. By inserting $N-1$ resolutions of the identity, however, this integral may be converted to the following form

$$Z = \int dx_1 \int dx_2 \cdots \int dx_N \; \rho\left(x_1, x_2; \frac{\beta}{N}\right) \rho\left(x_2, x_3; \frac{\beta}{N}\right) \cdots \rho\left(x_N, x_1; \frac{\beta}{N}\right) \tag{9.12}$$

While this additional integration appears to be a detriment, the ability to use a high-temperature ($\frac{\beta}{N}$) form of the density matrix

$$\rho\left(x, x'; \frac{\beta}{N}\right) = \sqrt{\frac{mN}{2\pi\beta\hbar^2}}\, e^{-\frac{mN}{2\beta\hbar^2}\left(x-x'\right)^2 - \frac{\beta}{N}\left(\frac{V(x)+V(x')}{2}\right)} \tag{9.13}$$

renders this path-integral formulation a net win. By combining the N time slices, the partition function takes the following form (in 1-D):

$$\begin{aligned} Z &= \left(\frac{mN}{2\pi\beta\hbar^2}\right)^{\frac{N}{2}} \int dx_1 \int dx_2 \cdots \int dx_N e^{-\frac{\beta}{N}\left[\frac{mN^2}{2\beta^2\hbar^2}\sum_i^N (x_i - x_{i+1})^2 + \sum_i^N V(x_i)\right]} \\ &\propto \int e^{-\beta V_{eff}} \end{aligned} \tag{9.14}$$

with the implied cyclic condition $x_{N+1} = x_1$. Here, $V(x)$ is the potential function on which the "beads" move (the electronic potential generated by Q-CHEM). The resulting integral, as shown in the last line above, is nothing more than a *classical* configuration integral in an N-dimensional space. The effective potential appearing above describes an N-bead "ring polymer," of which neighboring beads are harmonically coupled. The exponentially scaling, non-local nuclear problem has, therefore, been mapped onto an entirely classical problem, which is amenable to standard treatments of configuration sampling. These methods typically involve (thermostated) molecular dynamics or Monte Carlo sampling; only the latter is currently implemented in Q-CHEM. Importantly, N is reasonably small when the temperature is not too low: room-temperature systems involving H atoms typically are converged with roughly 30 beads. Therefore, fully quantum mechanical nuclear distributions may be obtained at a cost only roughly 30 times a classical simulation.

Path integral Monte Carlo (PIMC) is an entirely new job type in Q-CHEM and is activated by setting JOBTYP to PIMC.

9.10.1 Classical Sampling

The 1-bead limit of the above expressions is simply classical configuration sampling. When the temperature (controlled by the PIMC_TEMP keyword) is high or only heavy atoms are involved, the classical limit is often appropriate. The path integral machinery (with 1 "bead") may be utilized to perform classical Boltzmann sampling. The 1D partition function, for example, is simply

$$Z = \int dx\, e^{-\beta V(x)} \tag{9.15}$$

9.10.2 Quantum Sampling

Using more beads includes more quantum mechanical delocalization (at a cost of roughly N times the classical analog). This main input variable—the number of time slices (beads)—is controlled by the PIMC_NBEADSPERATOM keyword. The ratio of the inverse temperature to beads ($\frac{\beta}{N}$) 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 at least $\approx 10^5$, depending on the desired level of statistical convergence. A warmup run, in which the ring polymer is allowed to equilibrate (without accumulating statistics) can be performed by setting the PIMC_WARMUP_MCMAX keyword.

Much like *ab initio* molecular dynamics simulations, the main results of PIMC jobs in Q-CHEM are not in the job output file. Rather, they are compiled in the "PIMC" subdirectory of the user's scratch directory (*$QCSCRATCH/PIMC*). Therefore, PIMC jobs should *always* be run with the *-save* option. The output files do contain some useful information, however, including a basic data analysis of the simulation. Average energies (thermodynamic estimator), bond lengths (less than 5Å), bond length standard deviations and errors are printed at the end of the output file. The *$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: xyz-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 the ordering of the data.

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 naive 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 utilizes the simplest of these options: Levy flights. An n-bead snippet ($n < N$) of the ring polymer is first chosen at random, with the length controlled by the PIMC_SNIP_LENGTH keyword. Between the endpoints of this snippet, 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 autocorrelation 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.

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 1-bead limit is purely classical sampling; the infinite-bead limit is 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. Recommend setting to at least 100000 for converged simulations.

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 utilized. For Cartesian and normal-mode moves, the maximum displacement is adjusted during the warmup 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 utilized.

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 warmup 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.)

9.10.3 Examples

Example 9.18 Path integral Monte Carlo simulation of H_2 at room temperature

```
$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    10000     !Equilibration run
   PIMC_MCMAX           100000    !Production run
   PIMC_MOVETYPE        2         !Levy flights
   PIMC_SNIP_LENGTH     10        !Moves 8 beads per MC step (10-endpts)
$end
```

Example 9.19 Classical Monte Carlo simulation of a water molecule at 500K

```
$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    10000   !Equilibration run
   PIMC_MCMAX           100000  !Production run
   PIMC_MOVETYPE        0       !Cartesian displacements (ok for 1 bead)
   PIMC_ACCEPT_RATE     40      !During warmup, adjusts step size to 40% acceptance
$end
```

References and Further Reading

[1] Geometry Optimization (Appendix A).

[2] D. R. Yarkony, *J. Phys. Chem. A* **105**, 6277 (2001).

[3] B. G. Levine, J. D. Coe, and T. J. Martinez, *J. Phys. Chem. B* **112**, 405 (2008).

[4] S. Maeda, K. Ohno, and K. Morokuma, *J. Chem. Theory Comput.* **6**, 1538 (2010).

[5] B. G. Levine, C. Ko, J. Quenneville, and T. J. Martínez, *Mol. Phys.* **104**, 1039 (2006).

[6] N. Minezawa and M. S. Gordon, *J. Phys. Chem. A* **115**, 7901 (2011).

[7] X. Zhang and J. M. Herbert, *J. Phys. Chem. B* **118**, 7806 (2014).

[8] X. Zhang and J. M. Herbert, *J. Chem. Phys.* **141**, 064104 (2014).

[9] K. Fukui, *J. Phys. Chem.* **74**, 4161 (1970).

[10] K. Ishida, K. Morokuma, and A. Komornicki, *J. Chem. Phys.* **66**, 215 (1977).

[11] M. W. Schmidt, M. S. Gordon, and M. Dupuis, *J. Am. Chem. Soc.* **107**, 2585 (1985).

[12] G. Mills and H. H. Jónsson, *Phys. Rev. Lett.* **72**, 1124 (1994).

[13] G. Henkelman and H. Jónsson, *J. Chem. Phys.* **113**, 9978 (2000).

[14] E. Weinan, W. Ren, and E. Vanden-Eijnden, *Phys. Rev. B* **66**, 052301 (2002).

[15] A. Behn, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon, *J. Chem. Phys.* **135**, 224108 (2011).

[16] S. M. Sharada, P. M. Zimmerman, A. T. Bell, and M. Head-Gordon, *J. Chem. Theory Comput.* **8**, 5166 (2012).

[17] B. Peters, A. Heyden, A. T. Bell, and A. Chakraborty, *J. Chem. Phys.* **120**, 7877 (2004).

[18] J. Baker, *J. Comput. Chem.* **7**, 385 (1986).

[19] Y. Kumeda, D. J. Wales, and L. J. Munro, *Chem. Phys. Lett.* **341**, 185 (2001).

[20] J. Baker, A. Kessi, and B. Delley, *J. Chem. Phys.* **105**, 192 (1996).

[21] G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay, *J. Am. Chem. Soc.* **114**, 8191 (1992).

[22] G. Henkelman and H. Jónsson, *J. Chem. Phys.* **111**, 7010 (1999).

[23] A. Heyden, B. Peters, A. T. Bell, and F. J. Keil, *J. Phys. Chem. B* **109**, 1857 (2005).

[24] A. Heyden, A. T. Bell, and F. J. Keil, *J. Chem. Phys.* **123**, 224101 (2005).

[25] J. M. Herbert and M. Head-Gordon, *J. Chem. Phys.* **121**, 11542 (2004).

[26] J. M. Herbert and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **7**, 3269 (2005).

[27] P. Pulay and G. Fogarasi, *Chem. Phys. Lett.* **386**, 272 (2004).

[28] C. M. Aikens et al., *Theor. Chem. Acc.* **110**, 233 (2004).

[29] J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley, *Int. J. Quantum Chem. Symp.* **13**, 225 (1979).

[30] P. P. Kombrath, J. Kong, T. R. Furlani, and M. Head-Gordon, *Mol. Phys.* **100**, 1755 (2002).

[31] R. P. Steele and J. C. Tully, *Chem. Phys. Lett.* **500**, 167 (2010).

[32] M. Karplus, R. N. Porter, and R. D. Sharma, *J. Chem. Phys.* **43**, 3259 (1965).

[33] R. Porter, *Annu. Rev. Phys. Chem.* **25**, 317 (1974).

[34] R. Porter, L. Raff, and W. H. Miller, *J. Chem. Phys.* **63**, 2214 (1975).

[35] A. Brown, B. J. Braams, K. Christoffel, Z. Jin, and J. M. Bowman, *J. Chem. Phys.* **119**, 8790 (2003).

[36] D. S. Lambrecht, G. N. I. Clark, T. Head-Gordon, and M. Head-Gordon, *J. Phys. Chem. A* **115**, 5928 (2011).

[37] E. Ramos-Cordoba, D. S. Lambrecht, and M. Head-Gordon, *Faraday Discuss.* **150**, 345 (2011).

[38] G. Czako, A. L. Kaledin, and J. M. Bowman, *J. Chem. Phys.* **132**, 164103 (2010).

Chapter 10

Molecular Properties and Analysis

10.1 Introduction

Q-CHEM has incorporated a number of molecular properties and wavefunction analysis tools, summarized as follows:

- Population analysis for ground and excited states
- Multipole moments for ground and excited states
- Calculation of molecular intracules
- Vibrational analysis (including isotopic substitution)
- Interface to the Natural Bond Orbital package
- Molecular orbital symmetries
- Orbital localization
- Localized Orbital Bonding Analysis
- Data generation for 2-D or 3-D plots
- Orbital visualization using the MOLDEN and MACMOLPLT programs
- Natural transition orbitals for excited states
- NMR shielding tensors and chemical shifts
- Quantum transport modeling in the Landauer approximation

In addition, Chapter 12 features analysis tools based on absolutely localized orbitals (energy decomposition and charge-transfer analysis).

10.2 Wavefunction Analysis

Q-CHEM performs a number of standard wavefunction analyses by default. Switching the *$rem* variable WAVEFUNCTION_ANALYSIS to FALSE will prevent the calculation of all wavefunction analysis features (described in this section). Alternatively, each wavefunction analysis feature may be controlled by its *$rem* variable. (The NBO package which is interfaced with Q-CHEM is capable of performing more sophisticated analyses. See later in this chapter and the NBO manual for details).

WAVEFUNCTION_ANALYSIS

Controls the running of the default wavefunction analysis tasks.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Perform default wavefunction analysis.

FALSE Do not perform default wavefunction analysis.

RECOMMENDATION:

None

Note: WAVEFUNCTION_ANALYSIS has no effect on NBO, solvent models or vibrational analyses.

10.2.1 Population Analysis

The one-electron charge density, usually written as

$$\rho(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}\phi_\mu(\mathbf{r})\phi_\nu(\mathbf{r}) \tag{10.1}$$

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

$$N = 2\sum_{a}^{N/N22} |\psi_a(\mathbf{r})|^2 \, d\mathbf{r} \tag{10.2}$$

We can substitute the atomic orbital (AO) basis expansion of ψ_a into Eq. (10.2) to obtain

$$N = \sum_{\mu\upsilon} P_{\mu\upsilon}S_{\mu\upsilon} = \sum_{\mu} (\mathbf{PS})_{\mu\mu} = \mathrm{Tr}(\mathbf{PS}) \tag{10.3}$$

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 of the 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 a Mulliken population analysis [5]. Q-CHEM performs a Mulliken population analysis by default.

POP_MULLIKEN

Controls running of Mulliken population analysis.

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE (or 1)

OPTIONS:

FALSE	(or 0) Do not calculate Mulliken Population.
TRUE	(or 1) Calculate Mulliken population
2	Also calculate shell populations for each occupied orbital.
-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 a Löwdin population analysis instead of a Mulliken.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Do not calculate Löwdin Populations.
TRUE	Run Löwdin Population analyses instead of Mulliken.

RECOMMENDATION:

None

Although conceptually simple, Mulliken population analyses suffer from a heavy 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 [6], 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).

A more stable alternative to Mulliken or Löwdin charges are the so-called "CHELPG" charges ("Charges from the Electrostatic Potential on a Grid") [7]. These are the atom-centered charges 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 [7], 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 [8]. (For any particular geometry, however, numerical values of the charges are quite similar to those obtained using the original algorithm.) Note, however, that the Breneman-Wiberg approach uses a Cartesian grid and becomes expensive for large systems. (It is especially expensive when CHELPG charges are used in QM/MM-Ewald calculations [9].) For that reason, an alternative procedure based on atom-centered Lebedev grids is also available [9], 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.

CHELPG_HEAD

Sets the “head space” [7] (radial extent) of the CHELPG grid.

TYPE:

INTEGER

DEFAULT:

30

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 [7].

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 [7]), 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_H

Sets the Lebedev grid to use for hydrogen atoms.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Corresponding to a number of points in a Lebedev grid.

RECOMMENDATION:

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_HA

Sets the Lebedev grid to use for non-hydrogen atoms.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Corresponding to a number of points in a Lebedev grid (see Section 4.3.12.

RECOMMENDATION:

None.

Finally, Hirschfeld population analysis [10] provides yet another definition of atomic charges in molecules via a Stockholder prescription. The charge on atom A, q_A, is defined by

$$q_A = Z_A - \int d\mathbf{r} \frac{\rho_A^0(\mathbf{r})}{\sum_B \rho_B^0(\mathbf{r})} \rho(\mathbf{r}), \tag{10.5}$$

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. Following SCF completion, the Hirshfeld analysis proceeds by running another SCF calculation to obtain the atomic densities. Next numerical quadrature is used to evaluate the integral in Eq. (10.5). Neutral ground-state atoms are used, and 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.

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 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 default.

The next 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 [11], 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 MO and for the overall electron density. 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.

PRINT_RADII_GYRE

Controls printing of MO centroids and radii of gyration.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

TRUE	(or 1) Calculate the centroid and radius of gyration for each MO and density.
FALSE	(or 0) Do not calculate these quantities.

RECOMMENDATION:

None

10.2.2 Multipole Moments

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 wavefunction analysis requested.
TYPE:
INTEGER
DEFAULT:
4
OPTIONS:
n Calculate moments to nth order.
RECOMMENDATION:
Use default unless higher multipoles are required.

CIS_MOMENTS
Controls calculation of excited-state (CIS or TDDFT) multipole moments
TYPE:
LOGICAL/INTEGER
DEFAULT:
FALSE (or 0)
OPTIONS:
FALSE (or 0) Do not calculate excited-state moments.
TRUE (or 1) 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.

10.2.3 Symmetry Decomposition

Q-CHEM's default is to write the SCF wavefunction 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

10.2.4 Localized Orbital Bonding Analysis

Localized Orbital Bonding Analysis (LOBA) [12] is a technique developed by Dr. Alex Thom and Eric Sundstrom at Berkeley with Prof. Martin Head-Gordon. Inspired by the work of Rhee and Head-Gordon [13], 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 measured in %

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.)

10.2.5 Excited-State Analysis

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 [14]. To do this, we define a density matrix for the excited electron,

$$\mathbf{D}_{ab}^{elec} = \sum_i (\mathbf{X}+\mathbf{Y})_{ai}^{\dagger} (\mathbf{X}+\mathbf{Y})_{ib} \tag{10.6}$$

and a density matrix for the hole that is left behind in the occupied space:

$$\mathbf{D}_{ij}^{hole} = \sum_a (\mathbf{X}+\mathbf{Y})_{ia}(\mathbf{X}+\mathbf{Y})_{aj}^{\dagger} \tag{10.7}$$

The quantities $\mathbf{X}$ and $\mathbf{Y}$ are the transition density matrices, *i.e.*, the components of the TDDFT eigenvector [15]. 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 $\mathbf{D}^{elec}$ and $\mathbf{D}^{hole}$ into the AO basis, one can write

$$\Delta q = \sum_\mu (\mathbf{D}^{elec}\,\mathbf{S})_{\mu\mu} = -\sum_\mu (\mathbf{D}^{hole}\,\mathbf{S})_{\mu\mu} \tag{10.8}$$

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 [16], $\Delta q = -1$. For full TDDFT calculations, Δq may be slightly different than -1.

Comparison of Eq. (10.8) to Eq. (10.3) suggests that the quantities $(\mathbf{D}^{elec}\,\mathbf{S})$ and $(\mathbf{D}^{hole}\,\mathbf{S})$ are amenable to to population analyses of precisely the same sort used to analyze the ground-state density matrix. In particular, $(\mathbf{D}^{elec}\,\mathbf{S})_{\mu\mu}$ represents the μth AO's contribution to the excited electron, while $(\mathbf{D}^{hole}\,\mathbf{S})_{\mu\mu}$ is a contribution to the hole. The sum of these quantities,

$$\Delta q_\mu = (\mathbf{D}^{elec}\,\mathbf{S})_{\mu\mu} + (\mathbf{D}^{hole}\,\mathbf{S})_{\mu\mu} \tag{10.9}$$

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/INTEGER

DEFAULT:

FALSE

OPTIONS:

FALSE	(or 0) Do not perform particle/hole population analysis.
TRUE	(or 1) 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), which were introduced in Section 6.11.2. (NTOs can also be requested without excited-state population analysis. Some practical aspects of calculating and visualizing NTOs are discussed below, in Section 10.5.2.)

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

10.3 Interface to the NBO Package

Q-CHEM incorporates the Natural Bond Orbital package (v. 5.0 and 6.0) for molecular properties and wavefunction analysis. The NBO 5.0 package is invoked by setting the *$rem* variable NBO to TRUE and is initiated after the SCF wavefunction is obtained.

Note: If switched on for a geometry optimization, the NBO package will only be invoked at the end of the last optimization step.

Users should consult the NBO User's Manual for options and details relating to exploitation of the features offered in this package. The NBO 6.0 package [17, 18] can be downloaded by the user from http://nbo6.chem.wisc.edu, and can be invoked by: (a) setting the NBOEXE environment variable, and (b) include both NBO=TRUE and RUN_NBO6=TRUE in the qchem input file.

NBO analysis is also available for excited states calculated using CIS or TDDFT. Excited-state NBO analysis is still in its infancy, 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, and may require specification of additional NBO parameters in the *$nbo* section that is described below. Consult Ref. 19 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

The general format for passing options from Q-CHEM to the NBO program is shown below:

```
$nbo
   {NBO program keywords, parameters and options}
$end
```

Note: (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

10.4 Orbital Localization

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 [20, 21] minimizes the radial extent of the localized orbitals, *i.e.*, $\sum_i \langle ii||\mathbf{r}_1 - \mathbf{r}_2|^2|ii\rangle$, and although is relatively fast, does not separate σ and π orbitals, leading to two 'banana-orbitals' in the case of a double bond [22]. Pipek-Mezey localized orbitals [22] 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 [23] 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 the work of Dr. Joe Subotnik [24] and later Prof. Young-Min Rhee and Westin Kurlancheek with Prof. Martin Head-Gordon at Berkeley has, through use of the Resolution of the Identity approximation, reduced the formal cost may be asymptotically reduced to cubic scaling with the number of occupied orbitals.

BOYSCALC

Specifies the Boys localized orbitals are to be calculated

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not perform localize the occupied space.

1 Allow core-valence mixing in Boys localization.

2 Localize core and valence separately.

RECOMMENDATION:

None

ERCALC

Specifies the 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
```

10.5 Visualizing and Plotting Orbitals and Densities

Q-CHEM can generate orbital and density data in several formats suitable for plotting with a third-party visualization program.

10.5.1 Visualizing Orbitals Using MOLDEN and MACMOLPLT

Upon request, Q-CHEM will generate an input file for MOLDEN, a freely-available molecular visualization program [25, 26]. 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 [27, 28]. 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 10.1 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 10.2 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
```

10.5.2 Visualization of Natural Transition Orbitals

For excited states calculated using the CIS, RPA, or TDDFT methods, construction of Natural Transition Orbitals (NTOs), as described in Section 6.11.2, 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. 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 [29].)

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 2.7. 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 6.11.2), 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 10.3 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
```

10.5.3 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.

By default, the output from the *$plots* command is one (or several) ASCII 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 10.5.4.)

An example of the use of the *$plots* option is the following input deck:

Example 10.4 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 $x_{\rm min}$ $x_{\rm max}$

N_y $y_{\rm min}$ $y_{\rm max}$

N_z $z_{\rm min}$ $z_{\rm max}$

A line with 4 integers indicating how many things to plot:

$N_{\rm MO}$ $N_{\rm Rho}$ $N_{\rm Trans}$ $N_{\rm DA}$

An optional line with the integer list of MO's to evaluate (only if $N_{\rm MO} > 0$)

MO(1) MO(2) . . . MO($N_{\rm MO}$)

An optional line with the integer list of densities to evaluate (only if $N_{\rm Rho} > 0$)

Rho(1) Rho(2) . . . Rho($N_{\rm Rho}$)

An optional line with the integer list of transition densities (only if $N_{\rm Trans} > 0$)

Trans(1) Trans(2) . . . Trans($N_{\rm Trans}$)

An optional line with states for detachment/attachment densities (if $N_{\rm DA} > 0$)

DA(1) DA(2) . . . DA($N_{\rm DA}$)

$end

Line 1 of the *$plots* keyword section is reserved for comments. Lines 2–4 list the number of one dimension points and the range of the grid (note that coordinate ranges are in Angstroms, while all output is in atomic units). Line 5 must contain 4 non-negative integers indicating the number of: molecular orbitals ($N_{\rm 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 $1 \rightarrow N_{\text{DA}}$.

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* and *plot.detach.alpha*;
- unrestricted unrelaxed attachment/detachment analysis is sent to *plot.attach.alpha*, *plot.detach.alpha*, *plot.attach.beta* and *plot.detach.beta*;
- restricted relaxed attachments/detachment analysis is plotted in *plot.attach.rlx.alpha* and *plot.detach.rlx.alpha*; and finally
- unrestricted relaxed attachment/detachment analysis is sent to *plot.attach.rlx.alpha*, *plot.detach.rlx.alpha*, *plot.attach.rlx.beta* and *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 wavefunction 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 Direct Generation of "Cube" Files

As an alternative to the output format discussed above, all of the *$plots* data may be output directly to a sub-directory named *plots* in the job's scratch directory, which must therefore be saved using the *–save* option described in Section 2.7. The plotting data in this sub-directory are not written in the *plot.** format described above, but rather in the form of so-called "cube" file, one for each orbital or density that is requested. The "cube" format is a standard one for volumetric data, and consists of a small header followed by the orbital or density values at each grid point, in ASCII format. Because the grid coordinates themselves are not printed (their locations are implicit from information contained in the header), each individual cube file is much smaller than the corresponding *plot.** file would be. Cube files can be read by many standard (and freely-available) graphics programs, including MacMolPlt [27, 28] and VMD [30, 31]. VMD, in particular, is recommended for generation of high-quality images for publication. Cube files for the MOs and densities requested in the *$plots* section are requested by setting MAKE_CUBE_FILES to TRUE, with the *$plots* section specified as described in Section 10.5.3.

MAKE_CUBE_FILES

Requests generation of cube files for MOs, NTOs, or NBOs.

TYPE:

LOGICAL

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 10.5 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 6.11.2 and 10.5.2) 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 highest occupied NTO (HONTO) $\rightarrow$ lowest unoccupied NTO (LUNTO) excitation amplitude is always the most significant, 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 10.6 Generating cube files for the HONTO-to-LUNTO excitation 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 --> 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 10.7 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
```

10.5.5 NCI Plots

We have implemented the non-covalent interaction (NCI) plots from Weitao Yang's group [32, 33]. To generate these plots, one can set the PLOT_REDUCED_DENSITY_GRAD *$rem* variable to TRUE (see the nci-c8h14.in input example in *$QC/samples* directory).

10.6 Electrostatic Potentials

Q-CHEM can evaluate electrostatic potentials on a grid of points. Electrostatic potential evaluation is controlled by the *$rem* variable IGDESP, as documented below.

IGDESP

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, esp for each point.

TYPE:

INTEGER

DEFAULT:

none no electrostatic potential evaluation

OPTIONS:

-1	read grid input via the *$plots* section of the input deck
0	Generate the ESP values at all nuclear positions.
+n	read n grid points in bohrs (!) from the ASCII file ESPGrid.

RECOMMENDATION:

None

The following example illustrates the evaluation of electrostatic potentials on a grid, note that IANLTY must also be set to 200.

Example 10.8 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
   IGDESP     -1
$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
```

```
      0  0  0  0
      0
   $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 [34] that sought to model the ESP not by direct fitting, but by fitting to the multipole moments. 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 the following *$rem* option should be set:

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.

10.7 Spin and Charge Densities at the Nuclei

Gaussian basis sets violate nuclear cusp conditions [35–37]. This may lead to large errors in wavefunction at nuclei, particularly for spin density calculations [38]. This problem can be alleviated by using an averaging operator that compute wavefunction density based on constraints that wavefunction must satisfy near Coulomb singularity [39, 40]. The derivation of operators is based on hyper virial theorem [41] and presented in Ref. 39. Application to molecular spin densities for spin-polarized [40] and DFT [42] wavefunctions 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.* is shown to be adequate, with final answer being relatively insensitive to the exact choice of r_0 [39, 40]. 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 wavefunctions. 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.

10.8 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

10.9 Distributed Multipole Analysis

Distributed Multipole Analysis (DMA) [43] is a method to represent the electrostatic potential of a molecule in terms of a multipole expansion around a set of points. The points of expansion are the atom centers and (optionally) bond midpoints. Current implementation performs expansion into charges, dipoles, quadrupoles and octupoles. See also Sections 11.5 and 12.7.

DO_DMA

Specifies whether to perform Distributed Multipole Analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Turn off DMA.
TRUE	Turn on DMA.

RECOMMENDATION:

None

DMA_MIDPOINTS

Specifies whether to include bond midpoints into DMA expansion.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE	Do not include bond midpoints.
TRUE	Include bond midpoint.

RECOMMENDATION:

None

10.10 Intracules

The many dimensions of electronic wavefunctions 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 wavefunction 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. 44. Q-CHEM's intracule package was developed by Aaron Lee and Nick Besley, and can compute the following intracules for or HF wavefunctions:

- 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.

10.10.1 Position Intracules

The intracule density, $I(\mathbf{u})$, represents the probability for the inter-electronic vector $\mathbf{u} = \mathbf{u}_1 - \mathbf{u}_2$:

$$I(\mathbf{u}) = \int \rho(\mathbf{r}_1\mathbf{r}_2)\ \delta(\mathbf{r}_{12} - \mathbf{u})\ d\mathbf{r}_1\,\mathbf{dr}_2 \tag{10.10}$$

where $\rho(\mathbf{r}_1, \mathbf{r}_2)$ is the two-electron density. A simpler quantity is the spherically averaged intracule density,

$$P(u) = \int I(\mathbf{u})\mathrm{d}\Omega_{\mathbf{u}}\ , \tag{10.11}$$

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 [44]. If the molecular orbitals are expanded within a basis set

$$\psi_a(\mathbf{r}) = \sum_{\mu} c_{\mu a}\phi_\mu(\mathbf{r}) \tag{10.12}$$

The quantity $P(u)$ can be expressed as

$$P(u) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma}(\mu\nu\lambda\sigma)_{\mathrm{P}} \tag{10.13}$$

where $\Gamma_{\mu\nu\lambda\sigma}$ is the two-particle density matrix and $(\mu\nu\lambda\sigma)_{\mathrm{P}}$ is the position integral

$$(\mu\nu\lambda\sigma)_{\mathrm{P}} = \int \phi_\mu^*(\mathbf{r})\ \phi_\nu(\mathbf{r})\ \phi_\lambda^*(\mathbf{r}+\mathbf{u})\phi_\sigma(\mathbf{r}+\mathbf{u})\ d\mathbf{r}\ d\Omega \tag{10.14}$$

and $\phi_\mu(\mathbf{r})$, $\phi_\nu(\mathbf{r})$, $\phi_\lambda(\mathbf{r})$ and $\phi_\sigma(\mathbf{r})$ are basis functions. For HF wavefunctions, the position intracule can be decomposed into a Coulomb component,

$$P_{\mathrm{J}}(u) = \frac{1}{2}\sum_{\mu\nu\lambda\sigma} D_{\mu\nu}D_{\lambda\sigma}(\mu\nu\lambda\sigma)_{\mathrm{P}} \tag{10.15}$$

and an exchange component,

$$P_{\rm K}(u) = -\frac{1}{2}\sum_{\mu\nu\lambda\sigma}\left[D^{\alpha}_{\mu\lambda}D^{\alpha}_{\nu\sigma} + D^{\beta}_{\mu\lambda}D^{\beta}_{\nu\sigma}\right](\mu\nu\lambda\sigma)_{\rm P} \tag{10.16}$$

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. 45.

Some of the moments of $P(u)$ are physically significant [46], for example

$$\int_0^\infty u^0 P(u)du = \frac{n(n-1)}{2} \tag{10.17}$$

$$\int_0^\infty u^0 P_{\rm J}(u)du = \frac{n^2}{2} \tag{10.18}$$

$$\int_0^\infty u^2 P_{\rm J}(u)du = nQ - \mu^2 \tag{10.19}$$

$$\int_0^\infty u^0 P_{\rm K}(u)du = -\frac{n}{2} \tag{10.20}$$

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.

10.10.2 Momentum Intracules

Analogous quantities can be defined in momentum space; $\bar{I}(\mathbf{v})$, for example, represents the probability density for the relative momentum $\mathbf{v} = \mathbf{p}_1 - \mathbf{p}_2$:

$$\bar{I}(\mathbf{v}) = \int \pi(\mathbf{p}_1, \mathbf{p}_2)\ \delta(\mathbf{p}_{12} - \mathbf{v})d\mathbf{p}_1 d\mathbf{p}_2 \tag{10.21}$$

where $\pi(\mathbf{p}_1, \mathbf{p}_2)$ momentum two-electron density. Similarly, the spherically averaged intracule

$$M(v) = \int \bar{I}(\mathbf{v})\mathrm{d}\Omega_{\mathbf{v}} \tag{10.22}$$

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

$$M(v) = \sum_{\mu\nu\lambda\sigma}\Gamma_{\mu\nu\lambda\sigma}\,(\mu\nu\lambda\sigma)_{\rm M} \tag{10.23}$$

where $\Gamma_{\mu\nu\lambda\sigma}$ is the two-particle density matrix and $(\mu\nu\lambda\sigma)_{\rm M}$ is the momentum integral [47]

$$(\mu\nu\lambda\sigma)_{\rm M} = \frac{v^2}{2\pi^2}\int \phi^*_\mu(\mathbf{r})\phi_\nu(\mathbf{r}+\mathbf{q})\phi^*_\lambda(\mathbf{u}+\mathbf{q})\phi_\sigma(\mathbf{u})j_0(qv)\ d\mathbf{r}\ d\mathbf{q}\ d\mathbf{u} \tag{10.24}$$

The momentum integrals only possess four-fold permutational symmetry, *i.e.*,

$$(\mu\nu\lambda\sigma)_{\rm M} = (\nu\mu\lambda\sigma)_{\rm M} = (\sigma\lambda\nu\mu)_{\rm M} = (\lambda\sigma\mu\nu)_{\rm M} \tag{10.25}$$

$$(\nu\mu\lambda\sigma)_{\rm M} = (\mu\nu\sigma\lambda)_{\rm M} = (\lambda\sigma\nu\mu)_{\rm M} = (\sigma\lambda\mu\nu)_{\rm M} \tag{10.26}$$

and therefore generation of $M(v)$ is roughly twice as expensive as $P(u)$. Momentum intracules can also be decomposed into Coulomb $M_{\mathrm{J}}(v)$ and exchange $M_{\mathrm{K}}(v)$ components:

$$M_J(v) = \frac{1}{2}\sum_{\mu\nu\lambda\sigma} D_{\mu\nu}D_{\lambda\sigma}(\mu\nu\lambda\sigma)_{\mathrm{M}} \tag{10.27}$$

$$M_{\mathrm{K}}(v) = -\frac{1}{2}\sum_{\mu\nu\lambda\sigma}\left[D^{\alpha}_{\mu\lambda}D^{\alpha}_{\nu\sigma} + D^{\beta}_{\mu\lambda}D^{\beta}_{\nu\sigma}\right](\mu\nu\lambda\sigma)_{\mathrm{M}} \tag{10.28}$$

Again, the even-order moments are physically significant [47]:

$$\int_0^\infty v^0 M(v)dv = \frac{n(n-1)}{2} \tag{10.29}$$

$$\int_0^\infty u^0 M_{\mathrm{J}}(v)dv = \frac{n^2}{2} \tag{10.30}$$

$$\int_0^\infty v^2 P_{\mathrm{J}}(v)dv = 2nE_{\mathrm{T}} \tag{10.31}$$

$$\int_0^\infty v^0 M_{\mathrm{K}}(v)dv = -\frac{n}{2} \tag{10.32}$$

where n is the number of electrons and E_{T} is the total electronic kinetic energy. Currently, Q-CHEM can compute $M(v)$, $M_{\mathrm{J}}(v)$ and $M_{\mathrm{K}}(v)$ 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.

10.10.3 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 [48],

$$W_2(\mathbf{r}_1,\mathbf{p}_1,\mathbf{r}_2,\mathbf{p}_2) = \frac{1}{\pi^6}\int \rho_2(\mathbf{r}_1+\mathbf{q}_1,\mathbf{r}_1-\mathbf{q}_1,\mathbf{r}_2+\mathbf{q}_2,\mathbf{r}_2-\mathbf{q}_2)e^{-2i(\mathbf{p}_1\cdot\mathbf{q}_1+\mathbf{p}_2\cdot\mathbf{q}_2)}d\mathbf{q}_1 d\mathbf{q}_2 \tag{10.33}$$

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

$$W(u,v) = \int W_2(\mathbf{r}_1,\mathbf{p}_1,\mathbf{r}_2,\mathbf{p}_2)\delta(\mathbf{r}_{12}-\mathbf{u})\delta(\mathbf{p}_{12}-\mathbf{v})d\mathbf{r}_1\, d\mathbf{r}_2\, d\mathbf{p}_1\, d\mathbf{p}_2\, d\Omega_{\mathbf{u}}\, d\Omega_{\mathbf{v}} \tag{10.34}$$

If the orbitals are expanded in a basis set, then $W(u,v)$ can be written as

$$W(u,v) = \sum_{\mu\nu\lambda\sigma} \Gamma_{\mu\nu\lambda\sigma}\,(\mu\nu\lambda\sigma)_{\mathrm{W}} \tag{10.35}$$

where $(\mu\nu\lambda\sigma)_{\mathrm{W}}$ is the Wigner integral

$$(\mu\nu\lambda\sigma)_{\mathrm{W}} = \frac{v^2}{2\pi^2}\int\!\!\int \phi_\mu^*(\mathbf{r})\phi_\nu(\mathbf{r}+\mathbf{q})\phi_\lambda^*(\mathbf{r}+\mathbf{q}+\mathbf{u})\phi_\sigma(\mathbf{r}+\mathbf{u})j_0(q\,v)\;d\mathbf{r}\;dq\;d\Omega_{\mathbf{u}} \tag{10.36}$$

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,

$$\begin{aligned}[ssss]_{\mathrm{W}} \;=\;& \frac{u^2v^2}{2\pi^2}\int\!\!\int \exp\left[-\alpha|\mathbf{r}-\mathbf{A}|^2-\beta|\mathbf{r}+\mathbf{q}-\mathbf{B}|^2-\gamma|\mathbf{r}+\mathbf{q}+\mathbf{u}-\mathbf{C}|^2-\delta|\mathbf{r}+\mathbf{u}-\mathbf{D}|^2\right]\times\\ & j_0(qv)\;d\mathbf{r}\;d\mathbf{q}\;d\Omega_{\mathbf{u}}\end{aligned} \tag{10.37}$$

can be expressed as

$$[ssss]_{\mathrm{W}} = \frac{\pi u^2v^2\;e^{-(R+\lambda^2u^2+\mu^2v^2)}}{2(\alpha+\delta)^{3/2}(\beta+\gamma)^{3/2}}\int e^{-\mathbf{P}\cdot\mathbf{u}}j_0\left(|\mathbf{Q}+\eta\mathbf{u}|v\right)\;d\Omega_u \tag{10.38}$$

or alternatively

$$[ssss]_{\mathrm{W}} = \frac{2\pi^2u^2v^2e^{-(R+\lambda^2u^2+\mu^2v^2)}}{(\alpha+\delta)^{3/2}(\beta+\gamma)^{3/2}}\sum_{n=0}^{\infty}(2n+1)i_n(P\,u)j_n(\eta uv)j_n(Qv)P_n\left(\frac{\mathbf{P}\cdot\mathbf{Q}}{P\,Q}\right) \tag{10.39}$$

Two approaches for evaluating $(\mu\nu\lambda\sigma)_{\mathrm{W}}$ have been implemented in Q-CHEM, full details can be found in Ref. 49. 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 [50–52] 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 [53]. 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.

10.10.4 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, use 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.

10.10.5 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 10.9 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 10.10 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 $i_n(x)$.

```
$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   30   0.0   15.0
   v_points   20   0.0   10.0
$end
```

10.11 Vibrational Analysis

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 Raman [54] and infrared activities. Vibrational frequencies are calculated by either using an analytic Hessian (if available; see Table 9.1) or, numerical finite difference of the gradient. The default setting in Q-CHEM is to use the highest analytical derivative order available for the requested theoretical method.

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 [55]. 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 $\mathcal{O}(N^2M/N_{\rm seg})$, where $N_{\rm seg}$ is the number of segments. This option is invoked automatically by the program.

Following a vibrational analysis, Q-CHEM computes useful statistical thermodynamic properties at standard temperature and pressure, including: zero-point vibration energy (ZPVE) and, translational, rotational and vibrational, entropies and enthalpies.

The performance of various *ab initio* theories in determining vibrational frequencies has been well documented; see Refs. 56–58.

10.11.1 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 *$rem* variables EXCHANGE, CORRELATION (if required) (Chapter 4) and BASIS (Chapter 7). Since the default type of job (JOBTYPE) is a single point energy (SP) calculation, the JOBTYPE *$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 4.3.11) 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 default.

CPSCF_NSEG

Controls the number of segments used to calculate the CPSCF equations.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not solve the CPSCF equations in segments.

n User-defined. Use n segments when solving the CPSCF equations.

RECOMMENDATION:

Use default.

Example 10.11 An EDF1/6-31+G* optimization, followed by a vibrational analysis. Doing the vibrational analysis at a stationary point is necessary for the results to be valid.

```
$molecule
   O  1
   C  1  co
   F  2  fc  1  fco
   H  2  hc  1  hcp  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
```

10.12 Anharmonic Vibrational Frequencies

Computing vibrational spectra beyond the harmonic approximation has become an active area of research owing to the improved efficiency of computer techniques [59–62]. 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 [63, 64]. 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* wavefunction 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 [62, 65–68]. 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 [69]. 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 wavefunction, transition-optimized shifted Hermite (TOSH) theory [70], which uses first-order perturbation theory, which avoids the degeneracy problems of VPT2, but which incorporates anharmonic effects into the wavefunction, thus increasing the accuracy of the predicted anharmonic energies.

10.12.1 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 [71, 72], 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 [73]. It is also possible to include a subset of the harmonic vibrational modes with an anharmonic frequency calculation by invoking the ANHAR_SEL rem. 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 [74, 75]. 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 [76]. 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:
0 Full Hessian calculation
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

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 10.12 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
```

Example 10.13 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
```

10.12.2 Vibration Configuration Interaction Theory

To solve the nuclear vibrational Schrödinger equation, one can only use direct integration procedures for diatomic molecules [63, 64]. 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 wavefunction of the nth excited state of mode i is

$$\psi_i^{(n)} = \phi_i^{(n)} \prod_{j \neq i}^{m} \phi_j^{(0)} \tag{10.40}$$

where the $\phi_i^{(n)}$ represents the harmonic oscillator eigenfunctions for normal mode q_i. This can be expressed in terms of Hermite polynomials:

$$\phi_i^{(n)} = \left(\frac{\omega_i^{\frac{1}{2}}}{\pi^{\frac{1}{2}} 2^n n!} \right)^{\frac{1}{2}} e^{-\frac{\omega_i q_i^2}{2}} H_n(q_i \sqrt{\omega_i}) \tag{10.41}$$

With the basis function defined in Eq. (10.40), the nth wavefunction can be described as a linear combination of the Hermite polynomials:

$$\Psi^{(n)} = \sum_{i=0}^{n_1} \sum_{j=0}^{n_2} \sum_{k=0}^{n_3} \cdots \sum_{m=0}^{n_m} c_{ijk\cdots m}^{(n)} \psi_{ijk\cdots m}^{(n)} \tag{10.42}$$

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.*:

$$n = n_1 + n_2 + n_3 + \cdots + n_m \tag{10.43}$$

To determine this expansion coefficient $c^{(n)}$, we integrate the $\hat{H}$, as in Eq. (4.1), with $\Psi^{(n)}$ to get the eigenvalues

$$c^{(n)} = E_{\mathrm{VCI}(n)}^{(n)} = \langle \Psi^{(n)} | \hat{H} | \Psi^{(n)} \rangle \tag{10.44}$$

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. (10.44) 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.

10.12.3 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 [77–79], 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, $\hat{U}$,

$$\hat{U} = \sum_i^m \left(-\frac{1}{2} \frac{\partial^2}{\partial q_i^2} + \frac{{\omega_i}^2}{2} {q_i}^2 \right) \tag{10.45}$$

and a perturbed part, $\hat{V}$:

$$\hat{V} = \frac{1}{6} \sum_{ijk=1}^{m} \eta_{ijk} q_i q_j q_k + \frac{1}{24} \sum_{ijkl=1}^{m} \eta_{ijkl} q_i q_j q_k q_l \tag{10.46}$$

One can then apply second-order perturbation theory to get the ith excited state energy:

$$E^{(i)} = \hat{U}^{(i)} + \langle \Psi^{(i)} | \hat{V} | \Psi^{(i)} \rangle + \sum_{j \neq i} \frac{|\langle \Psi^{(i)} | \hat{V} | \Psi^{(j)} \rangle|^2}{\hat{U}^{(i)} - \hat{U}^{(j)}} \tag{10.47}$$

The denominator in Eq. (10.47) 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 [77–80]. An alternative solution has been proposed by Barone [62] 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

$$E_i^{\mathrm{VPT2}} = \sum_j \omega_j(n_j + 1/2) + \sum_{i\leq j} x_{ij}(n_i + 1/2)(n_j + 1/2) \tag{10.48}$$

where, if rotational coupling is ignored, the anharmonic constants x_{ij} are given by

$$x_{ij} = \frac{1}{4\omega_i\omega_j}\left(\eta_{iijj} - \sum_k^m \frac{\eta_{iik}\eta_{jjk}}{\omega_k^2} + \sum_k^m \frac{2(\omega_i^2 + \omega_j^2 - \omega_k^2)\eta_{ijk}^2}{[(\omega_i+\omega_j)^2 - \omega_k^2]\,[(\omega_i-\omega_j)^2 - \omega_k^2]}\right) \tag{10.49}$$

10.12.4 Transition-Optimized Shifted Hermite Theory

So far, every aspect of solving the nuclear wave equation has been considered, except the wavefunction. Since Schrödinger proposed his equation, the nuclear wavefunction has traditionally be expressed in terms of Hermite functions, which are designed for the harmonic oscillator case. Recently [70], a modified representation has been presented. 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

$$\hat{H} = -\frac{1}{2}\frac{\partial^2}{\partial q^2} + \frac{1}{2}\omega^2 q^2 + \eta_{iii}q^3 + \eta_{iiii}q^4 \tag{10.50}$$

and the wavefunction is expressed as in Eq. (10.41). Now, if we shift the center of the wavefunction 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 wavefunction. For a ground vibrational state, the wavefunction is written as

$$\phi^{(0)} = \left(\frac{\omega}{\pi}\right)^{\frac{1}{4}} e^{-\frac{\omega}{2}(q-\sigma)^2} \tag{10.51}$$

Similarly, for the first excited vibrational state, we have

$$\phi^{(1)} = \left(\frac{4\omega^3}{\pi}\right)^{\frac{1}{4}} (q-\sigma)\, e^{\frac{\omega}{2}(q-\sigma)^2} \tag{10.52}$$

Therefore, the energy difference between the first vibrational excited state and the ground state is

$$\Delta E_{\mathrm{TOSH}} = \omega + \frac{\eta_{iiii}}{8\omega^2} + \frac{\eta_{iii}\sigma}{2\omega} + \frac{\eta_{iiii}\sigma^2}{4\omega} \tag{10.53}$$

This is the fundamental vibrational frequency from first-order perturbation theory.

