QC-PBC 1.0 User’s Manual

Version 1.0

July, 2026

QC-PBC User's Manual

Version 1.0 editors:

Prof. Joonho Lee, Dr. Adam Rettig

The contributions of individual developers to each version are highlighted in "New Features" sections.

Published by:

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

Customer Support:

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

The information in this document applies to version 1.0 of QC-PBC.
This document version generated on July 6, 2026.

Contents

1 Introduction **7**

1.1 About This Manual . . . 7

1.1.1 Chapter Summaries . . . 7

1.2 Q-CHEM, Inc. . . . 8

1.2.1 Contact Information and Customer Support . . . 8

1.2.2 About the Company . . . 8

1.2.3 Company Mission . . . 8

1.3 Citing Q-CHEM . . . 8

2 Installation and Execution of QC-PBC **9**

2.1 Installing QC-PBC . . . 9

2.1.1 Downloading and Licensing . . . 9

2.1.2 QC-PBC Auxiliary files (*$QCAUX*) . . . 9

2.1.3 QC-PBC Run-time Environment Variables . . . 9

2.2 Running QC-PBC . . . 10

2.2.1 General Usage . . . 10

2.2.2 Testing and Exploring QC-PBC . . . 11

3 QC-PBC Inputs **12**

3.1 Atomistic Simulation Environment . . . 12

3.2 General Form . . . 12

3.3 Unit Cell Coordinate Input (*$unitcell*) . . . 13

3.3.1 Introduction . . . 13

3.3.2 Specifying the Molecular Coordinates Manually . . . 14

3.4 Job Specification: The *$rem* Input Section . . . 15

3.5 Batch Jobs: Multiple Inputs in a Single File . . . 17

3.6 QC-PBC Output File . . . 19

4 Self-Consistent Field Ground-State Methods **20**

4.1 Introduction . . . 20

4.2 Theoretical Background . . . 20

4.2.1 Molecular SCF and LCAO Approximations . . . 20

4.2.2 Periodic SCF and LCAO Approximations . . . 22

4.2.3 Hartree-Fock Theory . . . 24

4.3 Gaussian Plane Wave (GPW) . . . 26

4.3.1 Shellpair GPW evaluation . . . 27

4.4 Density Fitting (DF) . . . 29

4.5 Basic SCF Job Control . . . 30

4.5.1 Introduction . . . 30

4.5.2 Job Control . . . 30

4.5.3 Additional Options . . . 32
4.5.4 Examples . . . 35
4.6 SCF Initial Guess . . . 36
4.6.1 Introduction . . . 36
4.6.2 Initial Guess Types . . . 36
4.6.3 Reading MOs from Disk . . . 37
4.6.4 Basis Set Projection . . . 38
4.6.5 Advanced AUTOSAD Options . . . 39
4.7 Converging SCF Calculations . . . 41
4.7.1 Introduction . . . 41
4.7.2 Basic Convergence Control Options . . . 42
4.7.3 Direct Inversion in the Iterative Subspace (DIIS) . . . 43
4.7.4 Geometric Direct Minimization (GDM) . . . 44
4.8 Algorithms for Efficient Fock-Build . . . 46
4.8.1 Introduction . . . 46
4.8.2 Periodic Fourier-transformed Coulomb Method . . . 47
4.8.3 occ-RI-K Exchange Algorithm . . . 47
4.8.4 THC-K . . . 49
4.9 SCF Post-processing . . . 51
4.9.1 Density of States . . . 51
References and Further Reading . . . 55

5 Density Functional Theory **57**
5.1 Introduction . . . 57
5.2 Kohn-Sham Density Functional Theory . . . 57
5.3 Overview of Available Functionals . . . 59
5.3.1 Introduction . . . 59
5.3.2 Suggested Density Functionals for Extended Applications . . . 61
5.3.3 Exchange Functionals . . . 62
5.3.4 Correlation Functionals . . . 63
5.3.5 Exchange-Correlation Functionals . . . 65
5.3.6 Specialized Functionals . . . 70
5.3.7 User-Defined Density Functionals . . . 70
5.4 Basic DFT Job Control . . . 71
5.5 DFT Numerical Quadrature . . . 73
5.5.1 Introduction . . . 73
5.5.2 Angular Grids . . . 74
5.5.3 Standard Quadrature Grids . . . 75
5.5.4 Consistency Check and Cutoffs . . . 76
5.6 Range-Separated Hybrid Density Functionals . . . 77
5.6.1 Introduction . . . 77
5.6.2 Semi-Empirical RSH Functionals . . . 77
5.6.3 User-Defined RSH Functionals . . . 77
5.7 DFT Methods for van der Waals Interactions . . . 80
5.7.1 Non-Local Correlation (NLC) Functionals . . . 80
5.7.2 Empirical Dispersion Corrections: DFT-D . . . 83
References and Further Reading . . . 89

6 Wave Function-Based Correlation Methods **97**

6.1 Introduction . . . 97
6.2 Møller-Plesset Perturbation Theory . . . 98
6.3 MP2 Methods . . . 100
6.3.1 Introduction . . . 100
6.3.2 CD-MP2 Energies . . . 100
6.3.3 CD-MP2 Method with the Laplace Transformation . . . 102
6.3.4 Spin-Biased CD-MP2 Methods . . . 103
6.3.5 Regularized CD-MP2 Methods . . . 103
6.4 MP3 . . . 104
6.4.1 Introduction . . . 104
6.4.2 Examples . . . 105
6.5 Tensor Hypercontraction Methods . . . 105
6.6 Size-Consistent Brillouin-Wigner Perturbation Theory . . . 106
6.6.1 Introduction . . . 106
6.6.2 BW-s2 Job Control . . . 107
6.7 Direct Random Phase Approximation Methods . . . 108
6.7.1 Introduction . . . 108
6.7.2 dRPA Job Control . . . 109
6.8 Coupled-Cluster Methods . . . 109
6.8.1 Introduction . . . 109
6.8.2 Coupled Cluster Singles and Doubles (CCSD) . . . 110
6.8.3 Coupled Cluster Singles, Doubles and Triples (CCSDT) . . . 111
6.8.4 Cholesky Decomposition with CC (CD-CC) . . . 112
6.8.5 Job Control Options . . . 112
6.8.6 Examples . . . 115
References and Further Reading . . . 116

7 Basis Sets and Pseudopotentials **117**
7.1 Introduction to Basis Sets . . . 117
7.2 Crystalline Gaussian-Type Orbitals . . . 118
7.3 Built-In Basis Sets . . . 118
7.4 Basis Set Symbolic Representation . . . 119
7.4.1 Symbolic Representation Overview . . . 119
7.4.2 Customization . . . 124
7.5 User-Defined Basis Sets (*$basis*) . . . 125
7.5.1 Introduction . . . 125
7.5.2 Job Control . . . 125
7.5.3 Format for User-Defined Basis Sets . . . 126
7.6 Mixed Basis Sets . . . 127
7.7 Ghost Atoms and Basis Set Superposition Error . . . 127
7.8 Pseudopotentials . . . 130
7.8.1 Introduction & Overview . . . 130
7.8.2 GTH Pseudopotentials . . . 130
7.8.3 Built-In GTH Pseudopotentials . . . 131
7.8.4 User-Defined Pseudopotentials . . . 133
7.8.5 Forces and Vibrational Frequencies with Pseudopotentials . . . 134
References and Further Reading . . . 135

8 Equilibrium Geometry Optimization **137**

8.1 Equilibrium Geometries with QC-PBC . . . 137
8.1.1 Introduction . . . 137
8.1.2 Theoretical Background . . . 138
8.1.3 Eigenvector-Following (EF) Algorithm . . . 140
8.1.4 Delocalized Internal Coordinates . . . 141
8.2 Geometry Optimization Job Controls . . . 143
8.2.1 Job Control Overview . . . 143
8.2.2 *$rem* Job Control . . . 144
8.2.3 LIBOPT3 Job Control . . . 148
8.2.4 LIBOPT3 Job Examples . . . 157
References and Further Reading . . . 158

9 Fragment-based approaches, calculation of non-covalent interactions, and energy decomposition analysis. 159
9.1 Introduction . . . 159
9.2 Fragment Input . . . 159
9.2.1 Job Control . . . 160
9.2.2 Additional Options . . . 161
9.2.3 FRAGMO SCF Guess . . . 162
9.3 Energy Decomposition Analys . . . 164
9.3.1 Introduction . . . 164
9.3.2 Job Control . . . 166
9.3.3 Additional Options . . . 167
References and Further Reading . . . 168

Chapter 1

Introduction

1.1 About This Manual

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

Much of the syntax used in the QC-PBC input is borrowed from Q-CHEM. Furthermore, the features for Γ-point calculations are typically run through the same code as Q-CHEM internally. This makes the input keywords and corresponding options for those methods in QC-PBC identical to Q-CHEM. Nonetheless, we strive to provide self-contained description in this manual in case the users are unfamiliar with Q-CHEM. Some of the sections will be directly copied from the latest Q-CHEM manual.

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

While for moleculear systems, IQMOL, a graphical user interface designed for use with Q-CHEM, can be used and more information can be found on the `www.iqmol.org` websitge. IQMOL functions as a molecular structure builder, as an interface for local or remote submission of Q-CHEM jobs, and as a post-calculation visualization program for densities and molecular orbitals. We currently provide no such interface available for QC-PBC.

1.1.1 Chapter Summaries

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

Ch. 2: Procedures to install, test, and run QC-PBC on your machine.

Ch. 3: Overview of the QC-PBC command-line input.

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

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

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

Ch. 7: Using QC-PBC's built-in basis sets, or specifying a user-defined basis set, as well as pseudopotentials.

Ch. 8: Options available for running geometry optimization calculations.

Ch. 9: Fragment-based approaches for efficient calculations on large systems, calculation of non-covalent interactions, and energy decomposition analysis.

1.2 Q-CHEM, Inc.

1.2.1 Contact Information and Customer Support

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

1.2.2 About the Company

QC-PBC is distributed by Q-CHEM, Inc., the company responsible for distributing the widely-used Q-CHEM package for molecular modeling.

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 Anna Krylov (USC), Shirin Faraji (University of DÃijsseldorf), John Herbert (Ohio State), Peter Gill (University of Sydney), Yuezhi Mao (San Diego State University), 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.

1.2.3 Company Mission

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

1.3 Citing Q-CHEM

Users who publish papers based on QC-PBC calculations are asked to cite the official peer-reviewed literature citation for the software, which can be found on the website www.q-chem.com. The primary literature is extensively referenced throughout this manual, and users are urged to cite the original literature for particular theoretical methods. This is how our large community of academic developers gets credit for its effort.

Chapter 2

Installation and Execution of QC-PBC

2.1 Installing QC-PBC

2.1.1 Downloading and Licensing

The QC-PBC package is currently stored on our downloads page as a `.tar` file (it will be integrated properly with Q-Chem, to be distributed as an optional separate executable/binary within the Q-Chem installation package, before the 6.2 release). The file contains all of the executables, auxiliary files, and scripts needed to run Q-CHEM, as well as a set of sample jobs. The files can be downloaded from `http://downloads.q-chem.com/qcpbc/qcpbc_linux.tar.gz` (for Linux) or `http://downloads.q-chem.com/qcpbc/qcpbc_mac.tar.gz` (for Mac).

2.1.2 QC-PBC Auxiliary files (*$QCAUX*)

The `$QCAUX` environment variable determines the directory where QC-PBC searches for auxiliary files, specifically the basis sets and pseudopotentials required to run QC-PBC calculations. If not set explicitly, it defaults to `$QCPBC/qcaux`. The auxiliary files for QC-PBC are contained in QC-PBC's `$QCAUX`, so a separate download is not required.

2.1.3 QC-PBC Run-time Environment Variables

QC-PBC requires the following shell environment variables setup before running any calculations:

QCPBC	Defines the location of the QC-PBC directory structure.
QCAUX	Defines the location of the auxiliary information required by QC-PBC, which includes the license required to run QC-PBC.
QCSCRATCH	Defines the directory in which QC-PBC will store temporary files.

The user should also add `$QCPBC/bin` to their `PATH` variable, in order to run QC-PBC without the full path. Linux users must also add `$QCPBC/exe` to the `LD_LIBRARY_PATH` variable to load the dynamically linked libraries (this is temporary; the final version will be statically linked).

2.2 Running QC-PBC

2.2.1 General Usage

Once all of the command line arguments are set up properly, the user is now able to run QC-PBC. Currently, the best way to run QC-PBC is by using the QC-PBC Python script. This script is found in `$QCPBC/bin`, and can be run using the following format:

```
qcpbc -i infile -o outfile
qcpbc -i infile -o outfile -c savename
qcpbc -keep 1 -i infile -o outfile -c savename
```

where the `-i` command line flag indicates the name of the QC-PBC input file, and `-o` indicates the name of the desired output file. Additionally, `-nt` can be used to set the number of OpenMP threads used for the calculation. The `-c` argument indicates the name of the subdirectory in which to save scratch files, and the `-keep` argument can be used to keep the files in the scratch directory after completion of the job. It can be set to 1 (keep files) or 0 (do not keep files).

Parallel execution of QC-PBC can also be threaded across multiple processors on a single node using OpenMP. To run a calculation with OpenMP threads, specify the number of threads (*nthreads*) using the command option `-nt`. For example, to run with multiple threads, one would use the following:

```
qcpbc -nt nthreads -i infile -o outfile
```

2.2.1.1 Input File Formatting

QC-PBC uses a similar input file format to the Q-Chem program, with different sections denoted using `$rem` variable sections. An example input calculation for LiH at the rVV10/szv-gth level of theory is shown below:

```
$lattice
3
0.0 3.8370888985118827 3.8370888985118827
3.8370888985118827 0.0 3.8370888985118827
3.8370888985118827 3.8370888985118827 0.0
$end

$unitcell
ABSOLUTE
 0 1
Li 0.0 0.0 0.0
H 3.8370888985118827 0.0 0.0
$end

$rem
jobtype = sp
method = rVV10
basis = szv-gth
basis2 = szv-gth
```

```
kecut = 200
scf_convergence = 5
scf_algorithm = gdm
$end
```

with the `lattice` and `unitcell` sections defining the lattice and unit cell, respectively.

2.2.2 Testing and Exploring QC-PBC

QC-PBC is shipped with a small number of test jobs which are located in the `$QCPBC/libpbc/samples` directory. These sample jobs provide a good starting point for running QC-PBC jobs.

Chapter 3

QC-PBC Inputs

3.1 Atomistic Simulation Environment

The easiest way to make an input for QC-PBC is using the interface through Atomistic Simulation Environment (ASE), whose documentation available on https://wiki.fysik.dtu.dk/ase/about.html. Furthermore, the ASE interface can also directly import structures available from Materials Project (https://next-gen.materialsproject.org/) and offers a comprehensive interface for setting up the input for QC-PBC jobs. While the users can subsequently modify the input for customization, the ASE inputs use the most commonly used default input parameters for running self-consistent field calculations. In the end, the prepared input needs to be run by appropriate commandline commands that will be described in the next few sections.

3.2 General Form

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

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

Section Name	Description
$unitcell	Specifies the atoms and their coordinates within the unit cell (input file requisite).
$lattice	The lattice vectors that specify the unit cell (input file requisite).
$rem	Job specification and customization parameters (input file requisite).

Continued on next page

Table 3.1 – Continued from previous page

Section Name	Description
$aux_basis	User-defined auxiliary basis set for resolution-of-identity calculations (Chapter 7.5).
$basis	User-defined basis set information (Chapter 7).
$basis2	User-defined basis set information for basis2 job (Chapter 4.6.4).
$comment	User comments for inclusion into output file.
$pseudo	User-defined effective pseudopotentials (Section 7.8.4).
$geom_opt	Geometry optimization controls for LIBOPT3 calculations (Section 8.2).
$kpoints	User-defined set of **k**-points.
$xc_functional	User-defined DFT exchange-correlation functional (Section 5.3.7).

The keywords *$rem*, *$unitcell*, and *$lattice* are required in any QC-PBC input file

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

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

The second general aspect of QC-PBC input is that there are effectively two input sources:

- User input file (required)
- Internal program defaults and calculation results (built-in)

In general, users will need to enter variables for the *$unitcell*, *$lattice* 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 Unit Cell Coordinate Input (*$unitcell*)

3.3.1 Introduction

The *$unitcell* section communicates to the program the charge, spin multiplicity, and geometry of the unit cell being considered. The unit cell coordinates input begins with a keyword - either ABSOLUTE or RELATIVE, indicating whether geometry will be provided in absolute or relative coordinates. The next line consists of two integers: the net charge and the spin multiplicity of the molecule. The net charge can be any integer, including 0 for neutral unit cells, positive for cations, negative for anions (periodic systems are generally neutral). The multiplicity can be any integer as well (1 for a singlet, 2 for a doublet, 3 for a triplet, *etc.*). Each subsequent line of the molecular coordinate input corresponds to a

single atom in the molecule (or dummy atom), regardless of whether using Z-matrix internal coordinates or Cartesian coordinates.

Note: The coordinate system used for declaring an initial molecular geometry by default does not affect that used in a geometry optimization procedure. See Chapter 8.1 which discusses the geometry optimization packages in further detail.

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

3.3.2 Specifying the Molecular Coordinates Manually

Currently, coordinates in the *$unitcell* section must be given in cartesian coordinates. Unlike Q-CHEM, the rotation of the atoms matters due to the presence of the lattice, so atoms are not reoriented.

3.3.2.1 Cartesian Coordinates

QC-PBC 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 QC-PBC format for declaring a molecular geometry using Cartesian coordinates (in Ångstroms) is:

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

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

Note: If RELATIVE is specified, the coordinates of each atom are expressed as fractions of the lattice vectors.

Example 3.3.1 Atomic number Cartesian coordinate input for H_2O. The first line species the molecular charge (0) and multiplicity (1=singlet), respectively.

```
$unitcell
   ABSOLUTE
   0 1
   8   0.000000   0.000000  -0.212195
   1   1.370265   0.000000   0.848778
   1  -1.370265   0.000000   0.848778
$end
```

Example 3.3.2 Atomic symbol Cartesian coordinate input for H_2O.

```
$unitcell
   ABSOLUTE
   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. Atom positions can be expressed in absolute or relative coordinates.

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

3.3.2.2 Dummy Atoms

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

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

3.3.2.3 Ghost Atoms

Ghost atoms are supported and indicated by the identifier @ followed by the atomic symbol. (*e.g.*, $@H$, $@He$, ...) Ghost atoms are used to include the basis functions of an atom without including the corresponding electrons or nuclear charge.

Example 3.3.3 Atomic symbol Cartesian coordinate input for H_2O, including basis functions from a neighboring H_2O molecule.

```
$unitcell
   ABSOLUTE
   0 1
   O    0.000000   0.000000  -0.212195
   H    1.370265   0.000000   0.848778
   H   -1.370265   0.000000   0.848778
   @O   0.000000   1.000000  -0.212195
   @H   1.370265   1.000000   0.848778
   @H  -1.370265   1.000000   0.848778
$end
```

3.4 Job Specification: The *$rem* Input Section

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

```
REM_VARIABLE     VALUE     [ comment ]
```

or alternatively

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

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

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

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

Note:

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

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

REM_VARIABLE_NAME
A short description of what the variable controls.
TYPE:
The type of variable (INTEGER, LOGICAL or STRING)
DEFAULT:
The default value, if any.
OPTIONS:
A list of the options available to the user.
RECOMMENDATION:
A brief recommendation, where appropriate.

For example:

MADELUNG
Controls whether to use the Madelung correction for exact exchange energies.
TYPE:
LOGICAL
DEFAULT:
TRUE Add the Madelung correction to exact exchange energies.
OPTIONS:
FALSE Do not use Madelung correction.
RECOMMENDATION:
Madelung correction should generally be included when using methods including exact exchange unless a different finite size correction is employed.

If a default setting is indicated for a particular *$rem* variable, then it is not necessary to declare that variable in order for the default setting to be used. For example, the default value for the variable JOBTYPE is SP, indicating a single-point energy calculation, so to perform such a calculation the user does **not** need to set the JOBTYPE variable. To perform a geometry optimization, however, it is necessary to override this default by setting JOBTYPE = OPT.

QC-PBC provides defaults for most *$rem* variables, but the user will always have to stipulate a few others. In a single point energy calculation, for example, the minimum requirements will be BASIS (defining the basis set) and METHOD

(defining the level of theory for correlation and exchange). For example, METHOD = HF invokes a Hartree-Fock calculation, whereas METHOD = PBE specifies a DFT calculation using the PBE functional.

Example 3.3.4 Example of minimal *$rem* requirements to run a PBE/6-31G* single-point energy calculation.

```
$rem
  BASIS          6-31G*   Just a small basis set
  METHOD         pbe      The PBE density functional
$end
```

Note: While PSEUDO is not required, it is not recommended to run QC-PBC without a pseudopotential due to accuracy issues.

Note: While KECUT is not required, it is recommended to always specify this value as it is application dependent and the default value will not always be satisfactory.

The level of theory can alternatively be specified by setting values for two other *$rem* variables, EXCHANGE (defining the level of theory to treat exchange) and CORRELATION (defining the level of theory to treat electron correlation, if required).

For DFT calculations, METHOD specifies an exchange-correlation functional; see Section 5.4 for a list of supported functionals.

3.5 Batch Jobs: Multiple Inputs in a Single File

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

Some limitations should be kept in mind:

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

The following example requests a batch job consisting of (i) a PBE/6-31G* geometry optimization; followed by (ii) a PBE0/6-31G* single point job that begins from PBE MO coefficients (drastically reducing the number of iterations

required), and finally a MP2 energy calculation using 6-311G(d,p).

Example 3.5 Example of using information from previous jobs in a single input file.

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

$lattice
3
3.0 0.0 0.0
0.0 3.0 0.0
0.0 0.0 3.0
$end

$unitcell
   0 1
   H  0  0  0
   H  0  0  1.1
$end

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

@@@

$comment
   Energy at PBE0/6-31G*
$end

$lattice
3
3.0 0.0 0.0
0.0 3.0 0.0
0.0 0.0 3.0
$end

$unitcell
read
$end

$rem
   JOBTYPE       sp      ! Single point energy
   METHOD        pbe0
   BASIS         6-31G*
   SCF_GUESS     read    ! Use previous job's orbitals
$end

@@@

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

$lattice
3
3.0 0.0 0.0
0.0 3.0 0.0
0.0 0.0 3.0
$end

$unitcell
read
$end

$rem
```

3.6 QC-PBC Output File

When QC-PBC is invoked using

```
# qcpbc -i infile -o outfile
```

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

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

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

Chapter 4

Self-Consistent Field Ground-State Methods

4.1 Introduction

Theoretical "model chemistries"[8] involve two principal approximations. One must specify, first of all, the type of atomic orbital (AO) basis set that will be used to construct molecular orbitals (MOs), via the "linear combination of atomic orbitals" (LCAO) *ansatz*, available options for which are discussed in Chapter 7. Second, one must specify the manner in which the instantaneous interactions between electrons ("electron correlation") are to be treated. Self-consistent field (SCF) methods, in which electron correlation is described in a mean-field way, represent the simplest, most affordable, and most widely-used electronic structure methods. The SCF category of methods includes both Hartree-Fock (HF) theory as well as Kohn-Sham (KS) density functional theory (DFT). This Chapter summarizes QC-PBC's SCF capabilities, while Chapter 5 provides further details specific to DFT calculations. Chapter 6 describes the more sophisticated (but also more computationally expensive!) post-HF, wave function-based methods for describing electron correlation. If you are new to quantum chemistry, we recommend an introductory textbook such as Refs. 8, 23, or 10.

Section 4.2 provides the theoretical background behind SCF methods, including both HF and KS-DFT. In some sense, the former may be considered as a special case of the latter, and job-control *$rem* variables are much the same in both cases. Basic SCF job control is described in Section 4.5. Later sections introduce more specialized options that can be consulted as needed. Of particular note are the following:

- Initial guesses for SCF calculations (Section 4.6). Modification of the guess is recommended in cases where the SCF calculation fails to converge.
- Changing the SCF convergence algorithm (Section 4.7) is also a good strategy when the SCF calculation fails to converge.
- Several reduced-cost SCF methods are available for large systems (see Section 4.8).

4.2 Theoretical Background

4.2.1 Molecular SCF and LCAO Approximations

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

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

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

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

in atomic units, where

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

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

Separating the motions of the electrons from that of the nuclei, an idea originally due to Born and Oppenheimer,[2] yields the electronic Hamiltonian operator

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

The solution of the corresponding electronic Schrödinger equation,

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

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

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

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

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

The first approximation used to solve Eq. (4.5) is the independent-electron (mean-field) approximation, in which the wave function is approximated as an antisymmetrized product of one-electron functions, namely, the MOs. Each MO is determined by considering the electron as moving within an average field of all the other electrons. This affords the well-known Slater determinant wave function[20,21]

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

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

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

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

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

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

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

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

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

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

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

and $\boldsymbol{\varepsilon}$ is a diagonal matrix containing the orbital energies. Generalizing to an unrestricted formalism by introducing separate spatial orbitals for α and β spin in Eq. (4.7) yields the Pople-Nesbet equations [16]

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

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

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

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

4.2.2 Periodic SCF and LCAO Approximations

In periodic applications, the well known Bloch's theorem states that due to the translational symmetry of the system, the MOs used within the LCAO procedure must be periodic along the lattice, up to some phase factor:

$$\chi^{\mathbf{k}}(\mathbf{r} + \mathbf{T}) = e^{i\mathbf{k}\cdot\mathbf{T}}\chi^{\mathbf{k}}(\mathbf{r}) \tag{4.15}$$

Calculation	***$rem* Variable JOBTYPE**	
Single point energy (default)	SINGLE_POINT or SP	
Force (energy + gradient)	FORCE	
Equilibrium structure search	OPTIMIZATION or OPT	(Ch. 8)
Energy decomposition analysis	EDA	(Sections **??** and **??**)

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

where $\mathbf{T}$ is any lattice vector built from an integer number t_i of each primitive lattice vector:

$$\mathbf{T} = \sum_{i}^{N_{\text{dim}}} \mathbf{A}_i t_i \tag{4.16}$$

A reframing of this equation allows the orbitals to be written as the product of a cell periodic function $u(\mathbf{r})$ and a plane wave with momentum $\mathbf{k}$ (called a $\mathbf{k}$-point or crystal momentum):

$$\Psi^{\mathbf{k}}(\mathbf{r}) = u^{\mathbf{k}}(\mathbf{r}) e^{i\mathbf{k}\cdot\mathbf{r}} \tag{4.17}$$

Any $\mathbf{k}$-point is a valid solution of the Schrödinger equation, so one must integrate over all $\mathbf{k}$-points (the Brillouin zone) in order to reach the thermodynamic limit. It may be shown that only $\mathbf{k}$-points within a single reciprocal space unit cell are unique, so one needs only integrate over a single reciprocal space unit cell (known as the first Brillouin zone). This is typically done numerically using a uniform grid of points in $\mathbf{k}$-space, known as a Monkhorst-Pack mesh. A typical periodic calculation may use on the order of dozens to thousands of $\mathbf{k}$-points.

Within the SCF procedure, the Fock matrix is block diagonal - there are no matrix elements between orbitals of different $\mathbf{k}$-points. This means that there are independent Roothaan-Hall equations for each $\mathbf{k}$-point which may be solved individually for each $\mathbf{k}$-point to get a set of MOs and orbital energies at each $\mathbf{k}$-point:

$$\mathbf{F}^{\mathbf{k}}\mathbf{C}^{\mathbf{k}} = \epsilon^{\mathbf{k}}\mathbf{S}^{\mathbf{k}}\mathbf{C}^{\mathbf{k}} \tag{4.18}$$

The Pople-Nesbet equations may be similarly generalized:

$$\mathbf{F}^{\alpha\mathbf{k}}\mathbf{C}^{\alpha\mathbf{k}} = \varepsilon^{\alpha\mathbf{k}}\mathbf{S}^{\mathbf{k}}\mathbf{C}^{\alpha\mathbf{k}} \tag{4.19}$$

$$\mathbf{F}^{\beta\mathbf{k}}\mathbf{C}^{\beta\mathbf{k}} = \varepsilon^{\beta\mathbf{k}}\mathbf{S}^{\mathbf{k}}\mathbf{C}^{\beta\mathbf{k}} \tag{4.20}$$

While the Fock matrix itself is block diagonal in $\mathbf{k}$-space, the elements of matrix $\mathbf{F}^{\mathbf{k_1}}$ depends on the MOs at all other $\mathbf{k}$-points. One must then perform the SCF procedure for each $\mathbf{k}$-point in parallel. At the end of the procedure, instead of a discrete set of orbital energies, one rather obtains a set of bands, that are continuous in the $\mathbf{k}$ dimension.

So the main changes required for periodic SCF calculations are:

1. A grid of $\mathbf{k}$-points must be determined before the calculation.
2. A basis set which obeys Bloch theorem at each $\mathbf{k}$-point must be used.
3. At each SCF iteration, the Fock matrix at each $\mathbf{k}$-pointmust be built and then solved.

Types of ground-state energy calculations currently available in QC-PBC are summarized in Table 4.1.

MP_MESH

The number of k-points along each of the 3 reciprocal lattice vectors included in the Monkhorst-Pack mesh.

TYPE:

LIST[INTEGER]

DEFAULT:

None

OPTIONS:

None	A Γ-point calculation is run
$[N_{k_1}, N_{k_2}, N_{k_3}]$	A **k**-point calculations is run with $N_{k_1} * N_{k_2} * N_{k_3}$ total **k**-points

RECOMMENDATION:

Periodic calculations should be converged with respect to the number of **k**-points, however the number required is highly system dependent. Therefore one should increase the mesh size until the energy is converged.

Example 4.1 Input for a TPSS calculation on AlN using a Monkhorst-Pack mesh with 4x4x4 = 64 total **k**-points.

```
$comment
Run DFT energy calculation with k-points
$end

$lattice
3 ! dimension
3.12858800000000 0.00000000000000 0.00000000000000
-1.56429400000000 2.70943700000000 0.00000000000000
0.00000000000000 0.00000000000000 5.01695500000000
$end

$unitcell
absolute
0 1
Al 1.5642955643 0.9031447635 2.5049004111
Al -0.0000015643 1.8062922365 5.0133779111
N 1.5642955643 0.9031447635 4.4184974889
N -0.0000015643 1.8062922365 1.9100199889
$end

$rem
    JOBTYPE     sp                  ! energy calculation
    METHOD      blyp                ! use the TPSS mgga functional
    BASIS       szv-gth             ! double zeta basis
    KECUT       1500                ! GPW planewave cutoff 1500 eV
    PSEUDO      gth-pbe             ! GPW requires pseudopotential in most cases
    MP_MESH     [4,4,4]             ! 4x4x4 k-point Monkhorst-Pack mesh
$end
```

View output online

4.2.3 Hartree-Fock Theory

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

Consider once more the Roothaan-Hall equations, Eq. (4.18), or the Pople-Nesbet equations, Eq. (4.19), which can be traced back to Eq. (4.8), in which the effective potential v_{eff} depends on the SCF methodology. In a restricted HF (RHF) formalism, the effective potential can be written as

$$v_{\text{eff}} = \sum_{\mathbf{k_2}} \sum_{i}^{N/2} \left[2\hat{J}_i^{\mathbf{k_2}}(1) - \hat{K}_i^{\mathbf{k_2}}(1) \right] - \sum_{A=1}^{M} \frac{Z_A}{r_{1A}} \tag{4.21}$$

where the Coulomb and exchange operators are defined as

$$\hat{J}_i^{\mathbf{k_2}}(1) = \int \psi_i^{\mathbf{k_2}*}(2) \frac{1}{r_{12}} \psi_i^{\mathbf{k_2}}(2)\, d\mathbf{r}_2 \tag{4.22}$$

and

$$\hat{K}_i^{\mathbf{k_2}}(1)\psi_p^{\mathbf{k}}(1) = \left[\int \psi_i^{\mathbf{k_2}*}(2) \frac{1}{r_{12}} \psi_p^{\mathbf{k}}(2)\, d\mathbf{r}_2 \right] \psi_i^{\mathbf{k_2}}(1) \tag{4.23}$$

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

$$F_{\mu\nu}^{\mathbf{k}} = H_{\mu\nu}^{\text{core}\mathbf{k}} + J_{\mu\nu}^{\mathbf{k}} - K_{\mu\nu}^{\mathbf{k}} \tag{4.24}$$

where the core Hamiltonian matrix elements

$$H_{\mu\nu}^{\text{core}\mathbf{k}} = T_{\mu\nu}^{\mathbf{k}} + V_{\mu\nu}^{\mathbf{k}} \tag{4.25}$$

consist of kinetic energy elements

$$T_{\mu\nu}^{\mathbf{k}} = \int \phi_\mu^{\mathbf{k}}(\mathbf{r})^* \left(-\frac{1}{2}\hat{\nabla}^2 \right) \phi_\nu^{\mathbf{k}}(\mathbf{r})\, d\mathbf{r} \tag{4.26}$$

and nuclear attraction elements

$$V_{\mu\nu}^{\mathbf{k}} = \int \phi_\mu^{\mathbf{k}}(\mathbf{r})^* \left(-\sum_A \frac{Z_A}{|\mathbf{R}_A - \mathbf{r}|} \right) \phi_\nu^{\mathbf{k}}(\mathbf{r})\, d\mathbf{r} \tag{4.27}$$

The Coulomb and exchange elements are given by

$$J_{\mu\nu}^{\mathbf{k}} = \sum_{\lambda\sigma\mathbf{k}_2} P_{\sigma\lambda}^{\mathbf{k}_2} \left(\mu^{\mathbf{k}}\nu^{\mathbf{k}} | \lambda^{\mathbf{k}_2}\sigma^{\mathbf{k}_2} \right) \tag{4.28}$$

and

$$K_{\mu\nu}^{\mathbf{k}} = \frac{1}{2} \sum_{\lambda\sigma} P_{\lambda\sigma}^{\mathbf{k}_2} \left(\mu^{\mathbf{k}}\lambda^{\mathbf{k}_2} | \sigma^{\mathbf{k}_2}\nu^{\mathbf{k}} \right) \tag{4.29}$$

respectively, where the density matrix elements are

$$P_{\mu\nu}^{\mathbf{k}} = 2 \sum_{i=1}^{N/2} C_{\mu i}^{\mathbf{k}} C_{\nu i}^{\mathbf{k}*} \tag{4.30}$$

and the two electron integrals are

$$\left(\mu^{\mathbf{k}_1}\nu^{\mathbf{k}_2} | \lambda^{\mathbf{k}_3}\sigma^{\mathbf{k}_4} \right) = \int\int \phi_\mu^{\mathbf{k}_1}(\mathbf{r}_1)^* \phi_\nu^{\mathbf{k}_2}(\mathbf{r}_1) \left(\frac{1}{r_{12}} \right) \phi_\lambda^{\mathbf{k}_3}(\mathbf{r}_2)^* \phi_\sigma^{\mathbf{k}_4}(\mathbf{r_2})\, d\mathbf{r}_1\, d\mathbf{r}_2\,. \tag{4.31}$$

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.

Substituting the matrix element in Eq. (4.24) back into the Roothaan-Hall equations, Eq. (4.18), and iterating until self-consistency is achieved will yield the RHF energy and wave function. Alternatively, one could have adopted the

unrestricted form of the wave function by defining separate α and β density matrices:

$$\begin{aligned} P_{\mu\nu}^{\alpha\mathbf{k}} &= \sum_{i=1}^{n_\alpha} C_{\mu i}^{\alpha\mathbf{k}} C_{\nu i}^{\alpha\mathbf{k}*} \\ P_{\mu\nu}^{\beta\mathbf{k}} &= \sum_{i=1}^{n_\beta} C_{\mu i}^{\beta\mathbf{k}} C_{\nu i}^{\beta\mathbf{k}*} \end{aligned} \tag{4.32}$$

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

$$F_{\mu\nu}^{\alpha\mathbf{k}} = H_{\mu\nu}^{\mathrm{core}\mathbf{k}} + J_{\mu\nu}^{\mathbf{k}} - K_{\mu\nu}^{\alpha\mathbf{k}} \,, \tag{4.33}$$

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

$$K_{\mu\nu}^{\alpha\mathbf{k}} = \sum_{\mathbf{k}_2}^{N_k} \sum_{\lambda}^{N} \sum_{\sigma}^{N} P_{\lambda\sigma}^{\alpha\mathbf{k}_2} \left(\mu^{\mathbf{k}} \lambda^{\mathbf{k}_2} | \sigma^{\mathbf{k_2}} \nu^{\mathbf{k}}\right) \tag{4.34}$$

4.3 Gaussian Plane Wave (GPW)

QC-PBC uses Gaussian type orbitals as the basis for all SCF calculations. To evaluate the Fock matrix elements described in the previous section, one must sum over the entire (infinite) periodic lattice. This can be slow and non-convergent for the two-electron integrals. QC-PBC uses the popular the Gaussian plane wave (GPW) approach for evaluating periodic matrix elements between GTOs is. With this method, products of orbitals are expanded in a auxiliary basis of plane waves:

$$\phi_p^{\mathbf{k_1}}(\mathbf{r})^* \phi_q^{\mathbf{k_2}}(\mathbf{r}) = \sum_{\mathbf{G}} C_{p^{\mathbf{k_1}} q^{\mathbf{k_2}}}^{\mathbf{G}} e^{i\mathbf{G}\cdot\mathbf{r}} \tag{4.35}$$

The fitting coefficients $C_{pq}^{\mathbf{G}}$ are easily obtained via fast Fourier transform (FFT) in $\mathcal{O}(N_g \log N_g)$ time, however this requires that $\phi_p(\mathbf{r})$ and $\phi_q(\mathbf{r})$ are evaluated on a uniform grid in real space. This form of density fitting will be more accurate if a larger auxiliary basis set is used (i.e. more plane waves and a finer real space grid). In practice, the size of the auxiliary planewave basis set is determined from a single parameter - a cutoff for the maximum value of $\mathbf{G}$ to be used.

GPW has the advantage that the electron repulsion integrals used in J and K are easy to evaluate as the coulomb operator is diagonal in $\mathbf{G}$-space.

$$(\mu^{\mathbf{k_1}} \nu^{\mathbf{k_2}} | \lambda^{\mathbf{k_3}} \sigma^{\mathbf{k_4}}) = \sum_{\mathbf{G}} C_{\mu^{\mathbf{k_1}} \nu^{\mathbf{k_2}}}^{\mathbf{G}} C_{\lambda^{\mathbf{k_3}} \sigma^{\mathbf{k_4}}}^{\mathbf{G}} \frac{4\pi}{|G|^2} \tag{4.36}$$

GPW however has the disadvantages of requiring basis functions to be stored on a grid in real space at a memory cost of $\mathcal{O}(N_k N)$ when utilizing sparsity. This memory requirement can become demanding for large systems and large numbers of $\mathbf{k}$-points. Additionally, if basis functions are too compact, the number of plane waves required for an accurate fit via GPW will be prohibitively large. GPW must therefore be used in conjunction with pseudopotentials to avoid explicit modelling of core electrons.

KECUT

Sets the kinetic energy cutoff of the auxiliary basis set in eV.

TYPE:

INTEGER

DEFAULT:

500

OPTIONS:

n for a kecut of n eV.

RECOMMENDATION:

KECUT should be set as low as possible while still maintaining the desired accuracy. For pseudopotential calculations, somewhere between 500 and 1000 eV is often adequate. One may verify a desired choice of KECUT by increasing the value and ensuring negligible change in the converged energy.

Example 4.2 Input for an energy calculation on MgO using the SCAN functional. GPW is used with a planewave cutoff of 1500 eV.

```
$comment
Run DFT energy calculation using GPW
$end

$lattice
3 ! dimension
2.56829206000000 -0.00000000000000 1.48280459000000
0.85609768000000 2.42140880000000 1.48280459000000
-0.00000001000000 -0.00000000000000 2.96560719000000
$end

$unitcell
absolute
0 1
Mg 0.0000000000 0.0000000000 0.0000000000
O 1.7121948650 1.2107044000 2.9656081850
$end

$rem
    JOBTYPE     sp                  ! energy calculation
    METHOD      scan                ! use the SCAN functional
    BASIS       szv-gth             ! double zeta basis
    KECUT       1500                ! GPW planewave cutoff 1500 eV
    PSEUDO      gth-pbe             ! GPW requires pseudopotential in most cases
    MP_MESH     [3,3,3]             ! 3x3x3 k-point Monkhorst-Pack mesh
$end
```

View output online

4.3.1 Shellpair GPW evaluation

Rather than evaluating the basis functions on the real space grid (which requires a large amount of memory), one may directly evaluate products of primitive gaussian basis functions (a shellpair). The product of two Gaussian functions is another Gaussian function, allowing one to directly evaluate this function (and optimally screen it). This approach may be further optimized by separating the primitive Gaussian type orbital shellpair into x,y,z components and evaluating each 1-dimensional component before recombining into the full grid. By contracting the x,y,z components individually with the density matrix P (see equation 4.37, the electron density $\rho(\mathbf{r})$ may be quickly evaluated with very little storage

cost. This approach was has been used by CP2K[26] for many years by developing basis sets with only a small number of primitive Gaussian functions shared between all basis function shells. For further reading on the implementation of this algorithm in QC-PBC, we direct users to Dinh et al.[3].

$$\begin{aligned}\rho(\mathbf{r}) \mathrel{+}= \sum_{\substack{l_{x_a},l_{y_a},l_{z_a}\\ l_{x_b},l_{y_b},l_{z_b}}} & P^{ab}_{l_{x_a},l_{y_a},l_{z_a},l_{x_b},l_{y_b},l_{z_b}} \\ & (x-X_a)^{l_{x_a}}(x-X_b)^{l_{x_b}} e^{-\alpha_a|x-X_a|^2} e^{-\alpha_b|x-X_b|^2} \\ & (y-Y_a)^{l_{y_a}}(y-Y_b)^{l_{y_b}} e^{-\alpha_a|y-Y_a|^2} e^{-\alpha_b|y-Y_b|^2} \\ & (z-Z_a)^{l_{z_a}}(z-Z_b)^{l_{z_b}} e^{-\alpha_a|z-Z_a|^2} e^{-\alpha_b|z-Z_b|^2},\end{aligned} \tag{4.37}$$

If the unitcell is orthorhombic, then the GTO shellpairs will perfectly separate along the real-space grid, however for non-orthorhombic cells, further transformations must be performed and the Gaussian cannot be perfectly separated, making this algorithm slightly more computationally intensive. This approach works for evaluating $\rho(\mathbf{r})$, and the reverse algorithm may be used to contract a potential $V(\mathbf{r})$ with shellpairs to get a fock matrix (i.e. J and xc). Additionally, one may take $\rho(\mathbf{r})$ computed via this approach and evaluate $\nabla\rho(\mathbf{r})$ numerically via FFT, greatly speeding up GGA functionals. However, as of yet there is no approach for evaluating exact exchange via this method.

Testing has shown this approach to be especially effective for systems with a large degree of sparsity (i.e. surfaces), as well as systems with a large number of **k**-points (since this approach sums over the real-space BvK cell). We suggest the use of CP2K basis sets with shellpair evaluation to maximize performance.

GPW_METHOD

Sets the method for evaluating the Fock matrix via GPW.

TYPE:

STRING

DEFAULT:

CGTO

OPTIONS:

CGTO Compute the basis functions $\phi^{\boldsymbol{k}}_{\mu}(\boldsymbol{r})$ on a real space grid at the start of the SCF.

SHELLPAIR Compute $\rho(\boldsymbol{r})$ on the fly using the shellpair-based algorithm.

RECOMMENDATION:

CGTO should be used for small compact unitcells. SHELLPAIR should be used for sparse unitcells, large k-point meshes, or when memory is limited.

SHELLPAIR_ORTHO

Use the separability of the xyz components of the shellpair to speed up the calculation if the unitcell is orthorhombic.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Utilize the separability of the shellpairs into 1d grids, only if the unitcell is orthorhombic.

FALSE Do not use utilize the separability of the shellpairs into 1d grids if the unitcell is orthorhombic.

RECOMMENDATION:

Use the default (TRUE) to ensure the best performance.

SHELLPAIR_GRAD_RECIP

Evaluate $\nabla\rho(\mathbf{r})$ numerically in reciprocal space via FFT.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Compute $\nabla\rho(\mathbf{r})$ numerically starting from $\rho(\mathbf{r})$.

FALSE Compute $\nabla\rho(\mathbf{r})$ analytically.

RECOMMENDATION:

Use the default (TRUE) unless a low KECUT value is used - which can lead to a large amount of numerical error.

Example 4.3 Input for an LDA energy calculation of graphite using GPW with $\rho(\mathbf{r})$ evaluated via the shellpair algorithm.

```
$lattice
3 ! dimension
2.46000   0.00000   0.00000
1.23000   2.13000   0.00000
0.00000   0.00000  20.00000
$end

$unitcell
absolute
0 1
C      0.000000    0.000000    0.000000
C      1.230000    0.710000    0.000000
$end

$rem
jobtype sp            ! sp energy
method lda            ! lda functional
basis szv-gth         ! double zeta basis set
kecut 1500            ! planewave cutoff 1500 eV
pseudo gth-pbe        ! GTH-PBE pseudopotential
mp_mesh [3, 3, 3]     ! 27 k-points
scf_convergence 5     ! stop SCF when error is below 1e-5
gpw_method shellpair  ! use shellpair algorithm
$end
```

View output online

4.4 Density Fitting (DF)

$$(\mu\nu|\lambda\sigma) = \sum_{P}(\mu\nu|P)(P|Q)^{-1}(Q|\lambda\sigma) \tag{4.38}$$

1. CGTO-DF can be employed to accelerate the generation of the two-electron three-center integral.
2. Recommendation: ω should be decreased for supercell calculation.

4.5 Basic SCF Job Control

4.5.1 Introduction

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

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

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

- Restricted (R): typically appropriate for closed shell molecules at their equilibrium geometry, where electrons occupy orbitals in pairs.
- Unrestricted (U): appropriate for radicals with an odd number of electrons, and also for molecules with even numbers of electrons where not all electrons are paired, *e.g.*, stretched bonds and diradicals.
- The use of complex orbitals : restricted (CR) and unrestricted (CU), appropriate for **k**-point calculations where orbitals are inherently complex.

Aspects of an SCF calculation such as the SCF guess, the use of efficient algorithms to construct the Fock matrix like occ-RI-K (see Section 4.8.3), are unaffected by the use of GEN_SCFMAN. Likewise, using GEN_SCFMAN does not make any difference to the post-SCF procedures such as correlated methods, excited state calculations and evaluation of molecular properties.

4.5.2 Job Control

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

METHOD

Specifies the exchange-correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

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

RECOMMENDATION:

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

BASIS

Specifies the basis sets to be used.

TYPE:

STRING

DEFAULT:

No default basis set

OPTIONS:

General, Gen	User defined (*$basis* keyword required).
Symbol	Use standard basis sets as per Chapter 7.
Mixed	Use a mixture of basis sets (see Chapter 7).

RECOMMENDATION:

Consult literature and reviews to aid your selection.

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

SCF_CONVERGENCE

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

TYPE:

INTEGER

DEFAULT:

8 For all calculations.

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	Closed-shell systems.
TRUE	Open-shell systems.

OPTIONS:

FALSE	Constrain the spatial part of the alpha and beta orbitals to be the same.
TRUE	Do not Constrain the spatial part of the alpha and beta orbitals.

RECOMMENDATION:

Use the default. Note that for unrestricted calculations on systems with an even number of electrons it is usually necessary to break α/β symmetry in the initial guess

The calculations using other more special orbital types are controlled by the following *$rem* variables (they are not

effective if GEN_SCFMAN = FALSE):

Example 4.4 Unrestricted DFT calculation for stretched H_2.

```
$unitcell
     ABSOLUTE
     0 1
     H 5 5 5
     H 5 5 7
$end

 $lattice
     3
     10 0 0
     0 10 0
     0 0 10
 $end

 $rem
     JOBTYPE            sp
     METHOD             pbe
     BASIS              szv-gth
     KECUT              700
     UNRESTRICTED       true
     INTERNAL_STABILITY true
     SCF_ALGORITHM      gdm
     SCF_CONVERGENCE    8
     THRESH             14
 $end
```

View output online

4.5.3 Additional Options

Listed below are a number of useful options to customize an SCF calculation. This is only a short summary of the function of these *$rem* variables. Several important sub-topics are discussed separately, including reduced scaling methods for large systems (Section 4.8), customizing the initial guess (Section 4.6), and converging the SCF calculation (Section 4.7).

S2THRESH

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

TYPE:

INTEGER

DEFAULT:

Same as THRESH.

OPTIONS:

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

RECOMMENDATION:

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

THRESH

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

TYPE:

INTEGER

DEFAULT:

8 For single point energies.

10 For optimizations and frequency calculations.

14 For coupled-cluster calculations.

OPTIONS:

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

RECOMMENDATION:

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

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 QC-PBC output file.

TYPE:

INTEGER

DEFAULT:

0 Minimal, concise, useful and necessary output.

OPTIONS:

0 Minimal, concise, useful and necessary output.

2 Level 0 plus component breakdown of SCF electronic energy each cycle.

RECOMMENDATION:

Default, unless debugging, then use level 2.

4.5.4 Examples

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

Example 4.5 Example QC-PBC input for a single point energy calculation on water. Note that the declaration of the single point *$rem* variable is redundant because it is the same as the QC-PBC default.

```
$lattice
    3 ! dimension
    2.16330216000000 3.76300538000000 0.00000000000000
    -2.16330216000000 3.76300538000000 0.00000000000000
    0.00000000000000 0.00000000000000 7.06770668000000
$end

$unitcell
    absolute
    0 1
    H 0.0000000000 2.5289709649 4.9523547219
    H 0.0000000000 4.9970397951 1.4185013819
    H 0.0000000000 3.4655001426 3.6501759182
    H 0.0000000000 4.0605106174 0.1163225782
    H 0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 2.0160395022 6.9469368180
    H 0.8004007935 2.0160395022 6.9469368180
    O 0.0000000000 2.5186751564 3.9565869413
    O 0.0000000000 5.0073356036 0.4227336013
    O 0.0000000000 5.0237523919 3.0758661592
    O 0.0000000000 2.5022583681 6.6097194992
$end

$rem
    JOBTYPE sp
    METHOD  pbe
    BASIS   dzvp-gth
    PSEUDO  gth-pbe
$end
```

View output online

Example 4.6 UKS/DZVP-GTH calculation on the Li atom. Note that since the number of α and β electron differ, MOs default to an unrestricted formalism.

```
$unitcell
    ABSOLUTE
    0 2
    Li 5 5 5
$end

$lattice
    3
    10 0 0
    0 10 0
    0 0 10
$end

$rem
    JOBTYPE     sp
    METHOD      pbe
    BASIS       dzvp-gth
    PSEUDO      gth-pbe
$end
```

View output online

4.6 SCF Initial Guess

4.6.1 Introduction

The Roothaan-Hall and Pople-Nesbet equations of SCF theory are non-linear in the molecular orbital coefficients. Like many mathematical problems involving non-linear equations, prior to the application of a technique to search for a numerical solution, an initial guess for the solution must be generated. If the guess is poor, the iterative procedure applied to determine the numerical solutions may converge very slowly, requiring a large number of iterations, or at worst, the procedure may diverge.