Meanwhile, We know from the first-order perturbation theory with an ordinary wavefunction within a QFF PES, the energy is

$$\Delta E_{\mathrm{VPT1}} = \omega + \frac{\eta_{iiii}}{8\omega^2} \tag{10.54}$$

The differences between these two wavefunctions are the two extra terms arising from the shift in Eq. (10.53). To determine the shift, we compare the energy with that from second-order perturbation theory:

$$\Delta E_{\mathrm{VPT2}} = \omega + \frac{\eta_{iiii}}{8\omega^2} - \frac{5{\eta_{iii}}^2}{24\omega^4} \tag{10.55}$$

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:

$$\sigma = -\frac{5}{12}\frac{\eta_{iii}}{\omega^3} \tag{10.56}$$

Because the only difference between this approach and the ordinary wavefunction is the shift in the normal coordinate, we call it "transition-optimized shifted Hermite" (TOSH) functions [70]. This approximation gives second-order accuracy at only first-order cost.

For polyatomic molecules, we consider Eq. (10.53), and propose that the energy of the ith mode be expressed as:

$$\Delta E_i^{\mathrm{TOSH}} = \omega_i + \frac{1}{8\omega_i}\sum_j \frac{\eta_{iijj}}{\omega_j} + \frac{1}{2\omega_i}\sum_j \eta_{iij}\sigma_{ij} + \frac{1}{4\omega_i}\sum_{j,k}\eta_{iijk}\sigma_{ij}\sigma_{ik} \tag{10.57}$$

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

$$\sigma_{ij} = \frac{(\delta_{ij}-2)(\omega_i+\omega_j)\eta_{iij}}{4\omega_i\omega_j^2(2\omega_i+\omega_j)} - \sum_k \frac{\eta_{kkj}}{4\omega_k\omega_j^2} \tag{10.58}$$

10.12.5 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 $2 * (N_{\rm Vib} + N_{\rm VCI})/N_{\rm Vib}N_{\rm VCI}$ 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 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 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 default, unless on a very flat potential, in which case a larger value should be used.

Example 10.14 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
```

10.12.6 Isotopic Substitutions

By default Q-CHEM calculates vibrational frequencies using the atomic masses of the most abundant isotopes (taken from the Handbook of Chemistry and Physics, 63$^{\mathrm{rd}}$ Edition). 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 (298.18 K and 1 atm respectively), or 1 if they are to be altered. Note that the temperatures should be specified in Kelvin (K) and pressures in atmospheres (atm).

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 10.15 An EDF1/6-31+G* optimization, followed by a vibrational analysis. 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         3-21G
   JOBTYPE       opt
   METHOD        hf
$end

@@@

$molecule
   read
$end

$rem
   BASIS         3-21G
   JOBTYPE       freq
   METHOD        hf
   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   ! H’s replaced with D’s
     4   2.01410
$end
```

10.13 NMR Shielding Tensors

NMR spectroscopy is a powerful technique to yield important information on molecular systems in chemistry and biochemistry. 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. Examples for such assignments are numerous, for example, assignments of solid-state spectra [81, 82]. The implementation within Q-CHEM uses gauge-including atomic orbitals (GIAOs) [83–85] to calculate the NMR chemical shielding tensors. This scheme has been proven to be reliable an accurate for many applications [86].

The shielding tensor, σ, is a second-order property depending on the external magnetic field, B, and the nuclear magnetic spin momentum, m_k, of nucleus k:

$$\Delta E = -m_j(1 - \sigma)B \tag{10.59}$$

Using analytical derivative techniques to evaluate σ, the components of this 3×3 tensor are computed as

$$\sigma_{ij} = \sum_{\mu\nu} P_{\mu\nu} \frac{\partial^2 h_{\mu\nu}}{\partial B_i \partial m_{j,k}} + \sum_{\mu\nu} \frac{\partial P_{\mu\nu}}{\partial B_i} \frac{\partial h_{\mu\nu}}{\partial m_{j,k}} \tag{10.60}$$

where i and j represent are Cartesian components.

To solve for the necessary perturbed densities, $\partial P/\partial B_{x,y,z}$, a new CPSCF method based on a density matrix based formulation [87, 88] is used. This formulation is related to a density matrix based CPSCF (D-CPSCF) formulation employed for the computation of vibrational frequencies [89]. Alternatively, an MO-based CPSCF calculation of shielding tensors can be chosen by the variable MOPROP. Features of the NMR package include:

- Restricted HF-GIAO and KS-DFT-GIAO NMR chemical shifts calculations
- LinK/CFMM support to evaluate Coulomb- and exchange-like matrices
- Density matrix-based coupled-perturbed SCF (D-CPSCF)
- DIIS acceleration
- Support of basis sets up to d functions
- Support of LSDA/GGA/Hybrid XC functionals

10.13.1 Job Control

The JOBTYPE must be set to NMR to request the NMR chemical shifts.

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 default.

D_SCF_MAX_2

Sets the maximum number of level-2 iterations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User defined.

RECOMMENDATION:

Use 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.

10.13.2 Using NMR Shielding Constants as an Efficient Probe 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. Schleyer and co. have employed this phenomenon to justify a new unique probe of aromaticity [90]. 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 Hartree-Fock/6-31G* level. The same NICS value for benzene was also reported in Ref. 90. 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. 90. 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 10.16 Calculation of the NMR NICS probe of furan with Hartree-Fock/6-31G* with Q-CHEM.

```
$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
  GH      0.02132       0.32584       0.00034 ! ghost at the ring center
$end

$rem
JobType           NMR
METHOD            HF
BASIS             mixed
SCF_Algorithm     DIIS
PURCAR 111
```

```
SEPARATE_JK           0
LIN_K                 0
CFMM_ORDER           15
GRAIN                 1
CFMM_PRINT            2
CFMMSTAT              1
PRINT_PATH_TIME       1
LINK_MAXSHELL_NUMBER  1
SKIP_SCFMAN           0
IGUESS               core       !  Core Hamiltonian Guess
SCF_Convergence       7
ITHRSH               10        ! Threshold
IPRINT               23

D_SCF_CONVGUIDE       0        !REM_D_SCF_CONVGUIDE
D_SCF_METRIC          2        !Metric...
D_SCF_STORAGE        50        !REM_D_SCF_STORAGE
D_SCF_RESTART         0        !REM_D_SCF_RESTART

PRINT_PATH_TIME       1
SYM_IGNORE            1
NO_REORIENT           1
$end

$basis
C 1
6-31G*
****
C 2
6-31G*
****
C 3
6-31G*
****
O 4
6-31G*
****
C 5
6-31G*
****
H 6
6-31G*
****
H 7
6-31G*
****
H 8
6-31G*
****
H 9
6-31G*
****
H 10
6-31G*
****
```

10.14 Linear-Scaling NMR Chemical Shifts: GIAO-HF and GIAO-DFT

The importance of nuclear magnetic resonance (NMR) spectroscopy for modern chemistry and biochemistry cannot be overestimated. Despite tremendous progress in experimental techniques, the understanding and reliable assignment of observed experimental spectra remains often a highly difficult task, so that quantum chemical methods can be extremely useful both in the solution and the solid state (*e.g.*, Refs. 81, 82, 87, 91, 92, and references therein).

The cost for the computation of NMR chemical shifts within even the simplest quantum chemical methods such as Hartree-Fock (HF) or density functional (DFT) approximations increases conventionally with the third power of the molecular size M, $\mathcal{O}(M^3)$, where $\mathcal{O}(\cdot)$ stands for the scaling order. Therefore, the computation of NMR chemical shieldings has so far been limited to molecular systems in the order of 100 atoms without molecular symmetry.

For larger systems it is crucial to reduce the increase of the computational effort to linear, which has been recently achieved by Kussmann and Ochsenfeld [87, 93]. In this way, the computation of NMR chemical shifts becomes possible at both HF or DFT level for molecular systems with 1000 atoms and more, while the accuracy and reliability of traditional methods is fully preserved. In our formulation we use gauge-including atomic orbitals (GIAOs) [83, 84, 94], which have proven to be particularly successful [95]. For example, for many molecular systems the HF (GIAO-HF) approach provides typically an accuracy of 0.2–0.4 ppm for the computation of ^{1}H NMR chemical shifts (*e.g.* Refs. 81, 82, 87, 91, 92).

NMR chemical shifts are calculated as second derivatives of the energy with respect to the external magnetic field $\mathbf{B}$ and the nuclear magnetic spin m_{Nj} of a nucleus N:

$$\sigma_{ij}^N = \frac{\partial^2 E}{\partial B_i \partial M_{Nj}} \tag{10.61}$$

where i, j are x, y, z coordinates.

For the computation of the NMR shielding tensor it is necessary to solve for the response of the one-particle density matrix with respect to the magnetic field, so that the solution of the coupled perturbed SCF (CPSCF) equations either within the HF or the DFT approach is required.

These equations can be solved within a density matrix-based formalism for the first time with only linear-scaling effort for molecular systems with a non-vanishing HOMO-LUMO gap [87]. The solution is even simpler in DFT approaches without explicit exchange, since present density functionals are not dependent on the magnetic field.

The present implementation of NMR shieldings in Q-CHEM employs the LinK (linear exchange K) method [96, 97] for the formation of exchange contributions [87]. 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 [98] is adapted [87]. In addition, a multitude of different approaches for the solution of the CPSCF equations can be selected within Q-CHEM.

The so far largest molecular system for which NMR shieldings have been computed, contained 1003 atoms and 8593 basis functions (GIAO-HF/6-31G*) without molecular symmetry [87].

10.15 Indirect Nuclear Spin–Spin Coupling Constants

Indirect nuclear spin–spin coupling constants (also known as scalar coupling or J-coupling) can be evaluated in Q-CHEM at the Hartree–Fock and density functional levels of theory[99, 100] (restricted wavefunctions only). Pure and hybrid functionals are supported. To perform a J-coupling calculation, set JOBTYPE to ISSC. Refer to Section 10.16.3 for more options.

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 $\mathbf{m}$,

$$\mathbf{J}^{AB} = \frac{\partial^2 E}{\partial \mathbf{m}_A \partial \mathbf{m}_B}. \tag{10.62}$$

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 time-consuming.

Some authors have recommended [101] calculating only the Fermi contact contribution, and skipping the other contributions for ^{1}H–^{1}H coupling constants. For that purpose, Q-CHEM allows you to skip the calculation of any of the four (FC, SD, PSO, DSO) contributions (the mixed SD-FC contributions is automatically calculated at no additional cost whenever both the SD and FC contributions are calculated). See section 10.16.3 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 conservatively.

Specialized basis sets are highly recommended in any J-coupling calculation. The pcJ-n basis set family [102] has been added to the basis set library.

Note that the Hartree–Fock level of theory is *not* suitable to obtain J-coupling constants of *any* degree of reliability. Instead, use GGA or hybrid density functionals.

Example 10.17 An example of 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
SYMMETRY          TRUE
MOPROP_CONV_1ST   6
$end
```

10.16 Linear–Scaling Computation of Electric Properties

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 TDSCF equations, where the perturbation is represented as the interaction of the molecule with a single Fourier component within the dipole approximation:

$$\hat{H}^{(S)} = \frac{1}{2}\hat{\mu}\mathcal{E}\left(e^{-i\omega t} + e^{+i\omega t}\right) \tag{10.63}$$

$$\hat{\mu} = -e\sum_{i=1}^{N}\hat{r}_i \tag{10.64}$$

Here, $\mathcal{E}$ is the E-field vector, ω the corresponding frequency, e the electronic charge and μ the dipole moment operator. Starting from Frenkel's variational principle the TDSCF equations can be derived by standard techniques of perturbation theory [103]. As a solution we yield the first ($\mathrm{P}^{\mathrm{x}}(\pm\omega)$) and second order (*e.g.* $\mathrm{P}^{\mathrm{xy}}(\pm\omega,\pm\omega)$) perturbed density matrices with which the following properties are calculated:

- Static polarizability: $\alpha_{\mathrm{xy}}(0;0) = \mathrm{Tr}\left[\mathbf{H}^{\mu_{\mathrm{x}}}\mathbf{P}^{\mathrm{y}}\left(\omega = 0\right)\right]$
- Dynamic polarizability: $\alpha_{\mathrm{xy}}(\pm\omega;\mp\omega) = \mathrm{Tr}\left[\mathbf{H}^{\mu_{\mathrm{x}}}\mathbf{P}^{\mathrm{y}}(\pm\omega)\right]$
- Static hyperpolarizability: $\beta_{\mathrm{xyz}}(0;0,0) = \mathrm{Tr}\left[\mathbf{H}^{\mu_{\mathrm{x}}}\mathbf{P}^{\mathrm{yz}}(\omega = 0,\omega = 0)\right]$
- Second harmonic generation: $\beta_{\mathrm{xyz}}(\mp2\omega;\pm\omega,\pm\omega) = \mathrm{Tr}\left[\mathbf{H}^{\mu_{\mathrm{x}}}\mathbf{P}^{\mathrm{yz}}(\pm\omega,\pm\omega)\right]$
- Electro-optical Pockels effect: $\beta_{\mathrm{xyz}}(\mp\omega;0,\pm\omega) = \mathrm{Tr}\left[\mathbf{H}^{\mu_{\mathrm{x}}}\mathbf{P}^{\mathrm{yz}}(\omega = 0,\pm\omega)\right]$
- Optical rectification: $\beta_{\mathrm{xyz}}(0;\pm\omega,\mp\omega) = \mathrm{Tr}\left[\mathbf{H}^{\mu_{\mathrm{x}}}\mathbf{P}^{\mathrm{yz}}(\pm\omega,\mp\omega)\right]$

where $\mathbf{H}^{\mu_{\mathrm{x}}}$ is the matrix representation of the x component of the dipole moments.

The TDSCF calculation is the most time consuming step and scales asymptotically as $\mathcal{O}(N^3)$ because of the AO/MO transformations. The scaling behavior of the two-electron integral formations, which dominate over a wide range because of a larger pre-factor, can be reduced by LinK/CFMM from quadratic to linear ($\mathcal{O}(N^2)\rightarrow\mathcal{O}(N)$).

Third-order properties can be calculated with the equations above after a second-order TDSCF calculation (MOPROP: 101/102) or by use of Wigner's $(2n+1)$ rule [104] (MOPROP: 103/104). Since the second order TDSCF depends on the first-order results, the convergence of the algorithm may be problematically. So we recommend the use of 103/104 for the calculation of first hyperpolarizabilities.

These optical properties can be computed for the first time using linear-scaling methods (LinK/CFMM) for all integral contractions [88]. Although the present implementation available in Q-CHEM still uses MO-based time-dependent SCF (TDSCF) equations both at the HF and DFT level, the pre-factor of this $\mathcal{O}(M^3)$ scaling step is rather small, so that the reduction of the scaling achieved for the integral contractions is most important. Here, all derivatives are computed analytically.

Further specifications of the dynamic properties are done in the section *$fdpfreq* in the following format:

```
$fdpfreq
   property
```

```
      frequencies
      units
   $end
```

The first line is only required for third order properties to specify the kind of first hyperpolarizability:

- `StaticHyper` Static Hyperpolarizability
- `SHG` Second harmonic generation
- `EOPockels` Electro-optical Pockels effect
- `OptRect` Optical rectification

Line number 2 contains the values (FLOAT) of the frequencies of the perturbations. 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 given frequencies:

- `au` Frequency (atomic units)
- `eV` Frequency (eV)
- `nm` Wavelength (nm) → Note that 0 nm will be treated as 0.0 *a.u.*
- `Hz` Frequency (Hertz)
- `cmInv` Wavenumber (cm^{-1})

10.16.1 Examples for Section *$fdpfreq*

Example 10.18 Static and Dynamic polarizabilities, atomic units:

```
$fdpfreq
   0.0 0.03 0.05
   au
$end
```

Example 10.19 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 10.20 Static first hyperpolarizability, second harmonic generation and electro-optical Pockels effect, wavelength in nm:

```
$fdpfreq
   StaticHyper SHG EOPockels
   1064
   nm
$end
```

10.16.2 Features of Mopropman

- Restricted/unrestricted HF and KS-DFT CPSCF/TDSCF
- LinK/CFMM support to evaluate Coulomb- and exchange-like matrices
- DIIS acceleration
- Support of LSDA/GGA/Hybrid XC functionals listed below
- Analytical derivatives

The following XC functionals are supported:

Exchange:

- Dirac
- Becke 88

Correlation:

- Wigner
- VWN (both RPA and No. 5 parameterizations)
- Perdew-Zunger 81
- Perdew 86 (both PZ81 and VWN (No. 5) kernel)
- LYP

10.16.3 Job Control

The following options can be used:

MOPROP

Specifies the job for mopropman.

TYPE:

INTEGER

DEFAULT:

0 Do not run mopropman.

OPTIONS:

1	NMR chemical shielding tensors.
2	Static polarizability.
3	Indirect nuclear spin–spin coupling tensors.
100	Dynamic polarizability.
101	First hyperpolarizability.
102	First hyperpolarizability, reading First order results from disk.
103	First hyperpolarizability using Wigner's $(2n+1)$ rule.
104	First hyperpolarizability using Wigner's $(2n+1)$ rule, reading first order results from disk.

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 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 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 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 last $\mathbf{G}\left[\mathbf{P}^{\mathrm{x}}\right]$ when calculating dynamic polarizabilities in order to call mopropman in a second run with MOPROP = 102.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 False

1 True

RECOMMENDATION:

None

10.17 Electronic Couplings for Electron Transfer and Energy Transfer

10.17.1 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) [105], fragment-charge difference (FCD) [106], Boys localization [107], and Edmiston-Ruedenbeg localization [108] schemes. For EET, options include fragment-excitation difference (FED) [109], fragment-spin difference (FSD) [110], occupied-virtual separated Boys localization [111] or Edmiston-Ruedenberg localization [108]. In all these schemes, a vertical excitation such as CIS, RPA or TDDFT is required, and the GMH, FCD, FED, FSD, Boys or ER coupling values are calculated based on the excited state results.

10.17.1.1 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

$$\begin{aligned} \Delta q_{mn} &= q^{\mathrm{D}}_{mn} - q^{\mathrm{A}}_{mn} \\ &= \int_{\mathbf{r}\in\mathrm{D}} \rho_{mn}(\mathbf{r}) d\mathbf{r} - \int_{\mathbf{r}\in\mathrm{A}} \rho_{mn}(\mathbf{r}) d\mathbf{r} \end{aligned} \tag{10.65}$$

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 a excitation difference matrix is similarly defined with elements

$$\begin{aligned} \Delta x_{mn} &= x^{\mathrm{D}}_{mn} - x^{\mathrm{A}}_{mn} \\ &= \int_{\mathbf{r}\in\mathrm{D}} \rho^{(mn)}_{\mathrm{ex}}(\mathbf{r}) d\mathbf{r} - \int_{\mathbf{r}\in\mathrm{A}} \rho^{(mn)}_{\mathrm{ex}}(\mathbf{r}) d\mathbf{r} \end{aligned} \tag{10.66}$$

where $\rho^{(mn)}_{\mathrm{ex}}(\mathbf{r})$ 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,

$$\begin{aligned} \Delta s_{mn} &= s^{\mathrm{D}}_{mn} - s^{\mathrm{A}}_{mn} \\ &= \int_{\mathbf{r}\in\mathrm{D}} \sigma_{(mn)}(\mathbf{r}) d\mathbf{r} - \int_{\mathbf{r}\in\mathrm{A}} \sigma_{(mn)}(\mathbf{r}) d\mathbf{r} \end{aligned} \tag{10.67}$$

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. (10.65), (10.66), and (10.67), 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,

$$\overline{\Delta q}_{mn} = (\Delta q_{mn} + \Delta q_{nm})/2 \tag{10.68}$$

$$\overline{\Delta x}_{mn} = (\Delta x_{mn} + \Delta x_{nm})/2 \tag{10.69}$$

$$\overline{\Delta s}_{mn} = (\Delta s_{mn} + \Delta s_{nm})/2 \tag{10.70}$$

The final coupling values are obtained as listed below:

- For GMH,

$$V_{\mathrm{ET}} = \frac{(E_2 - E_1)\left|\vec{\mu}_{12}\right|}{\sqrt{(\vec{\mu}_{11} - \vec{\mu}_{22})^2 + 4\left|\vec{\mu}_{12}\right|^2}} \tag{10.71}$$

- For FCD,

$$V_{\text{ET}} = \frac{(E_2 - E_1)\overline{\Delta q}_{12}}{\sqrt{(\Delta q_{11} - \Delta q_{22})^2 + 4\overline{\Delta q}_{12}^2}} \tag{10.72}$$

- For FED,

$$V_{\text{EET}} = \frac{(E_2 - E_1)\overline{\Delta x}_{12}}{\sqrt{(\Delta x_{11} - \Delta x_{22})^2 + 4\overline{\Delta x}_{12}^2}} \tag{10.73}$$

- For FSD,

$$V_{\text{EET}} = \frac{(E_2 - E_1)\overline{\Delta s}_{12}}{\sqrt{(\Delta s_{11} - \Delta s_{22})^2 + 4\overline{\Delta s}_{12}^2}} \tag{10.74}$$

Q-CHEM provides the option to control FED, FSD, FCD and GMH calculations after a single-excitation calculation, such as CIS, RPA, TDDFT/TDA and TDDFT. To obtain ET coupling values using GMH (FCD) scheme, one should set *$rem* variables STS_GMH (STS_FCD) to be TRUE. Similarly, a FED (FSD) calculation is turned on by setting the *$rem* variable STS_FED (STS_FSD) to be TRUE. 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.

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 *i*th atom to the *j*th 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 the CIS and TDDFT calculations (including SF 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.

Example 10.21 A GMH & FCD calculation to analyze electron-transfer couplings in an ethylene and a methaniminium cation.

```
$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 10.22 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
```

10.17.1.2 Multi-state treatments

When dealing with multiple charge or electronic excitation centers, diabatic states can be constructed with Boys [107] or Edmiston-Ruedenberg [108] 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 $N_{state} \times N_{state}$ in size:

$$|\Xi_I\rangle = \sum_{J=1}^{N_{states}} |\Phi_J\rangle\, U_{ji} \qquad\qquad I = 1 \ldots N_{states} \tag{10.75}$$

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 or TDDFT methods in Q-Chem. In analogy to orbital localization, Boys-localized diabatization corresponds to maximizing the charge separation between diabatic state centers:

$$f_{Boys}(\mathbf{U}) = f_{Boys}(\{\Xi_I\}) = \sum_{I,J=1}^{N_{states}} \left|\langle\Xi_I|\vec{\mu}|\Xi_I\rangle - \langle\Xi_J|\vec{\mu}|\Xi_J\rangle\right|^2 \tag{10.76}$$

Here, $\vec{\mu}$ represents the dipole operator. ER-localized diabatization prescribes maximizing self-interaction energy:

$$\begin{aligned} f_{ER}(\mathbf{U}) &= f_{ER}(\{\Xi_I\}) \qquad (10.77) \\ &= \sum_{I=1}^{N_{states}} \int d\vec{\mathcal{R}}_1 \int d\vec{\mathcal{R}}_2 \frac{\langle \Xi_I | \hat{\rho}(\vec{\mathcal{R}}_2) | \Xi_I \rangle \langle \Xi_I | \hat{\rho}(\vec{\mathcal{R}}_1) | \Xi_I \rangle}{|\vec{\mathcal{R}}_1 - \vec{\mathcal{R}}_2|} \end{aligned}$$

where the density operator at position $\vec{\mathcal{R}}$ is

$$\hat{\rho}(\vec{\mathcal{R}}) = \sum_j \delta(\vec{\mathcal{R}} - \vec{r}^{(j)}) \qquad (10.78)$$

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 [107].

As written down in Eq. (10.76), 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 [111] by separately localizing the occupied and virtual components of $\vec{\mu}$, $\vec{\mu}^{occ}$ and $\vec{\mu}^{virt}$:

$$\begin{aligned} f_{BoysOV}(\mathbf{U}) &= f_{BoysOV}(\{\Xi_I\}) \qquad (10.79) \\ &= \sum_{I,J=1}^{N_{states}} \left(\left| \langle \Xi_I | \vec{\mu}^{occ} | \Xi_I \rangle - \langle \Xi_J | \vec{\mu}^{occ} | \Xi_J \rangle \right|^2 + \left| \langle \Xi_I | \vec{\mu}^{virt} | \Xi_I \rangle - \langle \Xi_J | \vec{\mu}^{virt} | \Xi_J \rangle \right|^2 \right) \end{aligned}$$

where

$$|\Xi_I\rangle = \sum_{ia} t_i^{Ia} |\Phi_i^a\rangle \qquad (10.80)$$

and the occupied/virtual components are defined by

$$\langle \Xi_I | \, \vec{\mu} \, | \Xi_J \rangle = \underbrace{\delta_{IJ} \sum_i \vec{\mu}_{ii} - \sum_{aij} t_i^{Ia} t_j^{Ja} \vec{\mu}_{ij}}_{\langle \Xi_I | \, \vec{\mu}^{occ} \, | \Xi_J \rangle} + \underbrace{\sum_{iba} t_i^{Ia} t_i^{Jb} \vec{\mu}_{ab}}_{\langle \Xi_I | \, \vec{\mu}^{virt} \, | \Xi_J \rangle} \qquad (10.81)$$

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 into Coulomb (J), Exchange (K) and one-electron (O) components [112]:

$$\langle \Xi_P | \, H \, | \Xi_Q \rangle = \underbrace{\sum_{iab} t_i^{Pa} t_i^{Qb} F_{ab} - \sum_{ija} t_i^{Pa} t_j^{Qa} F_{ij}}_{O} + \underbrace{\sum_{ijab} t_i^{Pa} t_j^{Qb} \, (ia|jb)}_{J} - \underbrace{\sum_{ijab} t_i^{Pa} t_j^{Qb} \, (ij|ab)}_{K} \qquad (10.82)$$

BOYS_CIS_NUMSTATE

Define how many states to mix with Boys localized diabatization.

TYPE:

INTEGER

DEFAULT:

0 Do not perform Boys localized diabatization.

OPTIONS:

1 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.

TYPE:

INTEGER

DEFAULT:

0 Do not perform ER localized diabatization.

OPTIONS:

1 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 TD-DFT diabatic states.

Example 10.23 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
jobtype          sp
method           cis
cis_n_roots      4
cis_singles   false
cis_triplets   true
basis            6-31g**
scf_convergence 8
symmetry         false
rpa              false
sym_ignore true
sym_ignore true
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.
$end

$localized_diabatization
On the next line, list which excited adiabatic states we want to mix.
1 2 3 4
$end
```

10.17.2 Diabatic-State-Based Methods

10.17.2.1 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:

$$\mathrm{D^-A} \longrightarrow \mathrm{DA^-} \tag{10.83}$$

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

$$\mathbf{H} - E\mathbf{S} = \begin{pmatrix} H_{ii} - S_{ii}E & H_{if} - S_{if}E \\ H_{if} - S_{if}E & H_{ff} - S_{ff}E \end{pmatrix} = 0 \tag{10.84}$$

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. (10.84) can be obtained by using the Löwdin transformation:

$$\mathbf{H}_{\text{eff}} = \mathbf{S}^{-1/2}\mathbf{H}\mathbf{S}^{-1/2}, \tag{10.85}$$

where the off-diagonal element of the effective Hamiltonian matrix represents the *electronic coupling* for the reaction, and it is defined by

$$V = H_{if}^{\text{eff}} = \frac{H_{if} - S_{if}(H_{ii} + H_{ff})/2}{1 - S_{if}^2} \tag{10.86}$$

In a general case where the initial and final states are not normalized, the electronic coupling is written as

$$V = \sqrt{S_{ii}S_{ff}} \times \frac{H_{if} - S_{if}(H_{ii}/S_{ii} + H_{ff}/S_{ff})/2}{S_{ii}S_{ff} - S_{if}^2} \tag{10.87}$$

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. (10.87), and it is widely used to calculate the electron transfer coupling [113–116]. In the DC scheme, the coupling matrix element is calculated directly using charge-localized determinants (the "diabatic states" in electron transfer literatures). In electron transfer systems, it has been shown that such charge-localized states can be approximated by symmetry-broken unrestricted Hartree-Fock (UHF) solutions [113, 114, 117]. The adiabatic eigenstates are assumed to be the symmetric and antisymmetric 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 [118] and a calculation for Slater's determinants in H_{if} and S_{if} [115, 116].

10.17.2.2 Corresponding orbital transformation

Let $|\Psi_a\rangle$ and $|\Psi_b\rangle$ be two single Slater-determinant wavefunctions for the initial and final states, and $\mathbf{a}$ and $\mathbf{b}$ be the spin-orbital sets, respectively:

$$\mathbf{a} = (a_1, a_2, \cdots, a_N) \tag{10.88}$$

$$\mathbf{b} = (b_1, b_2, \cdots, b_N) \tag{10.89}$$

Since the two sets of spin-orbitals are not orthogonal, the overlap matrix $\mathbf{S}$ can be defined as:

$$\mathbf{S} = \int \mathbf{b}^\dagger \mathbf{a}\, d\tau. \tag{10.90}$$

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 [118]. In this approach, each set of spin-orbitals $\mathbf{a}$ and $\mathbf{b}$ are linearly transformed,

$$\hat{\mathbf{a}} = \mathbf{aV} \tag{10.91}$$

$$\hat{\mathbf{b}} = \mathbf{bU} \tag{10.92}$$

where $\mathbf{V}$ and $\mathbf{U}$ are the left-singular and right-singular matrices, respectively, in the singular value decomposition (SVD) of $\mathbf{S}$:

$$\mathbf{S} = \mathbf{U}\hat{\mathbf{s}}\mathbf{V}^\dagger \tag{10.93}$$

The overlap matrix in the new basis is now diagonal

$$\int \hat{\mathbf{b}}^\dagger \hat{\mathbf{a}} = \mathbf{U}^\dagger \left(\int \mathbf{b}^\dagger \mathbf{a} \right) \mathbf{V} = \hat{\mathbf{s}} \tag{10.94}$$

10.17.2.3 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-electron operator $\Omega^{(1)}$ and two-electron operator $\Omega^{(2)}$,

$$\Omega^{(1)} = \sum_{i=1}^{N} \omega(i) \tag{10.95}$$

$$\Omega^{(2)} = \frac{1}{2} \sum_{i,j=1}^{N} \omega(i,j) \tag{10.96}$$

where $\omega(i)$ and $\omega(i,j)$, for the molecular Hamiltonian, are

$$\omega(i) = h(i) = -\frac{1}{2}\nabla_i^2 + V(i) \tag{10.97}$$

and

$$\omega(i,j) = \frac{1}{r_{ij}} \tag{10.98}$$

The evaluation of matrix elements can now proceed:

$$S_{ab} = \langle \Psi_b | \Psi_a \rangle = \det(\mathbf{U}) \det(\mathbf{V}^\dagger) \prod_{i=1}^{N} \hat{s}_{ii} \tag{10.99}$$

$$\Omega_{ab}^{(1)} = \langle \Psi_b | \Omega^{(1)} | \Psi_a \rangle = \det(\mathbf{U}) \det(\mathbf{V}^\dagger) \sum_{i=1}^{N} \langle \hat{b}_i | \omega(1) | \hat{a}_i \rangle \cdot \prod_{j \neq i}^{N} \hat{s}_{jj} \tag{10.100}$$

$$\Omega_{ab}^{(2)} = \langle \Psi_b | \Omega^{(2)} | \Psi_a \rangle = \frac{1}{2} \det(\mathbf{U}) \det(\mathbf{V}^\dagger) \sum_{ij}^{N} \langle \hat{b}_i \hat{b}_j | \omega(1,2)(1 - P_{12}) | \hat{a}_i \hat{a}_j \rangle \cdot \prod_{k \neq i,j}^{N} \hat{s}_{kk} \tag{10.101}$$

$$H_{ab} = \Omega_{ab}^{(1)} + \Omega_{ab}^{(2)} \tag{10.102}$$

In an atomic orbital basis set, $\{\chi\}$, we can expand the molecular spin orbitals $\mathbf{a}$ and $\mathbf{b}$,

$$\mathbf{a} = \chi\mathbf{A}, \qquad \hat{\mathbf{a}} = \chi\mathbf{AV} = \chi\hat{\mathbf{A}} \tag{10.103}$$

$$\mathbf{b} = \chi\mathbf{B}, \qquad \hat{\mathbf{b}} = \chi\mathbf{BU} = \chi\hat{\mathbf{B}} \tag{10.104}$$

The one-electron terms, Eq. (10.99), can be expressed as

$$\begin{aligned}\Omega_{ab}^{(1)} &= \sum_i^N \sum_{\lambda\sigma} \hat{A}_{\lambda i} T_{ii} \hat{B}_{i\sigma}^\dagger \langle \chi_\sigma | \omega(1) | \chi_\lambda \rangle \\ &= \sum_{\lambda\sigma} G_{\lambda\sigma} \omega_{\sigma\lambda}\end{aligned} \tag{10.105}$$

where $T_{ii} = S_{ab}/\hat{s}_{ii}$ and define a generalized density matrix, $\mathbf{G}$:

$$\mathbf{G} \;=\; \hat{\mathbf{A}}\mathbf{T}\hat{\mathbf{B}}^\dagger \tag{10.106}$$

Similarly, the two-electron terms [Eq. (10.101)] are

$$\begin{aligned}\Omega_{ab}^{(2)} &= \frac{1}{2}\sum_{ij}\sum_{\lambda\sigma}\sum_{\mu\nu} \hat{A}_{\lambda i}\hat{A}_{\sigma j}\left(\frac{1}{\hat{s}_{ii}}\right) T_{jj}\hat{B}_{i\mu}^\dagger \hat{B}_{j\nu}^\dagger \langle \chi_\mu\chi_\nu | \omega(1,2) | \chi_\lambda\chi_\sigma \rangle \\ &= \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^L G_{\sigma\nu}^R \langle \mu\nu || \lambda\sigma \rangle\end{aligned} \tag{10.107}$$

where $\mathbf{G}^R$ and $\mathbf{G}^L$ are generalized density matrices as defined in Eq. (10.106) except T_{ii} in $\mathbf{G}^L$ 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

$$\Omega_{ab}^{(1)} = \sum_{\lambda\sigma} G_{\lambda\sigma}^\alpha \omega_{\sigma\lambda} + \sum_{\lambda\sigma} G_{\lambda\sigma}^\beta \omega_{\sigma\lambda} \tag{10.108}$$

$$\begin{aligned}\Omega_{ab}^{(2)} &= \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{L\alpha} G_{\sigma\nu}^{R\alpha} (\langle \mu\nu | \lambda\sigma \rangle - \langle \mu\nu | \sigma\lambda \rangle) + \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{L\beta} G_{\sigma\nu}^{R\alpha} \langle \mu\nu | \lambda\sigma \rangle \\ &\quad + \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{L\alpha} G_{\sigma\nu}^{R\beta} \langle \mu\nu | \lambda\sigma \rangle + \sum_{\lambda\sigma\mu\nu} G_{\lambda\mu}^{L\beta} G_{\sigma\nu}^{R\beta} (\langle \mu\nu | \lambda\sigma \rangle - \langle \mu\nu | \sigma\lambda \rangle)\end{aligned} \tag{10.109}$$

The resulting one- and two-electron contributions, Eqs. (10.108) and (10.109) can be easily computed in terms of generalized density matrices using standard one- and two-electron integral routines in Q-CHEM.

10.17.2.4 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 two 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 wavefunctions of individual donor and acceptor fragments to form a charge-localized wavefunction. We call this approach “1+1” since the zeroth order wavefunctions are composed of the HF wavefunctions of the two fragments.

For example, for the case shown in Example (10.83), we can use Q-CHEM to calculate two HF wavefunctions: those of anionic donor and of neutral acceptor and they jointly form the initial state. For the final state, wavefunctions of neutral donor and anionic acceptor are used. Then the coupling value is calculated via Eq. (10.87).

Example 10.24 To calculate the electron-transfer coupling for a pair of stacked-ethylene 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
   JOBTYPE              SP
   METHOD               HF
   BASIS                6-31G(d)
   SCF_PRINT_FRGM       FALSE
   SYM_IGNORE           TRUE
   SCF_GUESS            FRAGMO
   STS_DC               TRUE
$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 *$rem* variables in this example, SYM_IGNORE, SCF_GUESS and STS_DC must be set to be the values as in the example in order to perform DC calculation with "1+1" charge-localized states. An additional *$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" wavefunction as an initial guess to look for the charge-localized wavefunction by further

HF self-consistent field calculation. This approach would 'relax' the wavefunction constructed by "1+1" method and include the intermolecular interaction effects in the initial and final wavefunctions. However, this method may sometimes fail, leading to either convergence problems or a resulting HF wavefunction that cannot represent the desired charge-localized states. This is more likely to be a problem when calculations are performed with with diffusive basis functions, or when the donor and acceptor molecules are very close to each other.

Example 10.25 To calculate the electron-transfer coupling for a pair of stacked-ethylene with "relaxed" 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
   JOBTYPE               SP
   METHOD                HF
   BASIS                 6-31G(d)
   SCF_PRINT_FRGM        FALSE
   SYM_IGNORE            TRUE
   SCF_GUESS             FRAGMO
   STS_DC                RELAX
$end
```

To perform 'relaxed' DC calculation, set STS_DC to be RELAX.

10.18 Calculating the Population of Effectively Unpaired ("odd") Electrons with DFT

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 [119–121]. 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 6.3.1). 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 (RDM-1) corresponding to a single-determinant wavefunction (e.g., SCF or Kohn-Sham DFT) is idempotent:

$$\int \gamma_\sigma^{\mathrm{scf}}(1;2)\gamma_\sigma^{\mathrm{scf}}(2;1)\, d\mathbf{r}_2 = \rho_\sigma(\mathbf{r}_1)\,,\ \gamma_\sigma^{\mathrm{scf}}(1;2) = \sum_i^{\mathrm{occ}} \psi_{i\sigma}^{\mathrm{ks}}(1)\psi_{i\sigma}^{\mathrm{ks}}(2)\,, \tag{10.110}$$

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 RDM-1 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:

$$\gamma_\sigma^{\mathrm{scf}}(1;2)\gamma_\sigma^{\mathrm{scf}}(2;1) = \rho_\alpha(1)h_{\mathrm{X}\sigma\sigma}(1,2)\,,\ \int h_{\mathrm{X}\sigma\sigma}(1,2)\, d\mathbf{r}_2 = 1\,. \tag{10.111}$$

When RDM-1 includes electron correlation, it becomes nonidempotent:

$$D_\sigma(1) \equiv \rho_\sigma(1) - \int \gamma_\sigma(1;2)\gamma_\sigma(2;1)\, d\mathbf{r}_2 \geq 0\,. \tag{10.112}$$

The function $D_\sigma(1)$ measures the deviation from idempotency of the correlated RDM-1 and yields the density of effectively-unpaired (odd) electrons of spin σ at point $\mathbf{r}_1$ [119, 122]. 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 $\bar{N}_u$:

$$D_u(1) = 2\sum_\sigma D_\sigma(1)d\mathbf{r}_1,\ \bar{N}_u = \int D_u(1)d\mathbf{r}_1\,. \tag{10.113}$$

The appearance of a factor of 2 in Eq. (10.113) above is required for reasons discussed in reference [122]. In Kohn-Sham DFT, the SCF RDM-1 is always idempotent which impedes the analysis of odd electron formation at that level of the theory. Ref. [123] has proposed a remedy to this situation. It was noted that the correlated RDM-1 cross product entering Eq. (10.112) reflects an effective exchange (also known as cumulant exchange [120]). 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 [124]. The effective exchange described with the correlated RDM-1 cross product should be fairly localized as well. With this in mind, the following form of the correlated RDM-1 cross product was proposed [123]:

$$\gamma_\sigma(1;2)\,\gamma_\sigma(2;1) = \rho_\sigma(1)\bar{h}_{\mathrm{X}\sigma\sigma}^{\mathrm{eff}}(1,2)\,. \tag{10.114}$$

where 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 [125]. The latter describes left-right static correlation effects in terms of certain effective exchange-correlation hole [125]. The extra delocalization of the HF exchange hole alone is compensated by certain physically motivated real-space corrections to it [125]:

$$\bar{h}_{\mathrm{XC}\alpha\alpha}(1,2) = \bar{h}_{\mathrm{X}\alpha\alpha}^{\mathrm{eff}}(1,2) + f_{\mathrm{c}}(1)\,\bar{h}_{\mathrm{X}\beta\beta}^{\mathrm{eff}}(1,2)\,, \tag{10.115}$$

where the BR exchange hole $\bar{h}_{\mathrm{X}\sigma\sigma}^{\mathrm{eff}}$ 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 1:

$$\int \bar{h}_{\mathrm{X}\sigma\sigma}^{\mathrm{eff}}(1;2)d\mathbf{r}_2 = N_{X\sigma}^{\mathrm{eff}}(1)\ \leq 1\,. \tag{10.116}$$

The expression of the relaxed normalization $N^{\rm eff}_{{\rm X}\sigma}(\mathbf{r})$ is quite complicated, but it is possible to represent it in closed analytic form [126, 127]. The smaller the relaxed normalization $N^{\rm eff}_{X\alpha}(1)$, the more delocalized the corresponding exact-exchange hole [125]. The $\alpha-\alpha$ exchange hole is further deepened by a fraction of the $\beta-\beta$ exchange hole, $f_{\rm c}(1)\,\bar h^{\rm eff}_{{\rm X}\beta\beta}(1,2)$, which gives rise to left-right static correlation. The local correlation factor $f_{\rm c}$ in Eq.(10.115) governs this deepening and hence the strength of the static correlation at each point [125]:

$$f_{\rm c}(\mathbf{r}) = \min\left(f_\alpha(\mathbf{r}),\, f_\beta(\mathbf{r}),\, 1\right)\ , \quad 0 \le f_{\rm c}(\mathbf{r})\ \le 1\ , f_\alpha(\mathbf{r}) = \frac{1-N^{\rm eff}_{{\rm X}\alpha}(\mathbf{r})}{N^{\rm eff}_{{\rm X}\beta}(\mathbf{r})}\,. \tag{10.117}$$

Using Eqs. (10.117), (10.113), and (10.114), the density of odd electrons becomes:

$$D_\alpha(1) = \rho_\alpha(1)(1-N^{\rm eff}_{X\alpha}(1)) \equiv \rho_\alpha(1) f_{\rm c}(1)\, N^{\rm eff}_{X\beta}(1)\,. \tag{10.118}$$

The final formulas for the spin-summed odd electron density and the total mean number of odd electrons read:

$$D_u(1) = 4\, a^{\rm op}_{\rm nd}\, f_{\rm c}(1)\left[\rho_\alpha(1)\,N^{\rm eff}_{X\beta}(1)+\rho_\beta(1)\,N^{\rm eff}_{X\alpha}(1)\right]\ ,\ \ \bar N_u = \int D_u(\mathbf{r_1})\,d\mathbf{r_1}\,. \tag{10.119}$$

Here $a_{\rm c}^{\rm nd-opp} = 0.526$ is the SCF-optimized linear coefficient of the opposite-spin static correlation energy term of the B05 functional [125, 127] .

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 ($F^{\rm r}_A$) as:

$$F^{\rm r}_A = \int_{\Omega_A} D_u(\mathbf{r_1})\,d\mathbf{r}_1\,. \tag{10.120}$$

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 [128]. 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 $F^{\rm r}_A$ and on the Kohn-Sham orbitals. The calculation of $F^{\rm r}_A$ 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 EXCHANGE rem line for these functionals. 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 10.26 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

```
$comment
Stretched H2: example of B3LYP calculation of
the atomic population of odd electrons
with post-SCF RI-BM05 extra iteration.
$end

$molecule
0 1
```

```
H   0.  0.   0.0
H   0.  0.   1.5000
$end

$rem
JOBTYPE     SP
SCF_GUESS   CORE
METHOD B3LYP
BASIS    G3LARGE
purcar 222
THRESH    14
MAX_SCF_CYCLES   80
PRINT_INPUT     TRUE
SCF_FINAL_PRINT  1
INCDFT    FALSE
XC_GRID 000128000302
SYM_IGNORE    TRUE
SYMMETRY      FALSE
SCF_CONVERGENCE   9
$end

@@@

$comment
Now one RI-B05 extra-iteration after B3LYP
to generate the odd-electron atomic population and the
correlated bond order.
$end

$molecule
READ
$end

$rem
JOBTYPE    SP
SCF_GUESS  READ
EXCHANGE BM05
purcar 22222
BASIS G3LARGE
AUX_BASIS riB05-cc-pvtz
THRESH    14
PRINT_INPUT     TRUE
INCDFT    FALSE
XC_GRID 000128000302
SYM_IGNORE    TRUE
SYMMETRY      FALSE
MAX_SCF_CYCLES   0
SCF_CONVERGENCE   9
dft_cutoffs 0
1415 1
$end

@@@

$comment
Finally, a fully SCF run RI-B05 using the previous output as a guess.
The following input lines are obligatory here:
purcar 22222
```

```
AUX_BASIS riB05-cc-pvtz
dft_cutoffs 0
1415 1
$end

$molecule
READ
$end

$rem
JOBTYPE    SP
SCF_GUESS  READ
EXCHANGE BM05
purcar 22222
BASIS G3LARGE
AUX_BASIS riB05-cc-pvtz
THRESH    14
PRINT_INPUT     TRUE
INCDFT    FALSE
IPRINT   3
XC_GRID 000128000302
SYM_IGNORE    TRUE
SCF_FINAL_PRINT  1
SYMMETRY      FALSE
MAX_SCF_CYCLES   80
SCF_CONVERGENCE   8
dft_cutoffs 0
1415 1
$end
```

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.

10.19 Quantum Transport Properties via the Landauer Approximation

Quantum transport at the molecule level involves bridging two electrodes with a molecule or molecular system, and calculating the properties of the resulting molecular electronic device, including current-voltage curves, the effect of the electrodes on the molecular states, etc. For a general introduction to the field, the following references are useful [129, 130].

The quantum transport code in Q-CHEM is developed by Prof. Barry Dunietz (Kent State) and his group. This package is invoked by the *$rem* variable TRANS_ENABLE.

TRANS_ENABLE

Decide whether or not to enable the molecular transport code.

TYPE:

INTEGER

DEFAULT:

0 Do not perform transport calculations.

OPTIONS:

1 Perform transport calculations in the Landauer approximation.

-1 Print matrices for subsequent calls for `tranchem.exe` as a stand-alone post-processing utility, or for generating bulk model files.

RECOMMENDATION:

Use as required.

Output is provided in the Q-CHEM output file and in the following additional files:

- `transmission.txt` (transmission function in the requested energy window)
- `TDOS.txt`
- `current.txt` (I-V plot if set)
- `FAmat.dat` (Hamiltonian matrix for follow up calculations and analysis)
- `Smat.dat` (Hamiltonian matrix for follow up calculations and analysis)

T-Chem requires two parameter files:

- `Trans-model.para` (Specifies the cluster model of the molecular device)
- `Trans-method.para` (Specifies the method(s) used in the transport calculation)

All the numbers must be provided. The details of file formats are given in Table 10.1 and Table 10.2.

Input example	Explanation
`142 Totorb`	total number of atomic orbitals in the cluster model of the molecular junction
`22 lob`	number of AO's representing the left electrode (the first lob functions of input)
`22 rob`	number of AO's representing the right electrode (the last rob functions of input)
`22 lg`	size of the repeating unit of the left electrode
`22 rg`	size of the repeating unit of the right electrode
`22 ql`	must be set to the lg value (later used in NEGF optimizations)
`22 qr`	must be set to the rg value (later used in NEGF optimizations)

Table 10.1: Example of a `Trans-model.para` file, and the meaning of each input line. All lines must be present

Key points to note about the `Trans-model.para` file, as documented in Table 10.1:

- WARNING: Unphysical setting of the regions is not caught by the program and might (will!) produce unphysical transmission functions. For example, the cluster model can be accidently partitioned by mistake within the orbital space of an atom. In fact, of course, It should always be partitioned between atomic layers.
- The ordering of the atoms in the *$molecule* section of the Q-CHEM input file *IS* important and is assumed to be as follows: Repeating units (left) - Molecular Junction - Repeating units (right).
- The atoms are provided by sets of left electrode repeating units first, in order of most distant from the surface layer. Then comes the the molecular region, followed by the right electrode region which starts with the surface layer to end with the most distant layer. The atom order within each layer (each repeating unit) must be consistent.

Input example	**Keywords included on each line**
`1`	`enable`
`0`	`AB`
`-6.5 -8.5 -4.5 300`	`efermi-emin-emax-npoints`
`0 0 0.01 0.01 0.07`	`method-htype-device-bulkr_smear-green_const`
`0 1 1`	`readinHS-tot-start`
`4.0 4 100 1.0`	`vmax-numres-numlinear-lpart`
`1 2 100`	`printDOS-printIV-ipoints`
`298.15`	`IVtemp`

Table 10.2: Example of a `Trans-method.para` file, and the keywords that correspond to each input line. The supported values of each keyword is listed below. All lines must be present.

The allowed values of the various keywords that comprise each line of the `Trans-method.para` file follow.

- `enable`: Sets the type of calculation. Allowed values are:
  - 1: Landauer level
  - 3: SCF GF at V=0 (not yet enabled)
  - 4: Full NEGF (not yet enabled)
- `AB`: Determines spins. Only closed shell singlet (0) allowed at present
- `emin,emax,npoints`: Energy window for which transmission is calculated (eV), and number of points
- `method`: Scheme to calculate the electrode GFs:
  - 0: A wide band limit (WBL) with a constant parameter (greens_const)
  - 1: WBL following the procedure proposed by Ke-Baranger-Yang
  - 2: WBL following the procedure proposed by Lopez-Sancho (decimation). Recommended
  - 3: Full TB following the procedure proposed by Ke-Baranger-Yang

  - 4: Full TB following the procedure proposed by Lopez-Sancho (decimation). Recommended
  - Comment: At least a single unit of the repeating layer of the electrodes is required to be included in the junction region. Another layer has to be included if readinhs==0 and a total of two additional layers are needed if `method` $\neq 0$.

- `htype`: determines the electronic coupling terms used for the self energy calculations:
  - 0: all coupling integrals between the junction and electrode are set following the cluster model (no screening imposed)
  - 1: only coupling between neighboring repeating units of the electrode model is allowed
  - 2: set the coupling terms by the electrode models that are read in (readinhs=1) or by the sufficiently large cluster model, where `lob (rob)` $\geq$ `2lg (rg)`
- `device-smear`: Imaginary smearing (eV) added to the real Hamiltonian for device GF evaluation.
- `bulk-smear`: Imaginary smearing (eV) added to the real Hamiltonian for electrodes GF evaluation.
- `readinhs`:
  - 0: no electrode Hamiltonians are available
  - 1: pre-calculated electrode Hamiltonians will be used. Expected files are `FAmat2l.dat, FBmat2l.dat, FAmat2r.dat, FBmat2r.dat, Smat2l.dat, Smat2r.dat.totorb2` is the total number of basis functions in the electrode model (same size is assumed for both electrodes), and, `start` is the first basis function from which the TB integral are to be extracted from.
- `vmax, numres, numlinear, lpart`
  - vmax = voltage should be equal to emax-emin if IV is set
  - other parameters here define the integration path for the NEGF algorithm (not used in this version, place holders must be included)
- `printDOS`
  - 0: no total dos printing
  - 1: a TDOS (of the junction region) will be printed as `TDOS.txt`
- `printIV (and ipoints, IVtemp)`
  - 0: no I-V information is calculated
  - 1: I(V=Vmax) is printed in `current.txt`
  - 2: I-V for bias from 0 to `Vmax` with a grid of `ipoints` at `IVtemp` is evaluated and printed in `current.txt`

As an example, the sample Q-CHEM input that works with the files documented in Table 10.1 and Table 10.2 is given below.

Example 10.27 Quantum transport calculation applied to C_6 between two gold electrodes.

```
$molecule
  0 1
  Au -0.2 0 0
  Au  2.5 0 0
  C   4.8 0 0
  C   6.5 0 0
  C   8.2 0 0
  C   9.9 0 0
  C  11.6 0 0
  C  13.3 0 0
  Au 15.6 0 0
  Au 18.3 0 0
$end

$rem
  jobtype               SP
  method                B3LYP
  BASIS                 lanl2dz
  ECP                   lanl2dz
  GEOM_OPT_MAXCYC       200
  INCDFT                FALSE
  mem_static            8000
  max_scf_cycles        400
  MEM_TOTAL             32000
  MOLDEN_FORMAT         TRUE
  scf_convergence       10
  scf_algorithm         diis
  trans_enable          1
$end
```

References and Further Reading

[1] Ground-State Methods (Chapters 4 and 5).

[2] Excited-State Calculations (Chapter 6).

[3] Basis Sets (Chapter 7).

[4] NBO 5.0 manual: www.chem.wisc.edu/∼nbo5.

[5] A. Szabo and N. S. Ostlund, *Modern Quantum Chemistry*, Dover, 1996.

[6] P.-O. Löwdin, *J. Chem. Phys.* **18**, 365 (1950).

[7] C. M. Breneman and K. B. Wiberg, *J. Comput. Chem.* **11**, 361 (1990).

[8] J. M. Herbert, L. D. Jacobson, K. U. Lao, and M. A. Rohrdanz, *Phys. Chem. Chem. Phys.* **14**, 7679 (2012).

[9] Z. C. Holden, R. M. Richard, and J. M. Herbert, *J. Chem. Phys.* **139**, 244108 (2013).

[10] F. L. Hirshfeld, *Theor. Chem. Acc.* **44**, 129 (1977).

[11] C. F. Williams and J. M. Herbert, *J. Phys. Chem. A* **112**, 6171 (2008).

[12] A. J. W. Thom, E. J. Sundstrom, and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **11**, 11297 (2009).

[13] Y. M. Rhee and M. Head-Gordon, *J. Am. Chem. Soc.* **130**, 3878 (2008).

[14] R. M. Richard and J. M. Herbert, *J. Chem. Theory Comput.* **7**, 1296 (2011).

[15] A. Dreuw and M. Head-Gordon, *Chem. Rev.* **105**, 4009 (2005).

[16] S. Hirata, M. Nooijen, and R. J. Bartlett, *Chem. Phys. Lett.* **326**, 255 (2000).

[17] E. D. Glendening, C. R. Landis, and F. Weinhold, *J. Comput. Chem.* **34**, 1429 (2013).

[18] E. D. Glendening et al., 2013.

[19] F. Weinhold, in *Computational Methods in Photochemistry*, edited by A. G. Kutateladze, volume 13 of *Molecular and Supramolecular Photochemistry*, page 393, Taylor & Francis, 2005.

[20] S. F. Boys, *Rev. Mod. Phys.* **32**, 296 (1960).

[21] S. F. Boys, in *Quantum Theory of Atoms, Molecules, and the Solid State*, edited by P.-O. Löwdin, page 253, Academic, New York, 1966.

[22] J. Pipek and P. G. Mezey, *J. Chem. Phys.* **90**, 4916 (1989).

[23] C. Edmiston and K. Ruedenberg, *Rev. Mod. Phys.* **35**, 457 (1963).

[24] J. E. Subotnik, Y. Shao, W. Liang, and M. Head-Gordon, *J. Chem. Phys.* **121**, 9220 (2004).

[25] G. Schaftenaar and J. H. Noordik, *J. Comput.-Aided Mol. Design* **14**, 123 (2000).

[26] The MOLDEN program may be freely downloaded from www.cmbi.ru.nl/molden/molden.html.

[27] B. M. Bode and M. S. Gordon, *J. Mol. Graphics Mod.* **16**, 133 (1998).

[28] MACMOLPLT may be downloaded from www.sci.ameslab.gov/∼brett/MacMolPlt.

[29] A. W. Lange and J. M. Herbert, *J. Am. Chem. Soc.* **131**, 124115 (2009).

[30] W. Humphrey, A. Dalke, and K. Schulten, *J. Molec. Graphics* **14**, 33 (1996).

[31] The VMD program may be downloaded from www.ks.uiuc.edu/Research/vmd.

[32] E. R. Johnson et al., *J. Chem. Phys.* **123**, 024101 (2005).

[33] J. Contreras-García et al., *J. Chem. Theory Comput.* **7**, 625 (2011).

[34] A. C. Simmonett, A. T. B. Gilbert, and P. M. W. Gill, *Mol. Phys.* **103**, 2789 (2005).

[35] T. Kato, *Commun. Pure Appl. Math.* **10**, 151 (1957).

[36] R. T. Pack and W. B. Brown, *J. Chem. Phys.* **45**, 556 (1966).

[37] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.* **104**, 9908 (1996).

[38] D. M. Chipman, *Theor. Chem. Acc.* **76**, 73 (1989).

[39] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.* **105**, 1470 (1996).

[40] V. A. Rassolov and D. M. Chipman, *J. Chem. Phys.* **105**, 1479 (1996).

[41] J. O. Hirschfelder, *J. Chem. Phys.* **33**, 1462 (1960).

[42] B. Wang, J. Baker, and P. Pulay, *Phys. Chem. Chem. Phys.* **2**, 2131 (2000).

[43] A. J. Stone and M. Alderton, *Mol. Phys.* **56**, 1047 (1985).

[44] P. M. W. Gill, D. P. O'Neill, and N. A. Besley, *Theor. Chem. Acc.* **109**, 241 (2003).

[45] A. M. Lee and P. M. W. Gill, *Chem. Phys. Lett.* **313**, 271 (1999).

[46] P. M. W. Gill, *Chem. Phys. Lett.* **270**, 193 (1997).

[47] N. A. Besley, A. M. Lee, and P. M. W. Gill, *Mol. Phys.* **100**, 1763 (2002).

[48] N. A. Besley, D. P. O'Neill, and P. M. W. Gill, *J. Chem. Phys.* **118**, 2033 (2003).

[49] E. Wigner, *Phys. Rev.* **40**, 749 (1932).

[50] V. I. Lebedev, *Zh. Vychisl. Mat. Mat. Fix.* **16**, 293 (1976).

[51] V. I. Lebedev, *Sibirsk. Mat. Zh.* **18**, 132 (1977).

[52] V. I. Lebedev and D. N. Laikov, *Dokl. Math.* **366**, 741 (1999).

[53] J. Cioslowski and G. Liu, *J. Chem. Phys.* **105**, 4151 (1996).

[54] B. G. Johnson and J. Florián, *Chem. Phys. Lett.* **247**, 120 (1995).

[55] P. P. Korambath, J. Kong, T. R. Furlani, and M. Head-Gordon, *Mol. Phys.* **100**, 1755 (2002).

[56] C. W. Murray, G. J. Laming, N. C. Handy, and R. D. Amos, *Chem. Phys. Lett.* **199**, 551 (1992).

[57] A. P. Scott and L. Radom, *J. Phys. Chem.* **100**, 16502 (1996).

[58] B. G. Johnson, P. M. W. Gill, and J. A. Pople, *J. Chem. Phys.* **98**, 5612 (1993).

[59] A. Miani, E. Cancès, P. Palmieri, A. Trombetti, and N. C. Handy, *J. Chem. Phys.* **112**, 248 (2000).

[60] R. Burcl, N. C. Handy, and S. Carter, *Spectrochim. Acta A* **59**, 1881 (2003).

[61] K. Yagi, K. Hirao, T. Taketsuga, M. W. Schmidt, and M. S. Gordon, *J. Chem. Phys.* **121**, 1383 (2004).

[62] V. Barone, *J. Chem. Phys.* **122**, 014108 (2005).

[63] S. D. Peyerimhoff, in *Encyclopedia of Computational Chemistry*, edited by P. v. R. Schleyer et al., page 2646, Wiley, Chichester, United Kingdom, 1998.

[64] T. Carrington, Jr., in *Encyclopedia of Computational Chemistry*, edited by P. v. R. Schleyer et al., page 3157, Wiley, Chichester, United Kingdom, 1998.

[65] A. Adel and D. M. Dennison, *Phys. Rev.* **43**, 716 (1933).

[66] E. B. Wilson and J. J. B. Howard, *J. Chem. Phys.* **4**, 260 (1936).

[67] H. H. Nielsen, *Phys. Rev.* **60**, 794 (1941).

[68] J. Neugebauer and B. A. Hess, *J. Chem. Phys.* **118**, 7215 (2003).

[69] R. J. Whitehead and N. C. Handy, *J. Mol. Spect.* **55**, 356 (1975).

[70] C. Y. Lin, A. T. B. Gilbert, and P. M. W. Gill, *Theor. Chem. Acc.* **120**, 23 (2008).

[71] N. A. Besley and K. A. Metcalf, *J. Chem. Phys.* **126**, 035101 (2007).

[72] N. A. Besley and J. A. Bryan, *J. Phys. Chem. C* **112**, 4308 (2008).

[73] M. W. D. Hanson-Heine, M. W. George, and N. A. Besley, *J. Chem. Phys.* **136**, 224102 (2012).

[74] W. J. Zheng and B. R. Brooks, *Biophys. J.* **88**, 3109 (2005).

[75] H. L. Woodcock et al., *J. Chem. Phys.* **129**, 214109 (2008).

[76] A. Ghysels et al., *J. Comput. Chem.* **31**, 994 (2010).

[77] W. D. Allen et al., *Chem. Phys.* **145**, 427 (1990).

[78] I. M. Mills, in *Molecular Spectroscopy: Modern Research*, edited by K. N. Rao and C. W. Mathews, chapter 3.2, Academic Press, New York, 1972.

[79] D. A. Clabo, W. D. Allen, R. B. Remington, Y. Yamaguchi, and H. F. Schaefer III, *Chem. Phys.* **123**, 187 (1988).

[80] H. H. Nielsen, *Rev. Mod. Phys.* **23**, 90 (1951).

[81] C. Ochsenfeld, *Phys. Chem. Chem. Phys.* **2**, 2153 (2000).

[82] C. Ochsenfeld, S. P. Brown, I. Schnell, J. Gauss, and H. W. Spiess, *J. Am. Chem. Soc.* **123**, 2597 (2001).

[83] R. Ditchfield, *Mol. Phys.* **27**, 789 (1974).

[84] K. Wolinski, J. F. Hinton, and P. Pulay, *J. Am. Chem. Soc.* **112**, 8251 (1990).

[85] M. Häser, R. Ahlrichs, H. P. Baron, P. Weiss, and H. Horn, *Theor. Chem. Acc.* **83**, 455 (1992).

[86] T. Helgaker and M. J. K. Ruud, *Chem. Rev.* **99**, 293 (1990).

[87] C. Ochsenfeld, J. Kussmann, and F. Koziol, *Angew. Chem.* **116**, 4585 (2004).

[88] J. Kussmann and C. Ochsenfeld, *J. Chem. Phys.* **127**, 204103 (2007).

[89] C. Ochsenfeld and M. Head-Gordon, *Chem. Phys. Lett.* **270**, 399 (1997).

[90] P. von Schleyer, C. Maerker, A. Dransfield, H. Jiao, and N. J. R. v. E. Hommes, *J. Am. Chem. Soc.* **118**, 6317 (1996).

[91] S. P. Brown et al., *Angew. Chem. Int. Ed. Engl.* **40**, 717 (2001).

[92] C. Ochsenfeld et al., *Solid State Nucl. Mag.* **22**, 128 (2002).

[93] J. Kussmann and C. Ochsenfeld, *J. Chem. Phys.* **127**, 054103 (2007).

[94] F. London, *J. Phys. Radium* **8**, 397 (1937).

[95] J. Gauss, *Ber. Bunsenges. Phys. Chem.* **99**, 1001 (1995).

[96] C. Ochsenfeld, C. A. White, and M. Head-Gordon, *J. Chem. Phys.* **109**, 1663 (1998).

[97] C. Ochsenfeld, *Chem. Phys. Lett.* **327**, 216 (2000).

[98] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *Chem. Phys. Lett.* **230**, 8 (1994).

[99] T. Helgaker, M. Watson, and N. Handy, *J. Chem. Phys.* **113**, 9402 (2000).

[100] V. Sychrovský, J. Gräfenstein, and D. Cremer, *J. Chem. Phys.* **113**, 3530 (2000).

[101] T. Bally and P. R. Rablen, *J. Org. Chem* **76**, 4818 (2011).

[102] F. Jensen, *J. Chem. Theory Comput.* **2**, 1360 (2006).

[103] H. Sekino and R. J. Bartlett, *J. Chem. Phys.* **85**, 976 (1986).

[104] S. P. Karna and M. Dupuis, *J. Comput. Chem.* **12**, 487 (1991).

[105] R. J. Cave and M. D. Newton, *Chem. Phys. Lett.* **249**, 15 (1996).

[106] A. A. Voityuk and N. Rösch, *J. Chem. Phys.* **117**, 5607 (2002).

[107] J. E. Subotnik, S. Yeganeh, R. J. Cave, and M. A. Ratner, *J. Chem. Phys.* **129**, 244101 (2008).

[108] J. E. Subotnik, R. J. Cave, R. P. Steele, and N. Shenvi, *J. Chem. Phys.* **130**, 234102 (2009).

[109] C.-P. Hsu, Z.-Q. You, and H.-C. Chen, *J. Phys. Chem. C* **112**, 1204 (2008).

[110] Z.-Q. You and C.-P. Hsu, *J. Chem. Phys.* **133**, 074105 (2010).

[111] J. E. Subotnik, J. Vura-Weis, A. Sodt, and M. A. Ratner, *J. Phys. Chem. A* **114**, 8665 (2010).

[112] J. Vura-Weis, M. Wasielewski, M. D. Newton, and J. E. Subotnik, *J. Phys. Chem. C* **114**, 20449 (2010).

[113] K. Ohta, G. L. Closs, K. Morokuma, and N. J. Green, *J. Am. Chem. Soc.* **108**, 1319 (1986).

[114] A. Broo and S. Larsson, *Chem. Phys.* **148**, 103 (1990).

[115] A. Farazdel, M. Dupuis, E. Clementi, and A. Aviram, *J. Am. Chem. Soc.* **112**, 4206 (1990).

[116] L. Y. Zhang, R. A. Friesner, and R. B. Murphy, *J. Chem. Phys.* **107**, 450 (1997).

[117] M. D. Newton, *Chem. Rev.* **91**, 767 (1991).

[118] H. F. King, R. E. Stanton, H. Kim, R. E. Wyatt, and R. G. Parr, *J. Chem. Phys.* **47**, 1936 (1967).

[119] K. Takatsuka, T. Fueno, and K. Yamaguchi, *Theor. Chem. Acc.* **48**, 175 (1978).

[120] R. C. Bochicchio, *J. Mol. Struct. (Theochem)* **429**, 229 (1998).

[121] V. N. Staroverov and E. R. Davidson, *Chem. Phys. Lett.* **330**, 161 (2000).

[122] E. Proynov, *J. Mol. Struct. (Theochem)* **762**, 159 (2006).

[123] E. Proynov, F. Liu, and J. Kong, *Phys. Rev. Lett.*, submitted revision.

[124] A. D. Becke, *J. Chem. Phys.* **119**, 2972 (2003).

[125] A. D. Becke and E. R. Johnson, *J. Chem. Phys.* **122**, 154104 (2005).

[126] E. Proynov, Y. Shao, and J. Kong, *Chem. Phys. Lett.* **493**, 381 (2010).

[127] E. Proynov, F. Liu, Y. Shao, and J. Kong, *J. Chem. Phys.* **136**, 034102 (2012).

[128] A. D. Becke, *J. Chem. Phys.* **88**, 2547 (1988).

[129] S. Datta, *Quantum transport: Atom to transistor*, Cambridge University Press, Cambridge, 2005.

[130] M. Di Ventra, *Electron transport in nanoscale systems*, Cambridge University Press, Cambridge, 2008.

Chapter 11

Molecules in Complex Environments: Solvent Models, QM/MM and QM/EFP Features, Density Embedding

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 solvent models;
- QM/MM tools;
- EFP and QM/EFP approach (polarizable electrostatic embedding);
- Density embedding methods.

11.2 Chemical Solvent Models

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 super-molecular 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. Accurate prediction of solvation free energies is, however, crucial for modeling of chemical reactions and ligand/receptor interactions in solution.

Q-CHEM contains several different implicit solvent models, which differ greatly in their level of sophistication and realism. These are generally known as self-consistent reaction field (SCRF) models, because the continuum solvent establishes a reaction field (*i.e.*, additional terms in the solute Hamiltonian) that depends upon the solute electron density, and must therefore be updated self-consistently during the iterative

Model	Cavity		Non-Electrostatic Terms?	Supported Basis Sets	Derivatives Available?
	Construction	Discretization			
Kirkwood-Onsager	spherical	point charges	no	all	none
Langevin Dipoles	atomic spheres (user-definable)	dipoles in 3-d space	no	all	none
C-PCM	atomic spheres (user-definable)	point charges or smooth Gaussians	user-specified	all	1st and 2nd
SS(V)PE/ IEF-PCM	atomic spheres (user-definable)	point charges or smooth Gaussians	user-specified	all	1st
COSMO	predefined atomic spheres	point charges	none	all	1st and 2nd
Isodensity SS(V)PE	isodensity contour	point charges	none	all	none
SM8	predefined atomic spheres	generalized Born	automatic	6-31G* 6-31+G* 6-31+G**	1st
SM12	predefined atomic spheres	generalized Born	automatic	all	none

Table 11.1: Summary of implicit solvation models available in Q-CHEM, indicating how the solute cavity is constructed and discretized, whether non-electrostatic terms are (or can be) included, which basis sets are available for use with each model, and whether analytic first and second derivatives are available for optimizations and frequency calculations.

convergence of the wavefunction. The simplest and oldest of these models that is available in Q-CHEM is the Kirkwood-Onsager model [1–3], 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. More sophisticated models, which use a molecule-shaped cavity and the full molecular electrostatic potential, include the conductor-like screening model (COSMO) [4] and the closely-related conductor-like PCM (C-PCM) [5–7], along with the "surface and simulation of volume polarization for electrostatics" [SS(V)PE] model [8]. The latter is also known as the "integral equation formalism" (IEF-PCM) [9, 10].

The C-PCM and IEF-PCM/SS(V)PE are examples of what are called "apparent surface charge" SCRF models, although the term *polarizable continuum models* (PCMs), as popularized by Tomasi and coworkers [11], 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 [12–15]. This approach 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 [12, 13]. 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 [16, 17] as well as versions 8 and 12 of the SMx models developed at the University of Minnesota [18, 19]. SM8 and SM12 are based upon the generalized Born method for electrostatics, augmented with atomic surface tensions intended to capture non-electrostatic effects (cavitation, dispersion, exchange repulsion, and changes in solvent structure). Empirical corrections of this sort are also available for the PCMs mentioned above, but within SM8 and SM12 these parameters have been optimized to reproduce experimental solvation energies.

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, and Table 11.1 provides a summary of the implicit solvent models that are available in Q-CHEM. In general, these methods are available for any SCF level of electronic structure theory, though in the case of SM8 only certain basis sets are supported. Post-Hartree–Fock calculations can be performed by first running an SCF + PCM job, in which case the correlated wavefunction will employ MOs and Hartree-Fock energy levels that are polarized by the solvent. *Note that the job-control format for specifying implicit solvent models changed significantly starting in* Q-CHEM *version 4.2.1*. This change was made in an attempt to simply and unify the input notation for a large number of different models.

SOLVENT_METHOD

Sets the preferred solvent method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0	Do not use a solvation model.
ONSAGER	Use the Kirkwood-Onsager model (Section 11.2.1).
PCM	Use an apparent surface charge, polarizable continuum model (Section 11.2.2).
ISOSVP	Use the iso-density implementation of the SS(V)PE model (Section 11.2.5).
COSMO	Use COSMO (similar to C-PCM but with an outlying charge correction [20, 21]; see Section 11.2.6).
SM8	Use version 8 of the Cramer-Truhlar SMx model (Section 11.2.7.1).
SM12	Use version 12 of the SMx model (Section 11.2.7.2).
CHEM_SOL	Use the Langevin Dipoles model (Section 11.2.8).

RECOMMENDATION:

Consult the literature. 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 11.2.2.) Several versions of SM12 are available as well, as discussed in Section 11.2.7.2.

Before going into detail about each of these models, a few potential points of confusion warrant mention, with regards to nomenclature. First, "PCM" refers to a family of models that includes C-PCM and SS(V)PE/IEF-PCM (the latter two being completely equivalent [10]). One or the other of these models can be selected by additional job control variables in a *$pcm* input section, as described in Section 11.2.2. 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") [20, 21]. This is discussed in Section 11.2.6.

Two implementations of the SS(V)PE model are also available. The PCM implementation (which is requested by setting SOLVENT_METHOD = PCM) uses a solute cavity constructed from atom-centered spheres, as with most 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, as advocated by Chipman [8, 22, 23]. This is an appealing, one-parameter cavity construction, although it is unclear that this construction alone is superior in its accuracy to carefully-parameterized atomic radii [24], at least not without additional, non-electrostatic terms included [25, 26] that are unfortunately not yet available in Q-CHEM's implementation of the isodensity version of SS(V)PE. Moreover, analytic energy gradients are not available for the isodensity cavity construction, whereas they are available when the cavity is constructed from atom-centered spheres. One additional subtlety, which is discussed in detail in Ref. [14], is the fact that the PCM implementation of the equation for the SS(V)PE surface charges [Eq. (11.2)] uses an asymmetric $\mathbf{K}$ matrix. In contrast, Chipman's isodensity implementation uses a symmetrized $\mathbf{K}$ matrix. Although the symmetrized version is somewhat more computationally efficient when the number of surface charges is large, the asymmetric version is better justified, theoretically [14]. (This admittedly technical point is clarified in Section 11.2.2 and in particular in Table 11.2.)

Regarding the accuracy of these models for solvation free energies (ΔG_{298}), SM8 achieves 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 [27]. To achieve comparable accuracy with IEF-PCM/ SS(V)PE, non-electrostatic terms must be included [25, 26, 28]. The SM12 model does not improve upon SM8 in any statistical sense [19], but does lift one important restriction on the level of electronic structure that can be combined with these models. Specifically, the Generalized Born model used in SM8 is based on a variant of Mulliken-style atomic charges, and is therefore parameterized only for a few small basis sets, *e.g.*, 6-31G*. SM12, on the other hand, uses a variety of charge schemes that are stable with respect to basis-set expansion, and can therefore be combined with any level of electronic structure theory for the solute. Quantitative fluid-phase thermodynamics can also be obtained using Klamt's COSMO-RS approach [29, 30], where RS stands for "real solvent". The COSMO-RS approach is *not* included in Q-CHEM and requires the COSMOtherm program, which is licensed separately through COSMOlogic [31], but Q-CHEM can write the input files that are need by COSMOtherm.

The following sections provide more details regarding theory and job control for the various implicit solvent models that are available in Q-CHEM. In addition, recent review articles are available for PCM methods [11], SMx [27], and COSMO [32]. Formal relationships between various PCMs have been discussed in Refs. [14, 22].

11.2.1 Kirkwood-Onsager Model

The simplest implicit solvation model available in Q-CHEM is the Kirkwood-Onsager model [1–3], wherein the solute is placed inside of a spherical cavity that is surrounded by a homogeneous dielectric medium. This model is characterized by two parameters: the cavity radius, a, and the solvent dielectric constant, ε. The former is typically calculated according to

$$a = (3V_m/4\pi N_A)^{1/3} \tag{11.1}$$

where V_m is the solute's molar volume, usually obtained from experiment (molecular weight or density [33]), and N_A is Avogadro's number. 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 [34]. Alternatively, a is sometimes selected as 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 the van der Waals radii of the relevant atoms. Unfortunately, solvation energies are typically quite sensitive to the choice of a (and to the construction of the solute cavity, more generally).

Unlike older versions of the Kirkwood-Onsager model, in which the solute's electron distribution was described entirely in terms of its dipole moment, Q-CHEM's version can use multipoles of arbitrarily high order, including the Born (monopole) term [35] for charged solutes, in order to describe the solute's electrostatic potential. The solute–continuum electrostatic interaction energy is then computed using analytic expressions for the interaction of the point multipoles with a dielectric continuum.

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.

The Kirkwood-Onsager SCRF is requested by setting SOLVENT_METHOD = ONSAGER in the *$rem* section (along with normal job control variables for an energy or gradient calculation), and furthermore specifying several additional options in a *$solvent* input section, as described below. Of these, the keyword **CavityRadius** is required. The *$rem* variable CC_SAVEAMPL may save some time for CCSD calculations using the Kirkwood-Onsager model.

NOTE: The following three job control variables belong *only* in the *$solvent* section. Do not place them in the *$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 Desired cavity radius, in Ångstroms.
RECOMMENDATION:
Use Eq. (11.1).

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:
As per required solvent; the default 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 Onsager model applied at the Hartree-Fock level to 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    Onsager
$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 CCSD/Onsager calculation applied to 1,2-dichloroethane molecule

```
$comment
1,2-dichloroethane GAUCHE Conformation
$end

$molecule
0 1
C     0.6541334418569877  -0.3817051480045552   0.8808840579322241
C    -0.6541334418569877   0.3817051480045552   0.8808840579322241
Cl    1.7322599856434779   0.0877596094659600  -0.4630557359272908
H     1.1862455146007043  -0.1665749506296433   1.7960750032785453
H     0.4889356972641761  -1.4444403797631731   0.8058465784063975
Cl   -1.7322599856434779  -0.0877596094659600  -0.4630557359272908
H    -1.1862455146007043   0.1665749506296433   1.7960750032785453
H    -0.4889356972641761   1.4444403797631731   0.8058465784063975
$end

$rem
METHOD           CCSD
BASIS            6-31g**
N_FROZEN_CORE    FC
CC_SAVEAMPL      1         ! Save the CC amplitudes on disk
SOLVENT_METHOD   ONSAGER
$end

$solvent
CavityRadius     3.65  !  in Angstroms
Dielectric       8.93  !  methylene chloride
$end
```

Example 11.3 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
0 1
  H       0.000000     0.000000    -0.862674
  F       0.000000     0.000000     0.043813
$end

$rem
JOBTYPE           FORCE
METHOD            HF
BASIS             6-31G*
SOLVENT_METHOD    ONSAGER
SCF_GUESS         READ  ! read vacuum solution as a guess
$end

$solvent
CavityRadius    2.5
$end
```

11.2.2 Polarizable Continuum Models

Clearly, the Kirkwood-Onsager model is inappropriate if the solute is very non-spherical. Nowadays, a more general class of "apparent surface charge" SCRF solvation models are much more popular, to the extent that the generic term "polarizable continuum model" (PCM) is typically used to denote these methods [11]. Apparent surface charge PCMs improve upon the Kirkwood-Onsager model in two ways. Most importantly, they provide a much more realistic description of molecular shape, typically by constructing the solute cavity from a union of atom-centered spheres. 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 wavefunction must be determined self-consistently.

11.2.2.1 Formal Theory and Discussion of Different Models

The PCM literature has a long history [11] and there are several different models in widespread use; connections between these models have not always been appreciated [8, 10, 14, 22]. Chipman [8, 22] has shown how various PCMs can be formulated within a common theoretical framework; see Ref. [15] for a pedagogical introduction. The PCM takes the form of a set of linear equations,

$$\mathbf{K}\mathbf{q} = \mathbf{R}\mathbf{v} \;, \tag{11.2}$$

Model	Literature Refs.	Matrix $\mathbf{K}$	Matrix $\mathbf{R}$	Scalar f_ε
COSMO	[4]	$\mathbf{S}$	$-f_\varepsilon\mathbf{1}$	$\frac{\varepsilon-1}{\varepsilon+1/2}$
C-PCM	[5, 6]	$\mathbf{S}$	$-f_\varepsilon\mathbf{1}$	$\frac{\varepsilon-1}{\varepsilon}$
SS(V)PE/ IEF-PCM	[8, 10]	$\mathbf{S}-\frac{f_\varepsilon}{2\pi}\mathbf{DAS}$	$-f_\varepsilon\left(\mathbf{1}-\frac{1}{2\pi}\mathbf{DA}\right)$	$\frac{\varepsilon-1}{\varepsilon+1}$
Isodensity SS(V)PE	[8, 23]	$\mathbf{S}-\frac{f_\varepsilon}{4\pi}\left(\mathbf{DAS}+\mathbf{SAD}^\dagger\right)$	$-f_\varepsilon\left(\mathbf{1}-\frac{1}{2\pi}\mathbf{DA}\right)$	$\frac{\varepsilon-1}{\varepsilon+1}$

Table 11.2: 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. [15, 22, 23] 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. At the level of Eq. (11.2), COSMO and C-PCM differ only in the dielectric screening factor f_ε, although COSMO includes an additional outlying charge correction that goes beyond Eq. (11.2) [20, 21].

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 11.2 for the PCMs that are available in Q-CHEM.

The oldest PCM is the so-called D-PCM model of Tomasi and co-workers [36], but unlike the models listed in Table 11.2, 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) [8]. 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 Screening Model (COSMO) introduced by Klamt and Schüürmann [4]. Truong and Stefanovich [5] later implemented the same model with a slightly different dielectric scaling factor (f_ε in Table 11.2), and called this modification GCOSMO. The latter was implemented within the PCM formalism by Barone and Cossi *et al.* [6, 7], who called the model C-PCM (for "conductor-like" PCM). In each case, the dielectric screening factor has the form

$$f_\varepsilon = \frac{\varepsilon-1}{\varepsilon+x}\ , \tag{11.3}$$

where Klamt and Schüürmann proposed $x = 1/2$ but $x = 0$ was used in GCOSMO and C-PCM. The latter value is the correct choice for a single charge in a spherical cavity (*i.e.*, the Born ion model), although Klamt and co-workers suggest that $x = 1/2$ is a better compromise, given that the Kirkwood-Onsager analytical result is $x = \ell/(\ell+1)$ for an ℓth-order multipole centered in a spherical cavity [4, 21]. The distinction is irrelevant in high-dielectric solvents; the $x = 0$ and $x = 1/2$ values of f_ε differ by only 0.6% for water at 25°C, for example. Truong [5] argues that $x = 0$ does a better job of preserving Gauss' Law in low-dielectric solvents, but more accurate solvation energies (at least for neutral molecules, as compared to experiment) are sometimes obtained using $x = 1/2$ [6]. This result is likely highly sensitive to cavity construction, and in any case, both versions are available in Q-CHEM.

Whereas the original COSMO model introduced by Klamt and Schüürmann [4] corresponds to Eq. (11.2) with $\mathbf{K}$ and $\mathbf{R}$ as defined in Table 11.2, Klamt and co-workers later introduced a correction for outlying charge that goes beyond Eq. (11.2) [20, 21]. Klamt now consistently refers to this updated model as "COSMO" [32], and we shall adopt this nomenclature as well. COSMO, with the outlying charge correction, is available in Q-CHEM and is described in Section 11.2.6. In contrast, C-PCM consists entirely of Eq. (11.2) with matrices $\mathbf{K}$ and $\mathbf{R}$ as defined in Table 11.2, although it is possible to modify the dielectric screening factor to use the $x = 1/2$ value (as in COSMO) rather than the $x = 0$ value. Additional non-electrostatic terms can be added at the user's discretion, as discussed below, but there is no explicit outlying charge correction in C-PCM. These and other fine-tuning details for PCM jobs are controllable via the *$pcm* input section that is described in Section 11.2.3.

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 [8]. 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 term SS(V)PE is Chipman's notation [8], but this model is formally equivalent to the "integral equation formalism" (IEF-PCM) that was developed originally by Cancès *et al.* [9, 37]. 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 11.2 [8, 10]. This version requires only the electrostatic potential, and is thus preferred, and it is this version that we designate as SS(V)PE/IEF-PCM. The C-PCM model becomes equivalent to SS(V)PE in the limit $\varepsilon \to \infty$ [8, 14], which means that C-PCM must somehow include an *implicit* correction for volume polarization, even if this was not by design [20]. For $\varepsilon \gtrsim 50$, numerical calculations reveal that there is essentially no difference between SS(V)PE and C-PCM results [14]. 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.

In terms of the numerical implementation of SS(V)PE/IEF-PCM, there is some ambiguity as to how the integral equations that define this model should be converted into the finite-dimensional matrix equation that is given in Eq. (11.2). As discussed in Ref. [14], for non-spherical cavities discretization fails to preserve certain formally exact symmetries of the integral operators $\hat{S}$ and $\hat{D}$ whose discretized forms are the matrices $\mathbf{S}$ and $\mathbf{D}$. For the PCM implementation of SS(V)PE, Q-CHEM uses an asymmetric form, $\mathbf{K} = \mathbf{S} - (f_\varepsilon/2\pi)\mathbf{DAS}$ (see Table 11.2), as this recovers the correct $\varepsilon \to \infty$ limit even in discretized form [14]. (Note that $\mathbf{K}$ is not symmetric in this case since $\mathbf{D} \neq \mathbf{D}^\dagger$.) For more efficient solution of Eq. (11.2), Chipman [8, 22, 23]) has often used a symmetrized version with $\mathbf{K} = \mathbf{S} - (f_\varepsilon/4\pi)(\mathbf{DAS} + \mathbf{SAD}^\dagger)$ instead, and this is the form of $\mathbf{K}$ that is used in the isodensity implementation of SS(V)PE that is discussed in Section 11.2.5. For a more detailed discussion of the history of these models, see the lengthy and comprehensive review by Tomasi *et al.* 11. For a briefer discussion of the connections between these models, see Refs. 14, 15, 22.

11.2.2.2 Cavity Construction and Discretization

Construction of the cavity surface is a crucial aspect of PCMs, and computed properties are quite sensitive to the details of the cavity construction. Traditionally [38, 39], and by default in Q-CHEM, solute cavities are constructed from a union of atom-centered spheres whose radii are 1.2 times the atomic van der Waals radii suggested by Bondi [40]. (This 20% augmentation is intended to mimic the fact that solvent molecules cannot approach all the way to the van der Waals radius of the solute atoms, but it is not clear that this is an optimal value. The default value in Q-CHEM is 1.2, but this value can be altered by the user.) Once the atomic radii are selected, the cavity surface is discretized using atom-centered Lebedev grids [41–43] of the same sort that are used to perform the numerical integrations in DFT. 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 [12].

A long-standing (though not well-publicized) problem with the aforementioned discretization procedure is that 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 [12]. It is also a fundamental hindrance to molecular dynamics calculations [13]. Recently, Lange and Herbert [12, 13] (building upon earlier work by York and Karplus [44]) 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 [13, 15]. 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 [12, 13]. 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. 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 wavefunctions can be performed in conjunction with these solvent models, in which case the correlated wavefunction calculation will utilize 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).

Researchers who use these PCMs are asked to cite Refs. 13, 14, which provide the details of Q-CHEM's implementation. (We point the reader in particular to Ref. 13, 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 11.2) as well as how the cavity was constructed (*e.g.*, "Bondi radii multiplied by 1.2"). Absent these details, PCM calculations will be difficult to reproduce in other electronic structure programs.

11.2.3 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.