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

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

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

1. Core Hamiltonian (CORE)
2. On-the-fly (automated) Superposition of Atomic Densities (AUTOSAD)
3. Reading previously obtained MOs from disk. (READ)
4. Basis set projection (BASIS2)

The first two of these guesses are built-in, and are briefly described in Section 4.6.2. The option of reading MOs from disk is described in Section 4.6.3. Finally, QC-PBC's novel basis set projection method is discussed in Section **??**.

4.6.2 Initial Guess Types

Core Hamiltonian The core Hamiltonian guess simply obtains the guess MO coefficients by diagonalizing the core Hamiltonian matrix in Eq. (4.25). It is also commonly known as the one-electron guess, as it completely ignores interelectronic interactions. Although the guess is exact for one-electron systems, the lack of repulsion effects leads to incorrect shell structure of atoms as well as all electrons crowding onto the heaviest atom in the system; see Ref. 13 for a discussion. Due to these effects, the core guess is typically extremely inaccurate and should only be used as a last resort; much better alternatives are provided by SAD.

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

case the AUTOSAD density is simply equivalent to the pretabulated SAD density. The pretabulated SAD guess is based on Hartree-Fock atomic densities whereas AUTOSAD uses the SCF method specified in *$rem*.

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:
AUTOSAD For all calculations.
OPTIONS:
CORE Diagonalize core Hamiltonian
AUTOSAD On-the-fly superposition of atomic densities
READ Read previous MOs from disk
FRAGMO Construct guess MOs from fragments (See Sec. 9.2.3)
RECOMMENDATION:
AUTOSAD guess for almost all calculations. If AUTOSAD fails, use the core Hamiltonian guess as a last resort.

4.6.3 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 QC-PBC with three command line variables:.

```
localhost-1> qcpbc -i job1.in -o job1.out -c save
localhost-2> qcpbc -i job2.in -o job2.out -c 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 QC-PBC command line variable is not necessary.
   (3) As for the SAD guess, QC-PBC 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 **??**.

Example 4.7 Input for a PBE calculation on water using the autosad guess. This avoids the use of QC-PBC's core guess, which is often poor.

```
$lattice
    3 ! dimension
    2.16330216000000 3.76300538000000 0.00000000000000
    -2.16330216000000 3.76300538000000 0.00000000000000
    0.00000000000000 0.00000000000000 7.06770668000000
$end

$unitcell
    absolute
    0 1
    H 0.0000000000 2.5289709649 4.9523547219
    H 0.0000000000 4.9970397951 1.4185013819
    H 0.0000000000 3.4655001426 3.6501759182
    H 0.0000000000 4.0605106174 0.1163225782
    H 0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 2.0160395022 6.9469368180
    H 0.8004007935 2.0160395022 6.9469368180
    O 0.0000000000 2.5186751564 3.9565869413
    O 0.0000000000 5.0073356036 0.4227336013
    O 0.0000000000 5.0237523919 3.0758661592
    O 0.0000000000 2.5022583681 6.6097194992
$end

$rem
    JOBTYPE     sp
    METHOD      pbe
    BASIS       dzvp-gth
    PSEUDO      gth-pbe
    SCF_GUESS   autosad
$end
```

View output online

4.6.4 Basis Set Projection

QC-PBC 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 converged MOs from the small basis are projected into the large basis by computing the overlap between the two basis sets.
- The density matrix for the large basis set is computed from the MOs, and the target SCF calculation commences.

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

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.

Note: SCF_GUESS = BASIS_PROJ must be set to enable basis set projection guesses, even if the BASIS2 variable is set.

Example 4.8 Input where a SZV-GTH basis set projection is used to generate a good initial guess for a calculation employing the much larger QZV2P-GTH basis set.

```
$lattice
    3 ! dimension
    2.16330216000000 3.76300538000000 0.00000000000000
    -2.16330216000000 3.76300538000000 0.00000000000000
    0.00000000000000 0.00000000000000 7.06770668000000
$end

$unitcell
    absolute
    0 1
    H 0.0000000000 2.5289709649 4.9523547219
    H 0.0000000000 4.9970397951 1.4185013819
    H 0.0000000000 3.4655001426 3.6501759182
    H 0.0000000000 4.0605106174 0.1163225782
    H 0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 2.0160395022 6.9469368180
    H 0.8004007935 2.0160395022 6.9469368180
    O 0.0000000000 2.5186751564 3.9565869413
    O 0.0000000000 5.0073356036 0.4227336013
    O 0.0000000000 5.0237523919 3.0758661592
    O 0.0000000000 2.5022583681 6.6097194992
$end

$rem
    JOBTYPE sp
    METHOD  pbe
    BASIS   qzv2p-gth
    BASIS2  szv-gth
    PSEUDO  gth-pbe
$end
```

View output online

4.6.5 Advanced AUTOSAD Options

While AUTOSAD should usually run without any additional configuration, there are cases where it may be necessary or desired to change the guess generation. AUTOSAD defaults to running each atom in the gas phase as a neutral

atom with the gas phase spin multiplicity. In general, *$rem* variables selected by the user will be carried through to AUTOSAD calculation, but the following options will by default be overridden for the AUTOSAD calculation only:

$rem variable	AUTOSAD default
JOBTYPE	SP
METHOD	PBE
SCF_ALGORITHM	DIIS_GDM
SCF_CONVERGENCE	7
POSTSCF_METHOD	NONE
INPUT_BOHR	FALSE
UNRESTRICTED	TRUE
MAX_SCF_CYCLES	200

To modify these (or any other) *$rem* variables for the AUTOSAD calculation, the *$rem_autosad* section can be used analogously to the *$rem_frgm* section. *$rem_autosad* variables will overwrite any conflicting *$rem* variables for the AUTOSAD calculation, and append any additional *$rem* variables. Variables listed in *$rem*, but neither overridden by default nor by *$rem_autosad*will still be used. Finally, the additional *$rem* variables below provide additional control related to charge, multiplicity, and periodicity. In the event that a periodic lattice is desired for AUTOSAD, the *$lattice_autosad* section can be used to provide a separate lattice for the AUTOSAD computation.

SCF_PRINT_AUTOSAD

Whether to print the AUTOSAD job to the standard out, or redirect to scratch.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE AUTOSAD job output is printed to the standard out.

FALSE AUTOSAD job output is redirected to scratch folder.

RECOMMENDATION:

FALSE is recommended unless troubleshooting a failed guess

AUTOSAD_GAMMA

Runs AUTOSAD only at gamma point. For FALSE, AUTOSAD_ISOLATED must be also set to false. If using k-points AUTOSAD, the mp-mesh may not be changed between guess and main calculation

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE AUTOSAD job uses k-points.

FALSE AUTOSAD job is a Γ-point calculation.

RECOMMENDATION:

TRUE is recommended

AUTOSAD_ATOMS

A list of atom types for which non-default autosad charges and multiplicities should be run. Must be run with the AUTOSAD_CHARGES and AUTOSAD_MULTIPLICITIES *$rem* variables

TYPE:

LIST[STRING]

DEFAULT:

(empty)

OPTIONS:

(empty) AUTOSAD runs with default charges and multiplicities.

$[A, B, \dots]$ AUTOSAD runs listed atom types with non-default charges and multiplicities.

RECOMMENDATION:

The default is generally recommended

AUTOSAD_CHARGES

A list of the non-default charges in the same order as AUTOSAD_ATOMS

TYPE:

LIST[INTEGER]

DEFAULT:

(empty)

OPTIONS:

(empty) AUTOSAD runs with default charges.

$[q_1, q_2, \dots]$ Following AUTOSAD_ATOMS, AUTOSAD runs atom A with charge q_1 and atom B with charge q_2.

RECOMMENDATION:

The default is generally recommended

AUTOSAD_MULTIPLICITIES

A list of the non-default multiplicities in the same order as AUTOSAD_ATOMS

TYPE:

LIST[INTEGER]

DEFAULT:

(empty)

OPTIONS:

(empty) AUTOSAD runs with default multiplicities

$[m_1, m_2, \dots]$ Following AUTOSAD_ATOMS, AUTOSAD runs atom A with multiplicity m_1 and atom B with multiplicity m_2

RECOMMENDATION:

The default is generally recommended

4.7 Converging SCF Calculations

4.7.1 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. QC-PBC features a number of SCF optimization algorithms which can be selected via the *$rem* variable SCF_ALGORITHM, including:

Methods that are based on extrapolation or interpolation:

- The highly successful DIIS procedures. These are the default (except for restricted open-shell SCF calculations) and are available for all orbital types (see Section 4.7.3). The damping[7] and level-shifting technique[6,9,19] can also be invoked together with DIIS (R, U only).

Methods that make use of orbital gradient:

- Geometric Direct Minimization (GDM) which is an improved and highly robust version of direct minimization (DM) and is the recommended fall-back when DIIS fails. Like DM, It can also be invoked after a few iterations with DIIS to improve the initial guess. GDM is available for all orbital types (see Section 4.7.4).
- GDM_LS: It is essentially a preconditioned (using orbital energy differences as the preconditioner) L-BFGS algorithm with line search, available for all orbital types

4.7.2 Basic Convergence Control Options

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

MAX_SCF_CYCLES

Controls the maximum number of SCF iterations permitted.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n $n > 0$ User-selected.

RECOMMENDATION:

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

SCF_ALGORITHM

Algorithm used for converging the SCF.

TYPE:

STRING

DEFAULT:

DIIS Pulay DIIS.

OPTIONS:

DIIS	Pulay DIIS.
GDM	Geometric Direct Minimization.
GDM_LS	L-BFGS algorithm with a GDM-like preconditioner and line search (see also GDM_QLS, which uses a quadratic line search algorithm)
DIIS_GDM	Use DIIS and then later switch to geometric direct minimization
HYBRID	Use a user defined sequence of SCF algorithms. See SCF_ALGORITHM_HYBRID, SCF_HYBRID_TOL, and SCF_HYBRID_MAXITER

RECOMMENDATION:

Use DIIS unless performing a restricted open-shell calculation, in which case GDM is recommended. If DIIS approaches the correct solution but fails to finally converge, GDM is the recommended fallback.

SCF_CONVERGENCE

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

TYPE:

INTEGER

DEFAULT:

5 For single point energy calculations

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

OPTIONS:

n Corresponding to 10^{-n}

RECOMMENDATION:

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

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

4.7.3 Direct Inversion in the Iterative Subspace (DIIS)

The SCF implementation of the Direct Inversion in the Iterative Subspace (DIIS) method[17,18] uses the property of an SCF solution that requires the density matrix to commute with the Fock matrix for each $\mathbf{k}$-point:

$$\mathbf{S^k P^k F^k} - \mathbf{F^k P^k S^k} = \mathbf{0} \; . \tag{4.39}$$

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

$$\mathbf{S^k P}_i^\mathbf{k} \mathbf{F}_i^\mathbf{k} - \mathbf{F}_i^\mathbf{k} \mathbf{P}_i^\mathbf{k} \mathbf{S^k} = \mathbf{e}_i^\mathbf{k} \tag{4.40}$$

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

$$\mathbf{F}_k^\mathbf{k} = \sum_{j=1}^{k-1} c_j \mathbf{F}_j^\mathbf{k} \tag{4.41}$$

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

$$Z = \left(\sum_k c_k \mathbf{e}_k^\mathbf{k} \right) \cdot \left(\sum_k c_k \mathbf{e}_k^\mathbf{k} \right) \tag{4.42}$$

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

$$\begin{pmatrix} \mathbf{e}_1 \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_1 \cdot \mathbf{e}_N & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \mathbf{e}_N \cdot \mathbf{e}_1 & \cdots & \mathbf{e}_N \cdot \mathbf{e}_N & 1 \\ 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} c_1 \\ \vdots \\ c_N \\ \lambda \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} \; . \tag{4.43}$$

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

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

$$\mathbf{F}_k^{\mathbf{k}} = \sum_{j=k-(L+1)}^{k-1} c_j \, \mathbf{F}_j^{\mathbf{k}} \; . \tag{4.44}$$

Here L is the size of the DIIS subspace, which is set using the *$rem* variable DIIS_SUBSPACE_SIZE. As the Fock matrix nears self-consistency, the linear matrix equations in Eq. (4.43) tend to become severely ill-conditioned and it is often necessary to reset the DIIS subspace (this is automatically carried out by the program).

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

4.7.4 Geometric Direct Minimization (GDM)

Geometric Direct Minimization (GDM) is an extremely robust SCF convergence algorithm that is only slightly less efficient than DIIS. In QC-PBC, we straightforwardly generalize this to handle $\mathbf{k}$-points. The GDM algorithm takes steps in an orbital rotation space that properly respects the hyperspherical geometry of the manifold of allowed SCF solutions. In other words, orbital rotations are variables that describe a space that is curved like a many-dimensional sphere. Just like the optimum flight paths for airplanes are not straight lines but great circles, so too are the optimum steps in orbital rotation space. GDM takes this correctly into account, which is the origin of its efficiency and its robustness. For full details see Ref. 24. GDM is a good alternative to DIIS for SCF jobs that exhibit convergence difficulties with DIIS. The GDM algorithm has been extended to restricted open-shell SCF calculations, and results indicate that it is much more efficient as compared to older direct-minimization methods.

Section 4.7.3 discussed the fact that DIIS can efficiently head towards the global SCF minimum in the early iterations. This can be true even if DIIS fails to converge in later iterations. For this reason, a hybrid scheme has been implemented which uses the DIIS minimization procedure to achieve convergence to an intermediate cutoff threshold. Thereafter, the geometric direct minimization algorithm is used. This scheme combines the strengths of the two methods quite nicely: the ability of DIIS to recover from initial guesses that may not be close to the global minimum, and the ability of GDM to robustly converge to a local minimum, even when the local surface topology is challenging for DIIS. This is the recommended procedure with which to invoke GDM. 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 may run GDM without DIIS to obtain only a single Roothaan

step (which also serves up a properly orthogonalized set of orbitals).

Example 4.9 Input for a UHF calculation using geometric direct minimization (GDM) on the water cluster at Gamma point.

```
$lattice
    3 ! dimension
    2.16330216000000 3.76300538000000 0.00000000000000
    -2.16330216000000 3.76300538000000 0.00000000000000
    0.00000000000000 0.00000000000000 7.06770668000000
$end

$unitcell
    absolute
    0 1
    H 0.0000000000 2.5289709649 4.9523547219
    H 0.0000000000 4.9970397951 1.4185013819
    H 0.0000000000 3.4655001426 3.6501759182
    H 0.0000000000 4.0605106174 0.1163225782
    H 0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 5.5099712578 3.4130834780
    H -0.8004007935 2.0160395022 6.9469368180
    H 0.8004007935 2.0160395022 6.9469368180
    O 0.0000000000 2.5186751564 3.9565869413
    O 0.0000000000 5.0073356036 0.4227336013
    O 0.0000000000 5.0237523919 3.0758661592
    O 0.0000000000 2.5022583681 6.6097194992
$end

$rem
    JOBTYPE sp
    METHOD  pbe
    SCF_ALGORITHM gdm
    UNRESTRICTED TRUE
    BASIS   qzv2p-gth
    PSEUDO  gth-pbe
$end
```

View output online

Example 4.10 Input for an RHF calculation using geometric direct minimization (GDM) on the diamond at [4,4,4] **k**-mesh. This is using B3LYP-D4 and GPW algorithm for the density fitting.

```
$lattice
3 ! specify dimension
0.0 3.370326545430162 3.370326545430162
3.370326545430162 0.0 3.370326545430162
3.370326545430162 3.370326545430162 0.0
$end

$unitcell
ABSOLUTE ! absolute/relative possible
 0 1
C 0.0 0.0 0.0
C 1.685163272715081 1.685163272715081 1.685163272715081
$end

$rem
jobtype sp
method b3lyp
dft_d 11
basis dzvp-gth
input_bohr true
scf_algorithm gdm
kecut 1500
scf_convergence 5
thresh 14
scf_max_cycles 200
scf_algorithm gdm
pseudo gth-pbe
do_bigmem true
mp_mesh = [4,4,4]
k_alg = mo_k
$end
```

View output online

4.8 Algorithms for Efficient Fock-Build

4.8.1 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 QC-PBC. QC-PBC currently supports the Gaussian-Planewave (GPW) density fitting approach to evaluate the Coulomb and exchange matrix. We also offer the occupied resolution-of-the-identity exchange (occ-RI-K) strategy to enable faster GPW-based K evaluations. Furthermore, we recently added a tensor hypercontraction (THC) based exchange algorithm to further speed up the construction of exchange with a dense **k**-grid.

The GPW approach[14] is commonly used in many GTO periodic codes due to its speed and ease of implementation.[11,22,26] This approach has been popularized by CP2K.[11] This method can be considered a form of density fitting, with plane waves as the auxiliary basis set. One can systematically improve the quality of the fitting procedure by increasing the number of plane waves through the energy cutoff.

In this approach, the pair density $\rho_{\mu_{\mathbf{k}_1}\nu_{\mathbf{k}_2}}(\mathbf{r}) = (\phi_{\mu_{\mathbf{k}_1}}(\mathbf{r}))^*\phi_{\nu_{\mathbf{k}_2}}(\mathbf{r})$ is expanded in a plane wave basis, where the

wavevector of plane waves, $\mathbf{G}$, is proportional to the reciprocal space lattice vector, $\{\mathbf{b}_i\}$,

$$\rho_{\mu_{\mathbf{k}_1}\nu_{\mathbf{k}_2}}(\mathbf{r}) = \sum_{\mathbf{G}} \tilde{\rho}_{\mu_{\mathbf{k}_1}\nu_{\mathbf{k}_2}}(\mathbf{G}) e^{i(\mathbf{G}+\mathbf{k}_2-\mathbf{k}_1)\cdot\mathbf{r}}. \tag{4.45}$$

The Fourier transform efficiently determines the fitting coefficients,

$$\tilde{\rho}_{\mu_{\mathbf{k}_1}\nu_{\mathbf{k}_2}}(\mathbf{G}) = \frac{1}{\sqrt{\Omega_0}} \int_{\Omega_0} \rho_{\mu_{\mathbf{k}_1}\nu_{\mathbf{k}_2}}(\mathbf{r}) e^{-i(\mathbf{G}+\mathbf{k}_2-\mathbf{k}_1)\cdot\mathbf{r}} d\mathbf{r}, \tag{4.46}$$

where Ω_0 denotes the volume of the unit cell in real space. The transformation can be carried out numerically using the fast Fourier transform (FFT) with numerical packages such as FFTW3.[4]

With the pair density and Coulomb operator in reciprocal space, the electron repulsion integrals (ERIs) can be evaluated in reciprocal space as follows

$$(\mu_{\mathbf{k}_1}\nu_{\mathbf{k}_2}|\lambda_{\mathbf{k}_3}\sigma_{\mathbf{k}_4}) = \sum_{\mathbf{G}} \tilde{\rho}_{\mu_{\mathbf{k}_1}\nu_{\mathbf{k}_2}}(\mathbf{G}) \frac{4\pi}{|\mathbf{G}+\mathbf{k}_{12}|^2} \tilde{\rho}_{\lambda_{\mathbf{k}_3}\sigma_{\mathbf{k}_4}}(-\mathbf{G}), \tag{4.47}$$

where the singularity at $\mathbf{G}+\mathbf{k}_{12} = \mathbf{0}$ is excluded from the summation. Alternatively, one may form the Hartree-like potential in reciprocal space, then transform it to a real space grid, and contract with the real-space pair density.

There are several drawbacks to the GPW strategy. In the GPW framework, the use of pseudopotentials is necessary since highly localized orbitals in the core region require a prohibitively large number of grid points and plane waves for converged numerical results. Even for second-row atoms, the use of a pseudopotential is necessary.[5,25] Additionally, the storage advantage of GTO basis sets is somewhat negated as a plane wave representation must be stored, leading to storage requirements similar to plane-wave codes. QC-PBC supports energies and efficient analytical gradients for GPW-based algorithms.

4.8.2 Periodic Fourier-transformed Coulomb Method

The periodic Fourier-transformed Coulomb (FTC) Method has been implemented in QC-PBC to perform pure density functional theory with traditional molecular Gaussian basis sets and dense $\mathbf{k}$-mesh sampling.

For all-electron calculations, it is recommended to turn on the Becke atom-centered grid for the exachnge calculation.

FTC_RI_J
Controls the use of periodic Fourier-transformed Coulomb (FTC) Method for constructing the Coulomb matrix

TYPE:
BOOL

DEFAULT:
FALSE Use occ-RI-K.

OPTIONS:
TRUE Use

RECOMMENDATION:
Use occ-RI-K unless running a bands job where virtual energies are desired, then use MOK.

4.8.3 occ-RI-K Exchange Algorithm

The occupied orbital RI-K (occ-RI-K) algorithm[12,15] is a new scheme for building the exchange matrix (K) partially in the MO basis using the RI approximation. occ-RI-K typically matches current alternatives in terms of both the accuracy

(energetics identical to standard RI-K) and convergence (essentially unchanged relative to conventional methods). On the other hand, this algorithm exhibits significant speedups of 10-100x over typical applications of RI, with the speedup increasing with the size of the AO basis set. Thus occ-RI-K helps to make larger basis set hybrid DFT calculations more feasible, which is quite desirable for achieving improved accuracy in DFT calculations with modern functionals. As QC-PBC uses the GPW framework by default (a form of RI), there is no loss in accuracy via use of the occ-RI-K algorithm.

The idea of the occ-RI-K formalism comes from a simple observation that the exchange energy E_K and its gradient can be evaluated from the diagonal elements of the exchange matrix in the occupied-occupied block K_{ii}, and occupied-virtual block K_{ia}, respectively, rather than the full matrix in the AO representation, $K_{\mu\nu}$. Mathematically,

$$\begin{aligned} E_K &= -\sum_{\mu\nu\boldsymbol{k}} P^{\boldsymbol{k}}_{\mu\nu} K^{\boldsymbol{k}}_{\mu\nu} \\ &= -\sum_{\mu\nu\boldsymbol{k}} c^{\boldsymbol{k}}_{\mu i} K^{\boldsymbol{k}}_{\mu\nu} c^{\boldsymbol{k}}_{\nu i} \\ &= -\sum_{i\boldsymbol{k}} K^{\boldsymbol{k}}_{ii} \end{aligned} \tag{4.48}$$

and

$$\frac{\partial E_K}{\partial \Delta^{\boldsymbol{k}}_{ai}} = 2K^{\boldsymbol{k}}_{ai} \tag{4.49}$$

where Δ is a skew-symmetric matrix used to parameterize the unitary transformation U, which represents the variations of the MO coefficients as follows:

$$U = e^{(\Delta - \Delta^T)}. \tag{4.50}$$

From Eq. 4.48 and 4.49 it is evident that the exchange energy and gradient need just $K_{i\nu}$ rather than $K_{\mu\nu}$.

In regular RI-K one has to compute two quartic terms,[27] whereas there are three quartic terms for the occ-RI-K algorithm. The speedup of the latter with respect to former can be explained from the following ratio of operations; refer to Ref. 15 for details.

$$\frac{\text{\# of RI-K quartic operations}}{\text{\# of occ-RI-K quartic operations}} = \frac{oNX^2 + oN^2X}{o^2X^2 + o^2NX + o^2NX} = \frac{N(X+N)}{o(X+2N)} \tag{4.51}$$

With a conservative approximation of $X \approx 2N$, the speedup is $\frac{3}{4}(N/o)$. The occ-RI-K algorithm also involves some cubic steps which should be negligible in the very large molecule limit. Tests in the Ref. 15 suggest that occ-RI-K for small systems with large basis will gain less speed than a large system with small basis, because the cubic terms will be more dominant for the former than the latter case.

In the course of SCF iteration, the occ-RI-K method does not require us to construct the exact Fock matrix explicitly. Rather, $k_{i\nu}$ contributes to the Fock matrix in the mixed MO and AO representations ($F_{i\nu}$) and yields orbital gradient and DIIS error vectors for converging SCF. On the other hand, since occ-RI-K does not provide exactly the same unoccupied eigenvalues, the diagonalization updates can differ from the conventional SCF procedure. In Ref. 15, occ-RI-K was found to require, on average, the same number of SCF iterations to converge and to yield accurate energies.

Note that the occ-RI-K virtual orbital energies are not computed correctly as they do not contribute to the overall energy. One should therefore not use any virtual bands computed via occ-RI-K (e.g. when running a postscf correlated method). This may be remedied via a single computation of the entire Fock matrix after SCF convergence, by setting CORRECT_KVIR to true;

K_ALG

Controls the use of the occ-RI-K approximation for constructing the exchange matrix

TYPE:

STRING

DEFAULT:

OCCRIK

OPTIONS:

OCCRIK	Use occ-RI-K.
MOK	Use MO-K.
AOK	Use AO-K.
THC-AOK	Use THC-AO-K.
THC-OOK	Use THC-oo-K.

RECOMMENDATION:

Use occ-RI-K unless running a bands job where virtual energies are desired, then use MOK.

CORRECT_KVIR

Controls whether a single MO-K fock build is performed after SCF convergence to correct the occ-RI-K virtual energies.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE	Correct virtual energies if occ-RI-K is used.
FALSE	Do not correct virtual energies.

RECOMMENDATION:

If virtual orbital energies are desired, use TRUE, otherwise use FALSE as it is quicker.

4.8.4 THC-K

4.8.4.1 THC-AO-K

The exchange energy can approximated as the product of 5 two-index tensors through the tensor hypercontraction (THC) framework:

$$K_{\mu\nu}^{\mathbf{k_1}} = \sum_{j\mathbf{k_2}} (\mu^{\mathbf{k_1}} j^{\mathbf{k_2}} | j^{\mathbf{k_2}} \nu^{\mathbf{k_1}}) \approx \sum_{j\mathbf{k_2}PQ} X_{\mu^{\mathbf{k_1}} P} X_{j^{\mathbf{k_1}} P} W_{PQ} X_{i^{\mathbf{k_2}} Q} X_{\nu^{\mathbf{k_1}} Q} \tag{4.52}$$

The interpolative separable density fitting approach to THC frames this factorization as a form of density fitting, where the products of the cell periodic part of atomic orbitals are fit via "interpolation vectors," $\{\xi(\mathbf{r})\}$, using weights given by the orbitals evaluated at specific "interpolation points":

$$u_\mu^{\mathbf{k_1}}(\mathbf{r})^* u_\nu^{\mathbf{k_2}}(\mathbf{r}) \approx \sum_P \xi_P(\mathbf{r}) u_\mu^{\mathbf{k_1}}(\mathbf{r_P})^* u_\nu^{\mathbf{k_2}}(\mathbf{r_Q}) \tag{4.53}$$

The set of interpolation points is typically determined via a k-means centroid voronoi tesselation algorithm, or via a QR decomposition. A larger number of interpolating points and interpolating vectors will lead to a more accurate fit, and therefore a higher quality solution. In practice, the number of interpolating points N_{ISDF} is set via a constant used to scale the number of basis functions:

$$N_{ISDF} = c_{ISDF} N \tag{4.54}$$

Once the interpolation points are determined, the interpolation vectors ($\xi_P(\mathbf{r})$), are determined via a least squares fit. Finally, the exchange matrix may be computed via:

$$K_{\mu\nu}^{\mathbf{k_1}} = \sum_{j\mathbf{k_2}} u_{\mu}^{\mathbf{k_1}}(\mathbf{r_P})^* u_{\nu}^{\mathbf{k_2}}(\mathbf{r_P}) M_{PQ}^{\mathbf{k_2}-\mathbf{k_1}} u_{\lambda}^{\mathbf{k_1}}(\mathbf{r_Q})^* u_{\sigma}^{\mathbf{k_2}}(\mathbf{r_Q}) P_{\nu\lambda}^{\mathbf{k_2}} \tag{4.55}$$

In the above, the quantity $M_{PQ}^{\mathbf{k_2}-\mathbf{k_1}}$ is the two electron integral between two interpolating vectors.

$$M_{PQ}^{\mathbf{k_2}-\mathbf{k_1}} = \int\int \frac{1}{|r_{12}|} e^{i(\mathbf{k_2}-\mathbf{k_1})\cdot\mathbf{r_1}} \xi_P(\mathbf{r_1}) \xi_Q(\mathbf{r_2}) e^{i(\mathbf{k_1}-\mathbf{k_2})\cdot\mathbf{r_2}} d\mathbf{r_1} d\mathbf{r_2} \tag{4.56}$$

In the above, the two electron integral depends only on the difference between $\mathbf{k_1}$ and $\mathbf{k_2}$, allowing an overall reduction in scaling for computing exchange, down to $\mathcal{O}(N_k \log(N_k) N^2 N_g)$, which can lead to reduced compute time for large **k**-point meshes.

4.8.4.2 THC-oo-K

Instead of fitting products of atomic orbitals, one may instead fit only products of occupied orbitals, since those are the only quantities that contribute to the exchange energy:

$$u_i^{\mathbf{k_1}}(\mathbf{r})^* u_j^{\mathbf{k_2}}(\mathbf{r}) \approx \sum_P \xi_P(\mathbf{r}) u_i^{\mathbf{k_1}}(\mathbf{r_P})^* u_j^{\mathbf{k_2}}(\mathbf{r_Q}) \tag{4.57}$$

This leads to a greatly reduced number of required interpolating points as the number of quantities fit is reduced from N^2 to N_{occ}^2. This is reflected in N_{ISDF} which is set via:

$$N_{ISDF} = c_{ISDF} N_{occ} \tag{4.58}$$

However, in THC-oo-K but the fit must be performed each iteration for the new occupied orbitals. Additionally, there is an additional term in the THC-oo-K Fock matrix arising from the orbital derivative of the ISDF fit. The scaling of the THC-oo-K algorithm is therefore reduced to $\mathcal{O}(N_k \log(N_k) N_{occ}^2 N_g)$, but with a larger prefactor than THC-AO-K. There is therefore a tradeoff in compute time vs. THC-AO-K, that is more beneficial with large basis sets.

It is recommended to use THC based algorithms when large **k**-point meshes are used (over 100 **k**-points). Specifically, THC-AO-K should be used for small basis sets, and THC-oo-K should be used for large basis sets.

CISDF

Controls the accuracy of the ISDF fit via the c_{ISDF} parameter. Actual value is divided by 1000.

TYPE:

INTEGER

DEFAULT:

55000 Corresponds to $c_{ISDF} = 55.0$

OPTIONS:

n Use $c_{ISDF} = n/1000$

RECOMMENDATION:

Use the default unless convergence issues are encountered, then increase.

4.9 SCF Post-processing

4.9.1 Density of States

4.9.1.1 Introduction

The density of states (DOS) is a quantity which describes the number of states accessible at a given energy. The DOS is a central quantity in the study of condensed matter and provides insight into a material's electronic properties. Currently there are two methods implemented in QC-PBC. The tetrahedron method of Blöchl et. al,[1] and the histogram method.

The histogram method is a sum over normalized Gaussians.

$$D(E) = \frac{1}{N_k\sqrt{2\pi\sigma^2}} \sum_i \sum_{\mathbf{k}} \exp\left(-\left(\frac{E-\epsilon_i(\mathbf{k})}{2\sigma}\right)^2\right). \tag{4.59}$$

This approach is straightforward; however it can require a very fine $\mathbf{k}$-mesh to converge, and the user must ensure that σ is sufficiently small as to not interfere with the interpretation of the density of states.

The tetrahedron, on the other hand method, works by integrating a linear interpolating function. The Monkhorst-Pack mesh is first decomposed into tetrahedra (a cube can be subdivided into 6 non-overlapping tetrahedra). The DOS contribution of a given tetrahedron and band index is computed analytically[1]. So, the DOS calculation is transformed from a sum over $\mathbf{k}$ to a sum over tetrahedra.

The advantage of the tetrahedron method is that it produces sharp features and requires less $\mathbf{k}$-point sampling than the histogram method. Additionally, it does not require any parameter tuning.

The DOS module produces an output file 'DOS.dat' inside the scratch directory containing the results. The first column of the output is the energy values, the second is the alpha density of states, and the third (present only for spin unrestricted calculations) is the beta density of states. All energy units in the DOS output are in electron volts.

4.9.1.2 Job Control

The following *$rem* variable must be specified in order to run a density of states calculation; however the user is advised to consult the additional options for fine control over the methodology, output window, and resolution.

DOS

Indicates whether a DOS calculation will be run after SCF.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Perform density of states calculation.

FALSE Do not perform density of states calculation.

RECOMMENDATION:

Set to TRUE when a DOS calculation is desired.

Example 4.11 Input for a standard density of states calculation of diamond with a [2,2,2] k-mesh.