11.2.3.1 *$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 with $f_\varepsilon = (\varepsilon - 1)/\varepsilon$.
COSMO	Original conductor-like screening model with $f_\varepsilon = (\varepsilon - 1)/(\varepsilon + 1/2)$.
SSVPE	SS(V)PE model with an asymmetric $\mathbf{K}$ matrix.

RECOMMENDATION:

The SS(V)PE model is more sophisticated model than either C-PCM or COSMO, and probably more appropriate for low-dielectric solvents, but it is also more computationally demanding. In high-dielectric solvents there is little difference between these models. Note that the keyword **COSMO** in this context simply affects the dielectric screening factor f_ε; to obtain the outlying charge correction suggested by Klamt [20, 21], one should use SOLVENT_METHOD = COSMO rather than SOLVENT_METHOD = PCM. (See Section 11.2.6.)

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. 13, 14 for a discussion of these differences. The **Fixed** option uses the Variable Tesserae Number (VTN) algorithm of Li and Jensen [45], with Lebedev grid points. VTN uses point charges with no switching function or Gaussian blurring, and is therefore subject to discontinuities in geometry optimizations. It is not recommended, except to make contact with other calculations in the literature.

SwitchThresh

The 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.

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 van der Waals radii scaled by a factor of 1.2. The most widely-used set of van der Waals radii are those determined from crystallographic data by Bondi [40], although the radius for hydrogen was later adjusted to 1.1 Å [46]. Bondi's analysis was later extended to the whole main group [47], and this entire extended set of van der Waals radii is available in Q-CHEM. (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 [48]. 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. Finally, the user may specify his or her own radii for cavity construction using a *$van_der_waals* input section, the format for which is described in Section 11.2.8. 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, in which the hydrogen atoms do not get their own spheres and the heavy-atom radii are increased to compensate [24].

Scaling the atomic radii is a crude way to account for the fact that solvent molecules should not typically penetrate all the way to the van der Waals radii of the solute. Another way to account for this effect is to employ a "solvent-accessible" cavity surface, which is constructed from the van der Waals surface by adding a certain value to each atomic radius. This value is the presumed radius of a solvent molecule. (The value 1.4 Å is often used for water.) The capability to specify a solvent "probe" radius to be added to the atomic radii of choice is available in Q-CHEM.

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.
FF	Use Lennard-Jones radii from a molecular mechanics force field.
UFF	Use radii form the Universal Force Field.
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 *$rem* variable, and also to define the atom types in the *$molecule* section (see Section 11.3). This is not required for UFF radii.

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:

f Use a scaling factor of $f > 0$.

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.

SASrad

Form a “solvent accessible” surface with the given solvent probe radius.

INPUT SECTION: **$pcm**

TYPE:

FLOAT

DEFAULT:

None

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 the user should consult the literature regarding the use of solvent-accessible surfaces.

Historically, discretization of the cavity surface has involved “tessellation” methods that divide the cavity surface area into finite polygonal “tesserae”. (The GEPOL algorithm [49] 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 [12–14] 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 [13, 14]. The default for MM atoms (in MM/PCM or QM/MM/PCM jobs) is $N = 110$ Lebedev points per atomic sphere, whereas the default for QM atoms is $N = 302$. (This represents a change relative to Q-CHEM versions earlier than 4.2.1, where the default for QM atoms was $N = 590$.) These default values exhibit good rotational invariance and absolute solvation energies that, in most cases, lie within ∼0.5–1.0 kcal/mol of the $N \to \infty$ limit [13], although exceptions (especially where charged solutes are involved) can be found [14]. The acceptable values for the number of Lebedev points per sphere are N = 26, 50, 110, 194, 302, 434, 590, 770, 974, 1202, 1454, 1730, 2030, 2354, 2702, 3074, 3470, 3890, 4334, 4802, or 5294.

HPoints

The number of Lebedev grid points to be placed on H atoms in the QM system.

INPUT SECTION: **$pcm**

TYPE:

INTEGER

DEFAULT:

302

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.

HeavyPoints

The number of Lebedev grid points to be placed non-hydrogen atoms in the QM system.

INPUT SECTION: **$pcm**

TYPE:

INTEGER

DEFAULT:

302

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.

MMHPoints

The number of Lebedev grid points to be placed on H 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.

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.

Especially for complicated molecules, the user may want to visualize the cavity surface. This can be accomplished by setting **PrintLevel** $\geq$ 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 [50, 51], 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. 12.)

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 *$pcm* input selection requests the construction of a solute cavity consisting of a single, fixed sphere. This is generally not recommended but is occasionally useful for making contact with the results of Born models in the literature, or the Kirkwood-Onsager model discussed in Section 11.2.1. In this case, the cavity radius and its center must also be specified in the *$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 Angstroms.
RECOMMENDATION:
None.

CavityCenter
Specifies the center of the spherical solute cavity.
INPUT SECTION: **$pcm**
TYPE:
FLOAT
DEFAULT:
0.0 0.0 0.0
OPTIONS:
x y z Coordinates of the cavity center, in Angstroms.
RECOMMENDATION:
The format is **CavityCenter** followed by three floating-point values, delineated by spaces. Uses the same coordinate system as the *$molecule* section.

11.2.3.2 Examples

The following example shows a very basic PCM job. The solvent dielectric is specified in the *$solvent* section, which is described below.

Example 11.4 A basic example of using the PCMs: optimization of trifluoroethanol in water.

```
$rem
JOBTYPE         OPT
BASIS           6-31G*
METHOD          B3LYP
SOLVENT_METHOD  PCM
$end

$pcm
Theory          CPCM
Method          SWIG
Solver          Inversion
```

```
HeavyPoints     194
HPoints         194
Radii           Bondi
vdwScale        1.2
$end

$solvent
Dielectric 78.39
$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
```

The next example uses a single spherical cavity and should be compared to the Kirkwood-Onsager job, Example 11.1 on page 572.

Example 11.5 PCM with a single spherical cavity, applied to 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    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 van der Waals 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 11.2.8).

Example 11.6 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       1.3862000000     0.0000000000     0.0000000000
          C       0.6931000000     1.2004844147     0.0000000000
          C      -0.6931000000     1.2004844147     0.0000000000
          C      -1.3862000000     0.0000000000     0.0000000000
          C      -0.6931000000    -1.2004844147     0.0000000000
          C       0.6931000000    -1.2004844147     0.0000000000
          H       2.4618000000     0.0000000000     0.0000000000
          H       1.2309000000     2.1319813390     0.0000000000
          H      -1.2309000000     2.1319813390     0.0000000000
          H      -2.4618000000     0.0000000000     0.0000000000
          H      -1.2309000000    -2.1319813390     0.0000000000
          H       1.2309000000    -2.1319813390     0.0000000000
$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
$end
```

11.2.3.3 *$solvent* section

The solvent for PCM calculations is specified using the *$solvent* section, as documented below. In addition, the *$solvent* section can be used to incorporate non-electrostatic interaction terms into the solvation energy. (The **Theory** keyword in the *$pcm* section specifies only how the electrostatic interactions are handled.) The general form of the *$solvent* input section is shown below. The *$solvent* section was used above to specify parameters for the Kirkwood-Onsager SCRF model, and will be used again below to specify the solvent for SMx calculations (Section 11.2.7); in each case, the particular options that can be listed in the *$solvent* section depend upon the value of the *$rem* variable SOLVENT_METHOD.

```
$solvent
   NonEls  <Option>
   NSolventAtoms <Number unique of solvent atoms>
   SolventAtom <Number1> <Number2> <Number3> <SASrad>
   SolventAtom <Number1> <Number2> <Number3> <SASrad>
   . . .
   <Keyword>  <parameter/option>
   . . .
$end
```

The keyword `SolventAtom` requires multiple parameters, whereas all other keywords require only a single parameter. In addition to any (optional) non-electrostatic parameters, the *$solvent* section is also used to specify the solvent's dielectric constant. If non-electrostatic interactions are ignored, then this is the only keyword that is necessary in the *$solvent* section. For non-equilibrium TD-DFT/C-PCM calculations (Section 6.3.4), the optical dielectric constant should be specified in the *$solvent* section as well.

Dielectric

The (static) dielectric constant of the PCM solvent.

INPUT SECTION: **$solvent**

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

ε Use a dielectric constant of $\varepsilon > 0$.

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

The optical dielectric constant of the PCM solvent.

INPUT SECTION: **$solvent**

TYPE:

FLOAT

DEFAULT:

1.78

OPTIONS:

ε_∞ Use an optical dielectric constant of $\varepsilon_\infty > 0$.

RECOMMENDATION:

The default corresponds to water at 25°C. Note that $\varepsilon_\infty = n^2$, where n is the solvent's index of refraction.

The non-electrostatic interactions currently available in Q-CHEM are based on the work of Cossi *et al.* [52], and are computed outside of the SCF procedure used to determine the electrostatic interactions. The non-electrostatic 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 non-electrostatic interactions is highly empirical and probably needs to be considered on a case-by-case basis. Following Ref. 52, 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 solvent-accessible surface.

The following keywords (in the *$solvent* section) are used to define non-electrostatic parameters for PCM calculations.

NonEls

Specifies what type of non-electrostatic 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
SASrad (FLOAT):	Probe radius (in Å) for defining the solvent accessible surface

RECOMMENDATION:

If not using LJ or LJCav, 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 non-electrostatic interactions.

Example 11.7 Optimization of trifluoroethanol in water using both electrostatic and non-electrostatic PCM interactions. OPLSAA parameters are used in the Lennard-Jones potential for dispersion and repulsion.

```
$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

$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
```

11.2.4 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 11.2), but in any case they have dimension $N_{grid} \times N_{grid}$, 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 $\mathcal{O}(N_{grid}^3)$ in CPU time and $\mathcal{O}(N_{grid}^2)$ 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 $\mathcal{O}(N_{grid}^3)$ cost of inverting $\mathbf{K}$ is often prohibitively expensive.

To avoid this bottleneck, Lange and Herbert [15] have developed an iterative conjugate gradient (CG) solver for Eq. (11.2) whose cost scales as $\mathcal{O}(N_{grid}^2)$ in CPU time and $\mathcal{O}(N_{grid})$ 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 [53]. Together, these features lend themselves to a solution of Eq. (11.2) whose cost scales as $\mathcal{O}(N_{grid})$ in both memory and CPU time, for sufficiently large systems [15]. Currently, these options are available only for C-PCM, not for SS(V)PE/IEF-PCM.

Listed below are job control variables for the CG solver, which should be specified within the *$pcm* input section. Researchers who use this feature are asked to cite the original SWIG PCM references [13, 14] as well as the reference for the CG solver [15].

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 for large MM/PCM or QM/MM/PCM calculations.

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 $\mathcal{O}(N_{grid}^2)$ 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 $\mathcal{O}(N_{grid})$ 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 $\mathcal{O}(N_{grid})$, 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 $r/d <=$ **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 `QMMMPCM_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.

11.2.5 Iso-Density Implementation of SS(V)PE

11.2.5.1 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 [23], 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 11.2, and it is based on Chipman's "symmetrized" form of the $\mathbf{K}$ matrix [14, 23].) 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 11.2.2, 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.

Although cavitation, dispersion, and and specific hydrogen-bonding corrections for use with the iso-density SS(V)PE model have been reported in the literature [25, 26], these are not yet available in Q-CHEM. Energy gradients are also not available for this implementation of SS(V)PE, although they are available for the implementation described in Section 11.2.2 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 wavefunctions can also take advantage of molecular orbitals that are polarized using SS(V)PE. Researchers who use the iso-density SS(V)PE feature are asked to cite Ref. 8.

An iso-density 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 *$rem* variables, the iso-density SS(V)PE code is an interface between Q-CHEM and a stand-alone FORTRAN code written by Chipman [23], and some *$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 runs a gas-phase calculation and after convergence runs the SS(V)PE computation.

1 does 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 iso-density 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 solvation module 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 Q-CHEM solvation computation.

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.

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
```

11.2.5.2 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 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.

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

real number specifying the constant.

RECOMMENDATION:

The default value 78.39 is appropriate for water solvent.

ISHAPE

A flag to set the shape of the cavity surface.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 use the electronic iso-density surface.

1 use a spherical cavity surface.

RECOMMENDATION:

Use the default surface.

RHOISO

Value of the electronic iso-density contour used to specify the cavity surface. (Only relevant for ISHAPE = 0).

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).

TYPE:

FLOAT

DEFAULT:

Half the distance between the outermost atoms plus 1.4 Angstroms.

OPTIONS:

Real number specifying the radius in bohr (if positive) or in Angstroms (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.

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).

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).

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.

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).

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 iso-density 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.

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 Angstroms.

RECOMMENDATION:

The default value is recommended unless the single-center integrations procedure fails.

TRANX, TRANY, TRANZ

x, y, and z value of user-specified translation (only relevant if ITRNGR is set to 5 or 6

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.

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.

ROTTHE ROTPHI ROTCHI

Euler angles (θ, ϕ, χ) in degrees for user-specified rotation of the cavity surface. (relevant if IROTGR=3)

TYPE:

FLOAT

DEFAULT:

0,0,0

OPTIONS:

θ, ϕ, χ in degrees

RECOMMENDATION:

None.

IOPPRD

Specifies the choice of system operator form.

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 iso-density 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 iso-density surface is found to be too far from the outer iso-density surface.

Some molecules, for example C_{60}, can have a hole in the middle. Such molecules have two different "legal" iso-density 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

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.

11.2.6 COSMO

According to Table 11.2, 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

$$\mathbf{q} = -f_\varepsilon \mathbf{S}^{-1}\mathbf{v} \ , \tag{11.4}$$

and as discussed in Section 11.2.3 the user has the option to choose either the original value suggested by Klamt [4, 20], $f_\varepsilon = (\varepsilon - 1)/(\varepsilon + 1/2)$, or else $f_\varepsilon = (\varepsilon - 1)/\varepsilon$ as in, *e.g.*, Refs. [5, 7, 14]. More importantly, however, COSMO differs from C-PCM in that the former includes a correction for outlying charge that goes beyond Eq. (11.4), whereas C-PCM consists of nothing more than induced surface charges computed (self-consistently) according to Eq. (11.4)

Upon solution of Eq. (11.4), the outlying charge correction in COSMO [20, 21] 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 1.95 R, 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 11.2.2 and 11.2.3, 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), $G_{\text{pol}} = \frac{1}{2}\sum_i q_i'' v_i''$. (For comparison, when the C-PCM approach described in Section 11.2.2 is used, the electrostatic polarization energy is $G_{\text{pol}} = \frac{1}{2}\sum_i q_i v_i$, 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 [54].

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. COSMO calculations can also be used as a starting point for "COSMO-RS" calculations [30, 55], where RS stands for "real solvent". The COSMO-RS approach is *not* included in Q-CHEM and requires the COSMOtherm program, which is licensed separately through COSMOlogic [31]. Q-CHEM users who are interested in COSMOtherm can request special versions of Q-CHEM for the generation of σ-surface files that are needed by COSMOtherm.

11.2.7 SMx Models

The SMx models developed by Cramer, Truhlar, and co-workers at the University of Minnesota are a class of implicit solvation models that are designed to be "universal" in the sense that they can be applied to any solvent for which a small of descriptors is known [27]. (Note that the x in SMx is simply a version number; versions SM8 and SM12 are available in Q-CHEM.) In particular, 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 non-electrostatic interactions that are parameterized in terms of atomic surface tensions. The non-electrostatic interactions include cavitation, dispersion, and changes in solvent structure, and the treatment of these non-electrostatic effects is crucial to obtaining accurate (free) energies of solvation.

11.2.7.1 The SM8 Model

The SM8 model is described in detail in Ref. 18. 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, 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 [40], 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 [18].

In addition to GB electrostatics, there are several other contributions to the SM8 standard-state free energy of solvation. The first of these is called the electronic-nuclear-polarization (ENP) energy, or simply the electronic polarization (EP) energy if the solute geometry is assumed to be identical in the gas and solution phases. 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. The CDS contribution to the solvation energy is a sum of terms that are each proportional (with geometry-dependent proportionality constants called atomic surface tensions) to the solvent-accessible surface areas (SASAs) of the individual solute atoms. 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 11.2.2. The SASA is computed using the Analytic Surface Area (ASA) algorithm of Ref. [56] and Bondi's values for the van der Waals radii [40], or else a value of 2.0 Å if no Bondi radius is available. (Note that, as in the case of non-electrostatic interactions in PCMs, this means that a different molecular surface is used for the bulk electrostatics as compared to the non-electrostatic 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 semi-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 mole/liter 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 [57] 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). Basis sets supported for SM8 calculations in Q-CHEM are:

- 6-31G*
- 6-31+G*
- 6-31+G**

The charge mapping parameters are given in Ref. 57. Other basis sets should not be used in SM8 calculations.

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 gase. 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. 58.

An SM8 calculation is requested by setting SOLVENT_METHOD = SM8 in the *$rem* section, along with other job-control variables appropriate for a Hartree-Fock or DFT calculation. Additional variables for SMx calculations should be listed in a *$smx* input section; for SM8, the only additional variable that is required is the name of the solvent:

Solvent
    Sets the SMx solvent
INPUT SECTION: **$smx**
TYPE:
    STRING
DEFAULT:
    water
OPTIONS:
    Any name from the list of solvents given below.
RECOMMENDATION:
    NONE

1,1,1-trichloroethane
1,1,2-trichloroethane
1,1-dichloroethane
1,2,4-trimethylbenzene
1,4-dioxane
1-bromo-2-methylpropane
1-bromopentane
1-bromopropane
1-butanol
1-chloropentane
1-chloropropane
1-decanol
1-fluorooctane
1-heptanol
1-hexanol
1-hexene
1-hexyne
1-iodobutane
1-iodopentene
1-iodopropane
1-nitropropane
1-nonanol
1-octanol
1-pentanol
1-pentene
1-pentyne
1-propanol
2,2,20trifluoroethanol
2,2,4-trimethylpentane
2,4-dimethylpentane
2,4-dimethylpyridine
2,6-dimethylpyridine
2-bromopropane
2-chlorobutane
2-heptanone
2-hexanone
2-methylpentane
2-methylpyridine
2-nitropropane
2-octanone
2-pentanone
2-propanol
2-propen-1-ol
3-methylpyridine
3-pentanone
4-heptanone
4-methyl-2-pentanone
4-methylpyridine
bromoethane
bromooctane
butanal
butanoicacid
butanone
butanonitrile
butylethanoate
butylamine
butylbenzene
carbon disulfide
carbon tetrachloride
chlorobenzene
chlorotoluene
cis-1,2-dimethylcyclohexane
decalin
cyclohexane
cyclohexanone
cyclopentane
cyclopentanol
cyclopentanone
decane
dibromomethane
dibutyl ether
dichloromethane
diethyl ether
diethylsulfide
diethylamine
diiodomethane
dimethyldisulfide
dimethylacetamide
dimethylformamide
dimethylpyridine
dimethyl sulfoxide
dipropylamine
dodecane
E-1,2-dichloroethene
E-2-pentene
ethanethiol
ethanol
ethylethanoate
ethylmethanoate
ethylphenyl ether
ethylbenzene
ethylene glycol
fluorobenzene
formamide
formic acid
hexadecyliodide
m-ethylbenzoate
m-ethylethanoate
m-ethylmethanoate
m-ethylphenylketone
m-ethylpropanoate
m-ethylbutanoate
m-ethylcyclohexane
m-ethylformamide
m-xylene
heptane
hexadecane
hexane
nitrobenzene
nitroethane
nitromethane
methylaniline
nonane
octane
pentane
o-chlorotoluene
o-cresol
o-dichlorobenzene
o-nitrotoluene
o-xylene
pentadecane
pentanal
pentanoic acid
pentylethanoate
pentylamine
perfluorobenzene
phenyl ether
propanal
propanoic acid
propanonitrile
propylethanoate
propylamine
p-xylene
pyridine
pyrrolidine
sec-butanol
t-butanol
t-butylbenzene
tetrachloroethene
tetrahydrofuran
tetrahyrothiophenedioxide
tetralin
thiophene
thiophenol

5-nonanone
acetic acid
acetone
acetonitrile
aniline
anisole
benzaldehyde
benzene
benzonitrile
benzyl alcohol
bromobenzene
hexanoic acid
iodobenzene
iodoethane
iodomethane
isobutanol
isopropyl ether
isopropylbenzene
isopropyltoluene
m-cresol
mesitylene
methanol
toluene
$trans$-decalin
tribromomethane
tributylphosphate
trichloroethene
trichloromethane
triethylamine
undecane
water
Z-1,2-dichloroethene
other

The final choice, **Solvent** = “other”, requires an additional free-format file called “solvent_data” that should contain the float-point values of the following solvent descriptors:

Dielec dielectric constant, ϵ, of the solvent
SolN index of refraction at optical frequencies at 293 K, n_{20}^{D}
SolA Abraham’s hydrogen bond acidity, $\sum \alpha_2^H$
SolB Abraham’s hydrogen bond basicity, $\sum \beta_2^H$
SolG $\gamma = \gamma_m/\gamma^0$ (default is 0.0), where γ_m is the macroscopic surface tension at air/solvent interface at 298 K, and γ^0 is 1 $cal\ mol^{-1}\ ^{-2}$ (1 dyne/cm = 1.43932 cal mol^{-1} Å^{-2})
SolC aromaticity, ϕ : the fraction of non-hydrogenic solvent atoms that are aromatic carbon atoms
SolH electronegative “halogenicity”, ψ : the fraction of non-hydrogenic solvent atoms that are F, Cl or Br

For a desired solvent, these 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

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 $\mathrm{CH_3OH_2^+}$ 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. 59. 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. 60.

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 [61] is not yet available in Q-CHEM.

11.2.7.2 The SM12 Model

The SM12 model [19] 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 [62], which are based on Hirshfeld population analysis, or else charges derived from the electrostatic potential [63, 64], 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 [62]
MK	Merz-Singh-Kollman charges [63]
CHELPG	ChElPG charges [64]

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 11.8 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
jobtype sp
scf_guess core
method b3lyp
basis 6-31G*
sym_ignore true
solvent_method sm12
$end

$smx
solvent 1octanol
charges chelpg
$end
```

11.2.8 Langevin Dipoles Solvation 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 Jan Florián and Arieh Warshel [16, 17] at the University of Southern California. 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 original literature [16, 17].

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 11.4: 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 [16] 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: **$chemsol**
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: **$chemsol**
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: **$chemsol**
TYPE:
INTEGER
DEFAULT:
0
OPTIONS:
0 Limited printout
1 Full printout
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. 17. 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 Angstroms.

Example 11.9 A Langevin dipoles calculation on the methoxide anion. A customized value is specified for the radius of the C 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

$chemsol
ReadRadii
$end

$van_der_waals
2
1 2.5
$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 [65] (see Section 11.4).

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 are present are AMBER [66], CHARMM [67], and OPLSAA [68].

As implemented in Q-CHEM, the ONIOM model [69] 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

$$E_{total} = E_{total}^{MM} - E_{QM}^{MM} + E_{QM}^{QM} \tag{11.5}$$

where E_{total}^{MM} is the MM energy of the total system (*i.e.*, QM + MM subsystems), E_{QM}^{MM} is the MM energy of the QM subsystem, and E_{QM}^{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 [70] 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 [71]. In this approach, the total energy of the system is simply the sum of the subsystem energies,

$$E_{total} = E_{MM} + E_{QM} \tag{11.6}$$

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 [71] 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 $q_{nuclear} = 1 + q_{MM}$ (*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 wavefunctions.
- 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 wavefunction 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 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 11.2.2.
- Gaussian blurring of MM external charges is available for Janus QM/MM calculations.
- The user may add new MM atoms types and MM parameters.
- The user may define his/her own force field.

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 [72], 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, Q-CHEM will determine the MM bond connectivity 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, in which case Q-CHEM might classify them as bonded regardless of whether the appropriate MM bond parameters are available. To avoid such a scenario, the user can specify the bonds explicitly by setting the *$rem* variable USER_CONNECT = TRUE, in which case the *$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.

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 Å.

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 (in units of kcal/mol/degree) and equilibrium bond angle (in degrees) are the final entries in each `Bond` line.

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

$$E_{torsion}(\theta) = k_{torsion}[1 + \cos(m\theta - \phi)] \tag{11.7}$$

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 613 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. For example, 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

$$E_{MM} = E_{Coul} + E_{vdW} + E_{bond} + E_{angle} + E_{torsion} + E_{imptor} + E_{UreyBrad} \tag{11.8}$$

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

$$E_{Coul} = f_{scale}\frac{q_1 q_2}{r_{12}} \tag{11.9}$$

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:

$$E_{vdW}(r_{ab}) = f_{scale}\,\epsilon_{ab}\left[\left(\frac{\sigma_{ab}}{r_{ab}}\right)^{12} - 2\left(\frac{\sigma_{ab}}{r_{ab}}\right)^{6}\right] \tag{11.10}$$

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,

$$E_{bond}(r_{ab}) = k_{bond}(r_{ab} - r_{eq})^2 \tag{11.11}$$

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,

$$E_{angle}(\theta_{abc}) = k_{angle}(\theta_{abc} - \theta_{eq})^2 \tag{11.12}$$

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

$$E_{torsion}(\theta_{abcd}) = f_{scale}\sum_{m} k_{abcd}[1+\cos(m\theta_{abcd}-\phi)] \tag{11.13}$$

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:

$$E_{imptor}(\theta_{abcd}) = \frac{k_{abcd}}{N_{equiv}}[1+\cos(m\,\theta_{abcd}-\phi)] \tag{11.14}$$

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. (11.14), the following energy function is used:

$$E_{imptor}(\theta_{abcd}) = \frac{k_{abcd}}{N_{equiv}}\theta_{abcd}^2 \tag{11.15}$$

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

$$E_{UreyBrad}(r_{ac}) = k_{abc}(r_{ac}-r_{eq})^2 \tag{11.16}$$

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.* [73] and (independently) Riccardi *et al.* [74], as implemented in both the AMBER [75] and CHARMM [74, 76] programs. These approaches use Mulliken charges to represent the periodic images of the wavefunction, 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 [77]. The implementation in Q-CHEM thus allows for the use of CHELPG charges to represent the image wavefunctions, affording an algorithm that is stable in extended basis sets [77].

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 $\alpha_{\rm MM}$ and/or $\alpha_{\rm QM}$ can greatly increase the computational time, and the choices that are optimal for MM calculations need not be optimal for QM/MM calculations [77]. The cost of the MM Ewald summation scales as $\mathcal{O}(N_{\rm recip}N_{\rm atoms})$, where $N_{\rm recip}$ is the number of reciprocal-space lattice vectors that is used for the k-space sum. The QM portion of the calculation scales as $\mathcal{O}(N_{\rm recip}N_{\rm QM}N_{\rm atoms})$, where $N_{\rm QM}$ is the number of QM atoms (whereas $N_{\rm atoms} = N_{\rm QM} + N_{\rm MM}$ 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 $\alpha_{\rm MM} \approx 0.5$ Å^{-1}, can be found by solving the equation [77]

$$\alpha_{\rm MM} = 2C/L \tag{11.17}$$

where

$$C = \left[-\ln\left(10^{-\text{SCF\_CONVERGENCE}}\right)\right]^{1/2} \tag{11.18}$$

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 [77]

$$\frac{2CL^3\alpha_{\mathrm{QM}}^3}{\pi^{3/2}} + \frac{\alpha_{\mathrm{QM}}^2 L^2}{\pi^{1/2}} - \alpha_{\mathrm{QM}} L - 2C = 0 \; . \tag{11.19}$$

To perform an MM- or QM/MM-Ewald job, one must set MM_SUBTRACTIVE = 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. [77]). 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
```

11.3.3 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.

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. 78.

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.

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.

MM_SUBTRACTIVE

Specifies whether a subtractive scheme is used in the E_{Coul}, Eq. (11.9), 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.

11.3.4 QM/MM Examples

- QM/MM Example 1

  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 11.10 ONIOM optimization of water dimer.

```
$rem
method                  pbe0
basis                   6-31G*
qm_mm_interface         oniom
force_field             oplsaa
user_connect            true
jobtype                 opt
molden_format           true
$end
```

```
$qm_atoms
4 5 6
$end

$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
```

- QM/MM Example 2

  Features of this job:

  – Janus electronic embedding with a YingYang 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 11.11 Excited-state single-point QM/MM calculation on deoxyadenosine.

```
$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

$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
```

- QM/MM Example 3

  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 11.12 MM molecular dynamics with user-defined MM parameters.

```
$rem
basis           sto-3g
method          hf
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

$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

$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, `QMMMPCM_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 11.2.2.

11.4 Q-CHEM/CHARMM Interface

Q-CHEM can be used a QM back-end for QM/MM calculations using CHARMM package [65]. 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 utilize 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 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 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 default unless running calculations with CHARMM where charges on the QM region need to be saved.

IGDEFIELD

Triggers the calculation of the electrostatic potential and/or the electric field at the positions of the MM charges.

TYPE:

INTEGER

DEFAULT:

UNDEFINED

OPTIONS:

O	Computes ESP.
1	Computes ESP and EFIELD.
2	Computes EFIELD.

RECOMMENDATION:

Must use this *$rem* when IGDESP is specified.

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

Example 11.13 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 [78]. 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 [79]. 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

The Effective Fragment Potential (EFP) method is a computationally inexpensive way of modeling intermolecular interactions in non-covalently bound systems. The EFP approach can be viewed as a QM/MM scheme with no empirical parameters. Originally EFP was developed by Gordon's group [80, 81], and was implemented in GAMESS [82]. A review of the EFP theory and applications can be found in Ref. 83, 84. A related approach, also based on distributed multipoles, is called XPOL; it is described in Section 12.7.

A new implementation of the EFP method based on the libefp library (*www.libefp.org*) has been added to Q-CHEM.[85] The new EFP module is called EFPMAN2. EFPMAN2 can run calculations in parallel on a shared memory multi-core computer and is much faster than the previous EFP implementation in Q-CHEM. EFPMAN2 is interfaced with CCMAN and CCMAN2 modules to allow coupled cluster and EOM-CC calculations with EFP. CIS and TDDFT calculations with EFP are also available. EFP calculations using EFPMAN2 module are enabled by EFP_USE_EFPMAN2 keyword. The older version of EFP (EFPMAN module) will be discarded in Spring 2014.

11.5.1 Theoretical Background

The total energy of the system consists of the interaction energy of the effective fragments (E^{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):

$$E^{ef\text{-}ef} = E_{elec} + E_{pol} + E_{disp} + E_{ex\text{-}rep}. \tag{11.20}$$

The QM-EF interactions are computed as follows. The electrostatics and polarization parts of the EFP potential contribute to the quantum Hamiltonian via one-electron terms,

$$H'_{pq} = H_{pq} + \langle p|\hat{V}^{elec} + \hat{V}^{pol}|q\rangle \tag{11.21}$$

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 inter-molecular interaction energy [86]. 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 [80, 81, 87, 88]. 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.

Electrostatic interaction between an effective fragment and the QM part is described by perturbation $\hat{V}^{elec}$ of the *ab initio* Hamiltonian (see Eq. (11.21)). 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).

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 [89]. To account for the charge-penetration effect, the simple exponential damping of the charge-charge term is used [89, 90]. 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:

$$\begin{aligned} E_{kl}^{ch\text{-}ch} &= \left[1-(1+\alpha_k R_{kl}/2)e^{-\alpha_k R_{kl}}\right] q^k q^l/R_{kl} \quad , \ \text{if } \alpha_k=\alpha_l \\ \text{or} \quad &= \left(1-\frac{\alpha_l^2}{\alpha_l^2-\alpha_k^2}e^{-\alpha_k R_{kl}}-\frac{\alpha_k^2}{\alpha_k^2-\alpha_l^2}e^{-\alpha_l R_{kl}}\right) q^k q^l/R_{kl} \quad , \ \text{if } \alpha_k\neq\alpha_l \end{aligned} \tag{11.22}$$

Damping parameters are included in the potential of each fragment, but QM-EFP electrostatic interactions are currently calculated without damping corrections.

Alternatively, one can obtain the short-range charge-penetration energy using the spherical Gaussian overlap (SGO) approximation [91]:

$$E_{kl}^{pen} = -2\left(\frac{1}{-2ln|S_{kl}|}\right)^{\frac{1}{2}}\frac{S_{kl}^2}{R_{kl}} \tag{11.23}$$

where S_{kl} is the overlap integral between localized MOs k and l (calculated for the exchange-repulsion term, Eq. (11.32)). 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.

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 computed as [80]

$$E^{pol} = -\frac{1}{2}\sum_k \mu^k(F^{\text{mult},k}+F^{\text{ai,nuc},k})+\frac{1}{2}\sum_k \bar{\mu}^k F^{\text{ai,elec},k} \tag{11.24}$$

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

$$\mu^k = \alpha^k F^{\text{total},k} \tag{11.25}$$

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:

$$F^{\text{ai,total},k} = F^{mult,k}+F^{ind,k}+F^{\text{ai,elec},k}+F^{\text{ai,nuc},k} \tag{11.26}$$

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:

$$\hat{V}^{pol} = -\frac{1}{2}\sum_k\sum_a^{x,y,z}\frac{(\mu_a^k+\bar{\mu}_a^k)a}{R^3} \tag{11.27}$$

where R and a are the distance and its Cartesian components between an electron and the polarizability point k. In sum, the total polarization energy is computed self-consistently using a two level iterative procedure. The objectives of the higher and lower levels are to converge the wavefunction and induced dipoles for a given fixed wavefunction, respectively. In the absence of the *ab initio* region, the induced dipoles of the EF system are iterated until self-consistent 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[91]:

$$\begin{aligned} F &= F_0 f^{damp} \\ f^{damp} &= 1.0 - exp(-\sqrt{\alpha\beta}r^2)(1 + \sqrt{\alpha\beta}r^2) \end{aligned} \tag{11.28}$$

Dispersion provides a leading contribution to van der Waals and π-stacking interactions. The dispersion interaction is expressed as the inverse R dependence:

$$E^{disp} = \sum_n C_6 R^{-6} \tag{11.29}$$

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 [92].

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 [93] with parameter $b = 1.5$:

$$C_6^{kl} \rightarrow \left(1 - e^{-bR} \sum_{k=0}^{6} \frac{(bR)^k}{k!}\right) C_6^{kl} \tag{11.30}$$

or using interfragment overlap (so called overlap-based damping)[91]:

$$C_6^{kl} \rightarrow \left(1 - S_{kl}^2 \left(1 - 2\log|S_{kl}| + 2\log^2|S_{kl}|\right)\right) C_6^{kl} \tag{11.31}$$

QM-EFP dispersion interactions are currently disabled.

Exchange-repulsion originates from the Pauli exclusion principle, which states that the wavefunction 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 wavefunction-based formalism to account for this inherently quantum effect. Exchange-repulsion is the only non-classical component of EFP and the only one that is repulsive.

The exchange-repulsion interaction is derived as an expansion in the intermolecular overlap, truncated at the quadratic term [94, 95], 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:

$$
\begin{aligned}
E_{ij}^{exch} \quad = \quad & -4\sqrt{\frac{-2\ln|S_{ij}|}{\pi}}\frac{S_{ij}^2}{R_{ij}} \qquad (11.32)\\
& -2S_{ij}\left(\sum_{k\in A}F_{ik}^{A}S_{kj}+\sum_{l\in B}F_{jl}^{B}S_{il}-2T_{ij}\right)\\
& +2S_{ij}^2\left(\sum_{J\in B}-Z_J R_{iJ}^{-1}+2\sum_{l\in B}R_{il}^{-1}+\sum_{I\in A}-Z_I R_{Ij}^{-1}+2\sum_{k\in A}R_{kj}^{-1}-R_{ij}^{-1}\right)
\end{aligned}
$$

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. (11.32) is derived within an infinite basis set approximation, it requires a reasonably large basis set to be accurate [6-31+G* is considered to be the smallest acceptable basis set, 6-311++G(3df,2p) is recommended]. These factors make exchange-repulsion the most computationally expensive part of the EFP energy calculations of moderately sized systems.

Large systems require additional considerations. Since total exchange-repulsion energy is given by a sum of terms in Eq. (11.32) 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).

The QM-EFP exchange-repulsion energy is currently disabled.

11.5.2 Excited-State Calculations with EFP

Interface of EFP with EOM-CCSD (both via CCMAN and CCMAN2), CIS, CIS(D), and TDDFT has been developed [96, 97]. 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. (11.21). In the coupled-cluster calculation, the induced dipoles of the fragments are frozen at their HF values.

The transformed Hamiltonian $\bar{H}$ effectively includes Coulomb and polarization contributions from the EFP part. As $\bar{H}$ is diagonalized in an EOM calculation, the induced dipoles of the effective fragments are frozen at their reference state value, *i.e.*, the EOM equations are solved with a constant response of the EFP environment. To account for solvent response to electron rearrangement in the EOM target states (*i.e.*, excitation or ionization), a perturbative non-iterative correction is computed for each EOM root as follows. The one-electron density of the target EOM state (excited or ionized) is calculated and used to re-polarize the environment, *i.e.*, to recalculate the induced dipoles of the EFP part in the field of an EOM state. These dipoles are used to compute the polarization energy corresponding to this state.

The total energy of the excited state with inclusion of the perturbative response of the EFP polarization is:

$$
E_{\text{IP}}^{\text{EOM/EFP}} = E_{\text{EOM}} + \Delta E_{\text{pol}} \qquad (11.33)
$$

where E_{EOM} is the energy found from EOM-CCSD procedure and ΔE_{pol} has the following form:

$$\begin{aligned}\Delta E_{\mathrm{pol}} \quad = \quad & \frac{1}{2}\sum_{k}\sum_{a}^{x,y,z}\Big[-(\mu^{k}_{\mathrm{ex},a}-\mu^{k}_{\mathrm{gr},a})(F^{\mathrm{mult},k}_{a}+F^{\mathrm{nuc},k}_{a}) \\ & +(\tilde{\mu}^{k}_{\mathrm{ex},a}F^{\mathrm{ai},k}_{\mathrm{ex},a}-\tilde{\mu}^{k}_{\mathrm{gr},a}F^{\mathrm{ai},k}_{\mathrm{gr},a})-(\mu^{k}_{\mathrm{ex},a}-\mu^{k}_{\mathrm{gr},a}+\tilde{\mu}^{k}_{\mathrm{ex},a}-\tilde{\mu}^{k}_{\mathrm{gr},a})F^{\mathrm{ai},k}_{\mathrm{ex},a}\Big]\end{aligned} \tag{11.34}$$

where $F^{\mathrm{ai}}_{\mathrm{gr}}$ and $F^{\mathrm{ai}}_{\mathrm{ex}}$ are the fields due to the reference (HF) state and excited-state electronic densities, respectively. μ^{k}_{gr} and $\tilde{\mu}^{k}_{\mathrm{gr}}$ are the induced dipole and conjugated induced dipole at the distributed polarizability point k consistent with the reference-state density, while μ^{k}_{ex} and $\tilde{\mu}^{k}_{\mathrm{ex}}$ are the induced dipoles corresponding to the excited state density.

The first two terms in Eq. (11.34) 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 wavefunction of the excited state.

The EOM 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. (11.34) 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 EOM/EFP scheme is that it does not compromise multi-state nature of EOM and that the electronic wavefunctions 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.

EOM-CC/EFP scheme works with any type of the EOM excitation operator R_k currently supported in Q-CHEM, *i.e.*, spin-flipping (SF), excitation energies (EE), ionization potential (IP), electron affinity (EA) (see Section 6.7.9 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.

Implementation of CIS/EFP, CIS(D)/EFP, and TDDFT/EFP methods is similar to the implementation of EOM/EFP. Polarization correction as in Eq. 11.34 is calculated and added to the CIS or TDDFT excitation energies.

11.5.3 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 11.5.2). 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[63] (HLA) or by mono-valent groups of atoms from the neighboring fragment(s) called hereafter Cap Link Atom (CLA). 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:

$$P = A_1H + \sum_{i=2}^{N-1} HA_iH + HA_N \tag{11.35}$$

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.

$$P = A_1C^2 + \sum_{i=2}^{N-1} C^{i-1}A_iC^{i+1} + C^{N-1}A_N \tag{11.36}$$

Q-CHEM provides a two-step script, `prefefp.pl`, located in *$QC/bin* which takes a PDB file and breaks it into capped fragments in the GAMESS format, such that the EFP parameters for these capped fragments can be generated, as explained in Section 11.5.8. As the EFP parameters are generated for each capped fragment, the neighboring fragments have duplicated parameter points (overlapping areas) in both the HLA and CLA schemes due to the overlapping caps. Since multipole expansion points and polarizability expansion points are computed on each capped residue by the standard procedure, the multipole (and damping terms) and polarizabilities need to be removed ($C^{\emptyset}$) from the overlapping areas.

Equations (11.35) and (11.36) become:

$$P = A_1C^{\emptyset} + \sum_{i=2}^{N-1} C^{\emptyset}A_iC^{\emptyset} + C^{\emptyset}A_N \tag{11.37}$$

The details concerning this removing procedure are presented in Section 11.5.8.

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:

$$E^{\text{tot}}(P-L) = E^{\text{elec}}(P-L) + E^{\text{pol}}(P-L) \tag{11.38}$$

The electrostatics is an additive term; its contribution to fragment-fragment and ligand-fragment interaction is computed as follows:

$$E^{\text{elec}}(P-L) = \sum_{i}^{N} E^{\text{elec}}\left(C^{\emptyset}A_{i=1}C^{\emptyset} - L\right) \tag{11.39}$$

The polarization contribution in an EFP system (no QM) is:

$$E^{\text{pol}}(P-L) = -\frac{1}{2}\sum_{k\in P,L} \mu^k F^{mult,k} + \frac{1}{2}\sum_{k\in P} \mu^k F^{mult,k} \tag{11.40}$$

The first term is the polarization energy obtained upon convergence of the induced dipoles of the ligand ($\mu^k_{efp}(L)$) and all fragments ($\mu^k_{fefp}(A_i)$). The system is thus fully polarized, all fragments (A$_i$ or L) are polarizing each other until self-consistency.

$$\begin{aligned}\mu^{k}_{efp}(L) &= \sum_{k\in A_i} \alpha^k(F^{mult,k} + F^{ind,k}) \\ \mu^{k}_{fefp}(A_i) &= \sum_{j\neq i}\sum_{k\in L,A_j} \alpha^k(F^{mult,k} + F^{ind,k})\end{aligned} \tag{11.41}$$

The second term of Eq. (11.40) is the polarization of the protein by itself; this value has to be subtracted once the induced dipoles (Eq. 11.41) 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 Molecular Fragmentation with Conjugated Caps approach (MFCC) which avoids the issue of overlapping of saturated fragments and was developed in 2003 by Zhang[98, 99]. 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 ($C^{i+1}C^{i-1}$), the so-called “concaps”, with the ligand. $N-1$ concap fragments are actually used to subtract the overlapping effect.

$$P = A_1C^2 + \sum_{i=2}^{N-1} C^{i-1}A_iC^{i+1} + C^{N-1}A_N - \sum_{i}^{N-1} C^{i+1}C^{i-1} \tag{11.42}$$

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 (N capped fragments + $N-1$ 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:

$$E^{\rm elec}(P-L) = \sum_{i}^{N} E^{\rm elec}\left(C^{i-1}A_iC^{i+1} - L\right) - \sum_{i}^{N-1} E^{\rm elec}\left(C^{i-1}C^{i+1} - L\right) \tag{11.43}$$

The main advantage of MFCC is that the multipole expansion obtained on each capped residue or concap are kept during the $E^{\rm elec}(P-L)$ calculation. In the present implementation, there are no polarization contributions. The MFCC scheme is not yet available for QM/fEFP.

11.5.4 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, EOM-CCSD methods]. Theoretical details on the interface of EFP with EOM-CCSD and CIS(D) can be found in Refs. 96, 97.

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 are available for fragments, Euler angle format and XYZ format. In 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. The XYZ format is identical to the coordinate format used in GAMESS; i.e., the name of the fragment is provided on the first line followed by three lines specifying names and x,y,z coordinates of first three atoms of the fragment.

When using EFPMAN2 you can also specify positions and orientations of the fragments using (x, y, z) coordinates of the first three atoms of the fragment. To enable this feature use EFP_COORD_XYZ keyword. The format of input goes as follows: for each fragment the first line should contain the name of the fragment and the following three lines should specify coordinates of the three first atoms belonging to a fragment. The sample input can be found in the examples section below.

11.5.5 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 multipolar 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 11.5. EFP potentials in GAMESS format are supported by new EFPMAN2 module. They are stored in *$QCAUX/fraglib* directory. EFP potentials compatible with old EFPMAN module have a different format; they are stored in *$QCAUX/efp* directory.

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 by 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.[100]. 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, wavefunction, Fock matrix, *etc.*, were obtained using 6-311++G(3df,2p) basis set.

Table 11.5: 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
available with EFPMAN2 only	
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

11.5.6 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 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.

11.5.6.1 Generating EFP Parameters in GAMESS

EFP parameters can be generated in GAMESS using MAKEFP job (RUNTYP=MAKEFP). For EFP parameters calculations, 6-311++G(3df,2p) basis set is recommended. Original Stone's distributed multipole analysis (bigexp=0 in the group $stone is recommended for non-aromatic compound; 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 GAMESS manual for further details. Below are examples of the RUNTYP=MAKEFP GAMESS input file for water and benzene.

```
GAMESS input example for water.

 $control 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
 $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
 $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
```

11.5.6.2 Converting EFP Parameters to the Q-CHEM Library Format

This section describes converting EFP parameters to the Q-CHEM Library Format. This format is used with the old EFPMAN module and is not supported by the new EFPMAN2 module.

To convert GAMESS-generated `*.efp` files to the Q-CHEM library format, one can use the script `efp_g2qlib.pl` located in the *$QC/bin* directory. The script takes three command line arguments:

1. the name of the `.efp` file generated by GAMESS;
2. the basis set name used in RUNTYP = MAKEFP;
3. fragment name for the Q-CHEM library.

In this example, the script `$QC/bin/efp_g2qlib.pl` is used to convert `mywater_gms.efp` file generated in RUNTYP = MAKEFP run with 6-311++G(3df,2p) basis set:

```
efp_g2qlib.pl mywater_gms.efp '6-311++G(3df,2p)' MYWATER_L
```

Note: Do not forget to place the second command-line argument—namely, the basis set name used for the calculation of the wavefunction and Fock matrix—in single quotes: "6-311++G(3df,2p)"

Note: Remember to add "_L" at the end of the file to distinguish the library fragments from the user-defined ones.

Note: Keep the `.efp` extension for files in the Q-CHEM EFP library. If you change the extension of the file with EFP parameters, Q-CHEM will not be able to recognize the file.

Move the produced `MYWATER_L.efp` file to the *$QCAUX/efp* directory. Now the EFP parameters for the fragment with name MYWATER_L are accessible via Q-CHEM EFP library and can be requested in the EFP input:

```
$efp_fragments
...
MYWATER_L -0.61278300 -3.71417606 3.79298003 2.366406 1.095665 3.136731
...
$end
```

11.5.6.3 Converting EFP Parameters to the Q-CHEM Input Format

This section is relevant for the old EFPMAN module only.

GAMESS EFP input file can be converted into Q-CHEM input format using the script `efp_g2qinp.pl` located in the *$QC/bin* directory. The script takes the name of the GAMESS input file as a command line argument and produces the Q-CHEM input file for a EFP calculation with user-defined fragments.

The script `$QC/bin/efp_g2qinp.pl` is used to convert `mywater_gms.inp` GAMESS input file into Q-CHEM input file mywater.in:

```
efp_g2qinp.pl mywater_gms.in > mywater.in

Input (GAMESS EFP input file):

$efrag

 FRAGNAME=WATER
 A01O1          -3.4915850115   1.6130035943   6.3466249474
 A02H2          -2.6802987071   1.8846442441   5.9133847256
 A03H3          -4.1772171823   2.0219163332   5.8152398722
 FRAGNAME=WATER
 A01O1          -2.6611610192   2.9115390164   4.0238365827
 A02H2          -1.9087196288   3.4457621856   4.2848131325
 A03H3          -2.5308218271   2.7856450856   3.0821181464
$END

$WATER
 ...
$END

Output (Q-Chem EFP input):

$efp_fragments
WATER -3.48455373 1.65108563 6.29264698 -0.182579 0.622332 -1.458796
WATER -2.61176287 2.93438808 3.98574400 -1.137556 0.960145 0.226277
$end

$efp_params
fragment WATER
...
$end
```

After conversion, the *$efp_params* section of the Q-CHEM input file contains the line starting with the er_basis keyword. This line has to be edited manually. The basis set used to produce EFP parameters (MAKEEFP job) has to be specified. For example, if the EFP parameters were produced using 6-311++G(3df,2p) basis set, the line in the above example should be edited as follows:

```
$efp_params
fragment WATER
...
$end
```

The script reads Cartesian coordinates of fragments and EFP parameters only and creates *$efp_fragments* and *$efp_params* sections of the Q-CHEM input file. All other required Q-CHEM input sections (*$rem*, *$molecule*, *etc.*.) need to be added manually.

11.5.7 Converting PDB Coordinates into Q-CHEM EFP Input Format

This section is relevant for the old EFPMAN module only.

The commonly used format for 3D structural data of biological molecules is Protein Data Bank (PDB) file. The Cartesian coordinates of molecules from PDB file can be converted to Q-CHEM format for EFP fragments. The script `efp_pdb2qinp.pl` located in *$QC/bin* is designed to convert coordinates of solvent molecules from PDB file to the Q-CHEM format. However, only EFP fragments available in the Q-CHEM EFP library located in $QCAUX/efp directory will be recognized and converted by the script. Since PDB format has many variants some adjustments are required to original PBD file to make it suitable for conversion. Consider example of PDB file which contains four water molecules:

```
Example: Water.pdb

REMARK TIPS3P model
REMARK
ATOM     1  OH2 TIP   1    2.950  -0.088   2.278  1.00  0.00    WAT
ATOM     2  H1  TIP   1    3.539  -0.465   1.623  1.00  0.00    WAT
ATOM     3  H2  TIP   1    2.635  -0.852   2.764  1.00  0.00    WAT
ATOM     4  OH2 TIP   2   -4.195  -1.030  -2.858  1.00  0.00    WAT
ATOM     5  H1  TIP   2   -4.507  -1.544  -3.606  1.00  0.00    WAT
ATOM     6  H2  TIP   2   -3.809  -1.689  -2.279  1.00  0.00    WAT
ATOM     7  OH2 TIP   3    0.489   3.999   2.164  1.00  0.00    WAT
ATOM     8  H1  TIP   3    0.125   4.885   2.215  1.00  0.00    WAT
ATOM     9  H2  TIP   3   -0.239   3.441   2.441  1.00  0.00    WAT
ATOM    10  OH2 TIP   4   -4.118   1.169   2.350  1.00  0.00    WAT
ATOM    11  H1  TIP   4   -4.186   0.349   1.858  1.00  0.00    WAT
ATOM    12  H2  TIP   4   -4.667   1.776   1.851  1.00  0.00    WAT
```

The PDB file contains 4 water molecules with tag "TIP". Each molecule has 3 atoms with tags "OH1", "H1", and "H2". However, the Q-CHEM EFP library (*$QCAUX/efp*) contains EFP fragment with name WATER_L that has 3 atoms with tags "A01O1", "A02H2", "A03H3" (see section "labels" at the bottom of the library file WATER_L.efp). In order to match the water molecule in PBD file and EFP library, the following lines should be added to the PDB file:

```
Example: Modified Water.pdb

REMARK TIPS3P model
REMARK
REMARK EFPF TIP WATER_L
REMARK EFPA TIP OH2 A01O1
REMARK EFPA TIP H1 A02H2
REMARK EFPA TIP H2 A02H3
ATOM   1  OH2 TIP   1    2.950  -0.088   2.278  1.00  0.00    WAT
ATOM   2  H1  TIP   1    3.539  -0.465   1.623  1.00  0.00    WAT
ATOM   3  H2  TIP   1    2.635  -0.852   2.764  1.00  0.00    WAT
ATOM   4  OH2 TIP   2   -4.195  -1.030  -2.858  1.00  0.00    WAT
ATOM   5  H1  TIP   2   -4.507  -1.544  -3.606  1.00  0.00    WAT
ATOM    6  H2  TIP   2   -3.809  -1.689  -2.279  1.00  0.00    WAT
ATOM    7  OH2 TIP   3    0.489   3.999   2.164  1.00  0.00    WAT
ATOM    8  H1  TIP   3    0.125   4.885   2.215  1.00  0.00    WAT
ATOM    9  H2  TIP   3   -0.239   3.441   2.441  1.00  0.00    WAT
ATOM   10  OH2 TIP   4   -4.118   1.169   2.350  1.00  0.00    WAT
ATOM   11  H1  TIP   4   -4.186   0.349   1.858  1.00  0.00    WAT
ATOM   12  H2  TIP   4   -4.667   1.776   1.851  1.00  0.00    WAT
```

New lines start with keyword “REMARK” in order to prevent reading of these lines by other programs. The first line “REMARK EFPF TIP WATER_L” contains keyword “EFPF” which is recognized by the script and tells to match a PDB molecule with tag “TIP” and a Q-CHEM EFP library fragment with tag “WATER_L”. The general format for this line is as follows:

```
REMARK EFPF <PDB_TAG> <EFP_TAG>
```

where `<PDB_TAG>` is a tag of the molecule in PDB file and `<EFP_TAG>` is a name of the corresponding Q-CHEM EFP library fragment.

The next three lines contain keyword “EFPA” which maps each atom type from PDB file to corresponding atom in the Q-CHEM EFP fragment. For example, the line “REMARK EFPA TIP OH2 A01O1” maps the oxygen atom “OH1” from molecule with tag “TIP” in PDB file to atom “A01O1” of the corresponding EFP fragment. The general format for these lines is as follows:

```
REMARK EFPA <PDB_TAG> <PDB_ATOM> <EFP_ATOM>
```

where `<PDB_TAG>` is a tag for the molecule, `<PDB_ATOM>` is a tag for the atom in the PDB file, and `<EFP_ATOM>` is the name of the corresponding atom in the EFP fragment from the Q-CHEM EFP library.

Note: The names of all EFP fragments can be found in the directory *$QCAUX/efp*. The names of atoms in EFP fragments are listed in the last section. “labels” at the bottom of `*.efp` files.

```
Example Section "labels" from "$QCAUX/efp/WATER_L.efp":

fragment WATER_L
...
labels
A01O1 O 0.0000000000  0.0000000000  0.0664326840
A02H2 H 0.0000000000  0.7532000000 -0.5271673160
A03H3 H 0.0000000000 -0.7532000000 -0.5271673160
$end
```

Finally, when each atom of the solvent molecule is matched to the corresponding atom of EFP fragment the script converts the coordinates and produces the *$efp_fragments* section of the Q-CHEM EFP input file.

```
Example: Converted Water.pdb file
```

```
$efp_fragments
WATER_L -3.48394754 1.65103763 6.29199349 -0.183326 0.621595 -1.458096
WATER_L -2.61157557 2.93379532 3.98490930 -1.138655 0.960180  0.226963
WATER_L  0.42784805 4.01733780 2.18237025  1.279456 1.850972 -1.727770
WATER_L -4.15254211 1.15707665 2.29452048  1.903173 0.582438  1.562171
$end
```

Note: Since the solvent molecules converted from the PBD file are present in the Q-CHEM EFP library, the section *$efp_fragments* is not required in the input.

11.5.8 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 peptidic 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 (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 (cfr. 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'.

Note: 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.

11.5.9 Input keywords

EFP_USE_EFPMAN2
Enable calculation using new EFPMAN2 module
TYPE:
LOGICAL
DEFAULT:
TRUE
OPTIONS:
TRUE FALSE
RECOMMENDATION:
None

EFP_COORD_XYZ

Use coordinates of three atoms instead of Euler angles to specify position and orientation of the fragments (EFPMAN2 only)

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

None

EFP_DIRECT_POLARIZATION_DRIVER

Use direct solver for EFP polarization (EFPMAN2 only)

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 (EFPMAN2 only)

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 off 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 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_QM_ELEC_DAMP

Controls QM-EFP electrostatics screening in EFP

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 switch off electrostatic screening

1 use overlap based damping correction

RECOMMENDATION:

None

11.5.10 Examples

11.5.10.1 Samples using new EFPMAN2 module

Example 11.14 Basic EFP-only calculation of benzene dimer in XYZ 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)
jobtype sp
purecart 2222
efp_fragments_only true
efp_use_efpman2 1
efp_coord_xyz 1
$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 11.15 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)
jobtype sp
purecart 2222
efp_fragments_only true
efp_use_efpman2 1
efp_disp_damp 1
$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 11.16 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   0.0         0.0         0.2243
H  -1.4233      0.0        -0.8973
H   1.4233      0.0        -0.8973
$end

$rem
method hf
basis 6-31G(d)
jobtype sp
purecart 2222
efp_use_efpman2 1
efp_disp_damp 1
$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 11.17 EOM-IP-CCSD/EFP calculation; CN radical hydrated by 6 waters.

```
$molecule
-1 1
 C           1.0041224092   2.5040921109  -0.3254633433
 N           0.8162211575   2.3197739512   0.7806258675
$end

$rem
basis 6-31+G*
method eom-ccsd
efp_fragments_only false
purecart   2222
scf_convergence 8
ip_states = 4
EFP = 1
eom_fake_ipea   true
ccman2  false
efp_exrep 0
efp_use_efpman2 1
$end

$efp_fragments
 H2O_L      1.12736608   -1.43556954   -0.73517708   -1.45590530    2.99520330    0.11722
 H2O_L      1.25577919    0.62068648   -2.69876653    2.56168924    1.26470722    0.33910
 H2O_L      3.76006184   -1.03358049    0.45980636   -1.53852111    2.58787281   -1.98107
 H2O_L      4.81593067    2.87535152   -0.24524178   -1.86802100    0.73283467   -2.17837
 H2O_L      4.07402278    0.74020006   -1.92695949    2.21177738    1.69303397   -2.30505
 H2O_L      3.60104027    1.35547341    1.88776964    0.43895304    1.25442317    1.07742
$end
```

11.5.10.2 Samples using old EFPMAN module

Example 11.18 Old format: EFP energy computation of benzene dimer (both fragments are treated by EFP) with old EFPMAN module. EFP parameters are read from the EFP library (*$QCAUX/efp*). Note that a dummy atom (He) is placed in a *$molecule* section for a pure MM/EFP calculation.

```
$molecule
0 1
 He       5.0    5.0    5.0
$end

$rem
method hf
basis 6-31G(d)
jobtype sp
purecart 2222
efp_fragments_only true
efp_use_efpman2 0
EFP 1
$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 11.19 New format: EFP energy computation of benzene dimer (both fragments are treated by EFP) with old EFPMAN module. All EFP parameters are read from the EFP library (*$QCAUX/efp*).

```
$molecule
0 1
--
0 1
-0.30448173 -2.24210052 -0.29383131 -0.642499 1.534222 -0.568147
--
0 1
-0.60075437  1.36443336  0.78647823  3.137879 1.557344 -2.568550
$end

$rem
method hf
jobtype sp
purecart 2222
efp_fragments_only true
efp_input true
efp_use_efpman2 0
$end

$efp_fragments
BENZENE_L
BENZENE_L
$end
```

Example 11.20 New format: EFP energy and gradient computation of benzene dimer (both fragments are treated by EFP). All EFP parameters are read from the EFP library (*$QCAUX/efp*).

```
$molecule
0 1
--
0 1
-0.30448173 -2.24210052 -0.29383131 -0.642499 1.534222 -0.568147
--
0 1
-0.60075437  1.36443336  0.78647823  3.137879 1.557344 -2.568550
$end

$rem
method hf
jobtype force
purecart 2222
efp_fragments_only true
efp_input true
efp_use_efpman2 0
$end

$efp_fragments
BENZENE_L
BENZENE_L
$end
```

Example 11.21 New format: EFP optimization of water dimer (both fragments are treated by EFP). All the EFP parameters are read from the EFP library (*$QCAUX/efp*). Note that for EFP optimization only the new format (without dummy atom) is supported.

```
$molecule
0 1
--
0 1
-2.57701480 1.41334560 3.40415522 0.000000 1.000000  0.000000
--
0 1
-6.07701480 1.41334560 3.40415522 0.000000 0.000000  0.000000
$end

$rem
method hf
jobtype opt
purecart 2222
efp_fragments_only true
efp_input true
efp_use_efpman2 0
$end

$efp_fragments
WATER_L
WATER_L
$end
```

Example 11.22 QM/MM computation of one water molecule in the QM part and one water + two ammonia molecules in the MM part. The EFP parameters will be taken from the EFP library (*$QCAUX/efp*). Note that for QM/MM calculation this is the only format that is supported.

```
$molecule
0 1
O1   0.47586     0.56326     0.53843
H2   0.77272     1.00240     1.33762
H3   0.04955    -0.23147     0.86452
$end

$rem
method hf
basis 6-31G(d)
jobtype sp
purecart 2222
EFP 1
efp_use_efpman2 0
$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 11.23 Excited states of formaldehyde with 6 EFP water molecules by CIS(D).

```
$molecule
0 1
```

```
 C1              1.0632450881806   2.0267971791743   0.4338879750526
 O2              1.1154451117032   1.0798728186948   1.1542424552747
 H3              1.0944666250874   3.0394904220684   0.8360468907200
 H4              0.9836601903170   1.9241779934791  -0.6452234478151
$end

$rem
basis 6-31+G*
efp_fragments_only false
purecart   2222
scf_convergence 8
method   cis(d)
ee_singlets  2
ee_triplets  2
EFP = 1
efp_use_efpman2 0
$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 11.24 EOM-IP-CCSD/EFP calculation of CN radical hydrated by 6 waters. EOM_FAKE_IPEA keyword can be either TRUE or FALSE. QM_EFP and EFP_EFP exchange-repulsion should be turned off.

```
$molecule
-1 1
 C           1.0041224092   2.5040921109  -0.3254633433
 N           0.8162211575   2.3197739512   0.7806258675
$end

$rem
basis 6-31+G*
method eom-ccsd
efp_fragments_only false
purecart 2222
scf_convergence 8
ip_states = 4
EFP = 1
eom_fake_ipea true
efp_exrep 0
efp_qm_exrep 0
efp_use_efpman2 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 11.25 EFP gradient computation for adenine dimer using library parameters.

```
$molecule
0 1
--
0 1
-1.37701480 1.41334560 3.40415522 2.504756 3.030456 -3.010720
--
0 1
-1.34215671 0.11877860 6.80424961 -2.807868 3.052053 -2.714643
$end

$rem
jobtyp force
method hf
purecart 2222
EFP_INPUT true
EFP_FRAGMENTS_ONLY 1
MAX_SCF_CYCLES 0
print_input false
sym_ignore true
efp_use_efpman2 0
$end

$efp_fragments
ADENINE_L
ADENINE_L
$end
```

Example 11.26 EFP energy computation for benzene dimer. This is an old-style input for EFP-only job which uses He atom trick. It will work but is no longer necessary (see the example above).

```
$molecule
0 1
 He       5.0    5.0    5.0
$end

$rem
method hf
basis 6-31G(d)
jobtype sp
purecart 2222
efp_fragments_only true
efp_use_efpman2 0
EFP_ELEC 1           ! EFP-EFP electrostatics is ON
EFP_POL 1            ! EFP-EFP polarization is ON
EFP_EXREP 1          ! EFP-EFP exchange-repulsion is ON
EFP_DISP 1           ! EFP-EFP dispersion is ON
EFP_QM_ELEC 1        ! QM-EFP electrostatics is ON
EFP_QM_POL 1         ! QM-EFP polarization is ON
EFP_QM_EXREP 0       ! QM-EFP exchange-repulsion is OFF
EFP_ELEC_DAMP 2      ! EFP-EFP electrostatic damping is EXPONENTIAL
EFP_DISP_DAMP 1      ! EFP-EFP dispersion damping is ON
EFP_QM_ELEC_DAMP 0   ! QM-EFP electrostatic damping is OFF
```

```
$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
```

11.6 Density Embedding Scheme

The exact density embedding method adapted from the method of Manby, Miller, and co-workers [101] allows embedding calculations to extend beyond electrostatic embedding. This embedding scheme allows for the fragmentation of a system into two interacting subsystems, which can be treated at two different levels of quantum mechanics (QM/QM), for example coupled cluster embedded in DFT. This type of embedding fully accounts for polarization as well as quantum mechanical exchange, as calculated from the supermolecular embedding density and the exchange correlation functional used. The goal of this embedding theory is to perform, say, a DFT calculation on fragment 1 in the presence of fragment 2.

11.6.1 Theory

Conventional supermolecular KS-DFT calculations contain several non-additive terms consisting of kinetic energies and exchange-correlation effects. These non-additive terms are eliminated by applying a level-shift projection operator to maintain orthogonality between fragments when performing a 1-in-2 energy calculation.

The process begins by performing a KS-DFT calculation on the full system, or supermolecule. The KS-DFT calculation is then repeated for fragment 1-in-2, which is fragment 1 in the presence of the localized MOs of fragment 2 taken from the supermolecular full calculation. The Fock Matrix for this calculation is constructed as follows,

$$\mathbf{f}^{(1)} = \mathbf{h} + \mathbf{J}[\gamma^{(1)} + \gamma^{(2)}] - \nu_{xc}[\gamma^{(1)} + \gamma^{(2)}] + \mu\mathbf{P}^{(2)} \tag{11.44}$$

where P is the level-shift projection operator constructed as:

$$\mathbf{P}^{(2)}_{\alpha\beta} = [\mathbf{S}\gamma^{(2)}\mathbf{S}]_{\alpha\beta} \tag{11.45}$$

where $\gamma^{(2)}$ is the localized density of fragment 2, and 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:

$$E_{correction} = \mu * tr(\gamma^{(1)}\mathbf{P}^{(2)}) \tag{11.46}$$

Once the KS-DFT energy of fragment 1-in-2 is computed, a post KS 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-DFT method to yield the low-level QM additive energy of the fragment 2. These energies are then summed to yield the total energy.

11.6.2 Job Control for Density Embedding Calculations

To use density embedding on a system, one must split the supermolecular 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 PURCAR = 222. It is also recommended that users disable symmetry for calculations with SYMMETRY = FALSE, and SYM_IGNORE = TRUE. 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 [Eq. (11.44) and (11.46)].

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 11.27 Input for a CCSD(T)/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.

```
$molecule
0 1
--
0 1
  He      -4.63032        2.10289       -1.62399
--
0 1
  F       -8.00612        1.74605       -1.25581
  H       -7.40964        1.84462       -0.47756
$end

$rem
  JOBTYPE         SP
  EXCHANGE        PBE
  CORRELATION     PBE
  BASIS           mixed  ! Must specify basis sets per atom in $basis section below
  PURCAR          222
  MAXSCF          100
  SCF_ALGORITHM   DIIS
  INCFOCK         0
  SYMMETRY        false
  SYM_IGNORE      true
  CC_SYMMETRY     false
  CCMAN2          true
  EMBEDMAN        1      ! Turning density embedding on
  EMBED_THEORY    1      ! Running CCSD(T) on fragment 1 in 2, or He in FH
  EMBED_MU        7      ! Default value
  EMBED_THRESH    500    ! Default value for assigning MOs to fragments
$end

$basis
  he 1
  6-31G
  ****
  f 2
  6-31G
  ****
  h 3
  6-31G
  ****
$end

@@@

$molecule
0 1
--
0 1
   F       -8.00612        1.74605       -1.25581
   H       -7.40964        1.84462       -0.47756
--
0 1
   He      -4.63032        2.10289       -1.62399
$end

$rem
  JOBTYPE         SP
  EXCHANGE        PBE
  CORRELATION     PBE
  BASIS           mixed
  PURCAR          222
  MAXSCF          100
  SCF_ALGORITHM   DIIS
  INCFOCK         0
  SYMMETRY        false
  SYM_IGNORE      true
  EMBEDMAN        1
  EMBED_THEORY    0      ! There will be no post-DFT calculations for HF-in-He
$end

$basis
  f 1
  6-31G
  ****
  h 2
  6-31G
  ****
  he 3
  6-31G
  ****
$end
```

References and Further Reading

[1] J. G. Kirkwood, *J. Chem. Phys.* **2**, 767 (1934).

[2] L. Onsager, *J. Am. Chem. Soc.* **58**, 1486 (1936).

[3] J. G. Kirkwood, *J. Chem. Phys.* **7**, 911 (1939).

[4] A. Klamt and G. Schüürmann, *J. Chem. Soc. Perkin Trans. 2* , 799 (1993).

[5] T. N. Truong and E. V. Stefanovich, *Chem. Phys. Lett.* **240**, 253 (1995).

[6] V. Barone and M. Cossi, *J. Phys. Chem. A* **102**, 1995 (1998).

[7] M. Cossi, N. Rega, G. Scalmani, and V. Barone, *J. Comput. Chem.* **24**, 669 (2003).

[8] D. M. Chipman, *J. Chem. Phys.* **112**, 5558 (2000).

[9] E. Cancès, *J. Chem. Phys.* **107**, 3032 (1997).

[10] E. Cancès and B. Mennucci, *J. Chem. Phys.* **114**, 4744 (2001).

[11] J. Tomasi, B. Mennucci, and R. Cammi, *Chem. Rev.* **106**, 2999 (2005).

[12] A. W. Lange and J. M. Herbert, *J. Phys. Chem. Lett.* **1**, 556 (2010).

[13] A. W. Lange and J. M. Herbert, *J. Chem. Phys.* **133**, 244111 (2010).

[14] A. W. Lange and J. M. Herbert, *Chem. Phys. Lett.* **509**, 77 (2011).

[15] 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 *Many-Body Effects and Electrostatics in Multi-Scale Computations of Biomolecules*, edited by Q. Cui, P. Ren, and M. Meuwly, 2014.

[16] J. Florián and A. Warshel, *J. Phys. Chem. B* **101**, 5583 (1997).

[17] J. Florián and A. Warshel, *J. Phys. Chem. B* **103**, 10282 (1999).

[18] A. V. Marenich, R. M. Olson, C. P. Kelly, C. J. Cramer, and D. G. Truhlar, *J. Chem. Theory Comput.* **3**, 2011 (2007).

[19] A. V. Marenich, C. J. Cramer, and D. G. Truhlar, *J. Chem. Theory Comput.* **9**, 609 (2013).

[20] A. Klamt and V. Jonas, *J. Chem. Phys.* **105**, 9972 (1996).

[21] K. Baldridge and A. Klamt, *J. Chem. Phys.* **106**, 6622 (1997).

[22] D. M. Chipman, *Theor. Chem. Acc.* **107**, 80 (2002).

[23] D. M. Chipman and M. Dupuis, *Theor. Chem. Acc.* **107**, 90 (2002).

[24] V. Barone, M. Cossi, and J. Tomasi, *J. Chem. Phys.* **107**, 3210 (1997).

[25] A. Pomogaeva and D. M. Chipman, *J. Chem. Theory Comput.* **7**, 3952 (2011).

[26] A. Pomogaeva and D. M. Chipman, *J. Chem. Theory Comput.* **10**, 211 (2014).

[27] C. J. Cramer and D. G. Truhlar, *Acc. Chem. Res.* **41**, 760 (2008).

[28] A. Klamt et al., *Acc. Chem. Res.* **42**, 489 (2009).

[29] A. Klamt, *J. Phys. Chem.* **99**, 2225 (1995).

[30] A. Klamt, F. Eckert, and W. Arlt, *Annu. Rev. Chem. Biomol. Eng.* **1**, 101 (2010).

[31] http://www.cosmologic.de/index.php.

[32] A. Klamt, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **1**, 699 (2011).

[33] R. C. Weast, editor, *CRC Handbook of Chemistry and Physics*, Chemical Rubber Company, Boca Rotan, 70th edition, 1989.

[34] M. W. Wong, M. J. Frisch, and K. B. Wiberg, *J. Am. Chem. Soc.* **113**, 4776 (1991).

[35] M. Born, *Z. Phys.* **1**, 45 (1920).

[36] S. Miertuš, E. Scrocco, and J. Tomasi, *Chem. Phys.* **55**, 117 (1981).

[37] J. Tomasi, B. Mennucci, and E. Cancès, *J. Mol. Struct. (Theochem)* **464**, 211 (1999).

[38] R. Bonaccorsi, P. Palla, and J. Tomasi, *J. Am. Chem. Soc.* **106**, 1945 (1984).

[39] J. Tomasi and M. Persico, *Chem. Rev.* **94**, 2027 (1994).

[40] A. Bondi, *J. Phys. Chem.* **68**, 441 (1964).

[41] V. I. Lebedev, *Zh. Vychisl. Mat. Mat. Fix.* **16**, 293 (1976).

[42] V. I. Lebedev, *Sibirsk. Mat. Zh.* **18**, 132 (1977).

[43] V. I. Lebedev and D. N. Laikov, *Dokl. Math.* **366**, 741 (1999).

[44] D. M. York and M. Karplus, *J. Phys. Chem. A* **103**, 11060 (1999).

[45] H. Li and J. H. Jensen, *J. Comput. Chem.* **25**, 1449 (2004).

[46] R. S. Rowland and R. Taylor, *J. Phys. Chem.* **100**, 7384 (1996).

[47] M. Mantina, A. C. Chamberlin, R. Valero, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem. A* **113**, 5806 (2009).

[48] A. K. Rappé, C. J. Casewit, K. S. Colwell, W. A. Goddard III, and W. M. Skiff, *J. Am. Chem. Soc.* **114**, 10024 (1992).

[49] J. L. Pascual-Ahuir, E. Silla, and I. T. non, *J. Comput. Chem.* **15**, 1127 (1994).

[50] W. Humphrey, A. Dalke, and K. Schulten, *J. Molec. Graphics* **14**, 33 (1996).

[51] The VMD program may be downloaded from www.ks.uiuc.edu/Research/vmd.

[52] M. Cossi, B. Mennucci, and R. Cammi, *J. Comput. Chem.* **17**, 57 (1996).

[53] P. Li, H. Johnston, and R. Krasny, *J. Comput. Phys.* **228**, 3858 (2009).

[54] A. Schäfer, A. Klamt, D. Sattle, J. C. W. Lohrenz, and F. Eckert, *Phys. Chem. Chem. Phys.* **2**, 2187 (2000).

[55] A. Klamt, F. Eckert, and M. Hornig, *J. Comput.-Aided Mol. Design* **15**, 355 (2001).

[56] D. A. Liotard, G. D. Hawkins, G. C. Lynch, C. J. Cramer, and D. G. Truhlar, *J. Comput. Chem.* **16**, 422 (1995).

[57] C. P. Kelly, C. J. Cramer, and D. G. Truhlar, *J. Chem. Theory Comput.* **1**, 1133 (2005).

[58] T. Zhu, J. Li, D. A. Liotard, C. J. Cramer, and D. G. Truhlar, *J. Chem. Phys.* **110**, 5503 (1999).

[59] J. D. Thompson, C. J. Cramer, and D. G. Truhlar, *J. Chem. Phys.* **119**, 1661 (2003).

[60] C. J. Cramer and D. G. Truhlar, in *Free Energy Calculations and Rational Drug Design*, edited by M. R. Reddy and M. D. Erion, page 63, Kluwer/Plenum, New York, 2001.

[61] J. Li, C. J. Cramer, and D. G. Truhlar, *Int. J. Quantum Chem.* **77**, 264 (2000).

[62] A. V. Marenich, S. V. Jerome, C. J. Cramer, and D. G. Truhlar, *J. Chem. Theory Comput.* **8**, 527 (2012).

[63] U. C. Singh and P. A. Kollman, *J. Comput. Chem.* **7**, 718 (1986).

[64] C. M. Breneman and K. B. Wiberg, *J. Comput. Chem.* **11**, 361 (1990).

[65] H. L. Woodcock et al., *J. Comput. Chem.* **28**, 1485 (2007).

[66] J. Wang, P. Cieplak, and P. A. Kollman, *J. Comput. Chem.* **21**, 1049 (2000).

[67] N. Foloppe and A. D. MacKerell, *J. Comput. Chem.* **21**, 86 (2000).

[68] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, *J. Am. Chem. Soc.* **117**, 11225 (1996).

[69] T. Vreven and K. Morokuma, *Annu. Rep. Comp. Chem.* **2**, 35 (2006).

[70] H. M. Senn and W. Thiel, *Topics Curr. Chem.* **268**, 173 (2007).

[71] Y. Shao and J. Kong, *J. Phys. Chem. A* **111**, 3661 (2007).

[72] P. Ren and J. W. Ponder, *J. Phys. Chem. B* **107**, 5933 (2003).

[73] K. Nam, J. Gao, and D. M. York, *J. Chem. Theory Comput.* **1**, 2 (2005).

[74] D. Riccardi, P. Schaefer, and Q. Cui, *J. Phys. Chem. B* **109**, 17715 (2005).

[75] R. C. Walker, M. F. Crowley, and D. A. Case, *J. Comput. Chem.* **29**, 1019 (2008).

[76] B. R. Brooks et al., *J. Comput. Chem.* **30**, 1545 (2009).

[77] Z. C. Holden, R. M. Richard, and J. M. Herbert, *J. Chem. Phys.* **139**, 244108 (2013).

[78] D. Das et al., *J. Chem. Phys.* **117**, 10534 (2002).

[79] H. L. Woodcock et al., *J. Chem. Phys.* **129**, 214109 (2008).

[80] P. N. Day et al., *J. Chem. Phys.* **105**, 1968 (1996).

[81] M. S. Gordon et al., *J. Phys. Chem. A* **105**, 293 (2001).

[82] M. W. Schmidt et al., *J. Comput. Chem.* **14**, 1347 (1983).

[83] D. Ghosh et al., *J. Phys. Chem. A* **114**, 12739 (2010).

[84] M. S. Gordon, D. G. Fedorov, S. R. Pruitt, and L. V. Slipchenko, *Chem. Rev.* **112**, 632 (2012).

[85] I. A. Kaliman and L. V. Slipchenko.

[86] A. D. Buckingham, *Q. Rev. Chem. Soc.* **13**, 183 (1959).

[87] A. J. Stone, *Chem. Phys. Lett.* **83**, 233 (1981).

[88] A. J. Stone and M. Alderton, *Mol. Phys.* **56**, 1047 (1985).

[89] L. V. Slipchenko and M. S. Gordon, *J. Comput. Chem.* **28**, 276 (2007).

[90] M. A. Freitag, M. S. Gordon, J. H. Jensen, and W. J. Stevens, *J. Chem. Phys.* **112**, 7300 (2000).

[91] L. V. Slipchenko and M. S. Gordon, *Mol. Phys.* **107**, 999 (2009).

[92] I. Adamovic and M. S. Gordon, *Mol. Phys.* **103**, 379 (2005).

[93] K. T. Tang and J. P. Toennies, *J. Chem. Phys.* **80**, 3726 (1984).

[94] J. H. Jensen and M. S. Gordon, *Mol. Phys.* **89**, 1313 (1996).

[95] J. H. Jensen and M. S. Gordon, *J. Chem. Phys.* **108**, 4772 (1998).

[96] L. V. Slipchenko, *J. Phys. Chem. A* **114**, 8824 (2010).

[97] D. Kosenkov and L. V. Slipchenko, *J. Phys. Chem. A* **115**, 392 (2011).

[98] D. W. Zhang and J. Z. H. Zhang, *J. Chem. Phys.* **119**, 3599 (2003).

[99] D. W. Zhang, X. H. Chen, and J. Z. H. Zhang, *J. Comput. Chem.* **24**, 1846 (2003).

[100] J. C. Flick, D. Kosenkov, E. G. Hohenstein, C. D. Sherrill, and L. V. Slipchenko, *J. Chem. Theory Comput.* **8**, 2835 (2012).

[101] F. R. Manby, M. Stella, J. D. Goodpaster, and T. F. Miller III, *J. Chem. Theory Comput.* **8**, 2564 (2012).

Chapter 12

Methods Based on Absolutely-Localized Molecular Orbitals

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 were developed and implemented by Dr. Rustam Z. Khaliullin at the University of California–Berkeley, with Profs. Martin Head-Gordon and Alexis Bell; the open shell versions were developed by Paul Horn at Berkeley, working with Martin Head-Gordon. Others were developed by Dr. Leif D. Jacobson and Ka Un Lao working with Prof. John M. 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) [1].
- Constrained (locally-projected) SCF methods for molecular interactions (SCF MI methods) between closed shell fragments [1], and also open shell fragments [2].
- Single Roothaan-step (RS) correction methods that improve FRAGMO and SCF MI description of molecular systems [1, 2].
- Automated calculation of the BSSE with counterpoise correction method (full SCF and RS implementation).
- Energy decomposition analysis and charge transfer analysis [2–4].
- Analysis of intermolecular bonding in terms of complementary occupied-virtual pairs [2, 4, 5].
- The variational explicit polarization (XPol) method, a self-consistent, charge-embedded, monomer-based SCF calculation [6–8]
- Symmetry-adapted perturbation theory (SAPT), a monomer-based method for computing intermolecular interaction energies and decomposing them into physically-meaningful components [9, 10].

- XPol+SAPT (XSAPT), which extends the SAPT methodology to systems consisting of more than two monomers [7, 8, 11].
- XSAPT(KS)+D, a dispersion-corrected version of XSAPT that affords accurate intermolecular interaction energies at very low cost [12, 13].
- The electrostatically embedded many-body expansion [14] and the fragment molecular orbital method [15, 16], for decomposing large clusters into small numbers of monomers, facilitating larger calculations.

Another fragment-based approach, Effective Fragment Potential method, was developed by Prof. Lyudmila V. Slipchenko at Purdue University and Prof. Anna I. Krylov at USC; this method is described in Section 11.5.

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 12.1 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 12.2 Open shell fragment descriptors in the *$molecule* section.

```
$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 multifragment systems with large basis sets [3]. 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, SCF_GUESS, MEM_TOTAL, MEM_STATIC are not inherited from the parent job.

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.

```
$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         SP
METHOD          EDF1
BASIS           6-31(2+,2+)g(df,pd)
SCF_GUESS       FRAGMO
SCF_PRINT_FRGM  FALSE
$end

$rem_frgm
SCF_CONVERGENCE 2
THRESH          5
$end
```

12.4 Locally-Projected SCF Methods

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 [1, 2]. The method is based on the equations of the locally-projected SCF for molecular interactions (SCF MI) [1, 2, 17–19]. 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 [1].

The ALMO approximation significantly reduces the number of variational degrees of freedom of the wavefunction. 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.* [17], the other utilizes the locally-projected equations of Gianinetti *et al.* [18] 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 12.4 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
JOBTYPE          SP
METHOD           BP86
BASIS            6-31(+,+)G(d,p)
FRGM_METHOD      STOLL
$end

$rem_frgm
SCF_CONVERGENCE 2
THRESH           5
$end
```

12.4.1 Locally-Projected SCF Methods with Single Roothaan-Step Correction

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 [1]. 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 [1, 20, 21]. 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. Wavefunction 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 12.5 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
JOBTYPE         SP
METHOD          HF
BASIS           AUG-CC-PVTZ
FRGM_METHOD     GIA
FRGM_LPCORR     RS_EXACT_SCF
$end

$rem_frgm
SCF_CONVERGENCE 2
THRESH          5
$end
```

12.4.2 Roothaan-Step Corrections to the FRAGMO Initial Guess

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).

12.4.3 Automated Evaluation of the Basis-Set Superposition Error

Evaluation of the basis-set superposition error (BSSE) is automated in Q-CHEM. To calculate BSSE-corrected binding energies, specify fragments in the *$molecule* section and set JOBTYPE to BSSE. The BSSE jobs are not limited to the SCF energies and can be evaluated for multifragment systems at any level of theory. Q-CHEM separates the system into fragments as specified in the *$molecule* section and performs a series of jobs on (a) each fragment, (b) each fragment with the remaining atoms in the system replaced by the ghost atoms, and (c) on the entire system. Q-CHEM saves all calculated energies and prints out the uncorrected and the BSSE corrected binding 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.

Important. It is recommended to include PURECART keyword in all BSSE jobs. GENERAL basis cannot be used for the BSSE calculations in the current implementation. Use MIXED basis instead.

Example 12.6 Evaluation of the BSSE corrected intermolecular interaction energy

```
$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           MP2
BASIS            6-31(+,+)G(d,p)
$end
```

12.5 Energy Decomposition and Charge-Transfer Analysis

12.5.1 Energy Decomposition 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 [3], including the open shell generalization [2]. 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 [4]. The ALMO charge transfer scale, $DeltaQ$, 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 GENERAL basis sets 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.

Example 12.7 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
METHOD           EDF1
BASIS            6-31(+,+)g(d,p)
PURECART         1112
FRGM_METHOD      GIA
FRGM_LPCORR      RS_EXACT_SCF
EDA_BSSE         TRUE
$end
```

Example 12.8 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
  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
```

12.5.2 Analysis of Charge-Transfer Based on 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 [5], and the description of synergic bonding in metal complexes agrees well with simple Dewar-Chatt-Duncanson model [4, 22, 23].

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 VMD or a similar third party program capable of making 3D plots.