```
$lattice
3 ! dimension
2.18050236000000 0.00000000000000 1.25891346000000
0.72683412000000 2.05579703000000 1.25891346000000
-0.00000000000000 0.00000000000000 2.51782692000000
$end

$unitcell
absolute
0 1
C 2.5439194200 1.7988224012 4.4061971100
C 0.3634170600 0.2569746287 0.6294567300
$end

$rem
   jobtype  sp
   method  lda
   unrestricted  false
   basis  SZV-GTH
   pseudo  gth-pbe
   fft_ng 8
   dos  true
   dos_method  tetrahedron
   dos_steps  1001
   mp_mesh  [2,2,2]
$end
```

View output online

4.9.1.3 Additional Options

DOS_METHOD

The method by which the DOS is computed

TYPE:

STRING

DEFAULT:

TETRAHEDRON

OPTIONS:

TETRAHEDRON	Use the tetrahedron linear interpolation method of Blöchl et. al.
HISTOGRAM	Use a sum of broadened Gaussian functions (see DOS_SIGMA).

RECOMMENDATION:

Use TETRAHEDRON generally, for dispersionless bands use HISTOGRAM

DOS_SIGMA

HISTOGRAM method only: The standard deviation (in Hartree) of the Gaussian broadening applied in HISTOGRAM method

TYPE:

FLOAT

DEFAULT:

0.01

OPTIONS:

$\sigma > 0$ User selected

RECOMMENDATION:

User should select DOS_SIGMA appropriate for the k-mesh and energy resolution desired.

DOS_STEPS

The number of energy values where the DOS is evaluated

TYPE:

INTEGER

DEFAULT:

1001

OPTIONS:

$N > 0$ User selected

RECOMMENDATION:

User should select DOS_STEPS appropriate for the energy resolution desired.

DOS_MIN

How far below the Fermi level (in eV) will be sampled in a DOS calculation

TYPE:

FLOAT

DEFAULT:

-30.0

OPTIONS:

Energy in eV User selected

RECOMMENDATION:

User should select DOS_MIN appropriate for the energetic region being studied

DOS_MAX

How far above the Fermi level (in eV) will be sampled in a DOS calculation

TYPE:

FLOAT

DEFAULT:

+30.0

OPTIONS:

Energy in eV User selected

RECOMMENDATION:

User should select DOS_MAX appropriate for the energetic region being studied

DOS_NBANDS

TETRAHEDRON method only: How many bands will be included in the DOS calculation

TYPE:

INTEGER

DEFAULT:

3 * number of occupied bands

OPTIONS:

N > 0 User selected

RECOMMENDATION:

The user should select DOS_NBANDS appropriate for the energetic region studied

DOS_FERMI_WINDOW

Whether DOS_MIN/MAX are centered on the Fermi level.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE DOS_MIN and DOS_MAX are measured relative to Fermi level.

FALSE DOS_MIN and DOS_MAX are relative to absolute band energies.

RECOMMENDATION:

TRUE is appropriate in all cases unless the user is interested in absolute band energies

DOS_FERMI_ZERO

Whether the Fermi level is set to zero in the DOS.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Fermi level is defined as zero in DOS output.

FALSE Absolute band energies are used in DOS output.

RECOMMENDATION:

TRUE is appropriate in all cases unless the user is interested in absolute band energies

References and Further Reading

[1] Peter E. BlÃűchl, O. Jepsen, and O. K. Andersen. Improved tetrahedron method for brillouin-zone integrations. 49(23):16223–16233. DOI: 10.1103/PhysRevB.49.16223.

[2] M. Born and R. Oppenheimer. *Ann. Phys.*, 389:457, 1927. DOI: 10.1002/andp.19273892002.

[3] Hieu Q Dinh, Adam Rettig, Xintian Feng, and Joonho Lee. Efficient all-electron periodic fourier-transformed coulomb method. *arXiv preprint arXiv:2509.16460*, 2025.

[4] Matteo Frigo and Steven G Johnson. The design and implementation of fftw3. *Proc. IEEE*, 93(2):216–231, 2005.

[5] Stefan Goedecker, Michael Teter, and Jürg Hutter. Separable dual-space gaussian pseudopotentials. *Phys. Rev. B*, 54(3):1703, 1996.

[6] M. F. Guest and V. R. Saunders. *Mol. Phys.*, 28:819, 1974. DOI: 10.1080/00268977400102171.

[7] D. H. Hartree. *The Calculation of Atomic Structures*. John Wiley and Sons, 1957.

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

[9] I. H. Hillier and V. R. Saunders. *Proc. Roy. Soc. Ser. A*, 320:161, 1970. DOI: 10.1098/rspa.1970.0203.

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

[11] Thomas D. KÃijhne, Marcella Iannuzzi, Mauro Del Ben, Vladimir V. Rybkin, Patrick Seewald, Frederick Stein, Teodoro Laino, Rustam Z. Khaliullin, Ole SchÃijtt, Florian Schiffmann, Dorothea Golze, Jan Wilhelm, Sergey Chulkov, Mohammad Hossein Bani-Hashemian, ValÃl'ry Weber, Urban BorÅątnik, Mathieu Taillefumier, Alice Shoshana Jakobovits, Alfio Lazzaro, Hans Pabst, Tiziano MÃijller, Robert Schade, Manuel Guidon, Samuel Andermatt, Nico Holmberg, Gregory K. Schenter, Anna Hehn, Augustin Bussy, Fabian Belleflamme, Gloria Tabacchi, Andreas GlÃűÃ§, Michael Lass, Iain Bethune, Christopher J. Mundy, Christian Plessl, Matt Watkins, Joost VandeVondele, Matthias Krack, and JÃijrg Hutter. CP2K: An electronic structure and molecular dynamics software package - Quickstep: Efficient and accurate electronic structure calculations. *J. Chem. Phys.*, 152(19):194103, May 2020. ISSN 0021-9606. DOI: 10.1063/5.0007045. URL https://doi.org/10.1063/5.0007045.

[12] J. Lee, A. Rettig, X. Feng, E. Epifanovsky, and M. Head-Gordon. *J. Chem. Theory Comput.*, 18(12):7336–7349, 2022. DOI: 10.1021/acs.jctc.2c00742.

[13] S. Lehtola. *J. Chem. Theory Comput.*, 15:1593, 2019. DOI: 10.1021/acs.jctc.8b01089.

[14] Gerald Lippert, JÃijrg Hutter, and Michele Parrinello. A hybrid Gaussian and plane wave density functional scheme. *Mol. Phys.*, 92(3):477–488, October 1997. ISSN 0026-8976. DOI: 10.1080/002689797170220. URL https://doi.org/10.1080/002689797170220.

[15] S. Manzer, P. R. Horn, N. Mardirossian, and M. Head-Gordon. *J. Chem. Phys.*, 143:024113, 2015. DOI: 10.1063/1.4923369.

[16] J. A. Pople and R. K. Nesbet. *J. Chem. Phys.*, 22:571, 1954. DOI: 10.1063/1.1740120.

[17] P. Pulay. *Chem. Phys. Lett.*, 73:393, 1980. DOI: 10.1016/0009-2614(80)80396-4.

[18] P. Pulay. *J. Comput. Chem.*, 3:556, 1982. DOI: 10.1002/jcc.540030413.

[19] V. R. Saunders and I. H. Hillier. *Int. J. Quantum Chem.*, 7:699, 1973. DOI: 10.1002/qua.560070407.

[20] J. C. Slater. *Phys. Rev.*, 34:1293, 1929. DOI: 10.1103/PhysRev.34.1293.

[21] J. C. Slater. *Phys. Rev.*, 35:509, 1930. DOI: 10.1103/PhysRev.35.509.

[22] Qiming Sun, Xing Zhang, Samragni Banerjee, Peng Bao, Marc Barbry, Nick S. Blunt, Nikolay A. Bogdanov, George H. Booth, Jia Chen, Zhi-Hao Cui, Janus J. Eriksen, Yang Gao, Sheng Guo, Jan Hermann, Matthew R. Hermes, Kevin Koh, Peter Koval, Susi Lehtola, Zhendong Li, Junzi Liu, Narbe Mardirossian, James D. McClain, Mario Motta, Bastien Mussard, Hung Q. Pham, Artem Pulkin, Wirawan Purwanto, Paul J. Robinson, Enrico Ronca, Elvira R. Sayfutyarova, Maximilian Scheurer, Henry F. Schurkus, James E. T. Smith, Chong Sun, Shi-Ning Sun, Shiv Upadhyay, Lucas K. Wagner, Xiao Wang, Alec White, James Daniel Whitfield, Mark J. Williamson, Sebastian Wouters, Jun Yang, Jason M. Yu, Tianyu Zhu, Timothy C. Berkelbach, Sandeep Sharma, Alexander Yu. Sokolov, and Garnet Kin-Lic Chan. Recent developments in the PySCF program package. *J. Chem. Phys.*, 153 (2):024109, July 2020. ISSN 0021-9606. DOI: 10.1063/5.0006074. URL https://doi.org/10.1063/5.0006074.

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

[24] T. Van Voorhis and M. Head-Gordon. *Mol. Phys.*, 100:1713, 2002. DOI: 10.1080/00268970110103642.

[25] David Vanderbilt. Soft self-consistent pseudopotentials in a generalized eigenvalue formalism. *Phys. Rev. B*, 41 (11):7892, 1990.

[26] Joost VandeVondele, Matthias Krack, Fawzi Mohamed, Michele Parrinello, Thomas Chassaing, and JÃijrg Hutter. Quickstep: Fast and accurate density functional calculations using a mixed Gaussian and plane waves approach. *Comput. Phys. Commun.*, 167(2):103–128, April 2005. ISSN 0010-4655. DOI: 10.1016/j.cpc.2004.12.014. URL https://www.sciencedirect.com/science/article/pii/S0010465505000615.

[27] F. Weigend. *Phys. Chem. Chem. Phys.*, 4:4285, 2002. DOI: 10.1039/b204199p.

Chapter 5

Density Functional Theory

5.1 Introduction

DFT[80,83,111,200] has emerged as an accurate, alternative first-principles approach to quantum mechanical molecular investigations. DFT calculations account for the overwhelming majority of all quantum chemistry calculations, not only because of its proven chemical accuracy, but also because of its relatively low computational expense, comparable to Hartree-Fock theory but with treatment of electron correlation that is neglected in a HF calculation. These two features suggest that DFT is likely to remain a leading method in the quantum chemist's toolkit well into the future. QC-PBC contains fast, efficient and accurate algorithms for all popular density functionals, making calculations on large molecules possible and practical.

DFT is primarily a theory of electronic ground state structures based on the electron density, $\rho(\mathbf{r})$, as opposed to the many-electron wave function, $\Psi(\mathbf{r}_1, \ldots, \mathbf{r}_N)$. There are a number of distinct similarities and differences between traditional wave function approaches and modern DFT methodologies. First, the essential building blocks of the many-electron wave function Ψ are single-electron orbitals, which are directly analogous to the Kohn-Sham orbitals in the DFT framework. Second, both the electron density and the many-electron wave function tend to be constructed via a SCF approach that requires the construction of matrix elements that are conveniently very similar.

However, traditional *ab initio* approaches using the many-electron wave function as a foundation must resort to a post-SCF calculation (Chapter 6) to incorporate correlation effects, whereas DFT approaches incorporate correlation at the SCF level. Post-SCF methods, such as perturbation theory or coupled-cluster theory are extremely expensive relative to the SCF procedure. On the other hand, while the DFT approach is exact in principle, in practice it relies on modeling an unknown exchange-correlation energy functional. While more accurate forms of such functionals are constantly being developed, there is no systematic way to improve the functional to achieve an arbitrary level of accuracy. Thus, the traditional approaches offer the possibility of achieving a systematically-improvable level of accuracy, but can be computationally demanding, whereas DFT approaches offer a practical route, but the theory is currently incomplete.

5.2 Kohn-Sham Density Functional Theory

The density functional theory by Hohenberg, Kohn, and Sham[68,79] stems from earlier work by Dirac,[45] who showed that the exchange energy of a uniform electron gas can be computed exactly from the charge density along. However, while this traditional *density* functional approach, nowadays called "orbital-free" DFT, makes a direct connection to the density alone, in practice it is constitutes a direct approach where the necessary equations contain only the electron density, difficult to obtain decent approximations for the kinetic energy functional. Kohn and Sham sidestepped this

difficulty via an indirect approach in which the kinetic energy is computed exactly for a noninteracting reference system, namely, the Kohn-Sham determinant.[79] It is the Kohn-Sham approach that first made DFT into a practical tool for calculations.

Within the Kohn-Sham formalism,[79] the ground state electronic energy, E, can be written as

$$E = E_{\mathrm{T}} + E_{\mathrm{V}} + E_{\mathrm{J}} + E_{\mathrm{XC}} \tag{5.1}$$

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

$$\rho_\alpha(\mathbf{r}) = \sum_{\mathbf{k}} \sum_{i=1}^{n_\alpha} |\psi_i^{\mathbf{k}\alpha}|^2 \tag{5.2a}$$

$$\rho_\beta(\mathbf{r}) = \sum_{\mathbf{k}} \sum_{i=1}^{n_\beta} |\psi_i^{\mathbf{k}\beta}|^2 \tag{5.2b}$$

where n_α and n_β are the number of alpha and beta electron respectively, and $\psi_i^{\mathbf{k}}$ are the Kohn-Sham orbitals. Thus, the total electron density is

$$\rho(\mathbf{r}) = \rho_\alpha(\mathbf{r}) + \rho_\beta(\mathbf{r}) \tag{5.3}$$

Within a finite basis set, the density is represented by[135]

$$\rho(\mathbf{r}) = \sum_{\mathbf{k}} \sum_{\mu\nu} P_{\mu\nu}^{\mathbf{k}} \phi_\mu^{\mathbf{k}}(\mathbf{r}) \phi_\nu^{\mathbf{k}*}(\mathbf{r}) \ , \tag{5.4}$$

where the $P_{\mu\nu}$ are the elements of the one-electron density matrix; see Eq. (4.30) in the discussion of Hartree-Fock theory. The various energy components in Eq. (5.1) can now be written

$$\begin{aligned} E_{\mathrm{T}} &= \sum_{\mathbf{k}} \sum_{i=1}^{n_\alpha} \left\langle \psi_i^{\mathbf{k}\alpha} \left| -\frac{1}{2}\hat{\nabla}^2 \right| \psi_i^{\mathbf{k}\alpha} \right\rangle + \sum_{i=1}^{n_\beta} \left\langle \psi_i^{\mathbf{k}\beta} \left| -\frac{1}{2}\hat{\nabla}^2 \right| \psi_i^{\mathbf{k}\beta} \right\rangle \\ &= \sum_{\mathbf{k}} \sum_{\mu\nu} P_{\nu\mu}^{\mathbf{k}} \left\langle \phi_\mu^{\mathbf{k}}(\mathbf{r}) \left| -\frac{1}{2}\hat{\nabla}^2 \right| \phi_\nu^{\mathbf{k}}(\mathbf{r}) \right\rangle \end{aligned} \tag{5.5}$$

$$\begin{aligned} E_{\mathrm{V}} &= -\sum_{A=1}^{M} Z_A \int \frac{\rho(\mathbf{r})}{|\mathbf{r} - \mathbf{R}_A|} d\mathbf{r} \\ &= -\sum_{\mathbf{k}} \sum_{\mu\nu} P_{\nu\mu}^{\mathbf{k}} \sum_A \left\langle \phi_\mu^{\mathbf{k}}(\mathbf{r}) \left| \frac{Z_A}{|\mathbf{r} - \mathbf{R}_A|} \right| \phi_\nu^{\mathbf{k}}(\mathbf{r}) \right\rangle \end{aligned} \tag{5.6}$$

$$\begin{aligned} E_{\mathrm{J}} &= \frac{1}{2} \left\langle \rho(\mathbf{r}_1) \left| \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \right| \rho(\mathbf{r}_2) \right\rangle \\ &= \frac{1}{2} \sum_{\mathbf{k}_1\mathbf{k}_2} \sum_{\mu\nu} \sum_{\lambda\sigma} P_{\nu\mu}^{\mathbf{k}_1} P_{\sigma\lambda}^{\mathbf{k}_2} \left(\mu^{\mathbf{k}_1}\nu^{\mathbf{k}_1} | \lambda^{\mathbf{k}_2}\sigma^{\mathbf{k}_2}\right) \end{aligned} \tag{5.7}$$

$$E_{\mathrm{XC}} = \int f\left[\rho(\mathbf{r}), \hat{\boldsymbol{\nabla}}\rho(\mathbf{r}), \ldots\right] \rho(\mathbf{r}) \, d\mathbf{r} \ . \tag{5.8}$$

Minimizing E with respect to the unknown Kohn-Sham orbital coefficients yields a set of matrix equations exactly analogous to Pople-Nesbet equations of the UHF case, Eq. (4.19), but with modified Fock matrix elements [*cf.* Eq. (4.33)]

$$F_{\mu\nu}^{\mathbf{k}\alpha} = (H^{\mathrm{core}})_{\mu\nu}^{\mathbf{k}} + J_{\mu\nu}^{\mathbf{k}} - (F^{\mathrm{XC}\alpha})_{\mu\nu}^{\mathbf{k}} \tag{5.9a}$$

$$F_{\mu\nu}^{\mathbf{k}\beta} = (H^{\mathrm{core}})_{\mu\nu}^{\mathbf{k}} + J_{\mu\nu}^{\mathbf{k}} - (F^{\mathrm{XC}\beta})_{\mu\nu}^{\mathbf{k}} \ . \tag{5.9b}$$

Here, $\mathbf{F}^{\mathrm{XC}\alpha}$ and $\mathbf{F}^{\mathrm{XC}\beta}$ are the exchange-correlation parts of the Fock matrices and depend on the exchange-correlation functional used. UHF theory is recovered as a special case simply by taking $F_{\mu\nu}^{\mathrm{XC}\alpha} = K_{\mu\nu}^{\alpha}$, and similarly for β. Thus, the density and energy are obtained in a manner analogous to that for the HF method. Initial guesses are made for the MO coefficients and an iterative process is applied until self-consistency is achieved.

5.3 Overview of Available Functionals

5.3.1 Introduction

QC-PBC currently has more than 30 exchange functionals as well as more than 30 correlation functionals, and in addition over 150 exchange-correlation (XC) functionals, which refer to functionals that are not separated into exchange and correlation parts, either because the way in which they were parameterized renders such a separation meaningless (*e.g.*, B97-D[56] or ωB97X[32]) or because they are a standard linear combination of exchange and correlation (*e.g.*, PBE[120] or B3LYP[19,152]). User-defined XC functionals can be created as specified linear combinations of any of the 30+ exchange functionals and/or the 30+ correlation functionals.

KS-DFT functionals can be organized onto a ladder with five rungs, in a classification scheme ("Jacob's Ladder") proposed by John Perdew in 2001.[115,122] The first rung contains a functional that only depends on the (spin-) density ρ_σ, namely, the local spin-density approximation (LSDA). These functionals are exact for the infinite uniform electron gas (UEG), but are highly inaccurate for molecular properties whose densities exhibit significant inhomogeneity. To improve upon the weaknesses of the LSDA, it is necessary to introduce an ingredient that can account for inhomogeneities in the density: the density gradient, $\hat{\boldsymbol{\nabla}}\rho_\sigma$. These generalized gradient approximation (GGA) functionals define the second rung of Jacob's Ladder and tend to improve significantly upon the LSDA. Two additional ingredients that can be used to further improve the performance of GGA functionals are either the Laplacian of the density $\hat{\nabla}^2\rho_\sigma$, and/or the kinetic energy density,

$$\tau_\sigma = \sum_i^{n_\sigma} \left\| \hat{\boldsymbol{\nabla}} \psi_{i,\sigma} \right\|^2 . \tag{5.10}$$

While functionals that employ both of these options are available in QC-PBC, the kinetic energy density is by far the more popular ingredient and has been used in many modern functionals to add flexibility to the functional form with respect to both constraint satisfaction (non-empirical functionals) and least-squares fitting (semi-empirical parameterization). Functionals that depend on either of these two ingredients belong to the third rung of the Jacob's Ladder and are called meta-GGAs. These meta-GGAs often further improve upon GGAs in areas such as thermochemistry, kinetics (reaction barrier heights), and even non-covalent interactions.

Functionals on the fourth rung of Jacob's Ladder are called hybrid density functionals. This rung contains arguably the most popular density functional of our time, B3LYP, the first functional to see widespread application in chemistry. "Global" hybrid (GH) functionals such as B3LYP (as distinguished from the "range-separated hybrids" introduced below) add a constant fraction of "exact" (Hartree-Fock) exchange to any of the functionals from the first three rungs. Thus, hybrid LSDA, hybrid GGA, and hybrid meta-GGA functionals can be constructed, although the latter two types are much more common. As an example, the formula for the B3LYP functional, as implemented in QC-PBC, is

$$E_{xc}^{\text{B3LYP}} = c_x E_x^{\text{HF}} + (1 - c_x - a_x)\, E_x^{\text{Slater}} + a_x E_x^{\text{B88}} + (1 - a_c)\, E_c^{\text{VWN1RPA}} + a_c E_c^{\text{LYP}} \tag{5.11}$$

where $c_x = 0.20$, $a_x = 0.72$, and $a_c = 0.81$.

A more recent approach to introducing exact exchange into the functional form is via range separation. Range-separated hybrid (RSH) functionals split the exact exchange contribution into a short-range (SR) component and a long-range (LR) component, often by means of the error function (erf) and complementary error function ($\text{erfc} \equiv 1 - \text{erf}$):

$$\frac{1}{r_{12}} = \underbrace{\frac{\text{erfc}(\omega r_{12})}{r_{12}}}_{\text{SR}} + \underbrace{\frac{\text{erf}(\omega r_{12})}{r_{12}}}_{\text{LR}} \tag{5.12}$$

The first term on the right in Eq. (5.12) is singular but short-range, and decays to zero on a length scale of $\sim 1/\omega$, while the second term constitutes a non-singular, long-range background. An RSH XC functional can be expressed generically as

$$E_{xc}^{\text{RSH}} = c_{x,\text{SR}} E_{x,\text{SR}}^{\text{HF}} + c_{x,\text{LR}} E_{x,\text{LR}}^{\text{HF}} + (1 - c_{x,\text{SR}}) E_{x,\text{SR}}^{\text{DFT}} + (1 - c_{x,\text{LR}}) E_{x,\text{LR}}^{\text{DFT}} + E_c^{\text{DFT}} , \tag{5.13}$$

where the SR and LR parts of the Coulomb operator are used, respectively, to evaluate the HF exchange energies $E_{x,\mathrm{SR}}^{\mathrm{HF}}$ and $E_{x,\mathrm{LR}}^{\mathrm{HF}}$. The corresponding DFT exchange functional is partitioned in the same manner, but the correlation energy E_c^{DFT} is evaluated using the full Coulomb operator, r_{12}^{-1}. Of the two linear parameters in Eq. (5.13), $c_{x,\mathrm{LR}}$ is usually either set to 1 to define long-range corrected (LRC) RSH functionals (see Section 5.6) or else set to zero, which defines screened-exchange (SE) RSH functionals. On the other hand, the fraction of short-range exact exchange ($c_{x,\mathrm{SR}}$) can either be determined via least-squares fitting, theoretically justified using the adiabatic connection, or simply set to zero. As with the global hybrids, RSH functionals can be fashioned using all of the ingredients from the lower three rungs. The rate at which the local DFT exchange is turned off and the non-local exact exchange is turned on is controlled by the parameter ω. Large values of ω tend to lead to attenuators that are less smooth (unless the fraction of short-range exact exchange is very large), while small values of (*e.g.*, $\omega = 0.2$–0.4 bohr^{-1}) are the most common in semi-empirical RSH functionals.

The final rung on Jacob's Ladder contains functionals that use not only occupied orbitals (via exact exchange), but virtual orbitals as well (via methods such as MP2 or the random phase approximation, RPA). These double hybrids (DH) are the most expensive density functionals available in QC-PBC, but can also be very accurate. The most basic form of a DH functional is

$$E_{xc}^{\mathrm{DH}} = c_x E_x^{\mathrm{HF}} + (1 - c_x)\, E_x^{\mathrm{DFT}} + c_c E_x^{\mathrm{MP2}} + (1 - c_c)\, E_c^{\mathrm{DFT}} \,. \tag{5.14}$$

As with hybrids, the coefficients can either be theoretically motivated or empirically determined. In addition, double hybrids can use exact exchange both globally or via range-separation, and their components can be as primitive as LSDA or as advanced as in meta-GGA functionals. QC-PBC does not currently support the execution of double-hybrid density functionals.

Finally, the last major advance in KS-DFT in recent years has been the development of methods that are capable of accurately describing non-covalent interactions, particularly dispersion. All of the functionals from Jacob's Ladder can technically be combined with these dispersion corrections, although in some cases the combination is detrimental, particularly for semi-empirical functionals that were parameterized in part using data sets of non-covalent interactions, and already tend to overestimate non-covalent interaction energies. The most popular such methods available in QC-PBC are:

- Non-local correlation (NLC) functionals (Section 5.7.1), including those of Vydrov and Van Voorhis[168,170] (VV09 and VV10) and of Lundqvist and Langreth[43,44] (vdW-DF-04 and vdW-DF-10). For efficient evaluation of periodic NLC functionals, the revised VV10 NLC functional of Sabatini and coworkers (rVV10) should be used[146].
- Damped, atom–atom pairwise empirical dispersion potentials from Grimme and others[33,56,58,59,147,149] [DFT-D2, DFT-CHG, DFT-D3(0), DFT-D3(BJ), DFT-D3(CSO), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op)]; see Section 5.7.2.

Below, we categorize the functionals that are available in QC-PBC, including exchange functionals (Section 5.3.3), correlation functionals (Section 5.3.4), and exchange-correlation functionals (Section 5.3.5). Within each category the functionals will be categorized according to Jacob's Ladder. Exchange and correlation functionals can be invoked using the *$rem* variables EXCHANGE and CORRELATION, while the exchange-correlation functionals can be invoked either by setting the *$rem* variable METHOD or alternatively (in most cases, and for backwards compatibility with earlier versions of QC-PBC) by using the *$rem* variable EXCHANGE. Some caution is warranted here. While setting METHOD to PBE, for example, requests the Perdew-Burke-Ernzerhof (PBE) exchange-correlation functional,[120] which includes both PBE exchange and PBE correlation, setting EXCHANGE = PBE requests only the exchange component and setting CORRELATION = PBE requests only the correlation component. Setting both of these values is equivalent to specifying METHOD = PBE.

	Single-Point	**Optimization**	**Frequency**
Ground State	LSDA	LSDA	LSDA
	GGA	GGA	GGA
	meta-GGA	meta-GGA	meta-GGA
	GH	GH	GH
	RSH	RSH	RSH
	NLC	NLC	—
	DFT-D	DFT-D	—
TDDFT	LSDA	—	—
	GGA	—	—
	meta-GGA	—	—
	GH	—	—
	RSH	—	—
	—	—	—
	DFT-D	—	—

Table 5.1: Available analytic properties for SCF calculations.

Finally, Table 5.1 provides a summary, arranged according to Jacob's Ladder, of which categories of functionals are available with analytic first derivatives (for geometry optimizations) or second derivatives (for phonon/vibrational frequency calculations). If analytic derivatives are not available for the requested job type, QC-PBC will automatically generate them via finite difference. Tests of the finite-difference procedure, in cases where analytic second derivatives *are* available, suggest that finite-difference frequencies are accurate to < 1 cm^{-1}, except for very low-frequency, non-bonded modes.[96] Also listed in Table 5.1 are which functionals are available for excited-state time-dependent DFT (TDDFT) calculations, as described in Section **??**.

5.3.2 Suggested Density Functionals for Extended Applications