Example 12.9 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
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 12.10 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 to 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
JOBTYPE          SP
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
```

12.6 Job Control for Locally-Projected SCF Methods

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:

Experimental option. Use 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 angstroms 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 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 default. Increasing n improves convergence of the PCG algorithm but overall may slow down calculations.

12.7 The Explicit Polarization (XPol) Method

12.7.1 Theory

XPol is an approximate, fragment-based molecular orbital method that was developed as a "next-generation" force field [6, 24–26]. 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 10.2.1). This approach incorporates many-body polarization, at a cost that scales linearly with the number of fragments, but neglects the antisymmetry requirement of the total electronic wavefunction. 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 supersystem wavefunction is written as a direct product of fragment wavefunctions,

$$|\Psi\rangle = \prod_{A}^{N_{frag}} |\Psi_A\rangle, \tag{12.1}$$

where N_{frag} is the number of fragments. We assume here that the fragments are molecules and that covalent bonds remain intact. The fragment wavefunctions are antisymmetric 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 [6, 7]

$$E_{\text{XPol}} = \sum_A \left[2\sum_a \mathbf{c}_a^\dagger \left(\mathbf{h}^A + \mathbf{J}^A - \tfrac{1}{2}\mathbf{K}^A\right)\mathbf{c}_a + E_{nuc}^A \right] + E_{embed}. \tag{12.2}$$

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. (12.2),

$$E_{embed} = \tfrac{1}{2}\sum_{A}\sum_{B\neq A}\sum_{J\in B}\left(-2\sum_{a}\mathbf{c}_a^{\dagger}\mathbf{I}_J\mathbf{c}_a + \sum_{I\in A} L_{IJ}\right) q_J, \tag{12.3}$$

arise from the electrostatic embedding. The matrix $\mathbf{I}_J$ is defined by its AO matrix elements,

$$(\mathbf{I}_J)_{\mu\nu} = \left\langle \mu \left| \frac{1}{|\vec{r}-\vec{R}_J|} \right| \nu \right\rangle, \tag{12.4}$$

and L_{IJ} is given by

$$L_{IJ} = \frac{Z_I}{|\vec{R}_I - \vec{R}_J|}. \tag{12.5}$$

According to Eqs. (12.2) and (12.3), 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. (12.3) 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* [1], 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.* [6, 25, 26] uses Mulliken charges for the embedding charges q_J in Eq. (12.3), though other charge schemes could be envisaged. In non-minimal basis sets, the use of Mulliken charges is beset by severe convergence problems [7], and Q-CHEM's implementation of XPol offers the alternative of using either Löwdin charges or "CHELPG" charges [27], the latter being. derived from the electrostatic potential as discussed in Section 10.2.1. The CHELPG charges are found to be stable and robust, albeit with a somewhat larger computational cost as compared to Mulliken or Löwdin charges [7, 8]. An algorithm to compute CHELPG charges using atom-centered Lebedev grids rather than traditional Cartesian grids is available [28] (see Section 10.2.1), 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.

Researchers who use Q-CHEM's XPol code are asked to cite Refs. 7, and 8.

12.7.2 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 12.9. 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

$$V_{\mathrm{LJ}}(R_{ij}) = 4\epsilon_{ij}\left[\left(\frac{\sigma_{ij}}{R_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{R_{ij}}\right)^{6}\right], \tag{12.6}$$

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,

$$V_{\text{Buck}}(R_{ij}) = \epsilon_{ij}\left[Ae^{-B\frac{R_{ij}}{\sigma_{ij}}} - C\left(\frac{\sigma_{ij}}{R_{ij}}\right)^6\right] , \tag{12.7}$$

Here, A, B, and C are additional (dimensionless) constants, independent of atom type. In both Eq. (12.6) and Eq. (12.7), the parameters ϵ_{ij} and σ_{ij} are determined using the geometric mean of atomic well-depth and length-scale parameters. For example,

$$\sigma_{ij} = \sqrt{\sigma_i \sigma_j} . \tag{12.8}$$

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 Angstroms. 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.

12.7.3 Job Control Variables for XPol

XPol calculations are enabled by setting the *$rem* variable XPOL to TRUE. These calculations can be used in combination with Hartree-Fock theory and with most density functionals, a notable exception being that XPol is not yet implemented for meta-GGA functionals (Section 4.3.3). Combining XPol with solvation models (Section 11.2) or external charges (*$external_charges*) is also not available. Analytic gradients are available when Mulliken or Löwdin 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

XPOL_CHARGE_TYPE

Controls the type of atom-centered embedding charges for XPol calculations.

TYPE:

STRING

DEFAULT:

QLOWDIN

OPTIONS:

QLOWDIN	Löwdin charges.
QMULLIKEN	Mulliken charges.
QCHELPG	CHELPG charges.

RECOMMENDATION:

Problems with Mulliken charges in extended basis sets can lead to XPol convergence failure. Löwdin charges tend to be more stable, and CHELPG charges are both robust and provide an accurate electrostatic embedding. However, CHELPG charges are more expensive to compute, and analytic energy gradients are not yet available for this choice.

XPOL_MPOL_ORDER

Controls the order of multipole expansion that describes electrostatic interactions.

TYPE:

STRING

DEFAULT:

CHARGES

OPTIONS:

GAS	No electrostatic embedding; monomers are in the gas phase.
CHARGES	Charge embedding.
DENSITY	Density embedding.

RECOMMENDATION:

Should be set to GAS to do a dimer SAPT calculation (see Section 12.8).

XPOL_PRINT

Print level for XPol calculations.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

N	Integer print level

RECOMMENDATION:

Higher values prints more information

XPOL_OMEGA

Controls the range-separation parameter, ω, that is used in long-range-corrected DFT.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Use different ω values for different fragments.
FALSE	Use a single value of ω for all fragments.

RECOMMENDATION:

If FALSE, the *$rem* variable OMEGA should be used to specify the single value of ω. If TRUE, separate values for each fragment should be specified in an *$lrc_omega* input section. Values in the *$lrc_omega* section have the same units as the *$rem* variable OMEGA, namely, ω =OMEGA/1000, in atomic units.

12.7.4 Examples

XPol on its own is not a useful method (as 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 12.11 An XPol single point calculation on the water dimer using a Lennard-Jones 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
XPOL_CHARGE_TYPE QLOWDIN
$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 12.12 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
XPOL_CHARGE_TYPE QLOWDIN
$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
```

12.8 Symmetry-Adapted Perturbation Theory (SAPT)

12.8.1 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.* [9, 29]. Additional context can be found in a pair of more recent review articles [10, 30].

In SAPT, the Hamiltonian for the A $\cdots$ B dimer is written as

$$\hat{H} = \hat{F}^A + \hat{F}^B + \xi\hat{W}^A + \eta\hat{W}^B + \zeta\hat{V} \ , \tag{12.9}$$

where $\hat{W}^A$ and $\hat{W}^B$ are Møller-Plesset fluctuation operators for fragments A and B, whereas $\hat{V}$ consists of the intermolecular Coulomb operators. This part of the perturbation is conveniently expressed as

$$\hat{V} = \sum_{i\in A}\sum_{j\in B}\hat{v}(ij) \tag{12.10}$$

with

$$\hat{v}(ij) = \frac{1}{|\vec{r}_i - \vec{r}_j|} + \frac{\hat{v}_A(j)}{N_A} + \frac{\hat{v}_B(i)}{N_B} + \frac{V_0}{N_A N_B} \ . \tag{12.11}$$

The quantity V_0 is the nuclear interaction energy between the two fragments and

$$\hat{v}_A(j) = -\sum_{I\in A}\frac{Z_I}{|\vec{r}_j - \vec{R}_I|} \tag{12.12}$$

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 wavefunctions that are direct products of monomer wavefunctions, $|\Psi_0\rangle = |\Psi_A\rangle|\Psi_B\rangle$, the SAPT approach is based on a symmetrized Rayleigh-Schrödinger perturbation expansion [9, 29] with respect to the perturbation parameters ξ, η, and ζ in Eq. (12.9). The resulting interaction energy can be expressed as[9, 29]

$$E_{int} = \sum_{i=1}^{\infty}\sum_{j=0}^{\infty}\left(E^{(ij)}_{pol} + E^{(ij)}_{exch}\right) \ . \tag{12.13}$$

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 $E^{(ij)}_{pol}$ 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 $E^{(ij)}_{pol}$ has a corresponding exchange term, $E^{(ij)}_{exch}$, that arises from an antisymmetrizer $\hat{\mathcal{A}}_{AB}$ that is introduced in order to project away the Pauli-forbidden components of the interaction energy that would otherwise appear [9].

The version of SAPT that is implemented in Q-CHEM assumes that $\xi = \eta = 0$, an approach that is usually called SAPT(0) [30]. Within the SAPT(0) formalism, the interaction energy is formally expressed by the following symmetrized Rayleigh-Schrödinger expansion [9, 29]:

$$E_{int}(\zeta) = \frac{\langle\Psi_0|\zeta\hat{V}\hat{\mathcal{A}}_{AB}|\Psi(\zeta)\rangle}{\langle\Psi_0|\hat{\mathcal{A}}_{AB}|\Psi(\zeta)\rangle} \ , \tag{12.14}$$

The antisymmetrizer $\hat{\mathcal{A}}_{AB}$ in this expression can be written as

$$\hat{\mathcal{A}}_{AB} = \frac{N_A!N_B!}{(N_A + N_B)!}\,\hat{\mathcal{A}}_A\hat{\mathcal{A}}_B\big(\hat{1} + \hat{\mathcal{P}}^{AB} + \hat{\mathcal{P}}'\big) \ , \tag{12.15}$$

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. (12.15) denotes all of the three-electron and higher-order exchanges. This operator is neglected in what is known as the "single-exchange" approximation [9, 29], which is expected to be quite accurate at typical van der Waals and larger intermolecular separations, but sometimes breaks down at smaller intermolecular separations [31].

Only terms up to $\zeta = 2$ in Eq. (12.14)—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. (12.13)]:

$$E_{int}^{\mathrm{SAPT(0)}} = E_{elst}^{(1)} + E_{exch}^{(1)} + E_{pol}^{(2)} + E_{exch}^{(2)} \; . \tag{12.16}$$

The electrostatic part of the first-order energy correction is denoted $E_{elst}^{(1)}$ and represents the Coulomb interaction between the two monomer electron densities [9]. The quantity $E_{exch}^{(1)}$ is the corresponding first-order (*i.e.*, Hartree-Fock) exchange correction. Explicit formulas for these corrections can be found in Ref. 29. The second-order term from the polarization expansion, denoted $E_{pol}^{(2)}$ in Eq. (12.16), 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

$$E_{ind}^{(2)} = E_{ind}^{(2)}(A \leftarrow B) + E_{ind}^{(2)}(B \leftarrow A) \; , \tag{12.17}$$

where the notation $A \leftarrow B$, for example, indicates that the frozen charge density of B polarizes the density of A. In detail,

$$E_{ind}^{(2)}(A \leftarrow B) = 2 \sum_{ar} t_{ar} (w_B)_{ra} \tag{12.18}$$

where

$$(w_B)_{ar} = (\hat{v}_B)_{ar} + \sum_{b} (ar|bb) \tag{12.19}$$

and $t_{ar} = (w_B)_{ar}/(\epsilon_a - \epsilon_r)$. The second term in Eq. (12.17), in which A polarizes B, is obtained by interchanging labels [7]. The second-order dispersion correction has a form reminiscent of the MP2 correlation energy:

$$E_{disp}^{(2)} = 4 \sum_{abrs} \frac{(ar|bs)(ra|sb)}{\epsilon_a + \epsilon_b - \epsilon_r - \epsilon_s} \; . \tag{12.20}$$

The induction and dispersion corrections both have accompanying exchange corrections (exchange-induction and exchange-dispersion) [9, 29].

The similarity between Eq. (12.20) 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 5.5.1.

It is common to replace $E_{ind}^{(2)}$ and $E_{exch\text{-}ind}^{(2)}$ in Eq. (12.16) with their "response" (*resp*) analogues, which are the infinite-order correction for polarization arising from a frozen partner density [9, 29]. Operationally, this substitution involves replacing the second-order induction amplitudes, t_{ar} in Eq. (12.18), with amplitudes obtained from solution of the coupled-perturbed Hartree-Fock equations [32]. (The perturbation is simply the electrostatic potential of the other monomer.) In addition, it is common to correct the SAPT(0) binding energy for higher-order polarization effects by adding a correction term of the form [9, 30]

$$\delta E_{int}^{\mathrm{HF}} = E_{int}^{\mathrm{HF}} - \left(E_{elst}^{(1)} + E_{exch}^{(1)} + E_{ind,resp}^{(2)} + E_{exch\text{-}ind,resp}^{(2)} \right) \tag{12.21}$$

to the interaction energy. Here, E_{int}^{HF} is the counterpoise-corrected Hartree-Fock binding energy for $\mathrm{A} \cdots \mathrm{B}$. Both the response corrections and the $\delta E_{\mathrm{int}}^{\mathrm{HF}}$ correction have been implemented as options in Q-CHEM's implementation of SAPT.

It is tempting to replace Hartree-Fock MOs and eigenvalues in the SAPT(0) formulas with their Kohn-Sham counterparts, as a low-cost means of introducing monomer electron correlation. The resulting procedure is known as SAPT(KS) [33], and does offer an improvement on SAPT(0) for some strongly hydrogen-bonded systems [8]. Unfortunately, SAPT(KS) results are generally in poor agreement with benchmark dispersion energies [8], owing to incorrect asymptotic behavior of approximate exchange-correlation potentials [34]. 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 $\epsilon_{\text{HOMO}} = -\text{IP}$, where ϵ_{HOMO} is the HOMO energy and "IP" represents the ionization potential [35]. Monomer-specific values of ω, tuned using the individual monomer IPs, substantially improve SAPT(KS) dispersion energies, though the results are still not of benchmark quality [35]. Other components of the interaction energy, however, can be described quite accurately SAPT(KS) in conjunction with a tuned version of LRC-ωPBE [35]. Use of monomer-specific ω values is controlled by the variable XPOL_OMEGA in the *$rem* section, and individual values are entered via an *$lrc_omega* input section.

Finally, some discussion of basis sets is warranted. Typically, SAPT calculations are performed in the so-called dimer-centered basis set (DCBS) [36], which means that the combined $A + B$ basis set is used to calculate the zeroth-order wavefunctions 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) [36]. However, MCBS results are generally of poorer quality. As an efficient alternative to the DCBS, Jacobson and Herbert [7] introduced a projected ("proj") basis set, borrowing an idea from dual-basis MP2 calculations [37]. 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 "pseudo-canonicalized", 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 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.

12.8.2 Job Control for SAPT Calculations

Q-CHEM's implementation of SAPT(0) was designed from the start as a correction for XPol calculations, a functionality that is described in Section 12.9. As such, a SAPT calculation is requested by setting both of the *$rem* variable SAPT and XPOL to TRUE. (Alternatively, one may set RISAPT=TRUE to use the RI version of SAPT.) If one wishes to perform a traditional SAPT calculation based on gas-phase SCF monomer wavefunctions rather than XPol monomer wavefunctions, then the *$rem* variable XPOL_MPOL_ORDER should be set to GAS.

SAPT energy components are printed separately at the end of a SAPT job. If EXCHANGE = HF, then the calculation corresponds to SAPT(0), whereas a SAPT(KS) calculation is requested by specifying the desired density functional. [Note that meta-GGAs are not yet available for SAPT(KS) calculations in Q-CHEM.] At present, only single-point energies for closed-shell (restricted) calculations are possible. Frozen orbitals are also unavailable.

Researchers who use Q-CHEM's SAPT code are asked to cite Refs. 7 and 8.

SAPT

Requests a SAPT calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Run a SAPT calculation.
FALSE	Do not run SAPT.

RECOMMENDATION:

If SAPT is set to TRUE, one should also specify XPOL=TRUE and XPOL_MPOL_ORDER=GAS.

RISAPT

Requests an RI-SAPT calculation

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Compute four-index integrals using the RI approximation.
FALSE	Do not use RI.

RECOMMENDATION:

Set to TRUE if an appropriate auxiliary basis set is available, as RI-SAPT is much faster and affords negligible errors (as compared to ordinary SAPT) if the auxiliary basis set is matched to the primary basis set. (The former must be specified using AUX_BASIS.)

SAPT_ORDER

Selects the order in perturbation theory for a SAPT calculation.

TYPE:

STRING

DEFAULT:

SAPT2

OPTIONS:

SAPT1	First order SAPT.
SAPT2	Second order SAPT.
ELST	First-order Rayleigh-Schrödinger perturbation theory.
RSPT	Second-order Rayleigh-Schrödinger perturbation theory.

RECOMMENDATION:

SAPT2 is the most meaningful.

SAPT_EXCHANGE

Selects the type of first-order exchange that is used in a SAPT calculation.

TYPE:

STRING

DEFAULT:

S_SQUARED

OPTIONS:

S_SQUARED	Compute first order exchange in 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.

SAPT_BASIS

Controls the MO basis used for SAPT corrections.

TYPE:

STRING

DEFAULT:

MONOMER

OPTIONS:

MONOMER	Monomer-centered basis set (MCBS).
DIMER	Dimer-centered basis set (DCBS).
PROJECTED	Projected basis set.

RECOMMENDATION:

The DCBS is more costly than the MCBS and can only be used with XPOL_MPOL_ORDER=GAS (*i.e.*, it is not available for use with XPol). The PROJECTED choice is an efficient compromise that is available for use with XPol.

SAPT_CPHF

Requests that the second-order corrections $E^{(2)}_{ind}$ and $E^{(2)}_{exch\text{-}ind}$ be replaced by their infinite-order "response" analogues, $E^{(2)}_{ind,resp}$ and $E^{(2)}_{exch\text{-}ind,resp}$.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Evaluate the response corrections and use $E^{(2)}_{ind,resp}$ and $E^{(2)}_{exch\text{-}ind,resp}$
FALSE	Omit these corrections and use $E^{(2)}_{ind}$ and $E^{(2)}_{exch\text{-}ind}$.

RECOMMENDATION:

Computing the response corrections requires solving CPHF equations for pair of monomers, which is somewhat expensive but may improve the accuracy when the monomers are polar.

SAPT_DSCF

Request the $\delta E_{int}^{\mathrm{HF}}$ correction

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Evaluate this correction.

FALSE Omit this correction.

RECOMMENDATION:

Evaluating the $\delta E_{int}^{\mathrm{HF}}$ correction requires an SCF calculation on the entire (super)system. This corrections effectively yields a "Hartree-Fock plus dispersion" estimate of the interaction energy.

SAPT_PRINT

Controls level of printing in SAPT.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

N Integer print level

RECOMMENDATION:

Larger values generate additional output.

SAPT_DISP_CORR

Request an empirical dispersion potential instead of calculating $E_{disp}^{(2)}$ and $E_{exch\text{-}disp}^{(2)}$ directly

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use a dispersion force field.

FALSE calculate $E_{disp}^{(2)}$ and $E_{exch\text{-}disp}^{(2)}$.

RECOMMENDATION:

Using dispersion potentials reduces the scaling from $O(N^5)$ to $O(N^3)$ with respect to monomer size.

SAPT_DISP_VERSION

Controls which dispersion potential is used for SAPT

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1 Use the dispersion potentials from Hesselmann [12, 38].

2 Use the dispersion potentials from Podeszwa. [13, 39].

RECOMMENDATION:

The Hesselmann ("+D") correction corresponds to “generation 1” of XSAPT(KS)+D [12], while the Podeszwa ("+D2") potential is “generation 2” [13]. The two potentials were parameterized in very different ways. The latter was fit directly to $E^{(2)}_{disp}$ and $E^{(2)}_{exch\text{-}disp}$ components calculated using SAPT(DFT), and generally performs better [13].

Example 12.13 Example showing a SAPT(0) calculation using the RI approximation in a DCBS.

```
$rem
BASIS           AUG-CC-PVDZ
AUX_BASIS       RIMP2-AUG-CC-PVDZ
METHOD          HF
RISAPT          TRUE
XPOL            TRUE
XPOL_MPOL_ORDER GAS   ! gas-phase monomer wave functions
SAPT_BASIS      DIMER
SYM_IGNORE      TRUE
$end

$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
```

12.9 The XPol+SAPT (XSAPT) Method

12.9.1 Theory

XPol+SAPT, or "XSAPT", was proposed by Jacobson and Herbert [7, 8] as a low-order-scaling, systematically-improveable method applicable to large systems. The idea is simply 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 [7]. From another point of view, this approach 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 SAPT(0) or SAPT(KS) calculations.

In particular, the zeroth-order Hamiltonian for XSAPT is taken by the the sum of fragment Fock operators defined by the XPol procedure, and the perturbation is the usual SAPT intermolecular perturbation [Eq. (12.11)] less the intermolecular interactions contained in the XPol fragment Fock operators. A standard SAPT(0) correction is then computed for each pair of monomers, using Eq. (12.16) in conjunction with the modified perturbation, to obtain dimer interaction energy E_{int}^{AB}. The total XSAPT energy is then

$$E_{\text{XSAPT}} = \sum_A \left(\sum_a \Big[2\,\epsilon_a^A - \mathbf{c}_a^\dagger (\mathbf{J}^A - \tfrac{1}{2}\mathbf{K}^A)\mathbf{c}_a \Big] + E_{nuc}^A + \sum_{B>A} E_{int}^{AB} \right) . \tag{12.22}$$

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, and is available in Q-CHEM.

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). 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 [13] 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 δ_{int}^{HF} corrections, comparing XSAPT results to traditional SAPT based on gas-phase monomers.) An XSAPT calculation is requested by setting the *$rem* variables XPOL and SAPT equal to TRUE and also setting XPOL_MPOL_ORDER = CHARGES.

Researchers who use Q-CHEM's XPol+SAPT code are asked to cite Refs. 7 and 8. The latter contains a thorough discussion of the theory; a briefer summary can be found in Ref. 11.

Example 12.14 Example showing an XPol+SAPT(0) calculation using CHELPG charges and CPHF.

```
$rem
BASIS             CC-PVDZ
METHOD            HF
XPOL              TRUE
XPOL_MPOL_ORDER   CHARGES
XPOL_CHARGE_TYPE  QCHELPG
SAPT              TRUE
SAPT_CPHF         TRUE
SYM_IGNORE        TRUE
```

```
$end


$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
```

12.9.2 XSAPT(KS)+D

As mentioned above, the dispersion components of the (X)SAPT(KS) interaction energy are not of benchmark quality, even when tuned LRC functionals are employed [35]. The dispersion and exchange-dispersion terms are also the most expensive part of a SAPT(0) or SAPT(KS) calculation, scaling as the fourth and fifth powers, respectively, of monomer size, whereas other terms are cubic scaling at worst. Both the efficiency and the accuracy of XSAPT(KS) calculations are thus improved if $E^{(2)}_{disp} + E^{(2)}_{exch\text{-}disp}$ is replaced by an empirical atom–atom dispersion potential. The resulting method is called XSAPT(KS)+D [12, 13].

An XSAPT(KS)+D calculation is requested by setting the *$rem* variable SAPT_DISP_CORR equal to TRUE. There are two versions of the dispersion potential, a “first generation" (+D) version [12] and a second generation (+D2) version [13], and the choice between the two is dictated by the variable SAPT_DISP_VERSION. Although both versions perform similarly for total binding energies, the second-generation version (in which the dispersion potential was fit directly to high-accuracy dispersion energy benchmarks) does a much better job of reproducing accurate benchmarks for the various energy components [13]. As with XPol, the XSAPT and XSAPT(KS)+D methods do not function with a solvation model or with external changes. Currently, only single-point energies are available, and only for closed-shell (restricted) calculations with no frozen orbitals.

Researchers who use XSAPT(KS)+D are asked to cite Ref. 12, or Ref. 13 for XSAPT(KS)+D2.

Example 12.15 XSAPT(KS)+D2 calculation of benzene–methane interaction.

```
$rem
sym_ignore        true
exchange          gen
basis             aug-cc-pvtz
aux_basis         rimp2-aug-cc-pvtz
xpol              true ! must be set to true for sapt jobs too
xpol_mpol_order   charges
xpol_charge_type  qchelpg
xpol_omega        true
```

```
risapt            true
sapt_order        2
sapt_basis        projected
sapt_disp_corr    true
sapt_disp_version 2  ! +D2
thresh            14
scf_convergence    7
basis_lin_dep_thresh 12
lrc_dft           true
$end

$xc_functional
x   wPBE   1.0
c   PBE   1.0
$end

$lrc_omega
275
450
$end

$molecule
0 1
-- benzene
0 1
C     1.3932178    0.0362913   -0.6332803
C     0.7280364   -1.1884015   -0.6333017
C    -0.6651797   -1.2247077   -0.6332803
C    -1.3932041   -0.0362972   -0.6333017
C    -0.7280381    1.1884163   -0.6332803
C     0.6651677    1.2246987   -0.6333017
H     2.4742737    0.0644484   -0.6317240
H     1.2929588   -2.1105409   -0.6317401
H    -1.1813229   -2.1750081   -0.6317240
H    -2.4742614   -0.0644647   -0.6317401
H    -1.2929508    2.1105596   -0.6317240
H     1.1813026    2.1750056   -0.6317401
-- methane
0 1
C     0.0000000    0.0000000    3.0826195
H     0.5868776    0.8381742    3.4463772
H    -1.0193189    0.0891638    3.4463772
H     0.0000000    0.0000000    1.9966697
H     0.4324413   -0.9273380    3.4463772
$end
```

12.10 The Many-Body Expansion Method

12.10.1 Theory and Implementation Details

The many-body expansion (MBE) for a system of N monomers is given by

$$E = \sum_{I=1}^{N} E_I + \sum_{I}^{N} \sum_{J>I}^{N} \Delta E_{IJ} + \sum_{I}^{N} \sum_{J>I}^{N} \sum_{K>J}^{N} \Delta E_{IJK} + \cdots , \tag{12.23}$$

in which E_I represents the energy of monomer I, ΔE_{IJ} = $E_{IJ} - E_I - E_J$ is a two-body correction for dimer IJ, and ΔE_{IJK} = $E_{IJK} - \Delta E_{IJ} - \Delta E_{IK} - \Delta E_{JK} - E_I - E_J - E_k$ is a three-body correction for trimer IJK, etc. 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) [14]. Alternatively, since the monomer electron densities are available from the E_I terms as well, one could use these densities to compute the actual monomer–monomer Coulomb interactions, and this forms the basis of the fragment molecular orbital (FMO) method [15, 16]. ***Note that* Q-CHEM*'s* *present implementation of Eq.*** (12.23) ***is very preliminary***. In particular, the "bodies" (fragments) cannot be covalently bonded to one another, and therefore this is a method appropriate for non-covalent clusters of molecules. Moreover, individual subsystem calculations have not yet been parallelized across processors, hence the present implementation is not using the real power of the MBE. While significant speed-ups may be observed at the two-body level, three-body (and higher) calculations are presently useful only as proof-of-concept exercises, as the total wall time will often be larger than the corresponding supersystem calculation. Analytic gradients are not yet available. These deficiencies will be rectified in a future release of the program.

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 (CP) procedure corrects for this by computing all energies (including cluster and monomers) using the cluster basis set. (Note, however, that basis-set extrapolation is still necessary for high-quality binding energies; in $(H_2O)_6$, for example, a CP-corrected MP2/aug-cc-pVQZ calculation is still $\approx$ 1 kcal/mol from the MP2 basis-set limit [40]. Fortunately, the MBE allows for use of large basis sets in order to perform basis-set extrapolations in sizable clusters [40, 41].) Two low-cost CP corrections that are consistent with an n-body expansion have been proposed: the many-body CP correction, MBCP(n) [40, 41], and the n-body Valiron-Mayer function counterpoise correction, VMFC(n) [42]. 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) [40, 41].

Q-CHEM's implementation of the EE-MBE(n) approach (electrostatically-embedded n-body expansion) is designed to use self-consistent charges generated by XPol calculations. As such, a MBE calculation is requested by setting both MANY_BODY_INT and XPOL = TRUE. The variable MBE_ORDER sets the truncation order, n. If one wishes to use MBE(n) *without* charge embedding, then XPOL must still be set to TRUE but XPOL_MPOL_ORDER should be set to GAS. For EE-MBE(n), the user has the choice of Mulliken, Löwdin, or CHELPG (specified using XPOL_CHARGE_TYPE), and these charges can either be evaluated self-consistently or fixed at the outset. (In the latter case, they are gas-phase charges computed at the geometry that each monomer has in the cluster environment. As an alternative to point charges, density embedding is also available, in which Coulomb interactions are computed between proper monomer electron densities. The MBE with density embedding is equivalent to the original version of the fragment molecular orbital (FMO) [15]. (Many subsequent modifications to the FMO algorithm have been introduced [16, 43] but are not yet available in Q-CHEM. These include, in particular, the option to use point charges or approximate electron repulsion integrals to compute the Coulomb interactions between distant monomers [44, 45], which actually makes FMO more like EE-MBE at long range.) This density-embedded version of FMO is available in Q-CHEM by setting XPOL_MPOL_ORDER = DENSITY.

Researchers who use Q-CHEM's MBE code are asked to cite Ref. 46 and—if the MBCP(n) method is used—to cite Ref. 40 as well.

12.10.2 Job Control and Examples for Many-Body Calculations

The following *$rem* variables control MBE jobs.

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

MBE_ORDER

Controls the truncation order n for MBE.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

N	Order of MBE

RECOMMENDATION:

EE-MBE and FMO can be performed up to fifth and third order, respectively.

XPOL_FIX_MULLIKEN

Control to use self-consistent charge for EE-MBE.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Perform an EE-MBE without self-consistent charge.
FALSE	Perform an EE-MBE with self-consistent charge.

RECOMMENDATION:

The charges are derived from isolated monomers without self-consistent process. It is available to use with Mulliken charges, Löwdin charges and CHELPG charges

XPOL_FIX_TIP3P

Use charges corresponding to TIP3P water for EE-MBE.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Perform an EE-MBE with charges corresponding to TIP3P water.
FALSE	Do not perform an EE-MBE with charges corresponding to TIP3P water.

RECOMMENDATION:

Only available for water molecules

MANY_BODY_BSSE

Controls the type of many-body BSSE corrections.

TYPE:

STRING

DEFAULT:

MBCP

OPTIONS:

MBCP	Use many-body counterpoise correction.
VMFC	Use Valiron-Mayer function counterpoise correction.

RECOMMENDATION:

NONE.

MBE_BSSE_ORDER

Controls the order of many-body BSSE corrections.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n	Order of many-body BSSE corrections

RECOMMENDATION:

MBCP and VMFC can be performed up to third and fourth order, respectively.

Example 12.16 Example showing a 3-body EE-MBE calculation using self-consistent CHELPG charges.

```
$rem
sym_ignore        true
method            rimp2
basis             cc-pvdz
aux_basis         rimp2-cc-pvdz
n_frozen_core     fc
xpol              true
xpol_mpol_order   charges
xpol_charge_type  qchelpg
many_body_int     true
mbe_order         3
purecart          1111
```

```
scf_convergence   7
thresh            14
$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
$end
```

Example 12.17 Example showing a 3-body MBCP calculation.

```
$rem
sym_ignore        true
method            b3lyp
basis             sto-3g
xpol              true
xpol_mpol_order   gas
many_body_bsse    mbcp
mbe_bsse_order    3
purecart          1111
scf_convergence   7
thresh            14
$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
$end
```

References and Further Reading

[1] R. Z. Khaliullin, M. Head-Gordon, and A. T. Bell, *J. Chem. Phys.* **124**, 204105 (2006).

[2] P. R. Horn, E. J. Sundstrom, T. A. Baker, and M. Head-Gordon, *J. Chem. Phys.* **138**, 134119 (2013).

[3] R. Z. Khaliullin, E. A. Cobar, R. C. Lochan, A. T. Bell, and M. Head-Gordon, *J. Phys. Chem. A* **111**, 8753 (2007).

[4] R. Z. Khaliullin, A. T. Bell, and M. Head-Gordon, *J. Chem. Phys.* **128**, 184112 (2008).

[5] R. Z. Khaliullin, A. T. Bell, and M. Head-Gordon, *Chem. Eur. J* **15**, 851 (2009).

[6] W. Xie, L. Song, D. G. Truhlar, and J. Gao, *J. Chem. Phys.* **128**, 234108 (2008).

[7] L. D. Jacobson and J. M. Herbert, *J. Chem. Phys.* **134**, 094118 (2011).

[8] J. M. Herbert, L. D. Jacobson, K. U. Lao, and M. A. Rohrdanz, *Phys. Chem. Chem. Phys.* **14**, 7679 (2012).

[9] B. Jeziorski, R. Moszynski, and K. Szalewicz, *Chem. Rev.* **94**, 1887 (1994).

[10] K. Szalewicz, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2**, 254 (2012).

[11] L. D. Jacobson, R. M. Richard, K. U. Lao, and J. M. Herbert, *Annu. Rep. Comp. Chem.* **9**, 25 (2013).

[12] K. U. Lao and J. M. Herbert, *J. Phys. Chem. Lett.* **3**, 3241 (2012).

[13] K. U. Lao and J. M. Herbert, *J. Chem. Phys.* **139**, 034107 (2013).

[14] E. Dahlke and D. G. Truhlar, *J. Chem. Theory Comput.* **3**, 46 (2007).

[15] K. Kitaura, E. Ikeo, T. Asada, T. Nakano, and M. Uebayasi, *Chem. Phys. Lett.* **313**, 701 (1999).

[16] D. G. Fedorov and K. Kitaura, *J. Phys. Chem. A* **111**, 6904 (2007).

[17] H. Stoll, G. Wagenblast, and H. Preuss, *Theor. Chem. Acc.* **57**, 169 (1980).

[18] E. Gianinetti, M. Raimondi, and E. Tornaghi, *Int. J. Quantum Chem.* **60**, 157 (1996).

[19] T. Nagata, O. Takahashi, K. Saito, and S. Iwata, *J. Chem. Phys.* **115**, 3553 (2001).

[20] W. Z. Liang and M. Head-Gordon, *J. Phys. Chem. A* **108**, 3206 (2004).

[21] W. Z. Liang and M. Head-Gordon, *J. Chem. Phys.* **120**, 10379 (2004).

[22] E. A. Cobar, R. Z. Khaliullin, R. G. Bergman, and M. Head-Gordon, *Proc. Natl. Acad. Sci. USA* **104**, 6963 (2007).

[23] R. C. Lochan, R. Z. Khaliullin, and M. Head-Gordon, *Inorg. Chem.* **47**, 4032 (2008).

[24] J. Gao, *J. Chem. Phys.* **109**, 2346 (1998).

[25] W. Xie and J. Gao, *J. Chem. Theory Comput.* **3**, 1890 (2007).

[26] W. Xie, M. Orozco, D. G. Truhlar, and J. Gao, *J. Chem. Theory Comput.* **5**, 459 (2009).

[27] C. M. Breneman and K. B. Wiberg, *J. Comput. Chem.* **11**, 361 (1990).

[28] Z. C. Holden, R. M. Richard, and J. M. Herbert, *J. Chem. Phys.* **139**, 244108 (2013).

[29] B. Jeziorski et al., SAPT: A program for many-body symmetry-adapted perturbation theory calculations of intermolecular interaction energies, in *Methods and Techniques in Computational Chemistry: METECC-94*, edited by E. Clementi, volume B, chapter 3, pages 79–129, STEF, Cagliari, 1993.

[30] E. G. Hohenstein and C. D. Sherrill, *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2**, 304 (2012).

[31] K. U. Lao and J. M. Herbert, *J. Phys. Chem. A* **116**, 3042 (2012).

[32] J. A. Pople, R. Krishnan, H. B. Schlegel, and J. S. Binkley, *Int. J. Quantum Chem. Symp.* **13**, 225 (1979).

[33] H. L. Williams and C. F. Chabalowski, *J. Phys. Chem. A* , 646 (2001).

[34] A. J. Misquitta and K. Szalewicz, *Chem. Phys. Lett.* **357**, 301 (2002).

[35] K. U. Lao and J. M. Herbert, *J. Chem. Phys.* **140**, 044108 (2014).

[36] H. L. Williams, E. M. Mas, and K. Szalewicz, *J. Chem. Phys.* **103**, 7374 (1995).

[37] R. P. Steele, R. A. DiStasio, Jr., Y. Shao, J. Kong, and M. Head-Gordon, *J. Chem. Phys.* **125**, 074108 (2006).

[38] A. Hesselmann, *J. Phys. Chem. A* **115**, 11321 (2011).

[39] R. Podeszwa, K. Pernal, K. Patkowski, and K. Szalewicz, *J. Phys. Chem. Lett.* **1**, 550 (2010).

[40] R. M. Richard, K. U. Lao, and J. M. Herbert, *J. Phys. Chem. Lett.* **4**, 2674 (2013).

[41] R. M. Richard, K. U. Lao, and J. M. Herbert, *J. Chem. Phys.* **139**, 224102 (2013).

[42] M. Kamiya, S. Hirata, and M. Valiev, *J. Chem. Phys.* **128**, 074103 (2008).

[43] D. G. Fedorov and K. Kitaura, in *The Fragment Molecular Orbital Method: Practical Applications to Large Molecular Systems*, edited by D. G. Fedorov and K. Kitaura, chapter 2, page 5, CRC Press, Boca Rotan, FL, 2009.

[44] T. Nakano et al., *Chem. Phys. Lett.* **351**, 475 (2002).

[45] D. G. Fedorov, L. V. Slipchenko, and K. Kitaura, *J. Phys. Chem. A* **114**, 8742 (2010).

[46] R. M. Richard, K. U. Lao, and J. M. Herbert, *J. Chem. Phys.* **141**, 014108 (2014).

Appendix A

Geometry Optimization with Q-CHEM

A.1 Introduction

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 repeated calculation of energies, gradients and (possibly) Hessians at each optimization cycle until convergence is attained. The optimization step involves modifying the current geometry, utilizing current and previous energy, gradient and Hessian information to produce a revised geometry which is closer to the target stationary point than its predecessor was. The art of geometry optimization lies in calculating the step $\mathbf{h}$, the displacement from the starting geometry on that cycle, so as to converge in as few cycles as possible.

There are four main factors that influence the rate of convergence. These are:

- Initial starting geometry.
- Algorithm used to determine the step $\mathbf{h}$.
- Quality of the Hessian (second derivative) matrix.
- Coordinate system chosen.

The first of these factors is obvious: the closer the initial geometry is to the final converged geometry the fewer optimization cycles it will take to reach it. The second factor is again obvious: if a poor step $\mathbf{h}$ is predicted, this will obviously slow down the rate of convergence. The third factor is related to the second: the best algorithms make use of second derivative (curvature) information in calculating $\mathbf{h}$, and the better this information is, the better will be the predicted step. The importance of the fourth factor (the coordinate system) has been generally appreciated later on: a good choice of coordinates can enhance the convergence rate by an order of magnitude (a factor of 10) or more, depending on the molecule being optimized.

Q-CHEM includes a powerful suite of algorithms for geometry optimization written by Jon Baker and known collectively as OPTIMIZE. These algorithms have been developed and perfected over the past ten years and the code is robust and has been well tested. OPTIMIZE is a general geometry optimization package for locating both minima and transition states. It can optimize using Cartesian, *Z*-matrix coordinates or delocalized internal coordinates. The last of these are generated automatically from the Cartesian coordinates and are often found to be particularly effective. It also handles fixed constraints on distances, angles, torsions and out-of-plane bends, between any atoms in the molecule, whether or not the desired constraint is satisfied in the starting geometry. Finally it can freeze atomic positions, or any x, y, z Cartesian atomic coordinates.

OPTIMIZE is designed to operate with minimal user input. All that is required is the initial guess geometry, either in Cartesian coordinates (*e.g.*, from a suitable model builder such as HyperChem) or as a *Z*-matrix, the type of stationary point being sought (minimum or transition state) and details of any imposed constraints. All decisions as to the optimization strategy (what algorithm to use, what coordinate system to choose, how to handle the constraints) are made by OPTIMIZE.

Note particularly, that although the starting geometry is input in a particular coordinate system (as a *Z*-matrix, for example) these coordinates are not necessarily used during the actual optimization. The best coordinates for the majority of geometry optimizations are delocalized internals, and these will be tried first. Only if delocalized internals fail for some reason, or if conditions prevent them being used (*e.g.*, frozen atoms) will other coordinate systems be tried. If all else fails the default is to switch to Cartesian coordinates. Similar defaults hold for the optimization algorithm, maximum step size, convergence criteria, *etc.* You may of course override the default choices and force a particular optimization strategy, but it is not normally necessary to provide OPTIMIZE with anything other than the minimal information outlined above.

The heart of the OPTIMIZE package (for both minima and transition states) is Baker's eigenvector-following (EF) algorithm [1]. This was developed following the work of Cerjan and Miller [2] and Simons and coworkers [3, 4]. The Hessian mode-following option incorporated into this algorithm is capable of locating transition states by walking uphill from the associated minima. By following the lowest Hessian mode, the EF algorithm can locate transition states starting from any reasonable input geometry and Hessian.

An additional option available for minimization is Pulay's GDIIS algorithm [5], which is based on the well known DIIS technique for accelerating SCF convergence [6]. GDIIS must be specifically requested, as the EF algorithm is the default.

Although optimizations can be carried out in Cartesian or *Z*-matrix coordinates, the best choice, as noted above, is usually delocalized internal coordinates. These coordinates were developed by Baker *et al.* [7], and can be considered as a further extension of the natural internal coordinates developed by Pulay *et al.* [8, 9] and the redundant optimization method of Pulay and Fogarasi [10].

OPTIMIZE incorporates a very accurate and efficient Lagrange multiplier algorithm for constrained optimization. This was originally developed for use with Cartesian coordinates [11, 12] and can handle constraints that are not satisfied in the starting geometry. The Lagrange multiplier approach has been modified for use with delocalized internals [13]; this is much more efficient and is now the default. The Lagrange multiplier code can locate constrained transition states as well as minima.

A.2 Theoretical Background

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

$$E(\mathbf{x}_0 + \mathbf{h}) = E(\mathbf{x}_0) + \mathbf{h}^t \frac{dE(\mathbf{x}_0)}{d\mathbf{x}} + \frac{1}{2}\mathbf{h}^t \frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2}\mathbf{h} + \cdots \tag{A.1}$$

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

$$\frac{dE(\mathbf{x}_0 + \mathbf{h})}{d\mathbf{h}} = \frac{dE(\mathbf{x}_0)}{d\mathbf{x}} + \frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2}\mathbf{h} + \text{higher terms (ignored)} \tag{A.2}$$

from which

$$\mathbf{h} = -\mathbf{H}^{-1}\mathbf{g} \tag{A.3}$$

where

$$\frac{dE}{d\mathbf{x}} \equiv \mathbf{g} \text{ (gradient vector)}, \qquad \frac{d^2E}{d\mathbf{x_1}d\mathbf{x_2}} \equiv \mathbf{H} \text{ (Hessian matrix)} \tag{A.4}$$

Equation (A.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 all methodologies supported in Q-CHEM; however analytical second derivatives are not. Furthermore, even if they were, it would not necessarily be advantageous to use them as their evaluation is usually computationally demanding, and, efficient optimizations can in fact 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. (A.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. Diagonalization of the Hessian matrix can be considered to define a set of mutually orthogonal directions on the energy surface (the eigenvectors) together with the curvature along those 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 the others. 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. (A.3) is even a descent step (*i.e.*, a direction that will actually lower the energy). Similarly, for a transition state search, the Hessian must have one negative eigenvalue. Maintaining the Hessian eigenvalue structure is not difficult for minimization, but it can be a difficulty when trying to find a transition state.

In a diagonal Hessian representation the Newton-Raphson step can be written

$$\mathbf{h} = \sum \frac{-F_i}{b_i}\mathbf{u}_i \tag{A.5}$$

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.* [3], 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 then lead 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 an approximate Hessian is being improved by updating, then there is no guarantee that the Hessian eigenvalue structure will be retained 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 (one and only one negative eigenvalue) 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 initially and using the mechanics Hessian to begin the *ab initio* optimization) 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 [14], 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.

A.3 Eigenvector-Following (EF) Algorithm

The work of Cerjan and Miller [2], and later Simons and co-workers [3, 4], 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

$$\mathbf{h} = \sum_i \frac{-F_i}{(b_i - \lambda)} \mathbf{u}_i \tag{A.6}$$

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 [14]), along one of the local eigenmodes, depending on the value chosen for λ. References 2–4 all utilize the same basic approach of Eq. (A.6) but differ in the means of determining the value of λ.

The EF algorithm [1] utilizes the rational function approach presented in Refs. 4, yielding an eigenvalue equation of the form

$$\begin{pmatrix} \mathbf{H} & \mathbf{g} \\ \mathbf{g}^t & 0 \end{pmatrix} \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} \tag{A.7}$$

from which a suitable λ can be obtained. Expanding Eq. (A.7) yields

$$(\mathbf{H} - \lambda)\mathbf{h} + \mathbf{g} = 0 \tag{A.8}$$

and

$$\mathbf{g}^t\mathbf{h} = \lambda \tag{A.9}$$

In terms of a diagonal Hessian representation, Eq. (A.8) rearranges to Eq. (A.6), and substitution of Eq. (A.6) into the diagonal form of Eq. (A.9) gives

$$\sum_i \frac{-F_i^2}{(b_i - \lambda)} = \lambda \tag{A.10}$$

which can be used to evaluate λ iteratively.

The eigenvalues, λ, of the RFO equation Eq. (A.7) have the following important properties [4]:

- 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. (A.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

$$\begin{pmatrix} b_1 & F_1 \\ F_1 & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ 1 \end{pmatrix} \tag{A.11}$$

$$\begin{pmatrix} b_2 & & & F_2 \\ & \ddots & \mathbf{0} & \vdots \\ & \mathbf{0} & b_n & F_n \\ F_2 & \cdots & F_n & 0 \end{pmatrix} \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} \tag{A.12}$$

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. (A.11), which is always positive and approaches zero at convergence, and λ_n is the lowest eigenvalue of Eq. (A.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. (A.6) to give the final step

$$\mathbf{h} = \frac{-F_1}{(b_1 - \lambda_p)}\mathbf{u}_1 - \sum_{i=2}^{n} \frac{-F_i}{(b_i - \lambda_n)}\mathbf{u}_i \tag{A.13}$$

If this step is greater than the maximum allowed, it is scaled down. For minimization only one shift parameter, λ_n, would be used which would act on all modes.

In Eq. (A.11) and Eq. (A.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 perhaps 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. (A.11) is replaced by

$$\begin{pmatrix} b_k & F_k \\ F_k & 0 \end{pmatrix} \begin{pmatrix} h_k \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_k \\ 1 \end{pmatrix} \tag{A.14}$$

and Eq. (A.12) would now exclude the kth mode but include the lowest. 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. 1.

A.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 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 (second derivative) 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. Cartesians are an obvious choice as they can be defined for all systems and gradients and second derivatives are calculated directly in Cartesian coordinates. Unfortunately, Cartesians normally make a poor coordinate set for optimization as they are heavily coupled. Cartesians have been returning to favor later on because of their very general nature, 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 [12, 15]. 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 utilized 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 way a chemist thinks of molecular structure) 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 [16].