QC-PBC contains over 150 exchange-correlation functionals, not counting those that can be straightforwardly appended with a dispersion correction (such as B3LYP-D3). Therefore, we suggest a few functionals from the second through fourth rungs of Jacob's Ladder in order to guide functional selection. Periodic systems have not enjoyed the same level of broad and thorough benchmarking as molecular systems, so functionals cannot be recommended with the same level of certainty. Instead, we recommend functionals that are commonly used for periodic systems as well as well rounded functionals for molecular applications that are expected to transfer well to periodic systems. **The single recommended method from each category is indicated in bold.**

From the GGAs on Rung 2, we recommend:

- **revPBE-D3(BJ)**: METHOD = revPBE and DFT_D = D3_BJ
- B97-D3(BJ): METHOD = B97-D3 and DFT_D = D3_BJ
- BLYP-D3(BJ): METHOD = BLYP and DFT_D = D3_BJ
- PBE: METHOD = PBE

From the meta-GGAs on Rung 3, we recommend:

- **B97M-rV**: METHOD = B97M-rV

From the hybrid GGAs on Rung 4, we recommend:

- **revPBE0-D3(BJ)**: METHOD = revPBE0 and DFT_D = D3_BJ
- HSE-HJS: METHOD = HSE-HJS
- ωB97X-rV: METHOD = wB97X-V
- ωB97X-D3: METHOD = wB97X-D3
- ωB97X-D: METHOD = wB97X-D
- B3LYP-D3(BJ): METHOD = B3LYP and DFT_D = D3_BJ

From the hybrid meta-GGAs on Rung 4, we recommend:

- ω**B97M-rV**: METHOD = wB97M-rV

5.3.3 Exchange Functionals

Note: All exchange functionals in this section can be invoked using the *$rem* variable EXCHANGE. Popular and/or recommended functionals within each class are listed first and indicated in bold. The rest are in alphabetical order.

- Local Spin-Density Approximation (LSDA)
  - **Slater**: Slater-Dirac exchange functional (Xα method with $\alpha = 2/3$)[45]
  - SR_LSDA (BNL): Short-range version of the Slater-Dirac exchange functional[50]
- Generalized Gradient Approximation (GGA)
  - **PBE**: Perdew, Burke, and Ernzerhof exchange functional[120]
  - **B88**: Becke exchange functional from 1988[18]
  - **revPBE**: Zhang and Yang one-parameter modification of the PBE exchange functional[188]
  - AK13: Armiento-Kümmel exchange functional from 2013[12]
  - B86: Becke exchange functional (X$\alpha\beta\gamma$) from 1986[15]
  - G96: Gill exchange functional from 1996[48]
  - mB86: Becke "modified gradient correction" exchange functional from 1986[16]
  - mPW91: modified version (Adamo and Barone) of the 1991 Perdew-Wang exchange functional[6]
  - muB88 (μB88): Short-range version of the B88 exchange functional by Hirao and coworkers[70]
  - muPBE (μPBE): Short-range version of the PBE exchange functional by Hirao and coworkers[70]
  - srPBE: Short-range version of the PBE exchange functional by Goll and coworkers[52,53]
  - optB88: Refit version of the original B88 exchange functional (for use with vdW-DF-04) by Michaelides and coworkers[78]
  - OPTX: Two-parameter exchange functional by Handy and Cohen[64]
  - PBEsol: PBE exchange functional modified for solids[124]
  - PW86: Perdew-Wang exchange functional from 1986[116]
  - PW91: Perdew-Wang exchange functional from 1991[119]

- RPBE: Hammer, Hansen, and Norskov exchange functional (modification of PBE)[62]
- rPW86: Revised version (Murray *et al.*) of the 1986 Perdew-Wang exchange functional[109]
- SOGGA: Second-order GGA functional by Zhao and Truhlar[193]
- wPBE (ωPBE): Henderson *et al.* model for the PBE GGA short-range exchange hole[65]

◦ Meta-Generalized Gradient Approximation (meta-GGA)

- **TPSS**: Tao, Perdew, Staroverov, and Scuseria exchange functional[159]
- **revTPSS**: Revised version of the TPSS exchange functional[126]
- BLOC: Minor modification of the TPSS exchange functional that works best with TPSSloc correlation (both by Della Sala and coworkers)[39]
- modTPSS: One-parameter version of the TPSS exchange functional[123]
- oTPSS: TPSS exchange functional with 5 refit parameters (for use with oTPSS correlation) by Grimme and coworkers[51]
- PBE-GX: First exchange functional based on a finite uniform electron gas (rather than an infinite UEG) by Pierre-François Loos[98]
- PKZB: Perdew, Kurth, Zupan, and Blaha exchange functional[121]
- regTPSS: Regularized (fixed order of limits issue) version of the TPSS exchange functional[145]
- SCAN: Strongly Constrained and Appropriately Normed exchange functional[157]
- rSCAN: Regularized SCAN exchange[14,47]
- r++SCAN: Regularized SCAN with uniform density limit and coordinate scaling behavior[47]
- r2SCAN: Re-Regularized SCAN exchange[47]**? ?**
- r4SCAN: Regularized SCAN with exact constraints obeyed by SCAN[47]
- revSCAN: Revised SCAN exchange[104]
- TM: Tao-Mo exchange functional derived via an accurate modeling of the conventional exchange hole[158]
- regTM: Regularized TM exchange[112]
- revTM: Revised TM exchange[71]
- TASK: TASK exchange functional[13]
- mTASK: Modified TASK exchange functional[110]

5.3.4 Correlation Functionals

Note: All correlation functionals in this section can be invoked using the *$rem* variable CORRELATION. Popular and/or recommended functionals within each class are listed first and indicated in bold. The rest are in alphabetical order.

◦ Local Spin-Density Approximation (LSDA)

- **PW92**: Perdew-Wang parameterization of the LSDA correlation energy from 1992[117]
- **VWN5** (VWN): Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #5[167]
- srVWN: Short-range version of the VWN correlation functional by Toulouse and coworkers[160]
- Liu-Parr: Liu-Parr $\rho^{1/3}$ model from the functional expansion formulation[97]
- PK09: Proynov-Kong parameterization of the LSDA correlation energy from 2009[138]

- PW92RPA: Perdew-Wang parameterization of the LSDA correlation energy from 1992 with RPA values [117]
- srPW92: Short-range version of the PW92 correlation functional by Paziani and coworkers [113]
- PZ81: Perdew-Zunger parameterization of the LSDA correlation energy from 1981 [118]
- VWN1: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #1 [167]
- VWN1RPA: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #1 with RPA values [167]
- VWN2: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #2 [167]
- VWN3: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #3 [167]
- VWN4: Vosko-Wilk-Nusair parameterization of the LSDA correlation energy #4 [167]
- Wigner:Wigner correlation functional (simplification of LYP) [153,179]

◦ Generalized Gradient Approximation (GGA)

- BOP: "one-parameter progressive" correlation functional with parameters appropriate for B88 exchange [161]
- **PBE**: Perdew, Burke, and Ernzerhof correlation functional [120]
- **LYP**: Lee-Yang-Parr opposite-spin correlation functional [91]
- P86: Perdew-Wang correlation functional from 1986 based on the PZ81 LSDA functional [114]
- P86VWN5: Perdew-Wang correlation functional from 1986 based on the VWN5 LSDA functional [114]
- PBEloc: PBE correlation functional with a modified beta term by Della Sala and coworkers [38]
- PBEBOP: "one-parameter progressive" correlation functional with parameters appropriate for PBE exchange [161]
- PBEsol: PBE correlation functional modified for solids [124]
- srPBE: Short-range version of the PBE correlation functional by Goll and coworkers [52,53]
- PW91: Perdew-Wang correlation functional from 1991 [119]
- regTPSS: Slight modification of the PBE correlation functional (also called vPBEc) [145]

◦ Meta-Generalized Gradient Approximation (meta-GGA)

- **TPSS**:Tao, Perdew, Staroverov, and Scuseria correlation functional [159]
- **revTPSS**: Revised version of the TPSS correlation functional [126]
- B95: Becke's two-parameter correlation functional from 1995 [21]
- oTPSS: TPSS correlation functional with 2 refit parameters (for use with oTPSS exchange) by Grimme and coworkers [51]
- PK06: Proynov-Kong "tLap" functional with τ and Laplacian dependence [136]
- PKZB: Perdew, Kurth, Zupan, and Blaha correlation functional [121]
- SCAN: Strongly Constrained and Appropriately Normed correlation functional [157]
- rSCAN: Regularized SCAN correlation [14,47]
- r++SCAN: Regularized SCAN with uniform density limit and coordinate scaling behavior [47]
- r2SCAN: Re-Regularized SCAN correlation [47]**? ?**
- revSCAN: Revised SCAN correlation [104]
- TM: Tao-Mo correlation functional, representing a minor modification to the TPSS correlation functional [158]
- revTM: Revised TM correlation [71]
- rregTM: Revised regularized TM correlation [72]
- TPSSloc: The TPSS correlation functional with the PBE component replaced by the PBEloc correlation functional [38]

5.3.5 Exchange-Correlation Functionals

Note: All exchange-correlation functionals in this section can be invoked using the *$rem* variable METHOD. For backwards compatibility, all of the exchange-correlation functionals *except* for the ones marked with an asterisk can be used with the *$rem* variable EXCHANGE. Popular and/or recommended functionals within each class are listed first and indicated in bold. The rest are in alphabetical order.

- Local Spin-Density Approximation (LSDA)
  - **SPW92***: Slater LSDA exchange + PW92 LSDA correlation
  - LDA: Slater LSDA exchange + VWN5 LSDA correlation
  - SVWN5*: Slater LSDA exchange + VWN5 LSDA correlation
- Generalized Gradient Approximation (GGA)
  - **B97-D3(0)**: B97-D with a fitted DFT-D3(0) tail instead of the original DFT-D2 tail[58]
  - **B97-D**: 9-parameter dispersion-corrected (DFT-D2) functional by Grimme[56]
  - **PBE***: PBE GGA exchange + PBE GGA correlation
  - **BLYP***: B88 GGA exchange + LYP GGA correlation
  - **revPBE***: revPBE GGA exchange + PBE GGA correlation
  - BEEF-vdW: 31-parameter semi-empirical exchange functional developed via a Bayesian error estimation framework paired with PBE correlation and vdW-DF-10 NLC[176]
  - BOP: B88 GGA exchange + BOP "one-parameter progressive" GGA correlation[161]
  - BP86*: B88 GGA exchange + P86 GGA correlation
  - BP86VWN*: B88 GGA exchange + P86VWN5 GGA correlation
  - BPBE*: B88 GGA exchange + PBE GGA correlation
  - EDF1: Modification of BLYP to give good performance in the 6-31+G* basis set[9]
  - EDF2: Modification of B3LYP to give good performance in the cc-pVTZ basis set for frequencies[92]
  - GAM: 21-parameter non-separable gradient approximation functional by Truhlar and coworkers[185]
  - HCTH93 (HCTH/93): 15-parameter functional trained on 93 systems by Handy and coworkers[63]
  - HCTH120 (HCTH/120): 15-parameter functional trained on 120 systems by Boese *et al.*[30]
  - HCTH147 (HCTH/147): 15-parameter functional trained on 147 systems by Boese *et al.*[30]
  - HCTH407 (HCTH/407): 15-parameter functional trained on 407 systems by Boese and Handy[27]
  - HLE16 – HCTH/407 exchange functional enhanced by a factor of 1.25 + HCTH/407 correlation functional enhanced by a factor of 0.5[165]
  - KT1: GGA functional designed specifically for shielding constant calculations[75]
  - KT2: GGA functional designed specifically for shielding constant calculations[75]
  - KT3: GGA functional with improved results for main-group nuclear magnetic resonance shielding constants[76]
  - mPW91*: mPW91 GGA exchange + PW91 GGA correlation
  - N12: 21-parameter non-separable gradient approximation functional by Peverati and Truhlar[131]
  - OLYP*: OPTX GGA exchange + LYP GGA correlation
  - PBEOP: PBE GGA exchange + PBEOP "one-parameter progressive" GGA correlation[161]

- PBEsol*: PBEsol GGA exchange + PBEsol GGA correlation
- PW91*: PW91 GGA exchange + PW91 GGA correlation
- RPBE*: RPBE GGA exchange + PBE GGA correlation
- rVV10*: rPW86 GGA exchange + PBE GGA correlation + rVV10 non-local correlation[146]
- SOGGA*: SOGGA GGA exchange + PBE GGA correlation
- SOGGA11: 20-parameter functional by Peverati, Zhao, and Truhlar[134]

◦ Meta-Generalized Gradient Approximation (meta-GGA)

- **B97M-rV***: The 12-parameter combinatorially-optimized, dispersion-corrected (VV10) B97M-V density functional by Mardirossian and Head-Gordon[101] with the VV10 NLC functional replaced by the rVV10 NLC functional[103]
- **M06-L**: 34-parameter functional by Zhao and Truhlar[191]
- **TPSS***: TPSS meta-GGA exchange + TPSS meta-GGA correlation
- **revTPSS***: revTPSS meta-GGA exchange + revTPSS meta-GGA correlation
- BLOC*: BLOC meta-GGA exchange + TPSSloc meta-GGA correlation
- M11-L: 44-parameter dual-range functional by Peverati and Truhlar[130]
- mBEEF: 64-parameter exchange functional paired with the PBEsol correlation functional[177]
- MGGA_MS0: MGGA_MS0 meta-GGA exchange + regTPSS GGA correlation[154]
- MGGA_MS1: MGGA_MS1 meta-GGA exchange + regTPSS GGA correlation[155]
- MGGA_MS2: MGGA_MS2 meta-GGA exchange + regTPSS GGA correlation[155]
- MGGA_MVS: MGGA_MVS meta-GGA exchange + regTPSS GGA correlation[156]
- MN12-L: 58-parameter meta-nonseparable gradient approximation functional by Peverati and Truhlar[132]
- MN15-L: 58-parameter meta-nonseparable gradient approximation functional by Yu, He, and Truhlar[187]
- oTPSS*: oTPSS meta-GGA exchange + oTPSS meta-GGA correlation
- PKZB*: PKZB meta-GGA exchange + PKZB meta-GGA correlation
- revM06-L: 31-parameter revised M06-L functional[172]
- SCAN*: SCAN meta-GGA exchange + SCAN meta-GGA correlation
- rSCAN: rSCAN exchange + rSCAN correltaion
- r++SCAN: r++SCAN exchange + r++SCAN correlation
- r2SCAN: r2SCAN exchange + r2SCAN correlation
- r4SCAN: r4SCAN exchange + r2SCAN correlation
- revSCAN: revSCAN exchange + revSCAN correlation
- t-HCTH (τ-HCTH): 16-parameter functional by Boese and Handy[28]
- TM*: TM meta-GGA exchange + TM meta-GGA correlation[158]
- revTM: revTM exchange + revTM correlation
- regTM: regTM exchange + regTPSS correlation
- rregTM: regTM exchange + rregTM correlation
- TASK: TASK exchange + PW92 correlation
- mTASK: mTASK exchange + PW92 correlation

- VSXC: 21-parameter functional by Voorhis and Scuseria[163]

○ Global Hybrid Generalized Gradient Approximation (GH GGA)

- **B3LYP**: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange + 19% VWN1RPA LSDA correlation + 81% LYP GGA correlation[19,152]
- **PBE0**: 25% HF exchange + 75% PBE GGA exchange + PBE GGA correlation[7]
- **revPBE0**: 25% HF exchange + 75% revPBE GGA exchange + PBE GGA correlation
- **B97**: Becke's original 10-parameter density functional with 19.43% HF exchange[22]
- B1LYP: 25% HF exchange + 75% B88 GGA exchange + LYP GGA correlation[5]
- B1PW91: 25% HF exchange + 75% B88 GGA exchange + PW91 GGA correlation[5]
- B3LYP5: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[19,152]
- B3P86: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange+ 19% VWN1RPA LSDA correlation + 81% P86 GGA correlation
- B1LYP: 25% HF exchange + 75% B88 GGA exchange + LYP GGA correlation[5]
- B1PW91: 25% HF exchange + 75% B88 GGA exchange + PW91 GGA correlation[5]
- B3LYP5: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[19,152]
- B3P86: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange+ 19% VWN1RPA LSDA correlation + 81% P86 GGA correlation
- B3PW91: 20% HF exchange + 8% Slater LSDA exchange + 72% B88 GGA exchange+ 19% PW92 LSDA correlation + 81% PW91 GGA correlation[19]
- B5050LYP: 50% HF exchange + 8% Slater LSDA exchange + 42% B88 GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[148]
- B97-1: Self-consistent parameterization of Becke's B97 density functional with 21% HF exchange[63]
- B97-2: Re-parameterization of B97 by Tozer and coworkers with 21% HF exchange[181]
- B97-3: 16-parameter version of B97 by Keal and Tozer with $\approx$ 26.93% HF exchange[77]
- B97-K: Re-parameterization of B97 for kinetics by Boese and Martin with 42% HF exchange[29]
- BHHLYP: 50% HF exchange + 50% B88 GGA exchange + LYP GGA correlation
- HFLYP*: 100% HF exchange + LYP GGA correlation
- MPW1K: 42.8% HF exchange + 57.2% mPW91 GGA exchange + PW91 GGA correlation[99]
- MPW1LYP: 25% HF exchange + 75% mPW91 GGA exchange + LYP GGA correlation[6]
- MPW1PBE: 25% HF exchange + 75% mPW91 GGA exchange + PBE GGA correlation[6]
- MPW1PW91: 25% HF exchange + 75% mPW91 GGA exchange + PW91 GGA correlation[6]
- O3LYP: 11.61% HF exchange + $\approx$ 7.1% Slater LSDA exchange + 81.33% OPTX GGA exchange + 19% VWN5 LSDA correlation + 81% LYP GGA correlation[67]
- PBEh-3c: Low-cost composite scheme of Grimme and coworkers for use with the def2-mSVP basis set only[60]
- PBE50: 50% HF exchange + 50% PBE GGA exchange + PBE GGA correlation[24]
- SOGGA11-X: 21-parameter functional with 40.15% HF exchange by Peverati and Truhlar[128]
- WC04: Hybrid density functional optimized for the computation of ^{13}C chemical shifts[180]

- WP04: Hybrid density functional optimized for the computation of ^{1}H chemical shifts [180]
- X3LYP: 21.8% HF exchange + 7.3% Slater LSDA exchange + ≈ 54.24% B88 GGA exchange + ≈ 16.66% PW91 GGA exchange + 12.9% VWN1RPA LSDA correlation + 87.1% LYP GGA correlation [183]

◦ Global Hybrid Meta-Generalized Gradient Approximation (GH meta-GGA)

- **M06-2X**: 29-parameter functional with 54% HF exchange by Zhao and Truhlar [195]
- **M08-HX**: 47-parameter functional with 52.23% HF exchange by Zhao and Truhlar [194]
- **TPSSh**: 10% HF exchange + 90% TPSS meta-GGA exchange + TPSS meta-GGA correlation [151]
- **revTPSSh**: 10% HF exchange + 90% revTPSS meta-GGA exchange + revTPSS meta-GGA correlation [41]
- B1B95: 28% HF exchange + 72% B88 GGA exchange + B95 meta-GGA correlation [21]
- B3TLAP: 17.13% HF exchange + 9.66% Slater LSDA exchange + 72.6% B88 GGA exchange + PK06 meta-GGA correlation [136,137]
- BB1K: 42% HF exchange + 58% B88 GGA exchange + B95 meta-GGA correlation [197]
- BMK: Boese-Martin functional for kinetics with 42% HF exchange [29]
- dlDF: Dispersion-less density functional (based on the M05-2X functional form) by Szalewicz and coworkers [127]
- M05: 22-parameter functional with 28% HF exchange by Zhao, Schultz, and Truhlar [198]
- M05-2X: 19-parameter functional with 56% HF exchange by Zhao, Schultz, and Truhlar [199]
- M06: 33-parameter functional with 27% HF exchange by Zhao and Truhlar [195]
- M06-HF: 32-parameter functional with 100% HF exchange by Zhao and Truhlar [192]
- M08-SO: 44-parameter functional with 56.79% HF exchange by Zhao and Truhlar [194]
- MGGA_MS2h: 9% HF exchange + 91 % MGGA_MS2 meta-GGA exchange + regTPSS GGA correlation [155]
- MGGA_MVSh: 25% HF exchange + 75 % MGGA_MVS meta-GGA exchange + regTPSS GGA correlation [156]
- MN15: 59-parameter functional with 44% HF exchange by Truhlar and coworkers [186]
- MPW1B95: 31% HF exchange + 69% mPW91 GGA exchange + B95 meta-GGA correlation [189]
- MPWB1K: 44% HF exchange + 56% mPW91 GGA exchange + B95 meta-GGA correlation [189]
- PW6B95: 6-parameter combination of 28 % HF exchange, 72 % optimized PW91 GGA exchange, and re-optimized B95 meta-GGA correlation by Zhao and Truhlar [190]
- PWB6K: 6-parameter combination of 46 % HF exchange, 54 % optimized PW91 GGA exchange, and re-optimized B95 meta-GGA correlation by Zhao and Truhlar [190]
- revM06: 32-parameter functional with 40.41% HF exchange [173]
- SCAN0: 25% HF exchange + 75% SCAN meta-GGA exchange + SCAN meta-GGA correlation [69]
- t-HCTHh (τ-HCTHh): 17-parameter functional with 15% HF exchange by Boese and Handy [28]
- TPSS0: 25% HF exchange + 75% TPSS meta-GGA exchange + TPSS meta-GGA correlation [55]

◦ Range-Separated Hybrid Generalized Gradient Approximation (RSH GGA)

- **wB97X-rV** (ωB97X-V): 10-parameter combinatorially-optimized, dispersion-corrected (VV10) functional with 16.7% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$ [100], with VV10 replaced by rVV10 [103].

- **wB97X-D3** (ωB97X-D3): 16-parameter dispersion-corrected (DFT-D3(0)) functional with $\approx$ 19.57% SR HF exchange, 100% LR HF exchange, and $\omega = 0.25$ [94]
- **wB97X-D** (ωB97X-D): 15-parameter dispersion-corrected (DFT-CHG) functional with $\approx$ 22.2% SR HF exchange, 100% LR HF exchange, and $\omega = 0.2$ [33]
- CAM-B3LYP: Coulomb-attenuating method functional by Handy and coworkers [184]
- CAM-QTP00: Re-parameterized CAM-B3LYP designed to satisfy the IP-theorem for all occupied orbitals of the water molecule [164]
- CAM-QTP01: Re-parameterized CAM-B3LYP optimized to satisfy the valence IPs of the water molecule, 34 excitation states, and G2-1 atomization energies [73]
- HSE-HJS: Screened-exchange "HSE06" functional with 25% SR HF exchange, 0% LR HF exchange, and ω=0.11, using the updated HJS PBE exchange hole model [65,82]
- LC-rVV10*: LC-VV10 density functional with the VV10 NLC functional replaced by the rVV10 NLC functional [103]
- LC-wPBE08 (LC-ωPBE08): 0% SR HF exchange + 100% LR HF exchange + ωPBE GGA exchange + PBE GGA correlation (ω=0.45) [175]
- LRC-BOP (LRC-μBOP): 0% SR HF exchange + 100% LR HF exchange + μB88 GGA exchange + BOP GGA correlation (ω=0.47) [142,150]
- LRC-wPBE (LRC-ωPBE): 0% SR HF exchange + 100% LR HF exchange + ωPBE GGA exchange + PBE GGA correlation (ω=0.3) [143]
- LRC-wPBEh (LRC-ωPBEh): 20% SR HF exchange + 100% LR HF exchange + 80% ωPBE GGA exchange + PBE GGA correlation (ω=0.2) [144]
- N12-SX: 26-parameter non-separable GGA with 25% SR HF exchange, 0% LR HF exchange, and $\omega = 0.11$ [133]
- rCAM-B3LYP: Re-fit CAM-B3LYP with the goal of minimizing many-electron self-interaction error [37]
- wB97 (ωB97): 13-parameter functional with 0% SR HF exchange, 100% LR HF exchange, and $\omega = 0.4$ [32]
- wB97X (ωB97X): 14-parameter functional with $\approx$ 15.77% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$ [32]

◦ Range-Separated Hybrid Meta-Generalized Gradient Approximation (RSH meta-GGA)

- **wB97M-rV** (ωB97M-V): 12-parameter combinatorially-optimized, dispersion-corrected (VV10) functional with 15% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$ [102]
- M06-SX: local revM06-L functional with 33.5% SR HF exchange [174], with VV10 replaced by rVV10 [103].
- M11: 40-parameter functional with 42.8% SR HF exchange, 100% LR HF exchange, and $\omega = 0.25$ [129]
- MN12-SX: 58-parameter non-separable meta-GGA with 25% SR HF exchange, 0% LR HF exchange, and $\omega = 0.11$ [133]
- revM11: 22-parameter functional with 22.5% SR HF exchange, 100% LR HF exchange, and $\omega = 0.4$ [166]
- wM05-D (ωM05-D): 21-parameter dispersion-corrected (DFT-CHG) functional with $\approx$ 36.96% SR HF exchange, 100% LR HF exchange, and $\omega = 0.2$ [93]
- wM06-D3 (ωM06-D3): 25-parameter dispersion-corrected [DFT-D3(0)] functional with $\approx$ 27.15% SR HF exchange, 100% LR HF exchange, and $\omega = 0.3$ [94]

5.3.6 Specialized Functionals

- SRC1-R1: TDDFT short-range corrected functional [Eq. (1) in Ref. 25, 1st row atoms]
- SRC1-R2: TDDFT short-range corrected functional [Eq. (1) in Ref. 25, 2nd row atoms]
- SRC2-R1: TDDFT short-range corrected functional [Eq. (2) in Ref. 25, 1st row atoms]
- SRC2-R2: TDDFT short-range corrected functional [Eq. (2) in Ref. 25, 2nd row atoms]
- BR89: Becke-Roussel meta-GGA exchange functional modeled after the hydrogen atom[23]
- B94: meta-GGA correlation functional by Becke that uses the BR89 exchange functional to compute the Coulomb potential[20]
- B94hyb: modified version of the B94 correlation functional for use with the BR89B94hyb exchange-correlation functional[20]
- BR89B94h: 15.4% HF exchange + 84.6% BR89 meta-GGA exchange + BR89hyb meta-GGA correlation[20]
- BRSC: Exchange component of the original B05 exchange-correlation functional**?**
- MB05: Exchange component of the modified B05 (BM05) exchange-correlation functional[140]
- B05: A full exact-exchange Kohn-Sham scheme of Becke that uses the exact-exchange energy density (RI) and accounts for static correlation[139,141]**?**
- BM05 (XC): Modified B05 hyper-GGA scheme that uses MB05 instead of BRSC as the exchange functional[140]
- PSTS: Hyper-GGA (100% HF exchange) exchange-correlation functional of Perdew, Staroverov, Tao, and Scuseria[125]
- MCY2: Mori-Sánchez-Cohen-Yang adiabatic connection-based hyper-GGA exchange-correlation functional[36,95,107]

5.3.7 User-Defined Density Functionals

Users can also request a customized density functional consisting of any linear combination of exchange and/or correlation functionals available in QC-PBC. A "general" density functional of this sort is requested by setting EXCHANGE = GEN and then specifying the functional by means of an *$xc_functional* input section consisting of one line for each desired exchange (X) or correlation (C) component of the functional, and having the format shown below.

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

Each line requires three variables: X or C to designate whether this is an exchange or correlation component; the symbolic representation of the functional, as would be used for the EXCHANGE or CORRELATION keywords variables

as described above; and a real number coefficient for each component. Note that Hartree-Fock exchange can be designated either as “X" or as “K". Examples are shown below.

Example 5.1 Input for an energy calculation on NaCl using a variant of B3LYP with 50% exact exchange.