In 1979 Pulay *et al.* published a key paper, introducing what were termed natural internal coordinates into geometry optimization [8]. 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 used widely at a later stage. 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 [9]. For minimization, natural internals have become the coordinates of first choice [9, 12].

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 [10], who have developed a scheme for carrying out geometry optimization directly in the redundant coordinate space.

Baker *et al.* have developed a set of delocalized internal coordinates [7] which eliminate all of the above-mentioned difficulties. Building on some of the ideas in the redundant optimization scheme of Pulay and Fogarasi [10], 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 $\mathbf{B}$-matrix [17],

$$\Delta\mathbf{q} = \mathbf{B}\Delta\mathbf{X} \tag{A.15}$$

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 simply 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

$$\mathbf{G}(\mathbf{UR}) = (\mathbf{UR}) \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} \tag{A.16}$$

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. (A.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 [7].

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 can only increase.

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 [12]. This is because steps taken in internal coordinate space tend to be much larger when translated into Cartesian space, and, 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 metastable structure) and has both a good starting geometry and a decent Hessian, the user should carry out the optimization in Cartesian coordinates.

A.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 [11]. Baker's algorithm used both penalty functions and the classical method of Lagrange multipliers [18], 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 [12]. Q-CHEM includes the latest version of this algorithm, which has been modified to handle constraints directly in delocalized internal coordinates [13].

The essential problem in constrained optimization is to minimize a function of, for example, n variables $F(\mathbf{x})$ subject to a series of m constraints of the form $C_i(\mathbf{x}) = 0$, $i = l.\ m$. 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 of the 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 in a molecule) is given in Cartesians by $(R_{ij} - R_0) = 0$, with

$$R_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} \tag{A.17}$$

and R_0 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

$$L(\mathbf{x}, \lambda) = F(\mathbf{x}) - \sum_{i=1}^{m} \lambda_i C_i(\mathbf{x}) \tag{A.18}$$

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. (A.18) with respect to $\mathbf{x}$ and λ affords

$$\frac{dL(\mathbf{x}, \lambda)}{dx_j} = \frac{dF(\mathbf{x})}{dx_j} - \sum_{i=1}^{m} \lambda_i \frac{dC_i(\mathbf{x})}{dx_j} \tag{A.19}$$

$$\frac{dL(\mathbf{x}, \lambda)}{d\lambda_i} = -C_i(\mathbf{x}) \tag{A.20}$$

At a stationary point of the Lagrangian we have $\nabla\mathbf{L} = 0$, *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 ($\mathbf{x}$, λ) for which $\nabla\mathbf{L} = 0$ will give a possible solution to the constrained optimization problem in exactly the same way as finding an $\mathbf{x}$ for which $\mathbf{g} = \nabla\mathbf{F} = 0$ 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

$$\nabla^2\mathbf{L} = \begin{pmatrix} \frac{d^2L(\mathbf{x},\lambda)}{dx_j dx_k} & \frac{d^2L(\mathbf{x},\lambda)}{dx_j d\lambda_i} \\ \frac{d^2L(\mathbf{x},\lambda)}{dx_j d\lambda_i} & \frac{d^2L(\mathbf{x},\lambda)}{d\lambda_j d\lambda_i} \end{pmatrix} \tag{A.21}$$

where

$$\begin{aligned}\frac{d^2L(\mathbf{x},\lambda)}{dx_j dx_k} &= \frac{d^2F(\mathbf{x})}{dx_j dx_k} - \sum_{i=1}^{m} \lambda_i \frac{d^2C_i(\mathbf{x})}{dx_j dx_k} && \text{(A.22)}\\ \frac{d^2L(\mathbf{x},\lambda)}{dx_j d\lambda_i} &= \frac{-dC_i(\mathbf{x})}{dx_j} && \text{(A.23)}\\ \frac{d^2L(\mathbf{x},\lambda)}{d\lambda_j d\lambda_i} &= 0 && \text{(A.24)}\end{aligned}$$

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, given by Eq. (A.19), and Lagrangian second derivative matrix, given by Eq. (A.21), can be formed, and the optimization step calculated in a similar manner to that for a standard unconstrained optimization [11].

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, nm, has thus increased by m compared to the corresponding unconstrained case. The Lagrangian Hessian matrix, $\nabla^2\mathbf{L}$, has m extra modes compared to the standard (unconstrained) Hessian matrix, $\nabla^2\mathbf{F}$. 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 [11].

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. (A.11) is replaced by

$$\begin{pmatrix} b_1 & & & F_1 \\ & \ddots & \mathbf{0} & \vdots \\ & \mathbf{0} & b_m & F_m \\ F_1 & \cdots & F_m & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ \vdots \\ h_m \\ 1 \end{pmatrix} \tag{A.25}$$

and Eq. (A.12) by

$$\begin{pmatrix} b_{m+1} & & & F_{m+1} \\ & \ddots & \mathbf{0} & \vdots \\ & \mathbf{0} & b_{m+n} & F_{m+n} \\ F_{m+1} & \cdots & F_{m+n} & 0 \end{pmatrix} \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_{m+1} \\ \vdots \\ h_{m+n} \\ 1 \end{pmatrix} \tag{A.26}$$

where now the b_i are the eigenvalues of $\nabla^2\mathbf{L}$, with corresponding eigenvectors $\mathbf{u}_i$, and $F_i = \mathbf{u}_i^t\nabla\mathbf{L}$. Here Eq. (A.25) includes the m constraint modes along which a negative Lagrangian Hessian eigenvalue is required, and Eq. (A.26) includes all the other modes.

Equations (A.25) and (A.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 $m + l$.

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 Baker:1993b, 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, $\nabla \mathbf{C}_i$, are zero, and would result in rows and columns of zeros in the Lagrangian Hessian matrix.

A.6 Delocalized Internal Coordinates

We do not give further details of the optimization algorithms available in Q-CHEM for imposing constraints in Cartesian coordinates, as it is far simpler and easier to do this directly in delocalized internal coordinates.

At first sight it does not seem particularly straightforward to impose any constraints at all in delocalized internals, 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. In this instance consider a unit vector with unit component corresponding to the primitive internal (stretch, bend or torsion) that one wishes to keep constant. This vector is then projected on to the full set, $\mathbf{U}$, of active delocalized coordinates, normalized, and then all n, for example, delocalized internals are Schmidt orthogonalized in turn to this normalized, projected constraint vector. The last coordinate taken in the active space should drop out (since it will be linearly dependent on the other vectors and the constraint vector) leaving $(n - 1)$ active vectors and one constraint vector.

In more detail, the procedure is as follows (taken directly from Ref. 7). The initial (usually unit) constraint vector $\mathbf{C}$ is projected on to the set $\mathbf{U}$ of delocalized internal coordinates according to

$$\mathbf{C}^{\text{proj}} = \sum \langle \mathbf{C} | \mathbf{C} \mathbf{U}_k \mathbf{U}_k \rangle \, \mathbf{U}_k \tag{A.27}$$

where the summation is over all n 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

$$\mathbf{V} = \left\{ \mathbf{C}^{\text{proj}}, \mathbf{U}_k \; k = 1.n \right\} \tag{A.28}$$

This set of vectors is Schmidt orthogonalized according to the standard procedure,

$$\tilde{\mathbf{V}}_k = \alpha_k \left(\mathbf{V}_k - \sum_{l=1}^{k-1} \left\langle \mathbf{V}_k \middle| \mathbf{V}_k \tilde{\mathbf{V}}_l \tilde{\mathbf{V}}_l \right\rangle \tilde{\mathbf{V}}_l \right) \tag{A.29}$$

where the first vector taken, $\mathbf{V}_1$, is $\mathbf{C}^{\mathrm{proj}}$. The α_k in Eq. (A.29) is a normalization factor. As noted above, the last vector taken, $\mathbf{V}_{n+1} = \mathbf{U}_k$, 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.

In Cartesians, as already noted, standard internal constraints (bond distances, angles and torsions) are somewhat complicated non-linear functions of the x, y and z coordinates of the atoms involved. A torsion, for example, which involves four atoms, is a function of twelve different coordinates. In internals, on the other hand, each constraint is a coordinate in its own right and is therefore a simple linear function of just one coordinate (itself).

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

$$C_i(R_i) = R_i - R_0 \tag{A.30}$$

$$dC_i(R_i)/dR_i = 1; \quad dC_i(R_i)/dR_j = 0 \tag{A.31}$$

$$d^2C_i(R_i)/dR_i dR_j = 0 \tag{A.32}$$

where R_0 is the desired value of the constrained coordinate, and R_i is its current value. From Eq. (A.31) we see that the constraint normals, $dC_i(\mathbf{R})/dR_i$, are simply unit vectors and the Lagrangian Hessian matrix, Eq. (A.21), can be obtained from the normal Hessian matrix by adding m columns (and m rows) of, again, 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 [19], 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: Bohrs 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 simply be 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. Normally, unless the starting geometry is particularly poor in this regard, constraints are satisfied fairly early on in the optimization (and at more or less the same time for multiple constraints), and Lagrange multipliers only need to be used in the first half-dozen or so cycles of a constrained optimization in internal coordinates.

A.7 GDIIS

Direct inversion in the iterative subspace (DIIS) was originally developed by Pulay for accelerating SCF convergence [6]. Subsequently, Csaszar and Pulay used a similar scheme for geometry optimization, which they termed GDIIS [5]. 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 $\mathbf{x}_i$ generated in previous optimization cycles are linearly combined to find the "best" geometry on the current cycle

$$\mathbf{x}_n = \sum_{i=1}^{m} c_i \mathbf{x}_i \tag{A.33}$$

where the problem is to find the best values for the coefficients c_i.

If we express each geometry, $\mathbf{x}_i$, 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

$$\mathbf{r} = \sum c_i \mathbf{e}_i \tag{A.34}$$

and

$$\sum c_i = 1 \tag{A.35}$$

are satisfied, then the relation

$$\sum c_i \mathbf{x}_i = \mathbf{x}_f \tag{A.36}$$

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

$$\mathbf{e}_i = -\mathbf{H}^{-1}\mathbf{g}_i \tag{A.37}$$

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. (A.34), together with the constraint equation, Eq. (A.35), leads to a system of $(m+l)$ linear equations

$$\begin{pmatrix} B_{11} & \cdots & B_{1m} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ B_{m1} & \cdots & B_{mm} & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_m \\ -\lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \tag{A.38}$$

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. (A.38) are used to calculate an intermediate interpolated geometry

$$\mathbf{x}'_{m+1} = \sum c_i \mathbf{x}_i \tag{A.39}$$

and its corresponding interpolated gradient

$$\mathbf{g}'_{m+1} = \sum c_i \mathbf{g}_i \tag{A.40}$$

A new, independent geometry is generated from the interpolated geometry and gradient according to

$$\mathbf{x}_{m+1} = \mathbf{x}'_{m+1} - \mathbf{H}^{-1}\mathbf{g}'_{m+1} \tag{A.41}$$

Note: Convergence is theoretically guaranteed regardless of the quality of the Hessian matrix (as long as it is positive definite), and the original GDIIS algorithm used a static Hessian (*i.e.*, the original starting Hessian, often a simple unit matrix, remained unchanged during the entire optimization). However, updating the Hessian at each cycle generally results in more rapid convergence, and this is the default in OPTIMIZE.

Other modifications to the original method include limiting the number of previous geometries used in Eq. (A.33) and, subsequently, by neglecting earlier geometries, and eliminating any geometries more than a certain distance (default: 0.3 *a.u.*) from the current geometry.

References and Further Reading

[1] J. Baker, *J. Comput. Chem.* **7**, 385 (1986).

[2] C. J. Cerjan and W. H. Miller, *J. Chem. Phys.* **75**, 2800 (1981).

[3] J. Simons, P. Jørgensen, H. Taylor, and J. Ozment, *J. Phys. Chem.* **87**, 2745 (1983).

[4] A. Banerjee, N. Adams, J. Simons, and R. Shepard, *J. Phys. Chem.* **89**, 52 (1985).

[5] P. Csaszar and P. Pulay, *J. Mol. Struct. (Theochem)* **114**, 31 (1984).

[6] P. Pulay, *J. Comput. Chem.* **3**, 556 (1982).

[7] J. Baker, A. Kessi, and B. Delley, *J. Chem. Phys.* **105**, 192 (1996).

[8] P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs, *J. Am. Chem. Soc.* **101**, 2550 (1979).

[9] G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay, *J. Am. Chem. Soc.* **114**, 8191 (1992).

[10] P. Pulay and G. Fogarasi, *J. Chem. Phys.* **96**, 2856 (1992).

[11] J. Baker, *J. Comput. Chem.* **13**, 240 (1992).

[12] J. Baker and D. Bergeron, *J. Comput. Chem.* **14**, 1339 (1993).

[13] J. Baker, *J. Comput. Chem.* **18**, 1079 (1997).

[14] D. Poppinger, *Chem. Phys. Lett.* **35**, 550 (1975).

[15] J. Baker and W. J. Hehre, *J. Comput. Chem.* **12**, 606 (1991).

[16] H. B. Schlegel, *Theor. Chem. Acc.* **66**, 333 (1984).

[17] E. B. Wilson, J. C. Decius, and P. C. Cross, *Molecular Vibrations*, McGraw-Hill, New York, 1955.

[18] R. Fletcher, *Practial Methods of Optimization*, volume 2, Wiley, New York, 1981.

[19] S. Califano, *Vibrational States*, Wiley, London, 1976.

Appendix B

AOINTS

B.1 Introduction

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.

“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” [1].

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. 1–10), and it is not surprising that Q-CHEM’s AOINTS package is considered the most advanced of its kind.

B.2 Historical Perspective

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 [11] 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 [12] (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 [13].

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 [14–18] 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	11	ERIs with low L and low K
1969	Pople	13	ERIs with low L and high K
1976	Dupuis	19	Integrals with any L and low K
1978	McMurchie	20	Integrals with any L and low K
1982	Almlöf	21	Introduction of the direct SCF approach
1986	Obara	22	Integrals with any L and low K
1988	Head-Gordon	8	Integrals with any L and low K
1991	Gill	1, 6	Integrals with any L and any K
1994	White	14	J matrix in linear work
1996	Schwegler	18, 23	HF exchange matrix in linear work
1997	Challacombe	17	Fock matrix in linear work

B.3 AOINTS: Calculating ERIs with Q-CHEM

The area of molecular integrals with respect to Gaussian basis functions has recently been reviewed [2] 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

$$(\mathbf{ab}|\mathbf{cd}) = \int \phi_{\mathbf{a}}(\mathbf{r_1})\phi_{\mathbf{b}}(\mathbf{r_1})\theta(r_{12})\phi_{\mathbf{c}}(\mathbf{r_2})\phi_{\mathbf{d}}(\mathbf{r_2})d\mathbf{r_1}d\mathbf{r_2} \tag{B.1}$$

where the basis functions are contracted Gaussian's (CGTF)

$$\phi_{\mathbf{a}}(\mathbf{r}) = (x - A_x)^{a_x}(y - A_y)^{a_y}(z - A_z)^{a_z}\sum_{i=1}^{K_a} D_{ai}e^{-\alpha_i|\mathbf{r}-\mathbf{A}|^2} \tag{B.2}$$

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 B.1) between two charge distributions referred to as bras $(\mathbf{ab}|$ and kets $|\mathbf{cd})$.

B.4 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.

B.5 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 [7] 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.

B.6 Fundamental ERI

The fundamental ERI [2] and the basis of all ERI algorithms is usually represented

$$\begin{aligned}[\mathbf{0}]^{(0)} &= [ss|ss]^{(0)} \\ &= D_A D_B D_C D_D \int e^{-\alpha|\mathbf{r_1}-\mathbf{A}|^2} e^{-\beta|\mathbf{r_1}-\mathbf{B}|^2} \left[\frac{1}{r_{12}}\right] e^{-\gamma|\mathbf{r_2}-\mathbf{C}|^2} e^{-\delta|\mathbf{r}_2-\mathbf{D}|^2} d\mathbf{r_1} d\mathbf{r_2}\end{aligned} \tag{B.3}$$

which can be reduced to a one-dimensional integral of the form

$$[\mathbf{0}]^{(0)} = U(2\vartheta^2)^{1/2}\left(\frac{2}{\pi}\right)^{1/2} \int\limits_0^1 e^{-Tu^2} du \tag{B.4}$$

and can be efficiently computed using a modified Chebyshev interpolation scheme [5]. Equation (B.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.

B.7 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 [22, 24] and the newly developed tensor equations [3] are the basis for the modern approaches.

B.8 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

$$\begin{aligned}(ss|ss) &= \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} D_{Ai}D_{Bj}D_{Ck}D_{Dl} \\ &\quad \times \int e^{-\alpha_i|\mathbf{r_1}-\mathbf{A}|^2} e^{-\beta_j|\mathbf{r_1}-\mathbf{B}|^2}\left[\frac{1}{r_{12}}\right] e^{-\gamma_k|\mathbf{r_2}-\mathbf{C}|^2} e^{-\delta_l|\mathbf{r_2}-\mathbf{D}|^2} d\mathbf{r_1}d\mathbf{r_2} \\ &= \sum_{i=1}^{3}\sum_{j=1}^{3}\sum_{k=1}^{3}\sum_{l=1}^{3} [s_i s_j|s_k s_l] \end{aligned} \tag{B.5}$$

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 [25], 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 [13] is excellent for high contraction low angular momentum integral classes.

B.9 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 [26, 27] 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 [28]. This paved the way for the most significant breakthrough in the area with the linear scaling CFMM algorithm [14] leading to linear scaling DFT calculations [29].

Further breakthroughs have been made with traditional theory in the form of the QCTC [17, 30, 31] and ONX [18, 23] algorithms, while more radical approaches [15, 16] 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 [32] to become rate limiting in *ab initio* approaches in the medium term. However, divide-and-conquer techniques [33–36] and the recently developed quadratically convergent SCF algorithm [37] show great promise for reducing this problem.

B.10 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 [6] 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 [4]. 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.

B.11 More Efficient Hartree–Fock Gradient and Hessian Evaluations

Q-CHEM combines the Head-Gordon–Pople (HGP) method [8] and the COLD prism method [3] 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 II^x P term in the gradient evaluation, and the P II^{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 azt are shown below.

B.12 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

Basis Set	**AIX**			**Linux**		
	Gradient Evaluation: P II^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 II^{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 B.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.

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] P. M. W. Gill, M. Head-Gordon, and J. A. Pople, *J. Phys. Chem.* **94**, 5564 (1990).

[2] P. M. W. Gill, *Adv. Quantum Chem.* **25**, 142 (1994).

[3] T. R. Adams, R. D. Adamson, and P. M. W. Gill, *J. Chem. Phys.* **107**, 124 (1997).

[4] M. J. Frisch, B. G. Johnson, P. M. W. Gill, D. J. Fox, and R. H. Nobes, *Chem. Phys. Lett.* **206**, 225 (1993).

[5] P. M. W. Gill, B. G. Johnson, and J. A. Pople, *Int. J. Quantum Chem.* **40**, 745 (1991).

[6] P. M. W. Gill and J. A. Pople, *Int. J. Quantum Chem.* **40**, 753 (1991).

[7] P. M. W. Gill, B. G. Johnson, and J. A. Pople, *Chem. Phys. Lett.* **217**, 65 (1994).

[8] M. Head-Gordon and J. A. Pople, *J. Chem. Phys.* **89**, 5777 (1988).

[9] B. G. Johnson, P. M. W. Gill, and J. A. Pople, *Chem. Phys. Lett.* **206**, 229 (1993).

[10] B. G. Johnson, P. M. W. Gill, and J. A. Pople, *Chem. Phys. Lett.* **206**, 239 (1993).

[11] S. F. Boys, *Proc. Roy. Soc. Ser. A* **200**, 542 (1950).

[12] S. F. Boys, G. B. Cook, C. M. Reeves, and I. Shavitt, *Nature* **178** (1956).

[13] J. A. Pople and W. J. Hehre, *J. Comput. Phys.* **27**, 161 (1978).

[14] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *Chem. Phys. Lett.* **230**, 8 (1994).

[15] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill, *Chem. Phys. Lett.* **254**, 329 (1996).

[16] J. P. Dombroski, S. W. Taylor, and P. M. W. Gill, *J. Phys. Chem.* **100**, 6272 (1996).

[17] M. Challacombe and E. Schwegler, *J. Chem. Phys.* **106**, 5526 (1997).

[18] E. Schwegler and M. Challacombe, *J. Chem. Phys.* **105**, 2726 (1996).

[19] M. Dupuis, J. Rys, and H. F. King, *J. Chem. Phys.* **65**, 111 (1976).

[20] L. E. McMurchie and E. R. Davidson, *J. Comput. Phys.* **26**, 218 (1978).

[21] J. Almlöf, K. Faegri, and K. Korsell, *J. Comput. Chem.* **3**, 385 (1982).

[22] S. Obara and A. Saika, *J. Chem. Phys.* **84**, 3963 (1986).

[23] E. Schwegler and M. Challacombe, *J. Chem. Phys.* **106**, 9708 (1996).

[24] S. Obara and A. Saika, *J. Chem. Phys.* **89**, 1540 (1988).

[25] R. D. Adamson, Shell-pair economisation, Master's thesis, Massey University, Palmerston North, New Zealand, 1995.

[26] M. J. S. Dewar, *Org. Mass. Spect.* **28**, 305 (1993).

[27] M. J. S. Dewar, *The Molecular Orbital Theory of Organic Chemistry*, McGraw-Hill, New York, 1969.

[28] B. G. Johnson, *Development, Implementation, and Performance of Efficient Methodologies for Density Functional Calculations*, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1993.

[29] C. A. White, B. G. Johnson, P. M. W. Gill, and M. Head-Gordon, *Chem. Phys. Lett.* **253**, 268 (1996).

[30] M. Challacombe, E. Schwegler, and J. Almlöf, *J. Chem. Phys.* **104**, 4685 (1996).

[31] M. Challacombe, E. Schwegler, and J. Almlöf, Modern developments in Hartree-Fock theory: Fast methods for computing the Coulomb matrix, Technical report, University of Minnesota and Minnesota Supercomputer Institute, Minneapolis, MN, 1995.

[32] D. L. Strout and G. E. Scuseria, *J. Chem. Phys.* **102**, 8448 (1995).

[33] W. Yang, *Phys. Rev. A* **44**, 7823 (1991).

[34] W. Yang, *Phys. Rev. Lett.* **66**, 1438 (1991).

[35] W. Yang and T.-S. Lee, *J. Chem. Phys.* **103**, 5674 (1995).

[36] T.-S. Lee, D. M. York, and W. Yang, *J. Chem. Phys.* **105**, 2744 (1996).

[37] C. Ochsenfeld and M. Head-Gordon, *Chem. Phys. Lett.* **270**, 399 (1997).

Appendix C

Q-CHEM Quick Reference

C.1 Q-CHEM Text Input Summary

C.1.1 Keyword: *$molecule*

Four methods are available for inputing geometry information:

- *Z*-matrix (Angstroms and degrees):
  ```
  $molecule
     {Z-matrix}
     {blank line, if parameters are being used}
     {Z-matrix parameters, if used}
  $end
  ```

- Cartesian Coordinates (Angstroms):
  ```
  $molecule
     {Cartesian coordinates}
     {blank line, if parameter are being used}
     {Coordinate parameters, if used}
  $end
  ```

- Read from a previous calculation:
  ```
  $molecule
     read
  $end
  ```

Keyword	**Description**
$molecule	Contains the molecular coordinate input (input file requisite).
$rem	Job specification and customization parameters (input file requisite).
$end	Terminates each keyword section.
$basis	User-defined basis set information (see Chapter 7).
$comment	User comments for inclusion into output file.
$ecp	User-defined effective core potentials (see Chapter 8).
$empirical_dispersion	User-defined van der Waals parameters for DFT dispersion correction.
$external_charges	External charges and their positions.
$force_field_params	Force field parameters for QM/MM calculations (see Section‘11.3).
$intracule	Intracule parameters (see Chapter 10).
$isotopes	Isotopic substitutions for vibrational calculations (see Chapter 10).
$localized_diabatization	Information for mixing together multiple adiabatic states into diabatic states (see Chapter 10).
$multipole_field	Details of a multipole field to apply.
$nbo	Natural Bond Orbital package.
$occupied	Guess orbitals to be occupied.
$opt	Constraint definitions for geometry optimizations.
$pcm	Special parameters for polarizable continuum models (see Section 11.2.3).
$plots	Generate plotting information over a grid of points (see Chapter 10).
$qm_atoms	Specify the QM region for QM/MM calculations (see Section 11.3).
$solvent	Additional parameters and variables for implicit solvent models (see Section 11.2).
$svp	Special parameters for the SS(V)PE module.
$svpirf	Initial guess for SS(V)PE) module.
$van_der_waals	User-defined atomic radii for Langevin dipoles solvation (see Chapter 10).
$xc_functional	Details of user-defined DFT exchange-correlation functionals.
$cdft	Special options for the constrained DFT method as implemented.
$efp_fragments	Specifies labels and positions of EFP fragments.
$efp_params	Contains user-defined parameters for effective fragments.
$eom_user_guess	Contains user-defined guess for EOM-CC calculations.
$complex_ccman	Contains parameters for complex-scaled and CAP-augmented EOM-CC calculations.

Table C.1: Q-CHEM user input section keywords. See the *$QC/samples* directory with your release for specific examples of Q-CHEM input using these keywords.

Note: (1) Users are able to enter keyword sections in any order.
(2) Each keyword section must be terminated with the *$end* keyword.
(3) Not all keywords have to be entered, but *$rem* and *$molecule* are compulsory.
(4) Each keyword section will be described below.
(5) The entire Q-CHEM input is case–insensitive.
(6) Multiple jobs are separated by the string `@@@` on a single line.

- Read from a file:

```
$molecule
   read filename
$end
```

C.1.2 Keyword: *$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
```

C.1.3 Keyword: *$basis*

The format for the user–defined basis section is as follows:

$$\begin{array}{llllll}
\multicolumn{6}{l}{\texttt{\$basis}} \\
 & X & 0 & & & \\
 & L & K & scale & & \\
 & \alpha_1 & C_1^{L_{min}} & C_1^{L_{min}+1} & \ldots & C_1^{L_{max}} \\
 & \alpha_2 & C_2^{L_{min}} & C_2^{L_{min}+1} & \ldots & C_2^{L_{max}} \\
 & \vdots & \vdots & \vdots & \ddots & \vdots \\
 & \alpha_K & C_K^{L_{min}} & C_K^{L_{min}+1} & \ldots & C_K^{L_{max}} \\
\multicolumn{6}{l}{\star\star\star\star} \\
\multicolumn{6}{l}{\texttt{\$end}}
\end{array}$$

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.

C.1.4 Keyword: *$comment*

Note that the entire input deck is echoed to the output file, thus making the *$comment* keyword largely redundant.

```
$comment
   User comments - copied to output file
$end
```

C.1.5 Keyword: *$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.

C.1.6 Keyword: *$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.

C.1.7 Keyword: *$external_charges*

All input should be given in atomic units.

Update: While charges should indeed be listed in atomic units, the units for distances depend on the user input. If the structure is specified in Angstroms (the default), the coordinates for external charges should also be in Angstroms. If the structure is specified in atomic units, the coordinates for external charges should also be in atomic units. (See INPUT_BOHR.)

```
$external_charges
   x-coord1   y-coord1   z-coord1   charge1
   x-coord2   y-coord2   z-coord2   charge2
$end
```

C.1.8 Keyword: *$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
```

C.1.9 Keyword: *$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
```

C.1.10 Keyword: *$multipole_field*

Multipole fields are all in atomic units.

```
$multipole_field
   field_component1   value1
   field_component2   value2
   ...
$end
```

C.1.11 Keyword: *$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
```

C.1.12 Keyword: *$occupied*

```
$occupied
   1  2  3  4 ...  nalpha
   1  2  3  4 ...  nbeta
$end
```

C.1.13 Keyword: *$opt*

Note that units are in Angstroms 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
```

C.1.14 Keyword: *$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
```

C.1.15 Keyword: *$svpirf*

```
$svpirf
    <# point> <x point> <y point> <z point> <charge> <grid weight>
    <# point> <x normal> <y normal> <z normal>
$end
```

C.1.16 Keyword: *$plots*

`$plots`

One comment line

Specification of the 3–D mesh of points on 3 lines:

N_x x_{min} x_{max}
N_y y_{min} y_{max}
N_z z_{min} z_{max}

A line with 4 integers indicating how many things to plot:

N_{MO} N_{Rho} N_{Trans} N_{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`

C.1.17 Keyword: *$localized_diabatization*

```
$plots
```

One comment line.

One line with an an array of adiabatic states to mix together.

$< adiabat1 >$ $< adiabat2 >$ $< adiabat3 > \ldots$

```
$end
```

Note: We count adiabatic states such that the first excited state is $< adiabat >= 1$, the fifth is $< adiabat >= 5$, and so forth.

C.1.18 Keyword *$van_der_waals*

Note: all radii are given in angstroms.

```
$van_der_waals
   1
   atomic_number   VdW_radius
$end
```

(alternative format)

```
$van_der_waals
   2
   sequential_atom_number   VdW_radius
$end
```

C.1.19 Keyword: *$xc_functional*

```
$xc_functional
   X  exchange_symbol  coefficient
   X  exchange_symbol  coefficient
   ...
   C  correlation_symbol  coefficient
   C  correlation_symbol  coefficient
   ...
   K  coefficient
$end
```

C.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 angstroms; *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
```

C.2.1 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
```

C.3 *$rem* Variable List

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.

C.3.1 General

BASIS
BASIS_LIN_DEP_THRESH
EXCHANGE
CORRELATION
ECP
JOBTYPE
METHOD
PURECART

C.3.2 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

C.3.3 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

C.3.4 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

C.3.5 Correlated Methods

AO2MO_DISK
CD_ALGORITHM
CORE_CHARACTER
CORRELATION
MEM_STATIC
MEM_TOTAL
N_FROZEN_CORE
N_FROZEN_VIRTUAL
PRINT_CORE_CHARACTER

C.3.6 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, 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_FULLRESPONSE
CC_REF_PROP_TE

C.3.7 Perfect pairing, Coupled cluster valence bond, and related methods

CCVB_METHOD
GVB_N_PAIRS
GVB_ORB_MAX_ITER
GVB_ORB_CONV
GVB_AMP_SCALE
GVB_PRINT
CCVB_GUESS
GVB_LOCAL
GVB_RESTART
GVB_ORB_SCALE
GVB_DO_SANO

C.3.8 Excited States: CIS, TDDFT, SF-XCIS and SOS-CIS(D)

CIS_CONVERGENCE
CIS_GUESS_DISK_TYPE
CIS_RELAXED_DENSITY
CIS_STATE_DERIV
MAX_CIS_CYCLES
XCIS
CIS_GUESS_DISK
CIS_N_ROOTS
CIS_SINGLETS
CIS_TRIPLETS
RPA
SPIN_FLIP_XCIS

C.3.9 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_THRESHOLD
EOM_NGUESS_DOUBLES
EOM_DOEXDIAG
EOM_PRECONV_SINGLES
EOM_IPEA_FILTER
CC_REST_AMPL
CC_EOM_PROP
CC_STATE_TO_OPT
CC_EOM_PROP_TE
EOM_DAVIDSON_MAXVECTORS
EOM_DAVIDSON_MAX_ITER
EOM_NGUESS_SINGLES
EOM_PRECONV_DOUBLES
EOM_PRECONV_SD
EOM_FAKE_IPEA
CC_REST_TRIPLES
CC_TRANS_PROP
CC_EOM_PROP
CC_FULLRESPONSE

C.3.10 Geometry Optimizations

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

C.3.11 Vibrational Analysis

DORAMAN
CPSCF_NSEG
FDIFF_STEPSIZE
IDERIV
ISOTOPES
JOBTYPE
VIBMAN_PRINT
ANHAR
VCI
FDIFF_DER
MODE_COUPLING
IGNORE_LOW_FREQ
FDIFF_STEPSIZE_QFF

C.3.12 Reaction Coordinate Following

JOBTYPE
RPATH_COORDS
RPATH_DIRECTION
RPATH_MAX_CYCLES
RPATH_MAX_STEPSIZE
RPATH_PRINT
RPATH_TOL_DISPLACEMENT

C.3.13 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

C.3.14 Wavefunction Analysis and Molecular Properties

CHEMSOL
CHEMSOL_EFIELD
CHEMSOL_NN
CHEM_SOL_PRINT
CIS_RELAXED_DENSITY
IGDESP
INTRACULE
MULTIPOLE_ORDER
NBO
POP_MULLIKEN
PRINT_DIST_MATRIX
PRINT_ORBITALS
READ_VDW
SOLUTE_RADIUS
SOLVENT_DIELECTRIC
STABILITY_ANALYSIS
WAVEFUNCTION_ANALYSIS
WRITE_WFN

C.3.15 Symmetry

CC_SYMMETRY
SYM_IGNORE
SYMMETRY
SYMMETRY_DECOMPOSITION
SYM_TOL

C.3.16 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

C.3.17 Resource Control

MEM_TOTAL
MEM_STATIC
AO2MO_DISK
CC_MEMORY
INTEGRALS_BUFFER
MAX_SUB_FILE_NUM
DIRECT_SCF

C.3.18 Alphabetical Listing

MBE_ORDER

Controls the truncation order n for MBE.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

N Order of MBE

RECOMMENDATION:

EE-MBE and FMO can be performed up to fifth and third order, respectively.

SAPT_DISP_CORR

Request an empirical dispersion potential instead of calculating $E^{(2)}_{disp}$ and $E^{(2)}_{exch\text{-}disp}$ directly

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Use a dispersion force field.
FALSE	calculate $E^{(2)}_{disp}$ and $E^{(2)}_{exch\text{-}disp}$.

RECOMMENDATION:

Using dispersion potentials reduces the scaling from $O(N^5)$ to $O(N^3)$ with respect to monomer size.

SAPT_DISP_VERSION

Controls which dispersion potential is used for SAPT

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1	Use the dispersion potentials from Hesselmann [1, 2].
2	Use the dispersion potentials from Podeszwa. [3, 4].

RECOMMENDATION:

The Hesselmann ("+D") correction corresponds to “generation 1” of XSAPT(KS)+D [2], while the Podeszwa ("+D2") potential is “generation 2” [4]. The two potentials were parameterized in very different ways. The latter was fit directly to $E^{(2)}_{disp}$ and $E^{(2)}_{exch\text{-}disp}$ components calculated using SAPT(DFT), and generally performs better [4].

SAPT_PRINT

Controls level of printing in SAPT.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

N Integer print level

RECOMMENDATION:

Larger values generate additional output.

RISAPT

Requests an RI-SAPT calculation

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Compute four-index integrals using the RI approximation.

FALSE Do not use RI.

RECOMMENDATION:

Set to TRUE if an appropriate auxiliary basis set is available, as RI-SAPT is much faster and affords negligible errors (as compared to ordinary SAPT) if the auxiliary basis set is matched to the primary basis set. (The former must be specified using AUX_BASIS.)

SAPT_DSCF

Request the $\delta E_{int}^{\mathrm{HF}}$ correction

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Evaluate this correction.

FALSE Omit this correction.

RECOMMENDATION:

Evaluating the $\delta E_{int}^{\mathrm{HF}}$ correction requires an SCF calculation on the entire (super)system. This corrections effectively yields a "Hartree-Fock plus dispersion" estimate of the interaction energy.

SAPT_EXCHANGE

Selects the type of first-order exchange that is used in a SAPT calculation.

TYPE:

STRING

DEFAULT:

S_SQUARED

OPTIONS:

S_SQUARED	Compute first order exchange in 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.

SAPT_BASIS

Controls the MO basis used for SAPT corrections.

TYPE:

STRING

DEFAULT:

MONOMER

OPTIONS:

MONOMER	Monomer-centered basis set (MCBS).
DIMER	Dimer-centered basis set (DCBS).
PROJECTED	Projected basis set.

RECOMMENDATION:

The DCBS is more costly than the MCBS and can only be used with XPOL_MPOL_ORDER=GAS (*i.e.*, it is not available for use with XPol). The PROJECTED choice is an efficient compromise that is available for use with XPol.

SAPT_CPHF

Requests that the second-order corrections $E^{(2)}_{ind}$ and $E^{(2)}_{exch\text{-}ind}$ be replaced by their infinite-order “response” analogues, $E^{(2)}_{ind,resp}$ and $E^{(2)}_{exch\text{-}ind,resp}$.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Evaluate the response corrections and use $E^{(2)}_{ind,resp}$ and $E^{(2)}_{exch\text{-}ind,resp}$
FALSE	Omit these corrections and use $E^{(2)}_{ind}$ and $E^{(2)}_{exch\text{-}ind}$.

RECOMMENDATION:

Computing the response corrections requires solving CPHF equations for pair of monomers, which is somewhat expensive but may improve the accuracy when the monomers are polar.

SAPT_ORDER

Selects the order in perturbation theory for a SAPT calculation.

TYPE:

STRING

DEFAULT:

SAPT2

OPTIONS:

SAPT1	First order SAPT.
SAPT2	Second order SAPT.
ELST	First-order Rayleigh-Schrödinger perturbation theory.
RSPT	Second-order Rayleigh-Schrödinger perturbation theory.

RECOMMENDATION:

SAPT2 is the most meaningful.

MANY_BODY_BSSE

Controls the type of many-body BSSE corrections.

TYPE:

STRING

DEFAULT:

MBCP

OPTIONS:

MBCP	Use many-body counterpoise correction.
VMFC	Use Valiron-Mayer function counterpoise correction.

RECOMMENDATION:

NONE.

MBE_BSSE_ORDER

Controls the order of many-body BSSE corrections.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n	Order of many-body BSSE corrections

RECOMMENDATION:

MBCP and VMFC can be performed up to third and fourth order, respectively.

SAPT

Requests a SAPT calculation.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Run a SAPT calculation.
FALSE	Do not run SAPT.

RECOMMENDATION:

If SAPT is set to TRUE, one should also specify XPOL=TRUE and XPOL_MPOL_ORDER=GAS.

XPOL_MPOL_ORDER

Controls the order of multipole expansion that describes electrostatic interactions.

TYPE:

STRING

DEFAULT:

CHARGES

OPTIONS:

GAS	No electrostatic embedding; monomers are in the gas phase.
CHARGES	Charge embedding.
DENSITY	Density embedding.

RECOMMENDATION:

Should be set to GAS to do a dimer SAPT calculation (see Section 12.8).

XPOL_PRINT

Print level for XPol calculations.

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

N	Integer print level

RECOMMENDATION:

Higher values prints more information

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.

XPOL_CHARGE_TYPE

Controls the type of atom-centered embedding charges for XPol calculations.

TYPE:

STRING

DEFAULT:

QLOWDIN

OPTIONS:

QLOWDIN	Löwdin charges.
QMULLIKEN	Mulliken charges.
QCHELPG	CHELPG charges.

RECOMMENDATION:

Problems with Mulliken charges in extended basis sets can lead to XPol convergence failure. Löwdin charges tend to be more stable, and CHELPG charges are both robust and provide an accurate electrostatic embedding. However, CHELPG charges are more expensive to compute, and analytic energy gradients are not yet available for this choice.

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 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 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 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 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 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 default unless convergence problems are encountered. The threshold value 10^{-n} should always be smaller than the convergence criterion ADC_DAVIDSON_CONV.

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_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_EXTENDED

Deprecated! Activates the ADC(2)-x variant. This option is ignored unless ADC_ORDER is set to 2 and METHOD is not set.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	Activate ADC(2)-x.
FALSE	Do an ADC(2)-s calculation.

RECOMMENDATION:

Use METHOD instead.

ADC_NGUESS_DOUBLES

Controls the number of excited state guess vectors which are double excitations.

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. 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:

ADC_ORDER

Deprecated! Controls the order in perturbation theory of ADC. Only read, if METHOD is not set.

TYPE:

INTEGER

DEFAULT:

None

OPTIONS:

0 Activate ADC(0).
1 Activate ADC(1).
2 Activate ADC(2)-s or ADC(2)-x.
3 Activate ADC(3).

RECOMMENDATION:

Use METHOD instead.

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 Debug1: more status information, extended information on timings.
...

RECOMMENDATION:

ADC_PROP_ES2ES

Controls the calculation of transition properties between excited states (currently only transition dipole moments and oscillator strengths).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate state-to-state transition properties.
FALSE Do not compute transition properties between excited states.

RECOMMENDATION:

Set to TRUE, if state-to-state properties are required.

ADC_PROP_ES

Controls the calculation of excited state properties (currently only dipole moments).

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Calculate excited state properties.

FALSE Do not compute state properties.

RECOMMENDATION:

Set to TRUE, if properties are required or attachment/detachment densities of the excited states should be plotted.

ADC_SOS

Activates the SOS models in second order ADC calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE Perform a standard ADC calculation

OPTIONS:

TRUE Employ the SOS model, if METHOD is set to ADC(2) or ADC(2)-x.

FALSE Do not use the SOS model.

RECOMMENDATION:

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.

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 [5] for examples and discussion.

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.

RECOMMENDATION:

This variable need only be specified in the event that velocities are not specified explicitly in a *$velocity* section.

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.

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.

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 utilize in the dipole autocorrelation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute dipole autocorrelation function.
$1 \leq n \leq$ AIMD_STEPS	Compute dipole autocorrelation function for last n timesteps of the trajectory.

RECOMMENDATION:

If the DACF is desired, set equal to AIMD_STEPS.

AIMD_NUCL_SAMPLE_RATE

The rate at which sampling is performed for the velocity and/or dipole autocorrelation 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 autocorrelation 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/DACF are desired.

AIMD_NUCL_VACF_POINTS

Number of time points to utilize in the velocity autocorrelation function for an AIMD trajectory

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute velocity autocorrelation function.
$1 \leq n \leq$ AIMD_STEPS	Compute velocity autocorrelation function for last n time steps of the trajectory.

RECOMMENDATION:

If the VACF is desired, set equal to AIMD_STEPS.

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 wavefunction.
$-n$	Generates n random geometries sampled from the harmonic vibrational wavefunction.

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_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.

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.

AO2MO_DISK

Sets the scratch space size for individual program modules

TYPE:

INTEGER

DEFAULT:

2000 2 Gb

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

The minimum disk requirement of RI-CIS(D) is approximately $3SOVXD$. Again, the batching scheme will become more efficient with more available disk space. There is no simple formula for SOS-CIS(D) and SOS-CIS(D_0) disk requirement. However, because the disk space is abundant in modern computers, this should not pose any problem. Just put the available disk space size in this case. The actual disk usage information will also be printed in the output file.

AO2MO_DISK

Sets the amount of disk space (in megabytes) available for MP2 calculations.

TYPE:

INTEGER

DEFAULT:

2000 Corresponding to 2000 Mb.

OPTIONS:

n User-defined number of megabytes.

RECOMMENDATION:

Should be set as large as possible, discussed in Section 5.3.1.

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.

AUX_BASIS

Specifies the type of auxiliary basis to be used in a method that involves RI-fitting procedures.

TYPE:

STRING

DEFAULT:

No default is assigned. Must be defined in the input

OPTIONS:

Symbol. Choose among the auxiliary basis sets collected in the qchem qcaux basis library

RECOMMENDATION:

Try a few different types of aux bases first

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 7.

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:

FOPPROJECTION (when DUAL_BASIS_ENERGY=false)
OVPROJECTION (when DUAL_BASIS_ENERGY=true)

OPTIONS:

FOPPROJECTION	Construct the Fock matrix in the second basis
OVPROJECTION	Projects MO's 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

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 7.
Mixed	Use a mixture of basis sets (see Chapter 7).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

BOYSCALC

Specifies the Boys localized orbitals are to be calculated

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not perform localize the occupied space.
1	Allow core-valence mixing in Boys localization.
2	Localize core and valence separately.

RECOMMENDATION:

None

BOYS_CIS_NUMSTATE

Define how many states to mix with Boys localized diabatization.

TYPE:

INTEGER

DEFAULT:

0	Do not perform Boys localized diabatization.

OPTIONS:

1 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 n point charges are used.

RECOMMENDATION:

None

CAGE_RADIUS

Defines radius of the charged cage.

TYPE:

INTEGER

DEFAULT:

225

OPTIONS:

n radius is n/100 Å.

RECOMMENDATION:

None

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 nonlocal 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_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 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_DOV_THRESH

Specifies the 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:

DOUBLE

DEFAULT:

2502 Corresponding to 0.25

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.

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 $abc \times 10^{-de}$, *e.g.*, 2502 corresponds to 0.25

RECOMMENDATION:

Increase to 0.25, 0.5 or 0.75 for non convergent coupled-cluster calculations.

CC_DO_DYSON_EE

Whether excited 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

Whether ground 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_EOM_PROP

Whether or not the non-relaxed (expectation value) one-particle EOM-CCSD target state 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 electron density, and the total $\langle R^2 \rangle = \langle X^2 \rangle + \langle Y^2 \rangle + \langle Z^2 \rangle$ (in atomic units). Incompatible with JOBTYPE=FORCE, OPT, 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. Sometimes the equations for left and right eigenvectors converge to different sets of target states. In this case, the simultaneous iterations of left and right vectors will diverge, and the properties for several or all the target states may be incorrect! The problem can be solved by varying the number of requested states, specified with XX_STATES, or the number of guess vectors (EOM_NGUESS_SINGLES). 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 package by specifying the state with CC_STATE_TO_OPT and requesting NBO=TRUE and CC_EOM_PROP=TRUE.

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_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_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_FNO_USEPOP

Selection of the truncation scheme

TYPE:

INTEGER

DEFAULT:

1 OCCT

OPTIONS:

0 POVO

RECOMMENDATION:

None

CC_FULLRESPONSE

Fully relaxed properties (including orbital relaxation terms) will be computed. The variable CC_EOM_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. Only available for EOM/CI methods for which analytic gradients are available.

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 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 wavefunction? 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_PRECONV_FZ

In active space methods, whether to pre-converge other wavefunction 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:

$0 \rightarrow 7$ higher values can lead to deforestation…

RECOMMENDATION:

Increase if you need more output and don’t like trees

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 $\langle S^2 \rangle$. 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 $\langle S^2 \rangle$. 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 electron density, and the total $\langle R^2 \rangle = \langle X^2 \rangle + \langle Y^2 \rangle + \langle Z^2 \rangle$ (in atomic units). Incompatible with JOBTYPE=FORCE, OPT, FREQ.

TYPE:

LOGICAL

DEFAULT:

FALSE (no one-particle properties will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations need to be solved (lambda CCSD equations) for properties with the cost approximately the same as CCSD equations. Use 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:

INTEGER

DEFAULT:

1

OPTIONS:

0 Do apply restrictions

1 Do not apply 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_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_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

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.

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 for the EOM-CCSD target states will be calculated. By default, the transition dipole moment is calculated between the CCSD reference and the EOM-CCSD target states. In order to calculate transition dipole moment 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 (no transition dipole and oscillator strength will be calculated)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

Additional equations (for the left EOM-CCSD eigenvectors plus lambda CCSD equations in case if 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 $\vec{S}^{-1/2}$, 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 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_CRASHONFAIL

Whether the calculation should crash or not if the constraint iterations do not converge.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE	Crash if constraint iterations do not converge.
FALSE	Do not crash.

RECOMMENDATION:

Use default.

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_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 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 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:

Use 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 5.3.1).

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 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 non-hydrogen atoms.

TYPE:

INTEGER

DEFAULT:

NONE

OPTIONS:

N Corresponding to a number of points in a Lebedev grid (see Section 4.3.12.

RECOMMENDATION:

None.

CHELPG_HEAD

Sets the "head space" [6] (radial extent) of the CHELPG grid.

TYPE:

INTEGER

DEFAULT:

30

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 Corresponding to a number of points in a Lebedev grid.

RECOMMENDATION:

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.

CHOLESKY_TOL

Tolerance of Cholesky decomposition of two-electron integrals

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n to define tolerance of 10^{-n}

RECOMMENDATION:

2 - qualitative calculations, 3 - appropriate for most cases, 4 - quantitative (error in total energy typically less than 1e-6 hartree)

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_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:

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 TD-DFT 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) to hold a temporary array whose minimum size is OV × 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_MOMENTS

Controls calculation of excited-state (CIS or TDDFT) multipole moments

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE (or 0)

OPTIONS:

FALSE (or 0) Do not calculate excited-state moments.

TRUE (or 1) 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.

CIS_MULLIKEN

Controls Mulliken and Löwdin population analyses for excited-state particle and hole density matrices.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not perform particle/hole population analysis.

TRUE (or 1) 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 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_RELAXED_DENSITY

Use the relaxed CIS density for attachment/detachment density analysis

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 singlet or triplet in unrestricted calculations.

TYPE:

INTEGER

DEFAULT:

120

OPTIONS:

None

RECOMMENDATION:

If set to 120, the states with $\langle \hat{S}^2 \rangle > 1.20$ are treated as triplet states, with other states are treated as singlets.

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_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 no change order during an optimization

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

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.

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 5.3.2 for details).

RECOMMENDATION:

Use default, unless performing calculations on molecules with heavy elements.

CORRELATION

Specifies the correlation level of theory, either DFT or wavefunction-based.

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

None	No Correlation.
VWN	Vosko-Wilk-Nusair parameterization #5
LYP	Lee-Yang-Parr
PW91, PW	GGA91 (Perdew)
PW92	LSDA 92 (Perdew and Wang) [7]
PK09	LSDA (Proynov-Kong) [8]
LYP(EDF1)	LYP(EDF1) parameterization
Perdew86, P86	Perdew 1986
PZ81, PZ	Perdew-Zunger 1981
PBE	Perdew-Burke-Ernzerhof 1996
TPSS	The correlation component of the TPSS functional
B94	Becke 1994 correlation in fully analytic form
B94hyb	Becke 1994 correlation as above, but readjusted for use only within the hybrid scheme BR89B94hyb
PK06	Proynov-Kong 2006 correlation (known also as “tLap”
(B88)OP	OP correlation [9], optimized for use with B88 exchange
(PBE)OP	OP correlation [9], optimized for use with PBE exchange
Wigner	Wigner
MP2	
Local_MP2	Local MP2 calculations (TRIM and DIM models)
CIS(D)	MP2-level correction to CIS for excited states
MP3	
MP4SDQ	
MP4	
CCD	
CCD(2)	
CCSD	
CCSD(T)	
CCSD(2)	
QCISD	
QCISD(T)	
OD	
OD(T)	
OD(2)	
VOD	
VOD(2)	
QCCD	
VQCCD	

RECOMMENDATION:

Consult the literature and reviews for guidance.

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
CCSD(T)	CCMAN and CCMAN2
CCSD(2)	CCMAN
CCSD(fT)	CCMAN
CCSD(dT)	CCMAN
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	Do not solve the CPSCF equations in segments.
n	User-defined. Use n segments when solving the CPSCF equations.

RECOMMENDATION:

Use default.

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: CUTOFF=CUTOCC/100

DEFAULT:

50

OPTIONS:

0-200

RECOMMENDATION:

None

CUTVIR

Specifies virtual orbital cutoff

TYPE:

INTEGER: CUTOFF=CUTVIR/100

DEFAULT:

0	No truncation

OPTIONS:

0-100

RECOMMENDATION:

None

CVGLIN

Convergence criterion for solving linear equations by the conjugate gradient iterative method (relevant if LINEQ=1 or 2).

TYPE:

FLOAT

DEFAULT:

1.0E-7

OPTIONS:

Real number specifying the actual criterion.

RECOMMENDATION:

The default value should be used unless convergence problems arise.

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.

DFTVDW_ALPHA1

Parameter in XDM calculation with higher-order terms

TYPE:

INTEGER

DEFAULT:

83

OPTIONS:

10-1000

RECOMMENDATION:

none

DFTVDW_ALPHA2

Parameter in XDM calculation with higher-order terms.

TYPE:

INTEGER

DEFAULT:

155

OPTIONS:

10-1000

RECOMMENDATION:

none

DFTVDW_JOBNUMBER

Basic vdW job control

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not apply the XDM scheme.

1 add vdW as energy/gradient correction to SCF.

2 add VDW as a DFT functional and do full SCF (this option only works with C6 XDM formula).

RECOMMENDATION:

none

DFTVDW_KAI

Damping factor K for C6 only damping function

TYPE:

INTEGER

DEFAULT:

800

OPTIONS:

10-1000 default 800

RECOMMENDATION:

none

DFTVDW_METHOD

Choose the damping function used in XDM

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

1 use Becke's damping function including C6 term only.

2 use Becke's damping function with higher-order (C8,C10) terms.

RECOMMENDATION:

none

DFTVDW_MOL1NATOMS

The number of atoms in the first monomer in dimer calculation

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0-NATOMS default 0

RECOMMENDATION:

none

DFTVDW_PRINT

Printing control for VDW code

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

0 no printing.

1 minimum printing (default)

2 debug printing

RECOMMENDATION:

none

DFTVDW_USE_ELE_DRV

Specify whether to add the gradient correction to the XDM energy. only valid with Becke's C6 damping function using the interpolated BR89 model.

TYPE:

LOGICAL

DEFAULT:

1

OPTIONS:

1 use density correction when applicable (default).

0 do not use this correction (for debugging purpose)

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. 10).

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_RS6

Controls the strength of dispersion corrections, s_{r6}, in the Grimme's DFT-D3 method (see Table IV in Ref. 10).

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n Corresponding to $s_{r6} = n/1000$.

RECOMMENDATION:

NONE

DFT_D3_S6

Controls the strength of dispersion corrections, s_6, in Grimme's DFT-D3 method (see Table IV in Ref. 10).

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n Corresponding to $s_6 = n/1000$.

RECOMMENDATION:

NONE

DFT_D3_S8

Controls the strength of dispersion corrections, s_8, in Grimme's DFT-D3 method (see Table IV in Ref. 10).

TYPE:

INTEGER

DEFAULT:

1000

OPTIONS:

n Corresponding to $s_8 = n/1000$.

RECOMMENDATION:

NONE

DFT_D_A

Controls the strength of dispersion corrections in the Chai-Head-Gordon DFT-D scheme in Eq.(3) of Ref. 11.

TYPE:

INTEGER

DEFAULT:

600

OPTIONS:

n Corresponding to $a = n/100$.

RECOMMENDATION:

Use default, *i.e.*, $a = 6.0$

DFT_D

Controls the application of DFT-D or DFT-D3 scheme.

TYPE:

LOGICAL

DEFAULT:

None

OPTIONS:

FALSE	(or 0) Do not apply the DFT-D or DFT-D3 scheme
EMPIRICAL_GRIMME	dispersion correction from Grimme
EMPIRICAL_CHG	dispersion correction from Chai and Head-Gordon
EMPIRICAL_GRIMME3	dispersion correction from Grimme's DFT-D3 method (see Section 4.3.8)

RECOMMENDATION:

NONE

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

DIELST

The static dielectric constant.

TYPE:

FLOAT

DEFAULT:

78.39

OPTIONS:

real number specifying the constant.

RECOMMENDATION:

The default value 78.39 is appropriate for water solvent.

DIIS_ERR_RMS

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

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE, FALSE

RECOMMENDATION:

Use 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 default

DIIS_SEPARATE_ERRVEC

Control optimization of DIIS error vector in unrestricted calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE Use a combined alpha and beta error vector.

OPTIONS:

FALSE Use a combined alpha and beta error vector.

TRUE Use separate error vectors for the alpha and beta spaces.

RECOMMENDATION:

When using DIIS in Q-CHEM a convenient optimization for unrestricted calculations is to sum the alpha and beta 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 alpha and beta 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 to 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_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_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.

DIRECT_SCF

Controls direct SCF.

TYPE:

LOGICAL

DEFAULT:

Determined by program.

OPTIONS:

TRUE Forces direct SCF.

FALSE Do not use direct SCF.

RECOMMENDATION:

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

DMA_MIDPOINTS

Specifies whether to include bond midpoints into DMA expansion.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

FALSE	Do not include bond midpoints.
TRUE	Include bond midpoint.

RECOMMENDATION:

None

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_DMA

Specifies whether to perform Distributed Multipole Analysis.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE	Turn off DMA.
TRUE	Turn on DMA.

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.

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 default.

D_SCF_MAX_2

Sets the maximum number of level-2 iterations.

TYPE:

INTEGER

DEFAULT:

30

OPTIONS:

n User defined.

RECOMMENDATION:

Use default.

EA_STATES

Sets the number of attached target states roots to find. By default, α electron will be attached (see EOM_EA_ALPHA).

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

ECP

Defines the effective core potential and associated basis set to be used

TYPE:

STRING

DEFAULT:

No pseudopotential

OPTIONS:

General, Gen	User defined. (*$ecp* keyword required)
Symbol	Use standard pseudopotentials discussed above.

RECOMMENDATION:

Pseudopotentials are recommended for first row transition metals and heavier elements. Consul the reviews for more details.

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.

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_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_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

EFP_COORD_XYZ

Use coordinates of three atoms instead of Euler angles to specify position and orientation of the fragments (EFPMAN2 only)

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE FALSE

RECOMMENDATION:

None

EFP_DIRECT_POLARIZATION_DRIVER

Use direct solver for EFP polarization (EFPMAN2 only)

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 off 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 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 (EFPMAN2 only)

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_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 overlap based damping correction

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_USE_EFPMAN2

Enable calculation using new EFPMAN2 module

TYPE:

LOGICAL

DEFAULT:

TRUE

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

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 [Eq. (11.44) and (11.46)].

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_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, and IP
SD(ST)	EOM-CCSD(sT), available for IP

RECOMMENDATION:

None

EOM_DAVIDSON_CONVERGENCE

Convergence criterion for the RMS residuals of excited state vectors

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}

OPTIONS:

n Corresponding to 10^{-n} convergence criterion

RECOMMENDATION:

Use default. Should normally be set to the same value 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:

00105 Corresponding to 0.00001

OPTIONS:

$abcde$ Integer code is mapped to $abc \times 10^{-de}$

RECOMMENDATION:

Use 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 less than EOM_DAVIDSON_CONVERGENCE.

EOM_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

EOM_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

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.

TYPE:

LOGICAL

DEFAULT:

FALSE (use proper EOM-IP code)

OPTIONS:

FALSE, TRUE

RECOMMENDATION:

None

EOM_GPLMR_MSUBSIZE

Specifies the number of Krylov-space residuals in GPLMR.

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Generate n residuals at each iteration.

RECOMMENDATION:

Use default. The convergence is faster for larger n, but the memory usage and the overall cost will increase.

EOM_GPLMR

Specifies whether to engage GPLMR solver in EOM calculations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Use GPLMR.

RECOMMENDATION:

Not available in 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

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

EOM_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

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 (see also EOM_PRECONV_DOUBLES), 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.

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 calculations instead of EOM_PRECONV_SINGLES

EOM_PRECONV_SD

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

N perform N Davidson iterations pre-converging singles and doubles.

RECOMMENDATION:

Occasionally necessary to ensure a doubly excited state is 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

N perform N Davidson iterations pre-converging singles.

RECOMMENDATION:

Sometimes helps with problematic convergence.

EOM_REF_PROP_TE

Request for calculation of non-relaxed two-particle EOM-CC properties. The two-particle properties currently include $\langle S^2\rangle$. 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 $\langle S^2\rangle$. The variable CC_EOM_PROP must be also set to TRUE. Alternatively, CC_CALC_SSQ can be used to request $\langle S^2\rangle$ 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.

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_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 uniterated EPAOs based on atomic blocks of SPS.

OPTIONS:

n Optimize the EPAOs for up to n iterations.

RECOMMENDATION:

Use 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 default, unless convergence failure is encountered.

ERCALC

Specifies the 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.

TYPE:

INTEGER

DEFAULT:

0 Do not perform ER localized diabatization.

OPTIONS:

1 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_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 exchange-correlation functional for hybrid.

TYPE:

STRING

DEFAULT:

No default exchange functional

OPTIONS:

HF	Fock exchange
Slater, S	Slater (Dirac 1930)
Becke86, B86	Becke 1986
Becke, B, B88	Becke 1988
muB88	Short-range Becke exchange, as formulated by Song *et al.* [12]
Gill96, Gill	Gill 1996
GG99	Gilbert and Gill, 1999
Becke(EDF1), B(EDF1)	Becke (uses EDF1 parameters)
PW86,	Perdew-Wang 1986
rPW86,	Refitted PW86 for use in vdW-DF-10 and VV10
PW91, PW	Perdew-Wang 1991
PBE	Perdew-Burke-Ernzerhof 1996
TPSS	The nonempirical exchange-correlation scheme of Tao, Perdew, Staroverov, and Scuseria (requires also that the user specify "TPSS" for correlation)
TPSSH	The hybrid version of TPSS (with no input line for correlation)
PBE0, PBE1PBE	PBE hybrid with 25% HF exchange
PBEOP	PBE exchange + one-parameter progressive correlation
wPBE	Short-range ωPBE exchange, as formulated by Henderson *et al.* [13]
muPBE	Short-range μPBE exchange, due to Song *et al.* [12]
B97	Becke97 XC hybrid
B97-1	Becke97 re-optimized by Hamprecht *et al.*
B97-2	Becke97-1 optimized further by Wilson *et al.*
B3PW91, Becke3PW91, B3P	B3PW91 hybrid
B3LYP, Becke3LYP	B3LYP hybrid
B3LYP5	B3LYP based on correlation functional #5 of
HCTH	HCTH hybrid
HCTH-120	HCTH-120 hybrid
HCTH-147	HCTH-147 hybrid
HCTH-407	HCTH-407 hybrid
	Vosko, Wilk, and Nusair rather than their functional #3
BOP	B88 exchange + one-parameter progressive correlation
EDF1	EDF1
EDF2	EDF2
VSXC	VSXC meta-GGA, not a hybrid
BMK	BMK hybrid
M05	M05 hybrid
M052X	M05-2X hybrid
M06L	M06-L hybrid
M06HF	M06-HF hybrid
M06	M06 hybrid
M062X	M06-2X hybrid
M08HX	M08-HX hybrid
M08SO	M08-SO hybrid
M11L	M11-L hybrid
M11	M11 long-range corrected hybrid
SOGGA	SOGGA hybrid
SOGGA11	SOGGA11 hybrid
SOGGA11X	SOGGA11-X hybrid
BR89	Becke-Roussel 1989 represented in analytic form

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.

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 default, unless on a very flat potential, 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 default, unless on a very flat potential, in which case a larger value should be used. See FDIFF_STEPSIZE_QFF for third and fourth derivatives.

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.

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.

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.

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.

FSM_MODE

Specifies the method of interpolation

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

1	Cartesian
2	LST

RECOMMENDATION:

2. 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:

Undefined

OPTIONS:

N	number of perpendicular gradients per node

RECOMMENDATION:

4. 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:

Undefined

OPTIONS:

N	number of nodes in FSM calculation

RECOMMENDATION:

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 9.7).

FSM_OPT_MODE

Specifies the method of optimization

TYPE:

INTEGER

DEFAULT:

Undefined

OPTIONS:

1	Conjugate gradients
2	Quasi-Newton method with BFGS Hessian update

RECOMMENDATION:

2. The quasi-Newton method is more efficient when the number of nodes is high.

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. 14.

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; delocalized internals are more efficient.

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 default.

GEOM_OPT_HESSIAN

Determines the initial Hessian status.

TYPE:

STRING

DEFAULT:

DIAGONAL

OPTIONS:

DIAGONAL	Set up diagonal Hessian.
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 is expensive to compute.

GEOM_OPT_LINEAR_ANGLE

Threshold for near linear bond angles (degrees).

TYPE:

INTEGER

DEFAULT:

165 degrees.

OPTIONS:

n User-defined level.

RECOMMENDATION:

Use 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 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 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:

INTEGER

DEFAULT:

1

OPTIONS:

1 Make use of point group symmetry.
0 Do not make use of point group symmetry.

RECOMMENDATION:

Use 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 gradient component.

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 (GDISS default).

RECOMMENDATION:

Use 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.

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 (it's 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 wavefunction.

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

1 Use Unrestricted Implementation described in Ref. 15

RECOMMENDATION:

Only works for Unrestricted PP and no other GVB model.

GVB_ORB_CONV

The GVB-CC wavefunction 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)$(e^{t^p_{ij}} - 1)/(e^1 - 1)$

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)$(e^{t^p_{ij}} - 1)/(e^1 - 1)$

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 (ie 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 (ie 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

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.

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_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 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 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

ICVICK

Specifies whether to perform cavity check

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.

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.

IGDEFIELD

Triggers the calculation of the electrostatic potential and/or the electric field at the positions of the MM charges.

TYPE:

INTEGER

DEFAULT:

UNDEFINED

OPTIONS:

O Computes ESP.

1 Computes ESP and EFIELD.

2 Computes EFIELD.

RECOMMENDATION:

Must use this *$rem* when IGDESP is specified.

IGDESP

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, esp for each point.

TYPE:

INTEGER

DEFAULT:

none no electrostatic potential evaluation

OPTIONS:

-1 read grid input via the *$plots* section of the input deck

0 Generate the ESP values at all nuclear positions.

+n read n grid points in bohrs (!) from the ASCII file ESPGrid.

RECOMMENDATION:

None

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 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.

INTCAV

A flag to select the surface integration method.

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.

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 default unless the CASE operator is desired.

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

IOPPRD

Specifies the choice of system operator form.

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.

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

IROTGR

Rotation of the cavity surface integration grid.

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.

ISHAPE

A flag to set the shape of the cavity surface.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 use the electronic iso-density surface.

1 use a spherical cavity surface.

RECOMMENDATION:

Use the default surface.

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

ITRNGR

Translation of the cavity surface integration grid.

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 Angstroms.

RECOMMENDATION:

The default value is recommended unless the single-center integrations procedure fails.

JOBTYPE

Specifies the type of calculation.

TYPE:

STRING

DEFAULT:

SP

OPTIONS:

SP	Single point energy.
OPT	Geometry Minimization.
TS	Transition Structure Search.
FREQ	Frequency Calculation.
FORCE	Analytical Force calculation.
RPATH	Intrinsic Reaction Coordinate calculation.
NMR	NMR chemical shift calculation.
ISSC	Indirect nuclear spin–spin coupling calculation.
BSSE	BSSE calculation.
EDA	Energy decomposition analysis.

RECOMMENDATION:

Job dependent

LB94_BETA

Set the β parameter of LB94 xc potential

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n	Corresponding to $\beta = n/10000$.

RECOMMENDATION:

Use default, *i.e.*, $\beta = 0.05$

LINEQ

Flag to select the method for solving the linear equations that determine the apparent point charges on the cavity surface.

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.

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 measured in %

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.

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 a Löwdin population analysis instead of a Mulliken.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not calculate Löwdin Populations.

TRUE Run Löwdin Population analyses instead of Mulliken.

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) Use the long-range-corrected version of the requested functional.

RECOMMENDATION:

Long-range correction is available for any combination of Hartree-Fock, B88, and PBE exchange (along with any stand-alone correlation functional).

MAKE_CUBE_FILES

Requests generation of cube files for MOs, NTOs, or NBOs.

TYPE:

LOGICAL

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

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_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:

User-defined.

RECOMMENDATION:

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

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 symmetries, but the branching-plane updating method is more efficient for larger molecules. However, the latter does not work if the two states have different symmetries. If using branching plane updating method, GEOM_OPT_COORDS must be set to 0 in the *$rem* section (*i.e.*, this algorithm is available in Cartesian coordinates only). The penalty-constrained method converges slowly and is suggested only when the other methods do not work.

MECP_OPT

Determines whether we are doing MECP optimizations.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE	Do MECP optimizations.
FALSE	Don’t do MECP optimizations.

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 Default.

MECP_STATE1

Determines the first state for crossing.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

$[i,j]$ find the jth excited state with the total spin of 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

Determines the second state for crossing.

TYPE:

INTEGER/INTEGER ARRAY

DEFAULT:

None

OPTIONS:

$[i,j]$ find the jth excited state with the total spin of 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 individual program modules

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 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_STATIC

Sets the memory for Fortran AO integral calculation and transformation modules.

TYPE:

INTEGER

DEFAULT:

64 corresponding to 64 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

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.

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 default, or set to the physical memory of your machine. Note that if more than 1GB is specified for a CCMAN job, the memory is allocated as follows

12%	MEM_STATIC
50%	CC_MEMORY
35%	Other memory requirements:

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 default.

METHOD

Specifies the level of theory.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

HF Exact (Hartree-Fock).

RECOMMENDATION:

Use HF for Hartree-Fock calculations.

METHOD

Specifies the level of theory, either DFT or wavefunction-based.

TYPE:

STRING

DEFAULT:

HF No correlation, Hartree-Fock exchange

OPTIONS:

MP2	Sections 5.2 and 5.3
RI-MP2	Section 5.5
Local_MP2	Section 5.4
RILMP2	Section 5.5.1
ATTMP2	Section 5.6.1
ATTRIMP2	Section 5.6.1
ZAPT2	A more efficient restricted open-shell MP2 method [16].
MP3	Section 5.2
MP4SDQ	Section 5.2
MP4	Section 5.2
CCD	Section 5.7
CCD(2)	Section 5.8
CCSD	Section 5.7
CCSD(T)	Section 5.8
CCSD(2)	Section 5.8
CCSD(fT)	Section 5.8.3
CCSD(dT)	Section 5.8.3
QCISD	Section 5.7
QCISD(T)	Section 5.8
OD	Section 5.7
OD(T)	Section 5.8
OD(2)	Section 5.8
VOD	Section 5.9
VOD(2)	Section 5.9
QCCD	Section 5.7
QCCD(T)	
QCCD(2)	
VQCCD	Section 5.9

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:

MGC_AMPS

Choice of Amplitude Truncation

TYPE:

INTEGER

DEFAULT:

none

OPTIONS:

2≤ n ≤ 123456, a sorted list of integers for every amplitude
which will be iterated. Choose 1234 for PQ and 123456 for PH

RECOMMENDATION:

MGC_LOCALINTER

Pair filter on an intermediate.

TYPE:

BOOL

DEFAULT:

FALSE

OPTIONS:

Any nonzero value enforces the pair constraint on intermediates,
significantly reducing computational cost. Not recommended for ≤ 2 pair locality

RECOMMENDATION:

MGC_LOCALINTS

Pair filter on an integrals.

TYPE:

BOOL

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:

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:

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-defined 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.

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. (11.9), 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 default.

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.

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.

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 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 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 default. For large systems, limiting this number may be required to avoid memory exhaustion.

MOPROP

Specifies the job for mopropman.

TYPE:

INTEGER

DEFAULT:

0 Do not run mopropman.

OPTIONS:

1	NMR chemical shielding tensors.
2	Static polarizability.
3	Indirect nuclear spin–spin coupling tensors.
100	Dynamic polarizability.
101	First hyperpolarizability.
102	First hyperpolarizability, reading First order results from disk.
103	First hyperpolarizability using Wigner's $(2n + 1)$ rule.
104	First hyperpolarizability using Wigner's $(2n + 1)$ rule, reading first order results from disk.

RECOMMENDATION:

None

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*10.0^{-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*10.0^{-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 wavefunction analysis requested.

TYPE:

INTEGER

DEFAULT:

4

OPTIONS:

n Calculate moments to nth order.

RECOMMENDATION:

Use default unless higher multipoles are required.

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

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 nonlocal part of vdW-DF-10 (aka vdW-DF2)
VV09	the nonlocal part of VV09
VV10	the nonlocal 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 SG-1 grid

OPTIONS:

Same as for XC_GRID

RECOMMENDATION:

Use 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 nonlocal 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. See further details in Ref. [17].

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 semilocal exchange functional is used. $C = 0.0089$ is recommended when a long-range corrected (LRC) hybrid functional is used. See further details in Ref. [17].

NOCI_PRINT

Specify the debug print level of NOCI

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

RECOMMENDATION:

Increase this for more debug information

NPTLEB

The number of points used in the Lebedev grid for the single-center surface integration. (Only relevant if INTCAV=0).

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).

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.

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.

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:

Experimental option. Use 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 angstroms 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 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 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_FROZEN_CORE

Controls the number of frozen core orbitals

TYPE:

INTEGER

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 Sec. 6.6).

N_FROZEN_CORE

Sets the number of frozen core orbitals in a post-Hartree–Fock calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

FC Frozen Core approximation (all core orbitals frozen).

n Freeze n core orbitals.

RECOMMENDATION:

While the default is not to freeze orbitals, MP2 calculations are more efficient with frozen core orbitals. Use FC if possible.

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.

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_SOL

Specifies number of atoms included in the Hessian

TYPE:

INTEGER

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.

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

Sets the Coulomb attenuation parameter ω.

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

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 EPAO's after a conventional SCF.

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.

PBHT_ANALYSIS

Controls whether overlap analysis of electronic excitations is performed.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Do not perform overlap analysis

TRUE Perform overlap analysis

RECOMMENDATION:

None

PBHT_FINE

Increases accuracy of overlap analysis

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE

TRUE Increase accuracy of overlap analysis

RECOMMENDATION:

None

PHESS

Controls whether partial Hessian calculations are performed.

TYPE:

INTEGER

DEFAULT:

0 Full Hessian calculation

OPTIONS:

0 Full Hessian calculation

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 utilized.

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 warmup 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. Recommend setting to at least 100000 for converged simulations.

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 utilized. For Cartesian and normal-mode moves, the maximum displacement is adjusted during the warmup 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 1-bead limit is purely classical sampling; the infinite-bead limit is 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.

POP_MULLIKEN

Controls running of Mulliken population analysis.

TYPE:

LOGICAL/INTEGER

DEFAULT:

TRUE (or 1)

OPTIONS:

FALSE	(or 0) Do not calculate Mulliken Population.
TRUE	(or 1) Calculate Mulliken population
2	Also calculate shell populations for each occupied orbital.
-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 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_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 virtuals.

NVIRT Number of virtuals to print.

RECOMMENDATION:

Use TRUE unless more virtuals are desired.

PRINT_RADII_GYRE

Controls printing of MO centroids and radii of gyration.

TYPE:

LOGICAL/INTEGER

DEFAULT:

FALSE

OPTIONS:

TRUE (or 1) Calculate the centroid and radius of gyration for each MO and density.

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 utilized 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:

2111 Cartesian h-functions and pure g, f, d functions

OPTIONS:

$hgfd$ Use 1 for pure and 2 for Cartesian.

RECOMMENDATION:

This is pre-defined for all standard basis sets

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 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 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 default unless running calculations with CHARMM.

RADSPH

Sphere radius used to specify the cavity surface (Only relevant for ISHAPE=1).

TYPE:

FLOAT

DEFAULT:

Half the distance between the outermost atoms plus 1.4 Angstroms.

OPTIONS:

Real number specifying the radius in bohr (if positive) or in Angstroms (if negative).

RECOMMENDATION:

Make sure that the cavity radius is larger than the length of the molecule.

RAS_ACT_DIFF

Sets the number of alpha vs. beta electrons

TYPE:

Integer

DEFAULT:

None

OPTIONS:

n	user defined integer

RECOMMENDATION:

Set to 1 for an odd number of electrons or a cation, -1 for an anion. 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 RASCI.

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 RASCI.

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 RASCI.

RAS_DO_HOLE

Controls the presence of hole excitations in the RAS-CI wavefunction.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Include hole configurations (RAS1 to RAS2 excitations)

FALSE Do not include hole configurations

RECOMMENDATION:

None. Only works with RASCI.

RAS_DO_PART

Controls the presence of particle excitations in the RAS-CI wavefunction.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Include particle configurations (RAS2 to RAS3 excitations)

FALSE Do not include particle configurations

RECOMMENDATION:

None. Only works with RASCI.

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 RASCI.

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 RASCI.

RAS_NATORB_STATE

Allows to save the natural orbitals of a RAS-CI computed state.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

i Saves the natural orbitals for the i-th state

RECOMMENDATION:

None. Only works with RASCI.

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 RASCI.

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_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 RASCI.

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 RASCI.

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:

Only for RASCI, which at present only allows for the computation of systems with an even number of electrons. Thus, RAS_SPIN_MULT only can take odd values.

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.

RHOISO

Value of the electronic iso-density contour used to specify the cavity surface. (Only relevant for ISHAPE = 0).

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.

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.

ROKS_LEVEL_SHIFT

Introduce a level shift of N/100 Hartree to aid convergence.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No shift

N level shift of N/100 Hartree.

RECOMMENDATION:

Use in cases of problematic convergence.

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. You should also set UNRESTRICTED=TRUE

ROTTHE ROTPHI ROTCHI

Euler angles (θ, ϕ, χ) in degrees for user-specified rotation of the cavity surface. (relevant if IROTGR=3)

TYPE:

FLOAT

DEFAULT:

0,0,0

OPTIONS:

θ, ϕ, χ in degrees

RECOMMENDATION:

None.

RPATH_COORDS

Determines which coordinate system to use in the IRC search.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Use mass-weighted coordinates.

1 Use Cartesian coordinates.

2 Use *Z*-matrix coordinates.

RECOMMENDATION:

Use default.

RPATH_DIRECTION

Determines the direction of the eigen mode 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 eigen mode.

-1 Descend in the negative direction of the eigen mode.

RECOMMENDATION:

It is usually not possible to determine in which direction to go *a priori*, and therefore both directions will need to be considered.

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 thousandths of *a.u.*).

TYPE:

INTEGER

DEFAULT:

150 corresponding to a step size of 0.15 *a.u.*.

OPTIONS:

n Step size = n/1000.

RECOMMENDATION:

None.

RPATH_PRINT

Specifies the print output level.

TYPE:

INTEGER

DEFAULT:

2

OPTIONS:

n

RECOMMENDATION:

Use default, 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.

RECOMMENDATION:

Use 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:

None

SAVE_LAST_GPX

Save last $\mathbf{G}\,[\mathbf{P}^{\mathrm{x}}]$ when calculating dynamic polarizabilities in order to call mopropman in a second run with MOPROP = 102.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	False
1	True

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

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

DIIS Pulay DIIS.

OPTIONS:

DIIS	Pulay DIIS.
DM	Direct minimizer.
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).
GDM	Geometric Direct Minimization.
RCA	Relaxed constraint algorithm
RCA_DIIS	Use RCA initially, switching to DIIS for later iterations (see THRESH_RCA_SWITCH and MAX_RCA_CYCLES described later in this chapter)
ROOTHAAN	Roothaan repeated diagonalization.

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 is the recommended fallback option. If DIIS approaches the correct solution but fails to finally converge, DIIS_GDM is the recommended fallback.

SCF_CONVERGENCE

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

TYPE:

INTEGER

DEFAULT:

5	For single point energy calculations.
8	For geometry optimizations and vibrational analysis.
8	For SSG calculations, see Chapter 5.

OPTIONS:

User-defined

RECOMMENDATION:

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

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 and density matrices.

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 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 density (available only with standard basis sets)
GWH	For ROHF where a set of orbitals are required.
FRAGMO	For a fragment MO calculation

OPTIONS:

CORE	Diagonalize core Hamiltonian
SAD	Superposition of atomic density
GWH	Apply generalized Wolfsberg-Helmholtz approximation
READ	Read previous MOs from disk
FRAGMO	Superimposing converged fragment MOs

RECOMMENDATION:

SAD guess for standard basis sets. For general basis sets, it is best to use the BASIS2 *$rem*. Alternatively, try the GWH or core Hamiltonian guess. 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_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 and DFT exchange-correlation matrices) on each cycle.

RECOMMENDATION:

Proceed with care; can result in *extremely* large output files at level 2 or higher. 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.

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).

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.

SET_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.

SFX_AMP_OCC_A

Defines a customer 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 customer 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

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

SKIP_CIS_RPA

Skips the solution of the CIS, RPA, TDA or TDDFT equations for wavefunction 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.

SOLVENT_METHOD

Sets the preferred solvent method.

TYPE:

STRING

DEFAULT:

0

OPTIONS:

0	Do not use a solvation model.
ONSAGER	Use the Kirkwood-Onsager model (Section 11.2.1).
PCM	Use an apparent surface charge, polarizable continuum model (Section 11.2.2).
ISOSVP	Use the iso-density implementation of the SS(V)PE model (Section 11.2.5).
COSMO	Use COSMO (similar to C-PCM but with an outlying charge correction [18, 19]; see Section 11.2.6).
SM8	Use version 8 of the Cramer-Truhlar SMx model (Section 11.2.7.1).
SM12	Use version 12 of the SMx model (Section 11.2.7.2).
CHEM_SOL	Use the Langevin Dipoles model (Section 11.2.8).

RECOMMENDATION:

Consult the literature. 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 11.2.2.) Several versions of SM12 are available as well, as discussed in Section 11.2.7.2.

SOS_FACTOR

Controls the strength of the opposite-spin component of PT2 correlation energy.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n	Corresponding to $c_{os} = n/1000000$ in Eq. (4.65).

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

SPIN_FLIP_XCIS

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

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 wavefunction.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0	Do not compute the SSG wavefunction
1	Do compute the SSG wavefunction

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 $c_{ss} = n/1000000$ in Eq. (4.65).

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.

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 the CIS and TDDFT calculations (including SF 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.

SVP_CAVITY_CONV

Determines the convergence value of the iterative iso-density 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 solvation module 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 Q-CHEM solvation computation.

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.

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 runs a gas-phase calculation and after
convergence runs the SS(V)PE computation.

1 does 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

SYMMETRY

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

TYPE:

LOGICAL

DEFAULT:

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

SYM_IGNORE

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

TYPE:

LOGICAL

DEFAULT:

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

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

Use 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.

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:

Should be at least three greater than SCF_CONVERGENCE. Increase for more significant figures, at greater computational cost.

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.

TRANS_ENABLE

Decide whether or not to enable the molecular transport code.

TYPE:

INTEGER

DEFAULT:

0 Do not perform transport calculations.

OPTIONS:

1 Perform transport calculations in the Landauer approximation.

-1 Print matrices for subsequent calls for `tranchem.exe` as a stand-alone post-processing utility, or for generating bulk model files.

RECOMMENDATION:

Use as required.

TRANX, TRANY, TRANZ

x, y, and z value of user-specified translation (only relevant if ITRNGR is set to 5 or 6

TYPE:

FLOAT

DEFAULT:

0, 0, 0

OPTIONS:

x, y, and z relative to the origin in the appropriate units.

RECOMMENDATION:

None.

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

UNRESTRICTED

Controls the use of restricted or unrestricted orbitals.

TYPE:

LOGICAL

DEFAULT:

FALSE (Restricted) Closed-shell systems.

TRUE (Unrestricted) Open-shell systems.

OPTIONS:

TRUE (Unrestricted) Open-shell systems.

FALSE Restricted open-shell HF (ROHF).

RECOMMENDATION:

Use default unless ROHF is desired. Note that for unrestricted calculations on systems with an even number of electrons it is usually necessary to break alpha/beta 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_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:

INTEGER

DEFAULT:

0

OPTIONS:

0 MGEMM disabled

1 MGEMM enabled

RECOMMENDATION:

Use when having single-precision cards

VARTHRESH

Controls the temporary integral cut-off threshold. $tmp\_thresh = 10^{-\text{VARTHRESH}} \times DIIS\_error$

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.

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 $2 * (N_{\text{Vib}} + N_{\text{VCI}})/N_{\text{Vib}}N_{\text{VCI}}$ 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 default.

WANG_ZIEGLER_KERNEL

Controls whether to use the Wang-Ziegler non-collinear exchange-correlation kernel in a SFDFT calculation.

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 wavefunction analysis tasks.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Perform default wavefunction analysis.

FALSE Do not perform default wavefunction analysis.

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, use 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

XCIS

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:

1 SG-1 hybrid

OPTIONS:

0	Use SG-0 for H, C, N, and O, SG-1 for all other atoms.
1	Use SG-1 for all atoms.
2	Low Quality.
mn	The first six integers correspond to m radial points and the second six integers correspond to n angular points where possible numbers of Lebedev angular points are listed in section 4.3.11.
$-mn$	The first six integers correspond to m radial points and the second six integers correspond to n angular points where the number of Gauss-Legendre angular points $n = 2N^2$.

RECOMMENDATION:

Use 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) grid specified by XC_GRID (which defaults to SG-1, if not otherwise specified) for final cycles of the SCF.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE/FALSE

RECOMMENDATION:

The use of the smart grid can save some time on initial SCF cycles.

XOPT_SEAM_ONLY

Orders an intersection seam search only, no minimization is to perform.

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 setting this option to TRUE and use 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 CC_NLOWSPIN.
spin = 1	Addresses states with high spin, see also CC_NHIGHSPIN.

irrep	Specifies the irreducible representation to which
	the state belongs, for C_{2v} point group symmetry
	irrep = 1 for A_1, irrep = 2 for A_2,
	irrep = 3 for B_1, irrep = 4 for B_2.

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_FIX_MULLIKEN

Control to use self-consistent charge for EE-MBE.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE	Perform an EE-MBE without self-consistent charge.
FALSE	Perform an EE-MBE with self-consistent charge.

RECOMMENDATION:

The charges are derived from isolated monomers without self-consistent process. It is available to use with Mulliken charges, Löwdin charges and CHELPG charges

XPOL_FIX_TIP3P

Use charges corresponding to TIP3P water for EE-MBE.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Perform an EE-MBE with charges corresponding to TIP3P water.

FALSE Do not perform an EE-MBE with charges corresponding to TIP3P water.

RECOMMENDATION:

Only available for water molecules

XPOL_OMEGA

Controls the range-separation parameter, ω, that is used in long-range-corrected DFT.

TYPE:

BOOLEAN

DEFAULT:

FALSE

OPTIONS:

TRUE Use different ω values for different fragments.

FALSE Use a single value of ω for all fragments.

RECOMMENDATION:

If FALSE, the *$rem* variable OMEGA should be used to specify the single value of ω. If TRUE, separate values for each fragment should be specified in an *$lrc_omega* input section. Values in the *$lrc_omega* section have the same units as the *$rem* variable OMEGA, namely, ω =OMEGA/1000, in atomic units.

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 (4,2)=(M,N) 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. Hesselmann, *J. Phys. Chem. A* **115**, 11321 (2011).

[2] K. U. Lao and J. M. Herbert, *J. Phys. Chem. Lett.* **3**, 3241 (2012).

[3] R. Podeszwa, K. Pernal, K. Patkowski, and K. Szalewicz, *J. Phys. Chem. Lett.* **1**, 550 (2010).

[4] K. U. Lao and J. M. Herbert, *J. Chem. Phys.* **139**, 034107 (2013).

[5] J. M. Herbert and M. Head-Gordon, *J. Chem. Phys.* **121**, 11542 (2004).

[6] C. M. Breneman and K. B. Wiberg, *J. Comput. Chem.* **11**, 361 (1990).

[7] J. P. Perdew and Y. Wang, *Phys. Rev. B* **45**, 13244 (1992).

[8] E. Proynov and J. Kong, *Phys. Rev. A* **79**, 014103 (2009).

[9] T. Tsuneda, T. Suzumura, and K. Hirao, *J. Chem. Phys.* **110**, 10664 (1999).

[10] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg, *J. Chem. Phys.* **132**, 154104 (2010).

[11] J.-D. Chai and M. Head-Gordon, *Phys. Chem. Chem. Phys.* **10**, 6615 (2008).

[12] J. W. Song, T. Hirosawa, T. Tsuneda, and K. Hirao, *J. Chem. Phys.* **126**, 154105 (2007).

[13] T. M. Henderson, B. G. Janesko, and G. E. Scuseria, *J. Chem. Phys.* **128**, 194105 (2008).

[14] D. Das et al., *J. Chem. Phys.* **117**, 10534 (2002).

[15] G. J. O. Beran, B. Austin, A. Sodt, and M. Head-Gordon, *J. Phys. Chem. A* **109**, 9183 (2005).

[16] D. Jayatilaka and T. J. Lee, *Chem. Phys. Lett.* **199**, 211 (1992).

[17] O. A. Vydrov and T. Van Voorhis, *J. Chem. Phys.* **133**, 244103 (2010).

[18] A. Klamt and V. Jonas, *J. Chem. Phys.* **105**, 9972 (1996).

[19] K. Baldridge and A. Klamt, *J. Chem. Phys.* **106**, 6622 (1997).