```
$comment
Run custom B3LYP variant with 50% exact exchange (replacing b88 exchange)
$end

$lattice
3 ! dimension
3.42201500000000 0.00000000000000 1.97570200000000
1.14067100000000 3.22630600000000 1.97570200000000
0.00000000000000 0.00000000000000 3.95140200000000
$end

$unitcell
absolute
0 1
Na 0.0000000000 0.0000000000 0.0000000000
Cl 2.2813430000 1.6131530000 3.9514030000
$end


$rem
    JOBTYPE     sp                  ! Energy calculation
    METHOD      gen                 ! define our own dft functional
    BASIS       szv-gth             ! double zeta basis
    KECUT       1000                ! planewave cutoff 1000 eV
    PSEUDO      gth-pbe             ! use GTH pseudopotential fit with PBE
    MP_MESH     [3,3,3]             ! 3x3x3 k-point Monkhorst-Pack mesh
$end



$xc_functional
X slater 0.08
X b88 0.42
C vwn1rpa 0.19
C lyp 0.81
K 0.5
$end
```

View output online

5.4 Basic DFT Job Control

Basic SCF job control was described in Section 4.5 in the context of Hartree-Fock theory and is largely the same for DFT. The keywords METHOD and BASIS are required, although for DFT the former could be substituted by specifying EXCHANGE and CORRELATION instead.

METHOD

Specifies the exchange-correlation functional.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use METHOD = *NAME*, where *NAME* is either HF for Hartree-Fock theory or else one of the DFT methods listed in Section 5.3.5.

RECOMMENDATION:

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

EXCHANGE

Specifies the exchange functional (or most exchange-correlation functionals for backwards compatibility).

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

NAME Use EXCHANGE = *NAME*, where *NAME* is either:

1) One of the exchange functionals listed in Section 5.3.3

2) One of the XC functionals listed in Section 5.3.5 that is not marked with an asterisk.

3) GEN, for a user-defined functional (see Section 5.3.7).

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations are indicated in bold in Sections 5.3.5 and 5.3.3.

CORRELATION

Specifies the correlation functional.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

NAME Use CORRELATION = *NAME*, where *NAME* is one of the correlation functionals listed in Section 5.3.4.

RECOMMENDATION:

In general, consult the literature to guide your selection. Our recommendations are indicated in bold in Section 5.3.4.

The following *$rem* variables are related to the choice of the quadrature grid required to integrate the XC part of the functional, which does not appear in Hartree-Fock theory. Beyond the uniform grid used in the GPW approach, atom-centered DFT quadrature grids are described in Section 5.5.

GRID_TYPE_XC

Specifies the type of grid to use for DFT calculations.

TYPE:

INTEGER

DEFAULT:

11 Uniform grid for GPW calculation

OPTIONS:

n Use SG-n for all atoms, $n = 1, 2$, or 3

XY A string of two six-digit integers X and Y, where X is the number of radial points and Y is the number of angular points where possible numbers of Lebedev angular points, which must be an allowed value from Table 5.2 in Section 5.5.

$-XY$ Similar format for Gauss-Legendre grids, with the six-digit integer X corresponding to the number of radial points and the six-digit integer Y providing the number of Gauss-Legendre angular points, $Y = 2N^2$.

RECOMMENDATION:

For all-electron calculations, the atom-centered quadrature grid needs to be set explicitly. Users should consult Table 5.3 for the choice of quadrature grid.

BVK_CGTO

Whether CGTO is handled in Bloch orbital basis.

TYPE:

BOOL

DEFAULT:

False CGTO is computed in the **k**-point format.

OPTIONS:

True CGTO is computed in the BvK format.

RECOMMENDATION:

When all-electron periodic FTC is requested, this is automatically turned on. We recommend using CGTO in BvK format when **k**-point mesh is dense, and the memory is limited.

5.5 DFT Numerical Quadrature

5.5.1 Introduction

In practical DFT calculations, the forms of the approximate exchange-correlation functionals used are quite complicated, such that the required integrals involving the functionals generally cannot be evaluated analytically. QC-PBC evaluates these integrals through numerical quadrature directly applied to the exchange-correlation integrand. Several standard quadrature grids are available ("SG-n", $n = 0, 1, 2, 3$), with a default value that is automatically set according to the complexity of the functional in question.

The quadrature approach in QC-PBC 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. QC-PBC uses the nuclear partitioning of Becke,[17] though without the "atomic size adjustments" of Ref. 17. The atomic integrals are then evaluated through standard one-center numerical techniques. Thus, the exchange-correlation energy is obtained as

$$E_{\mathrm{XC}} = \sum_{A}^{\text{atoms}} \sum_{i \in A}^{\text{points}} w_{Ai} f(\mathbf{r}_{Ai}) \,, \tag{5.15}$$

where the function f is the aforementioned XC integrand and the quantities w_{Ai} are the quadrature weights. The sum over i runs over grid points belonging to atom A, which are located at positions $\mathbf{r}_{Ai} = \mathbf{R}_A + \mathbf{r}_i$, so this approach

No. Points	Degree ($\ell_{\max}$)	No. Points	Degree ($\ell_{\max}$)	No. Points	Degree ($\ell_{\max}$)
6	3	230	25	1730	71
18	5	266	27	2030	77
26	7	302	29	2354	83
38	9	350	31	2702	89
50	11	434	35	3074	95
74	13	590	41	3470	101
86	15	770	47	3890	107
110	17	974	53	4334	113
146	19	1202	59	4802	119
170	21	1454	65	5294	125
194	23				

Table 5.2: Lebedev angular quadrature grids available in QC-PBC.

requires only the choice of a suitable one-center integration grid (to define the $\mathbf{r}_i$), which is independent of nuclear configuration. These grids are implemented in QC-PBC in a way that ensures that the $E_{\rm XC}$ is rotationally-invariant, *i.e.*, that is does not change when the molecule undergoes rigid rotation in space.[74]

Quadrature grids are further separated into radial and angular parts. Within QC-PBC, the radial part is usually treated by the Euler-Maclaurin scheme proposed by Murray *et al.*,[108] which maps the semi-infinite domain $[0, \infty)$ onto $[0, 1)$ and applies the extended trapezoid rule to the transformed integrand. Alternatively, Gill and Chien proposed a radial scheme based on a Gaussian quadrature on the interval $[0, 1]$ with a different weight function.[34] This "MultiExp" radial quadrature is exact for integrands that are a linear combination of a geometric sequence of exponential functions, and is therefore well suited to evaluating atomic integrals. However, the task of generating the MultiExp quadrature points becomes increasingly ill-conditioned as the number of radial points increases, so that a "double exponential" radial quadrature[105,106] is used for the largest standard grids in QC-PBC,[105,106] namely SG-2 and SG-3.[42] (See Section 5.5.3.)

5.5.2 Angular Grids

For a fixed value of the radial spherical-polar coordinate r, a function $f(\mathbf{r}) \equiv f(r, \theta, \phi)$ has an exact expansion in spherical harmonic functions,

$$f(r, \theta, \phi) = \sum_{\ell=0}^{\infty} \sum_{m=-\ell}^{\ell} c_{\ell m} \mathrm{Y}_{\ell m}(\theta, \phi) \; . \tag{5.16}$$

Angular quadrature grids are designed to integrate $f(r, \theta, \phi)$ for fixed r, and are often characterized by their degree, meaning the maximum value of ℓ for which the quadrature is exact, as well as by their efficiency, meaning the number of spherical harmonics exactly integrated per degree of freedom in the formula. QC-PBC supports the following two types of angular grids.

- **Lebedev grids.** These are specially-constructed grids for quadrature on the surface of a sphere,[87–90] based on the octahedral point group. Lebedev grids available in QC-PBC are listed in Table 5.2. These grids typically have near-unit efficiencies, with efficiencies exceeding unity in some cases. A Lebedev grid is selected by specifying the number of grid points (from Table 5.2) using the *$rem* keyword XC_GRID, as discussed below.
- **Gauss-Legendre grids.** These are spherical direct-product grids in the two spherical-polar angles, θ and ϕ. Integration in over θ is performed using a Gaussian quadrature derived from the Legendre polynomials, while integration over ϕ is performed using equally-spaced points. A Gauss-Legendre grid is selected by specifying

Pruned Grid	Ref.	Parent Grid (N_r, N_Ω)	No. Grid Points (C atom)[a]	Default Grid for Which Functionals?[b]
SG-0	35	(23, 170)	1,390 (36%)	None
SG-1	49	(50, 194)	3,816 (39%)	LDA, most GGAs and hybrids
SG-2	42	(75, 302)	7,790 (34%)	Meta-GGAs; B95- and B97-based functionals
SG-3	42	(99, 590)	17,674 (30%)	Minnesota functionals

[a]Number in parenthesis is the fraction of points retained from the parent grid

[b]Reflects QC-PBC versions since v. 4.4.2

Table 5.3: Standard quadrature grids available in QC-PBC, along with the number of grid points for a carbon atom, showing the reduction in grid points due to pruning.

the total number of points, $2N^2$, to be used for the integration, which specifies a grid consisting of $2N_\phi$ points in ϕ and N_θ in θ, for a degree of $2N-1$. Gauss-Legendre grids exhibit efficiencies of only 2/3, and are thus lower in quality than Lebedev grids for the same number of grid points, but have the advantage that they are defined for arbitrary (and arbitrarily-large) numbers of grid points. This offers a mechanism to achieve arbitrary accuracy in the angular integration, if desired.

Combining these radial and angular schemes yields an intimidating selection of quadratures, so it is useful to standardize the grids. This is done for the convenience of the user, to facilitate comparisons in the literature, and also for developers wishing to compared detailed results between different software programs, because the total electronic energy is sensitive to the details of the grid, just as it is sensitive to details of the basis set. Standard quadrature grids are discussed next.

5.5.3 Standard Quadrature Grids

Four different "standard grids" are available in QC-PBC, designated SG-n, for $n = 0, 1, 2$, or 3; both quality and the computational cost of these grids increases with n. These grids are constructed starting from a "parent" grid $(N_r,\ N_\Omega)$ consisting of N_r radial spheres with N_Ω angular (Lebedev) grid points on each, then systematically pruning the number of angular points in regions where sophisticated angular quadrature is not necessary, such as near the nuclei where the charge density is nearly spherically symmetric and at long distance from the nuclei where it varies slowly. A large number of points are retained in the valence region where angular accuracy is critical. The SG-n grids are summarized in Table 5.3. While many electronic structure programs use some kind of procedure to delete unnecessary grid points in the interest of computational efficiency, QC-PBC's SG-n grids are notable in that the complete grid specifications are available in the peer-reviewed literature,[35,42,49] to facilitate reproduction of QC-PBC DFT calculations using other electronic structure programs. Just as computed energies may vary quite strongly with the choice of basis set, so too in DFT they may vary strongly with the choice of quadrature grid. In publications, users should always specify the grid that is used, and it is suggested to cite the appropriate literature reference from Table 5.3.

The SG-0 and SG-1 grids are designed for calculations on large molecules using GGA functionals. SG-1 affords integration errors on the order of ∼0.2 kcal/mol for medium-sized molecules and GGA functionals, including for demanding test cases such as reaction enthalpies for isomerizations. (Integration errors in total energies are no more than a few μhartree, or ∼0.01 kcal/mol.) The SG-0 grid was derived in similar fashion, and affords a root-mean-square error in atomization energies of 72 μhartree with respect to SG-1, while relative energies are reproduced well.[35] In either case, errors of this magnitude are typically considerably smaller than the intrinsic errors in GGA energies, and hence acceptable. As seen in Table 5.3, SG-1 retains $< 40\%$ of the grid points of its parent grid, which translates directly into cost savings.

Both SG-0 and SG-1 were optimized so that the integration error in the energy falls below a target threshold, but derivatives of the energy (including such properties as (hyper)polarizabilities[31]) are often more sensitive to the quality of the integration grid. Special care is required, for example, when imaginary vibrational frequencies are encountered, as low-frequency (but real) vibrational frequencies can manifest as imaginary if the grid is sparse. If imaginary frequencies are found, or if there is some doubt about the frequencies reported by QC-PBC, the recommended procedure is to perform the geometry optimization and vibrational frequency calculations again using a higher-quality grid. (The optimization should converge quite quickly if the previously-optimized geometry is used as an initial guess.)

SG-1 was the default DFT integration grid for all density functionals for QC-PBC versions 3.2–4.4. Beginning with QC-PBC v. 4.4.2, however, the default grid is functional-dependent, as summarized in Table 5.3. This is a reflection of the fact that although SG-1 is adequate for energy calculations using most GGA and hybrid functionals (although care must be taken for some other properties, as discussed below), it is *not* adequate to integrate many functionals developed since ∼2005. These include meta-GGAs, which are more complicated due to their dependence on the kinetic energy density (τ_σ in Eq. (5.10)) and/or the Laplacian of the density ($\nabla^2\rho_\sigma$). Functionals based on B97, along with the Minnesota suite of functionals,[195,196] contain relatively complicated expressions for the exchange inhomogeneity factor, and are therefore also more sensitive to the quality of the integration grid.[42,100,178] To integrate these modern density functionals, the SG-2 and SG-3 grids were developed,[42] which are pruned versions of the medium-quality (75, 302) and high-quality (99, 590) integration grids, respectively. Tests of properties known to be highly sensitive to the quality of the integration grid, such as vibrational frequencies, hyper-polarizabilities, and potential energy curves for non-bonded interactions, demonstrate that SG-2 is usually adequate for meta-GGAs and B97-based functionals, and in many cases is essentially converged with respect to an unpruned (250, 974) grid.[42] The Minnesota functionals are more sensitive to the grid, and while SG-3 is often adequate, it is not completely converged in the case of non-bonded interactions.[42]

Note:

1. SG-0 was re-optimized for QC-PBC v. 3.0, so results may differ slightly as compared to older versions of the program.

2. The SG-2 and SG-3 grids use a double-exponential radial quadrature,[42] whereas a general grid (selected by setting XC_GRID = XY, as described in Section 5.4) uses an Euler-MacLaurin radial quadrature. As such, *absolute* energies cannot be compared between, *e.g.*, SG-2 and XC_GRID = 000075000302, even though SG-2 uses a pruned (75, 302) grid. However, energy *differences* should be quite similar between the two.

3. As noted in Ref. 42, for Minnesota functionals some wiggles in potential energy surfaces may persist with the SG-3 grid, especially for longer-range non-bonded interactions. Although these are rarely problematic for energy differences, if the user wants to eliminate these oscillations then we recommend an unpruned (99, 590) grid, *i.e.*, XC_GRID = 000099000590.

5.5.4 Consistency Check and Cutoffs

Whenever QC-PBC calculates numerical density functional integrals, the electron density itself is also integrated numerically as a test of the quality of the numerical quadrature. The extent to which this numerical result differs from the number of electrons is an indication of the accuracy of the other numerical integrals. A warning message is printed whenever the relative error in the numerical electron count reaches 0.01%, indicating that the numerical XC results may not be reliable. If the warning appears on the first SCF cycle it is probably not serious, because the initial-guess density matrix is sometimes not idempotent. This is the case with the SAD guess discussed in Section 4.6, and also with a density matrix that is taken from a previous geometry optimization cycle, and in such cases the problem will likely correct itself in subsequent SCF iterations. If the warning persists, however, then one should consider either

using a finer grid or else selecting an alternative initial guess.

By default, QC-PBC will estimate the magnitude of various XC contributions on the grid and eliminate those determined to be numerically insignificant. QC-PBC uses specially-developed cutoff procedures which permits evaluation of the XC energy and potential in only $\mathcal{O}(N)$ work for large molecules. This is a significant improvement over the formal $\mathcal{O}(N^3)$ scaling of the XC cost, and is critical in enabling DFT calculations to be carried out on very large systems. In rare cases, however, the default cutoff scheme can be too aggressive, eliminating contributions that should be retained; this is almost always signaled by an inaccurate numerical density integral. An example of when this could occur is in calculating anions with multiple sets of diffuse functions in the basis. A remedy may be to increase the size of the quadrature grid.

5.6 Range-Separated Hybrid Density Functionals

5.6.1 Introduction

Whereas RSH functionals such as LRC-ωPBE are attempts to add 100% LR Hartree-Fock exchange with minimal perturbation to the original functional (PBE, in this example), other RSH functionals are of a more empirical nature and their range-separation parameters have been carefully parameterized along with all of the other parameters in the functional. These cases are functionals are discussed first, in Section 5.6.2, because their range-separation parameters should be taken as fixed. User-defined values of the range-separation parameter are discussed in Section 5.6.3.

5.6.2 Semi-Empirical RSH Functionals

Semi-empirical RSH functionals for which the range-separation parameter should be considered fixed include the ωB97, ωB97X, and ωB97X-D functionals developed by Chai and Head-Gordon;[32,33] ωB97X-V and ωB97M-V from Mardirossian and Head-Gordon;[100,102] M11 from Peverati and Truhlar;[129] ωB97X-D3, ωM05-D, and ωM06-D3 from Chai and coworkers;[93,94] and the screened exchange functionals N12-SX and MN12-SX from Truhlar and co-workers.[133] More recently, Mardirossian and Head-Gordon developed two RSH functionals, ωB97X-V and ωB97M-V, via a combinatorial approach by screening over 100,000 possible functionals in the first case and over 10 billion possible functionals in the second case. Both of the latter functionals use the VV10 non-local correlation functional in order to improve the description of non-covalent interactions and isomerization energies. ωB97M-V is a 12-parameter meta-GGA with 15% short-range exact exchange and 100% long-range exact exchange and is one of the most accurate functionals available through rung 4 of Jacob's Ladder, across a wide variety of molecular applications. This has been verified by benchmarking the functional on nearly 5000 data points against over 200 alternative functionals available in QC-PBC.[102] The use of these functionals for periodic applications via the replacement of the VV10 NLC with rVV10 is largely unbenchmarked, but likely suffers degraded performance due to missing 3-body terms in the rVV10 correction.

5.6.3 User-Defined RSH Functionals

As pointed out in Ref. 46 and elsewhere, the description of charge-transfer excited states within density functional theory (or more precisely, time-dependent DFT) requires full (100%) non-local HF exchange, at least in the limit of large donor–acceptor distance. Hybrid functionals such as B3LYP[19,152] and PBE0[8] that are well-established and in widespread use, however, employ only 20% and 25% HF exchange, respectively. While these functionals provide excellent results for many ground-state properties, they cannot correctly describe the distance dependence of charge-transfer excitation energies, which are enormously underestimated by most common density functionals. This is a serious problem in any case, but it is a *catastrophic* problem in large molecules and in non-covalent clusters, where

TDDFT often predicts a near-continuum of spurious, low-lying charge transfer states.[84,85] The problems with TDDFT's description of charge transfer are not limited to large donor–acceptor distances, but have been observed at ∼2 Å separation, in systems as small as uracil–$(H_2O)_4$.[84] Rydberg excitation energies also tend to be substantially underestimated by standard TDDFT.

One possible avenue by which to correct such problems is to parameterize functionals that contain 100% HF exchange, though few such functionals exist to date. An alternative option is to attempt to preserve the form of common GGAs and hybrid functionals at short range (*i.e.*, keep the 25% HF exchange in PBE0) while incorporating 100% HF exchange at long range, which provides a rigorously correct description of the long-range distance dependence of charge-transfer excitation energies, but aims to avoid contaminating short-range exchange-correlation effects with additional HF exchange. The separation is accomplished using the range-separation *ansatz* that was introduced in Section 5.3. In particular, functionals that use 100% HF exchange at long range ($c_{x,\mathrm{LR}} = 1$ in Eq. (5.13)) are known as "long-range-corrected" (LRC) functionals. An LRC version of PBE0 would, for example, have $c_{x,\mathrm{SR}} = 0.25$.

To fully specify an LRC functional, one must choose a value for the range separation parameter ω in Eq. (5.12). In the limit $\omega \to 0$, the LRC functional in Eq. (5.13) reduces to a non-RSH functional where there is no "SR" or "LR", because all exchange and correlation energies are evaluated using the full Coulomb operator, r_{12}^{-1}. Meanwhile the $\omega \to \infty$ limit corresponds to a new functional, $E_{xc}^{\mathrm{RSH}} = E_c + E_x^{\mathrm{HF}}$. Full HF exchange is inappropriate for use with most contemporary GGA correlation functionals, so the latter limit is expected to perform quite poorly. Values of $\omega > 1.0$ bohr^{-1} are likely not worth considering, according to benchmark tests.[86,143]

Evaluation of the short- and long-range HF exchange energies is straightforward,[10] so the crux of any RSH functional is the form of the short-range GGA exchange functional, and several such functionals are available in QC-PBC. These include short-range variants of the B88 and PBE exchange described by Hirao and co-workers,[70,150] called μB88 and μPBE in QC-PBC,[142] and an alternative formulation of short-range PBE exchange proposed by Scuseria and co-workers,[65] which is known as ωPBE. These functionals are available in QC-PBC thanks to the efforts of the Herbert group.[143,144] By way of notation, the terms "μPBE", "ωPBE", *etc.*, refer only to the short-range exchange functional, $E_{x,\mathrm{SR}}^{\mathrm{DFT}}$ in Eq. (5.13). These functionals could be used in "screened exchange" mode, as described in Section 5.3, as for example in the HSE03 functional,[66] therefore the designation "LRC-ωPBE", for example, should only be used when the short-range exchange functional ωPBE is combined with 100% Hartree-Fock exchange in the long range.

In general, LRC-DFT functionals have been shown to remove the near-continuum of spurious charge-transfer excited states that appear in large-scale TDDFT calculations.[86] However, certain results depend sensitively upon the value of the range-separation parameter ω,[85,86,143,144,162] especially in TDDFT calculations (Section **??**) and therefore the results of LRC-DFT calculations must therefore be interpreted with caution, and probably for a range of ω values. This can be accomplished by requesting a functional that contains some short-range GGA exchange functional (ωPBE or μPBE, in the examples mentioned above), in combination with setting the *$rem* variable LRC_DFT = TRUE, which requests the addition of 100% Hartree-Fock exchange in the long-range. Basic job-control variables and an example can be found below. The value of the range-separation parameter is then controlled by the variable OMEGA, as shown in the examples below.

LRC_DFT

Controls the application of long-range-corrected DFT

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE (or 0) Do not apply long-range correction.

TRUE (or 1) Add 100% long-range Hartree-Fock exchange to the requested functional.

RECOMMENDATION:

The *$rem* variable OMEGA must also be specified, in order to set the range-separation parameter.

OMEGA

Sets the range-separation parameter, ω, also known as μ, in functionals based on Hirao's RSH scheme.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

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

RECOMMENDATION:

None

HFK_SR_COEF

Sets the coefficient for short-range HF exchange

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

n Corresponding to $n/100000000$

RECOMMENDATION:

None

HFK_LR_COEF

Sets the coefficient for long-range HF exchange

TYPE:

INTEGER

DEFAULT:

100000000

OPTIONS:

n Corresponding to $n/100000000$

RECOMMENDATION:

None

Rohrdanz *et al.* [144] published a thorough benchmark study of both ground- and excited-state properties using the LRC-ωPBEh functional, in which the "h" indicates a short-range hybrid (*i.e.*, the presence of some short-range HF exchange).

Empirically-optimized parameters of $c_{x,\mathrm{SR}} = 0.2$ (see Eq. (5.13)) and $\omega = 0.2$ bohr^{-1} were obtained,[144] and these parameters are taken as the defaults for LRC-ωPBEh. Caution is warranted, however, especially in TDDFT calculations for large systems, as excitation energies for states that exhibit charge-transfer character can be rather sensitive to the precise value of ω.[85,144] In such cases (and maybe in general), the "tuning" procedure described in Section **??** is recommended.

By adding 100% Hartree-Fock exchange to the asymptotic Coulomb operator, LRC functionals guarantee that an electron and hole experience an asymptotic interaction potential $1/r$. This is correct for a molecule in the gas phase, but to simulate a material one might desire an asymptotic behavior of $1/(\varepsilon r)$, where ε is the (static) dielectric constant of the material. In conjunction with "optimal tuning" of the range-separation parameter, as described in Section **??**, such functionals have been shown to afford accurate fundamental gaps for organic photovoltaic materials,[81] and are naturally combined with polarizable continuum models (Section **??**) that employ the same dielectric constant.[26] These have come to be called *screened* RSH (sRSH) functionals.[81] An XC function of this type can be expressed generically as[11]

$$E_{xc}^{\mathrm{sRSH}} = c_{x,\mathrm{SR}} E_{x,\mathrm{SR}}^{\mathrm{HF}} + \varepsilon^{-1} E_{x,\mathrm{LR}}^{\mathrm{HF}} + (\varepsilon^{-1} - c_{x,\mathrm{SR}}) E_{x,\mathrm{SR}}^{\mathrm{DFT}} + (1 - \varepsilon^{-1}) E_{x,\mathrm{LR}}^{\mathrm{DFT}} + E_c^{\mathrm{DFT}} \,, \tag{5.17}$$

which should be compared to Eq. (5.13) that provides the generic form for an RSH functional. Although the RSH formalism allows for an arbitrary coefficient $c_{x,\mathrm{LR}}$ for the long-range Hartree-Fock exchange term, as in Eq. (5.13), this implies that the asymptotic electron–hole interaction has the form $c_{x,\mathrm{LR}}/r$ rather than $1/r$.[46] As such, LRC functionals are a particular class of RSH functionals where $c_{x,\mathrm{LR}} = 1$, ensuring proper asymptotic behavior in vacuum. Along the same lines, sRSH functionals set $c_{x,\mathrm{LR}} = \varepsilon^{-1}$ to ensure proper asymptotic behavior in a dielectric material. Using Eq. (5.17), users may construct sRSH functionals by means of a *$xc_functional* input section.

5.7 DFT Methods for van der Waals Interactions

This section describes two different procedures for obtaining a better description of dispersion (van der Waals) interactions in DFT calculations: non-local correlation functionals (Section 5.7.1) and empirical atom–atom dispersion potentials ("DFT-D", Section 5.7.2). For a pedagogical introduction to these methods, discussing the commonalities between them, see Ref. 54.

5.7.1 Non-Local Correlation (NLC) Functionals

From the standpoint of the electron density, the vdW interaction is a non-local one: even for two non-overlapping, spherically-symmetric charge densities (two argon atoms, say), the presence of molecule B in the non-covalent A$\cdots$B complex induces ripples in the tail of A's charge distribution, which are the hallmarks of non-covalent interactions.[40] (This is the fundamental idea behind the non-covalent interaction plots described in Section **??**; the vdW interaction manifests as large density gradients in regions of space where the density itself is small.) Semi-local GGAs that depend only on the density and its gradient cannot describe this long-range, correlation-induced interaction, and meta-GGAs at best describe it at middle-range via the Laplacian of the density and/or the kinetic energy density. A proper description of *long-range* electron correlation requires a non-local functional, *i.e.*, an exchange-correlation potential having the form

$$v_{\mathrm{c}}^{\mathrm{nl}}(\mathbf{r}) = \int f(\mathbf{r}, \mathbf{r}')\, d\mathbf{r}' \,. \tag{5.18}$$

In this way, a perturbation at a point $\mathbf{r}'$ (due to B, say) then induces an exchange-correlation potential at a (possibly far-removed) point $\mathbf{r}$ (on A).

While several generations of NLC functionals have been developed, most are developed for molecular systems and do not translate in a straightforward manner to periodic systems. QC-PBC therefore currently includes just one such functional that can describe dispersion interactions in periodic applications:

- rVV10 by Sabatini and coworkers.[146]

This functional is implemented in a self-consistent manner, and analytic gradients with respect to nuclear displacements are available.[169–171] The non-local correlation is governed by the *$rem* variable NL_CORRELATION, which can be set to the value: rvv10.

rVV10 is used in combination with PBE correlation, which must be added explicitly. In addition, the values of two parameters, C and b (see Ref. 171), must be specified for rVV10. These parameters are controlled by the *$rem* variables NL_VV_C and NL_VV_B, respectively. For instance, to invoke VV10 with $C = 0.0093$ and $b = 5.9$, the following input is used:

```
$rem
   CORRELATION         PBE
   NL_CORRELATION      rVV10
   NL_VV_C             93
   NL_VV_B             590
   ...
$end
```

In rVV10 neither C nor b are assigned a default value and must always be provided in the input.

Unlike local (LSDA) and semi-local (GGA and meta-GGA) functionals, for non-local functionals evaluation of the correlation energy requires a double integral over the spatial variables, as compared to the single integral [Eq. (5.8)] required for semi-local functionals:

$$E_{\mathrm{c}}^{\mathrm{nl}} = \int v_{\mathrm{c}}^{\mathrm{nl}}(\mathbf{r})\, d\mathbf{r} = \int f(\mathbf{r},\mathbf{r}')\, \rho(\mathbf{r})\, d\mathbf{r}\, d\mathbf{r}'. \tag{5.19}$$

In practice, this double integration is performed numerically on a quadrature grid.[169–171] By default, the uniform grid used by GPW is used.

NL_VV_C

Sets the parameter C in rVV10. This parameter is fit to asymptotic van der Waals C_6 coefficients.

TYPE:

INTEGER

DEFAULT:

No default for VV10

OPTIONS:

n Corresponding to $C = n/10000$

RECOMMENDATION:

$C = 0.0093$ is recommended when a semi-local exchange functional is used. $C = 0.0089$ is recommended when a long-range corrected (LRC) hybrid functional is used. For further details see Ref. 170.

NL_VV_B

Sets the parameter b in rVV10. This parameter controls the short range behavior of the non-local correlation energy.

TYPE:

INTEGER

DEFAULT:

No default

OPTIONS:

n Corresponding to $b = n/100$

RECOMMENDATION:

The optimal value depends strongly on the exchange functional used. $b = 5.9$ is recommended for rPW86. For further details see Ref. 170.

USE_RVV10

Used to turn on the rVV10 NLC functional

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE Not supported (VV10 inefficient for periodic systems)

TRUE Use rVV10 NLC

RECOMMENDATION:

Set to TRUE if the rVV10 NLC is desired.

Example 5.2 Input for an energy calculation on graphite using the PBE functional with the rVV10 NLC functional included.

```
$comment
Run PBE with rVV10 on graphite
$end

$lattice
3 ! dimension
2.46729100000000 0.00000000000000 0.00000000000000
-1.23364600000000 2.13673700000000 0.00000000000000
0.00000000000000 0.00000000000000 7.80307300000000
$end

$unitcell
absolute
0 1
C 0.0000000000 0.0000000000 1.9507682500
C 0.0000000000 0.0000000000 5.8523047500
C -0.0000015670 1.4244920456 1.9507682500
C 1.2336465670 0.7122449544 5.8523047500
$end

$rem
    JOBTYPE     sp          ! Energy calculation
    METHOD      pbe         ! Hartree-Fock
    BASIS       szv-gth     ! double zeta basis
    KECUT       1000        ! planewave cutoff 1000 eV
    PSEUDO      gth-pbe     ! use GTH pseudopotential fit with PBE
    MP_MESH     [4,4,4]     ! 4x4x4 k-point Monkhorst-Pack mesh
    USE_RVV10   true        ! use rvv10
    NL_VV_C     93          ! rvv10 C = 0.0093
    NL_VV_B     630         ! rvv10 B = 6.3
$end
```

View output online

5.7.2 Empirical Dispersion Corrections: DFT-D

A major development in DFT during the mid-2000s was the recognition that, first of all, semi-local density functionals do not properly capture dispersion (van der Waals) interactions, a problem that has been addressed only much more recently by the non-local correlation functionals discussed in Section 5.7.1; and second, that a cheap and simple solution to this problem is to incorporate empirical potentials of the form $-C_6/R^6$, where the C_6 coefficients are pairwise atomic parameters. This approach, which is an alternative to the use of a non-local correlation functional, is known as *dispersion-corrected DFT* (DFT-D).[57,61]

There are currently several unique DFT-D methods in QC-PBC. These are requested via the *$rem* variable DFT_D and are discussed below.

DFT_D

Controls the empirical dispersion correction to be added to a DFT calculation.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Do not apply any DFT-D scheme

4 DFT-D3(0) dispersion correction from Grimme *et al.* [58]

5 DFT-D3M(0) dispersion correction from Smith *et al.* [149]

6 DFT-D3(BJ) dispersion correction from Grimme *et al.* [59]

7 DFT-D3M(BJ) dispersion correction from Smith *et al.* [149]

8 DFT-D3(CSO) dispersion correction from Schröder *et al.* [147]

9 DFT-D3(op) dispersion correction from Witte *et al.* [182]

RECOMMENDATION:

Use D3(BJ) as a starting point, use other D3 varieties if necessary.

Grimme's DFT-D3 method[58] constitutes an improvement on his D2 approach, and is also available along with analytic first and second derivatives, for any density functional that is available in QC-PBC. The D3 correction includes a potential akin to that in D2 but including atomic C_8 terms as well:

$$E_{\text{D3,2-body}} = -\sum_{A}^{\text{atoms}} \sum_{B<A}^{\text{atoms}} \left[s_6 \left(\frac{C_{6,AB}}{R_{AB}^6} \right) f_{\text{damp},6}(R_{AB}) + s_8 \left(\frac{C_{8,AB}}{R_{AB}^8} \right) f_{\text{damp},8}(R_{AB}) \right] . \tag{5.20}$$

The total D3 dispersion correction consists of this plus a three-body term of the Axilrod-Teller-Muto (ATM) triple-dipole variety, so that the total D3 energy is $E_{\text{DFT-D3}} = E_{\text{KS-DFT}} + E_{\text{D3,2-body}} + E_{\text{ATM,3-body}}$

Several versions of DFT-D3 are available in QC-PBC, which differ in the choice of the two damping functions. Grimme's formulation,[58] which is now known as the "zero-damping" version [DFT-D3(0)], uses damping functions of the form

$$f_{\text{damp},n}^{\text{D3(0)}}(R_{AB}) = \left[1 + 6 \left(\frac{R_{AB}}{s_{r,n} R_{0,AB}} \right)^{-\beta_n} \right]^{-1} \tag{5.21}$$

for $n = 6$ or 8, $\beta_6 = 14$, and $\beta_8 = 16$.[58,94] The parameters $R_{0,AB}$ come from atomic van der Waals radii, $s_{r,6}$ is a functional-dependent parameter, and $s_{r,8} = 1$. Typically s_6 is set to unity and s_8 is optimized for the functional in question.

The more recent Becke–Johnson-damping version of DFT-D3,[59] DFT-D3(BJ), is designed to be finite (but non-zero) as $R_{AB} \to 0$. The damping functions used in DFT-D3(BJ) are

$$f_{\text{damp},n}^{\text{D3(BJ)}}(R_{AB}) = \frac{R_{AB}^n}{R_{AB}^n + (\alpha_1 R_{0,AB} + \alpha_2)^n} \tag{5.22}$$

where α_1 and α_2 are adjustable parameters fit for each density functional. As in DFT-D3(0), s_6 is generally fixed to unity and s_8 is optimized for each functional. DFT-D3(BJ) generally outperforms the original DFT-D3(0) version.[59]

The C_6-only (CSO) approach of Schröder *et al.* [147] discards the C_8 term in Eq. (5.20) and uses a damping function with one parameter, α_1:

$$f_{\text{damp},6}^{\text{D3(CSO)}}(R_{AB}) = \frac{C_{AB}^6}{R_{AB}^6 + (2.5\text{Å})^6} \left(s_6 + \frac{\alpha_1}{1 + \exp[R_{AB} - (2.5\text{Å}) R_{0,AB}]} \right) . \tag{5.23}$$

The DFT-D3(BJ) approach was re-parameterized by Smith *et al.* [149] to yield the "modified" DFT-D3(BJ) approach, DFT-D3M(BJ), whose parameterization relied heavily on non-equilibrium geometries. The same authors also introduces a modification DFT-D3M(0) of the original zero-damping correction, which introduces one additional parameter

(α_1) as compared to DFT-D3(0):

$$f_{\text{damp},n}^{\text{D3M(0)}}(R_{AB}) = \left[1 + 6\left(\frac{R_{AB}}{s_{r,n}R_{0,AB}} + \alpha_1 R_{0,AB}\right)^{-\beta_n}\right]^{-1} . \tag{5.24}$$

Finally, optimized power approach of Witte *et al.* [182] treats the exponent, β_6, as an optimizable parameter, given by

$$f_{\text{damp},n}^{\text{D3(op)}}(R_{AB}) = \frac{R_{AB}^{\beta_n}}{R_{AB}^{\beta_n} + (\alpha_1 R_{0,AB} + \alpha_2)^{\beta_n}} . \tag{5.25}$$

Note that $\beta_8 = \beta_6 + 2$.

To summarize this bewildering array of D3 damping functions:

- **DFT-D3(0)** is requested by setting DFT_D = D3_ZERO. The model depends on four scaling parameters (s_6, $s_{r,6}$, s_8, and $s_{r,8}$), as defined in Eq. (5.21).
- **DFT-D3(BJ)** is requested by setting DFT_D = D3_BJ. The model depends on four scaling parameters (s_6, s_8, α_1, and α_2), as defined in Eq. (5.22).
- **DFT-D3(CSO)** is requested by setting DFT_D = D3_CSO. The model depends on two scaling parameters (s_6 and α_1), as defined in Eq. (5.23).
- **DFT-D3M(0)** is requested by setting DFT_D = D3_ZEROM. The model depends on five scaling parameters (s_6, s_8, $s_{r,6}$, $s_{r,8}$, and α_1), as defined in Eq. (5.24).
- **DFT-D3M(BJ)** is requested by setting DFT_D = D3_BJM. The model depends on four scaling parameters (s_6, s_8, α_1, and α_2), as defined in Eq. (5.22).
- **DFT-D3(op)** is requested by setting DFT_D = D3_OP. The model depends on four scaling parameters (s_6, s_8, α_1, α_2, and β_6), as defined in Eq. (5.22).

The scaling parameters in these damping functions can be modified using the *$rem* variables described below.

Note:

1. DFT-D3(0) is defined for elements up to Pu ($Z = 94$).
2. The B97-D3(0), ωB97X-D3, ωM06-D3 functionals automatically employ the DFT-D3(0) dispersion correction when invoked by setting METHOD equal to B97-D3, wB97X-D3, or wM06-D3.
3. Currently, QC-PBC does not automatically populate the DFT-D parameters for functionals. It is therefore **highly** reccomended that the user specify the parameters manually for every DFT-D calculation.

DFT_D3_S6

The linear parameter s_6 in eq. (5.20). Used in all forms of DFT-D3.

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_6 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS6

The nonlinear parameter $s_{r,6}$ in Eqs. (5.21) and Eq. (5.24). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,6} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_S8

The linear parameter s_8 in Eq. (5.20). Used in DFT-D3(0), DFT-D3(BJ), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_8 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_RS8

The nonlinear parameter $s_{r,8}$ in Eqs. (5.21) and Eq. (5.24). Used in DFT-D3(0) and DFT-D3M(0).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $s_{r,8} = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A1

The nonlinear parameter α_1 in Eqs. (5.22), (5.23), (5.24), and (5.25). Used in DFT-D3(BJ), DFT-D3(CSO), DFT-D3M(0), DFT-D3M(BJ), and DFT-D3(op).

TYPE:

INTEGER

DEFAULT:

100000

OPTIONS:

n Corresponding to $\alpha_1 = n/100000$.

RECOMMENDATION:

NONE

DFT_D3_A2
The nonlinear parameter α_2 in Eqs. (5.22) and (5.25). Used in DFT-D3(BJ), DFT-D3M(BJ), and DFT-D3(op).
TYPE:
INTEGER
DEFAULT:
100000
OPTIONS:
n Corresponding to $\alpha_2 = n/100000$.
RECOMMENDATION:
NONE

DFT_D3_POWER
The nonlinear parameter β_6 in Eq. (5.25). Used in DFT-D3(op). Must be greater than or equal to 6 to avoid divergence.
TYPE:
INTEGER
DEFAULT:
600000
OPTIONS:
n Corresponding to $\beta_6 = n/100000$.
RECOMMENDATION:
NONE

The three-body interaction term, $E^{(3)}$,[58] must be explicitly turned on, if desired.

DFT_D3_3BODY
Controls whether the three-body interaction in Grimme's DFT-D3 method should be applied (see Eq. (14) in Ref. 58).
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 5.3 Input for a r^2-SCAN-D3(BJ) energy calculation on Argon. Note the DFT-D3 parameters are not included for this functional so they must be specified in the input.

```
$comment
Run r2-SCAN with DFT-D3(BJ) dispersion correction.
$end


$lattice
3 ! dimension
3.28425170000000 -0.00000000000000 1.89616357000000
1.09475156000000 3.09642182000000 1.89616357000000
-0.00000000000000 -0.00000000000000 3.79232614000000
$end

$unitcell
absolute
0 1
Ar 0.0000000000 0.0000000000 0.0000000000
$end


$rem
    JOBTYPE     sp          ! Energy calculation
    METHOD      r2scan      ! r2-scan mgga functional
    BASIS       szv-gth     ! double zeta basis
    KECUT       1000        ! planewave cutoff 1000 eV
    PSEUDO      gth-pbe     ! use GTH pseudopotential fit with PBE
    MP_MESH     [4,4,4]     ! 4x4x4 k-point Monkhorst-Pack mesh
    DFT_D       6           ! DFT-D3(BJ)
    DFT_D3_S6   100000
    DFT_D3_S8   78981.345
    DFT_D3_A1   49484.001
    DFT_D3_A2   573083.694
$end
```

View output online

References and Further Reading

[1] AOINTS (Appendix **??**).

[2] Molecular Properties Analysis (Chapter **??**).

[3] Basis Sets (Chapter 7) and Effective Core Potentials (Chapter **??**).

[4] Molecular Geometry and Critical Points (Chapter 8).

[5] C. Adamo and V. Barone. *Chem. Phys. Lett.*, 274:242, 1997. DOI: 10.1016/S0009-2614(97)00651-9.

[6] C. Adamo and V. Barone. *J. Chem. Phys.*, 108:664, 1998. DOI: 10.1063/1.475428.

[7] C. Adamo and V. Barone. *J. Chem. Phys.*, 110:6158, 1999. DOI: 10.1063/1.478522.

[8] C. Adamo, G. E. Scuseria, and V. Barone. *J. Chem. Phys.*, 111:2889, 1999. DOI: 10.1063/1.479571.

[9] R. D. Adamson, P. M. W. Gill, and J. A. Pople. *Chem. Phys. Lett.*, 284:6, 1998. DOI: 10.1016/S0009-2614(97)01282-7.

[10] R. D. Adamson, J. P. Dombroski, and P. M. W. Gill. *J. Comput. Chem.*, 20:921, 1999. DOI: 10.1002/(SICI)1096-987X(19990715)20:9<921::AID-JCC3>3.0.CO;2-K.

[11] B. Alam, A. F. Morrison, and J. M. Herbert. *J. Phys. Chem. C*, 124:24653, 2020. DOI: 10.1021/acs.jpcc.0c07932.

[12] R. Armiento and S. Kümmel. *Phys. Rev. Lett.*, 111:036402, 2013. DOI: 10.1103/PhysRevLett.111.036402.

[13] T. Aschebrock and S. Kümmel. *Phys. Rev. Research*, 1:033082, 2019. DOI: 10.1103/PhysRevResearch.1.033082.

[14] A. P. Bartok and J. R. Yates. *J. Chem. Phys.*, 150:161101, 2019. DOI: 10.1063/1.5094646.

[15] A. D. Becke. *J. Chem. Phys.*, 84:4524, 1986. DOI: 10.1063/1.450025.

[16] A. D. Becke. *J. Chem. Phys.*, 85:7184, 1986. DOI: 10.1063/1.451353.

[17] A. D. Becke. *J. Chem. Phys.*, 88:2547, 1988. DOI: 10.1063/1.454033.

[18] A. D. Becke. *Phys. Rev. A*, 38:3098, 1988. DOI: 10.1103/PhysRevA.38.3098.

[19] A. D. Becke. *J. Chem. Phys.*, 98:5648, 1993. DOI: 10.1063/1.464913.

[20] A. D. Becke. *Int. J. Quantum Chem. Symp.*, 28:625, 1994. DOI: 10.1002/qua.560520855.

[21] A. D. Becke. *J. Chem. Phys.*, 104:1040, 1996. DOI: 10.1063/1.470829.

[22] A. D. Becke. *J. Chem. Phys.*, 107:8554, 1997. DOI: 10.1063/1.475007.

[23] A. D. Becke and M. R. Roussel. *Phys. Rev. A*, 39:3761, 1989. DOI: 10.1103/PhysRevA.39.3761.

[24] Y. A. Bernard, Y. Shao, and A. I. Krylov. *J. Chem. Phys.*, 136:204103, 2012. DOI: 10.1063/1.4714499.

[25] N. A. Besley, M. J. G. Peach, and D. J. Tozer. *Phys. Chem. Chem. Phys.*, 11:10350, 2009. DOI: 10.1039/b912718f.

[26] S. Bhandari, M. S. Cheung, E. Geva, L. Kronik, and B. D. Dunietz. *J. Chem. Theory Comput.*, 14:6287, 2019. DOI: 10.1021/acs.jctc.8b00876.

[27] A. D. Boese and N. C. Handy. *J. Chem. Phys.*, 114:5497, 2001. DOI: 10.1063/1.1347371.

[28] A. D. Boese and N. C. Handy. *J. Chem. Phys.*, 116:9559, 2002. DOI: 10.1063/1.1476309.

[29] A. D. Boese and J. M. L. Martin. *J. Chem. Phys.*, 121:3405, 2004. DOI: 10.1063/1.1774975.

[30] A. D. Boese, N. L. Doltsinis, N. C. Handy, and M. Sprik. *J. Chem. Phys.*, 112:1670, 2000. DOI: 10.1063/1.480732.

[31] F. Castet and B. Champagne. *J. Chem. Theory Comput.*, 8:2044, 2012. DOI: 10.1021/ct300174z.

[32] J.-D. Chai and M. Head-Gordon. *J. Chem. Phys.*, 128:084106, 2008. DOI: 10.1063/1.2834918.

[33] J.-D. Chai and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 10:6615, 2008. DOI: 10.1039/b810189b.

[34] S.-H. Chien and P. M. W. Gill. *J. Comput. Chem.*, 24:732, 2003. DOI: 10.1002/jcc.10211.

[35] S.-H. Chien and P. M. W. Gill. *J. Comput. Chem.*, 27:730, 2006. DOI: 10.1002/jcc.20383.

[36] A. J. Cohen, P. Mori-Sánchez, and W. Yang. *J. Chem. Phys.*, 127:034101, 2007. DOI: 10.1063/1.2749510.

[37] A. J. Cohen, P. Mori-Sánchez, and W. Yang. *J. Chem. Phys.*, 126:191109, 2007. DOI: 10.1063/1.2741248.

[38] L. A. Constantin, E. Fabiano, and F. Della Sala. *Phys. Rev. B*, 86, 2012. DOI: 10.1103/PhysRevB.86.035130.

[39] L. A. Constantin, E. Fabiano, and F. Della Sala. *J. Chem. Theory Comput.*, 9:2256, 2013. DOI: 10.1021/ct400148r.

[40] J. Contreras-García, E. R. Johnson, S. Keinan, B. Chaudret, J.-P. Piquemal, D. N. Beratan, and W. Yang. *J. Chem. Theory Comput.*, 7:625, 2011. DOI: 10.1021/ct100641a.

[41] G. I. Csonka, J. P. Perdew, and A. Ruzsinszky. *J. Chem. Theory Comput.*, 6:3688, 2010. DOI: 10.1021/ct100488v.

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

[43] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist. *Phys. Rev. Lett.*, 92:246401, 2004. DOI: 10.1103/PhysRevLett.92.246401.

[44] M. Dion, H. Rydberg, E. Schröder, D. C. Langreth, and B. I. Lundqvist. *Phys. Rev. Lett.*, 95:109902, 2005. DOI: 10.1103/PhysRevLett.95.109902.

[45] P. A. M. Dirac. *P. Camb. Philos. Soc.*, 26:376, 1930. DOI: 10.1017/S0305004100016108.

[46] A. Dreuw, J. L. Weisman, and M. Head-Gordon. *J. Chem. Phys.*, 119:2943, 2003. DOI: 10.1063/1.1590951.

[47] J. W. Furness, A. D. Kaplan, J. Ning, J. P. Perdew, and J. Sun. *J. Chem. Phys.*, 156:034109, 2022. DOI: 10.1063/5.0073623.

[48] P. M. W. Gill. *Mol. Phys.*, 89:433, 1996. DOI: 10.1080/002689796173813.

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

[50] P. M. W. Gill, R. D. Adamson, and J. A. Pople. *Mol. Phys.*, 88:1005, 1996. DOI: 10.1080/00268979609484488.

[51] L. Goerigk and S. Grimme. *J. Chem. Theory Comput.*, 6:107, 2010. DOI: 10.1021/ct900489g.

[52] E. Goll, H.-J. Werner, and H. Stoll. *Phys. Chem. Chem. Phys.*, 7:3917, 2005. DOI: 10.1039/B509242F.

[53] E. Goll, H.-J. Werner, H. Stoll, T. Leininger, P. Gori-Giorgi, and A. Savin. *Chem. Phys.*, 329:276, 2006. DOI: 10.1016/j.chemphys.2006.05.020.

[54] Montgomery Gray and John M Herbert. 4.1 comparison to benchmarks. *Annual Reports on Computational Chemistry*, 20:31, 2024.

[55] S. Grimme. *J. Phys. Chem. A*, 109:3067, 2005. DOI: 10.1021/jp050036j.

[56] S. Grimme. *J. Comput. Chem.*, 27:1787, 2006. DOI: 10.1002/jcc.20495.

[57] S. Grimme. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.*, 1:211, 2011. DOI: 10.1002/wcms.30.

[58] S. Grimme, J. Antony, S. Ehrlich, and H. Krieg. *J. Chem. Phys.*, 132:154104, 2010. DOI: 10.1063/1.3382344.

[59] S. Grimme, S. Ehrlich, and L. Goerigk. *J. Comput. Chem.*, 32:1456, 2011. DOI: 10.1002/jcc.21759.

[60] S. Grimme, J. G. Brandenburg, C. Bannwarth, and A. Hansen. *J. Chem. Phys.*, 143:054107, 2015. DOI: 10.1063/1.4927476.

[61] S. Grimme, A. Hansen, J. G. Brandenburg, and C. Bannwarth. *Chem. Rev.*, 116:5105, 2016. DOI: 10.1021/acs.chemrev.5b00533.

[62] B. Hammer, L. B. Hansen, and J. K. Nørskov. *Phys. Rev. B*, 59:7413, 1999. DOI: 10.1103/PhysRevB.59.7413.

[63] F. A. Hamprecht, A. J. Cohen, D. J. Tozer, and N. C. Handy. *J. Chem. Phys.*, 109:6264, 1998. DOI: 10.1063/1.477267.

[64] N. C. Handy and A. J. Cohen. *Mol. Phys.*, 99:403, 2001. DOI: 10.1080/00268970010018431.

[65] T. M. Henderson, B. G. Janesko, and G. E. Scuseria. *J. Chem. Phys.*, 128:194105, 2008. DOI: 10.1063/1.2921797.

[66] J. Heyd, G. E. Scuseria, and M. Ernzerhof. *J. Chem. Phys.*, 118:8207, 2003. DOI: 10.1063/1.1564060.

[67] W.-M. Hoe, A. J. Cohen, and N. C. Handy. *Chem. Phys. Lett.*, 341:319, 2001. DOI: 10.1016/S0009-2614(01)00581-4.

[68] P. Hohenberg and W. Kohn. *Phys. Rev. B*, 136:864, 1964. DOI: 10.1103/PhysRev.136.B864.

[69] K. Hui and J.-D. Chai. *J. Chem. Phys.*, 144:044114, 2016. DOI: 10.1063/1.4940734.

[70] H. Iikura, T. Tsuneda, T. Yanai, and K. Hirao. *J. Chem. Phys.*, 115:3540, 2001. DOI: 10.1063/1.1383587.

[71] S. Jana, K. Sharma, and P. Samal. *J. Phys. Chem. A*, 123:6356, 2019. DOI: 10.1021/acs.jpca.9b02921.

[72] S. Jana, S. K. Behera, S. Smiga, L. A. Constantin, and P. Samal. *J. Chem. Phys.*, 155:024103, 2021. DOI: 10.1063/5.0051331.

[73] Y. Jin and R. J. Bartlett. *J. Chem. Phys.*, 145:034107, 2016. DOI: 10.1063/1.4955497.

[74] B. G. Johnson, P. M. W. Gill, and J. A. Pople. *Chem. Phys. Lett.*, 220:377, 1994. DOI: 10.1016/0009-2614(94)00199-5.

[75] T. W. Keal and D. J. Tozer. *J. Chem. Phys.*, 119:3015, 2003. DOI: 10.1063/1.1590634.

[76] T. W. Keal and D. J. Tozer. *J. Chem. Phys.*, 121:5654, 2004. DOI: 10.1063/1.1784777.

[77] T. W. Keal and D. J. Tozer. *J. Chem. Phys.*, 123:121103, 2005. DOI: 10.1063/1.2061227.

[78] J. Klimeš, D. R. Bowler, and A. Michaelides. *J. Phys. Condens. Matter*, 22:022201, 2010. DOI: 10.1088/0953-8984/22/2/022201.

[79] W. Kohn and L. J. Sham. *Phys. Rev. A*, 140:1133, 1965. DOI: 10.1103/PhysRev.140.A1133.

[80] W. Kohn, A. D. Becke, and R. G. Parr. *J. Phys. Chem.*, 100:12974, 1996. DOI: 10.1021/jp960669l.

[81] L. Kronik and S. Kümmel. *Adv. Mater.*, 30:1706560, 2018. DOI: 10.1002/adma.201706560.

[82] A. V. Krukau, O. A. Vydrov, A. F. Izmaylov, and G. E. Scuseria. *J. Chem. Phys.*, 125:224106, 2006. DOI: 10.1063/1.2404663.

[83] B. B. Laird, R. B. Ross, and T. Ziegler, editors. *Density-Functional Methods in Chemistry: An Overview*, volume 629 of *ACS Symposium Series*. American Chemical Society, Washington, D.C., 1996. DOI: 10.1021/bk-1996-0629.ch001.

[84] A. Lange and J. M. Herbert. *J. Chem. Theory Comput.*, 3:1680, 2007. DOI: 10.1021/ct700125v.

[85] A. W. Lange and J. M. Herbert. *J. Am. Chem. Soc.*, 131:124115, 2009. DOI: 10.1021/ja808998q.

[86] A. W. Lange, M. A. Rohrdanz, and J. M. Herbert. *J. Phys. Chem. B*, 112:6304, 2008. DOI: 10.1021/jp802058k.

[87] V. I. Lebedev. *Zh. Vychisl. Mat. Mat. Fix.*, 15:48, 1975. DOI: 10.1016/0041-5553(75)90133-0.

[88] V. I. Lebedev. *Zh. Vychisl. Mat. Mat. Fix.*, 16:293, 1976. DOI: 10.1016/0041-5553(76)90100-2.

[89] V. I. Lebedev. *Sibirsk. Mat. Zh.*, 18:132, 1977.

[90] V. I. Lebedev and D. N. Laikov. *Dokl. Math.*, 366:741, 1999.

[91] C. Lee, W. Yang, and R. G. Parr. *Phys. Rev. B*, 37:785, 1988. DOI: 10.1103/PhysRevB.37.785.

[92] C. Y. Lin, M. W. George, and P. M. W. Gill. *Aust. J. Chem.*, 57:365, 2004. DOI: 10.1071/CH03263.

[93] Y.-S. Lin, C.-W. Tsai, G.-D. Li, and J.-D. Chai. *J. Chem. Phys.*, 136:154109, 2012. DOI: 10.1063/1.4704370.

[94] Y.-S. Lin, G.-D. Li, S.-P. Mao, and J.-D. Chai. *J. Chem. Theory Comput.*, 9:263, 2013. DOI: 10.1021/ct300715s.

[95] F. Liu, E. Proynov, J.-G. Yu, T. R. Furlani, and J. Kong. *J. Chem. Phys.*, 137:114104, 2012. DOI: 10.1063/1.4752396.

[96] K.-Y. Liu, J. Liu, and J. M. Herbert. *J. Comput. Chem.*, 38:1678, 2017. DOI: 10.1002/jcc.24811.

[97] S. Liu and R. G. Parr. *J. Mol. Struct. (Theochem)*, 501:29, 2000. DOI: 10.1016/S0166-1280(99)00410-8.

[98] P.-F. Loos. *J. Chem. Phys.*, 146:114108, 2017. DOI: 10.1063/1.4978409.

[99] B. J. Lynch, P. L. Fast, M. Harris, and D. G. Truhlar. *J. Phys. Chem. A*, 104:4811, 2000. DOI: 10.1021/jp000497z.

[100] N. Mardirossian and M. Head-Gordon. *Phys. Chem. Chem. Phys.*, 16:9904, 2014. DOI: 10.1039/c3cp54374a.

[101] N. Mardirossian and M. Head-Gordon. *J. Chem. Phys.*, 142:074111, 2015. DOI: 10.1063/1.4907719.

[102] N. Mardirossian and M. Head-Gordon. *J. Chem. Phys.*, 144:214110, 2016. DOI: 10.1063/1.4952647.

[103] N. Mardirossian, L. R. Pestana, J. C. Womack, C.-K. Skylaris, T. Head-Gordon, and M. Head-Gordon. *J. Phys. Chem. Lett.*, 8:35, 2017. DOI: 10.1021/acs.jpclett.6b02527.

[104] P. D. Mezei, G. I. Csonka, and M. Kállay. *J. Chem. Theory Comput.*, 14:2469, 2018. DOI: 10.1021/acs.jctc.8b00072.

[105] M. Mitani. *Theor. Chem. Acc.*, 130:645, 2011. DOI: 10.1007/s00214-011-0985-x.

[106] M. Mitani and Y. Yoshioka. *Theor. Chem. Acc.*, 131:1169, 2012. DOI: 10.1007/s00214-012-1169-z.

[107] P. Mori-Sánchez, A. J. Cohen, and W. Yang. *J. Chem. Phys.*, 124:091102, 2006. DOI: 10.1063/1.2179072.

[108] C. W. Murray, N. C. Handy, and G. J. Laming. *Mol. Phys.*, 78:997, 1993. DOI: 10.1080/00268979300100651.

[109] É. D. Murray, K. Lee, and D. C. Langreth. *J. Chem. Theory Comput.*, 5:2754, 2009. DOI: 10.1021/ct900365q.

[110] B. Neupane, H. Tang, N. K. Nepal, S. Adhikari, and A. Ruzsinszky. *Phys. Rev. Materials*, 5:063803, 2021. DOI: 10.1103/PhysRevMaterials.5.063803.

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

[112] A. Patra, S. Jana, and P. Samal. *J. Chem. Phys.*, 153:184112, 2020. DOI: 10.1063/5.0025173.

[113] S. Paziani, S. Moroni, P. Gori-Giorgi, and G. B. Bachelet. *Phys. Rev. B*, 73:155111, 2006. DOI: 10.1103/PhysRevB.73.155111.

[114] J. P. Perdew. *Phys. Rev. B*, 33:8822, 1986. DOI: 10.1103/PhysRevB.33.8822.

[115] J. P. Perdew and K. Schmidt. Jacob's ladder of density functional approximations for the exchange-correlation energy. In V. Van Doren, C. Van Alsenoy, and P. Geerlings, editors, *Density Functional Theory and Its Applications to Materials*, volume 577 of *AIP Conference Proceedings*, pages 1–20. American Institute of Physics, 2001. DOI: 10.1063/1.1390175.

[116] J. P. Perdew and Y. Wang. *Phys. Rev. B*, 33:8800, 1986. DOI: 10.1103/PhysRevB.33.8800.

[117] J. P. Perdew and Y. Wang. *Phys. Rev. B*, 45:13244, 1992. DOI: 10.1103/PhysRevB.45.13244.

[118] J. P. Perdew and A. Zunger. *Phys. Rev. B*, 23:5048, 1981. DOI: 10.1103/PhysRevB.23.5048.

[119] J. P. Perdew, J. A. Chevary, S. H. Vosko, K. A. Jackson, M. R. Pederson, D. J. Singh, and C. Fiolhais. *Phys. Rev. B*, 46:6671, 1992. DOI: 10.1103/PhysRevB.46.6671.

[120] J. P. Perdew, K. Burke, and M. Ernzerhof. *Phys. Rev. Lett.*, 77:3865, 1996. DOI: 10.1103/PhysRevLett.77.3865.

[121] J. P. Perdew, S. Kurth, A. Zupan, and P. Blaha. *Phys. Rev. Lett.*, 82:2544, 1999. DOI: 10.1103/PhysRevLett.82.2544.

[122] J. P. Perdew, A. Ruzsinszky, J. Tao, V. N. Staroverov, G. E. Scuseria, and G. I. Csonka. *J. Chem. Phys.*, 123: 062201, 2005. DOI: 10.1063/1.1904565.

[123] J. P. Perdew, A. Ruzsinszky, J. Tao, G. I. Csonka, and G. E. Scuseria. *Phys. Rev. A*, 76:042506, 2007. DOI: 10.1103/PhysRevA.76.042506.

[124] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, O. A. Vydrov, G. E. Scuseria, L. A. Constantin, X. Zhou, and K. Burke. *Phys. Rev. Lett.*, 100:136406, 2008. DOI: 10.1103/PhysRevLett.100.136406.

[125] J. P. Perdew, V. N. Staroverov, J. Tao, and G. E. Scuseria. *Phys. Rev. A*, 78:052513, 2008. DOI: 10.1103/PhysRevA.78.052513.

[126] J. P. Perdew, A. Ruzsinszky, G. I. Csonka, L. A. Constantin, and J. Sun. *Phys. Rev. Lett.*, 103:026403, 2009. DOI: 10.1103/PhysRevLett.103.026403.

[127] K. Pernal, R. Podeszwa, K. Patkowski, and K. Szalewicz. *Phys. Rev. Lett.*, 103:263201, 2009. DOI: 10.1103/PhysRevLett.103.263201.

[128] R. Peverati and D. G. Truhlar. *J. Chem. Phys.*, 135:191102, 2011. DOI: 10.1063/1.3663871.

[129] R. Peverati and D. G. Truhlar. *J. Phys. Chem. Lett.*, 2:2810, 2011. DOI: 10.1021/jz201170d.

[130] R. Peverati and D. G. Truhlar. *J. Phys. Chem. Lett.*, 3:117, 2012. DOI: 10.1021/jz201525m.

[131] R. Peverati and D. G. Truhlar. *J. Chem. Theory Comput.*, 8:2310, 2012. DOI: 10.1021/ct3002656.

[132] R. Peverati and D. G. Truhlar. *Phys. Chem. Chem. Phys.*, 14:13171, 2012. DOI: 10.1039/c2cp42025b.

[133] R. Peverati and D. G. Truhlar. *Phys. Chem. Chem. Phys.*, 14:16187, 2012. DOI: 10.1039/c2cp42576a.

[134] R. Peverati, Y. Zhao, and D. G. Truhlar. *J. Phys. Chem. Lett.*, 2:1991, 2011. DOI: 10.1021/jz200616w.

[135] J. A. Pople, P. M. W. Gill, and B. G. Johnson. *Chem. Phys. Lett.*, 199:557, 1992. DOI: 10.1016/0009-2614(92)85009-Y.

[136] E. Proynov and J. Kong. *J. Chem. Theory Comput.*, 3:746, 2007. DOI: 10.1021/ct600372t.

[137] E. Proynov and J. Kong. In G. Vaysilov and T. Mineva, editors, *Theoretical Aspects of Catalysis*. Heron Press, Birmingham, UK, 2008.

[138] E. Proynov and J. Kong. *Phys. Rev. A*, 79:014103, 2009. DOI: 10.1103/PhysRevA.79.014103.

[139] E. Proynov, Y. Shao, and J. Kong. *Chem. Phys. Lett.*, 493:381, 2010. DOI: 10.1016/j.cplett.2010.05.029.

[140] E. Proynov, F. Liu, and J. Kong. *Chem. Phys. Lett.*, 525:150, 2012. DOI: 10.1016/j.cplett.2011.12.069.

[141] E. Proynov, F. Liu, Y. Shao, and J. Kong. *J. Chem. Phys.*, 136:034102, 2012. DOI: 10.1063/1.3676726.

[142] R. M. Richard and J. M. Herbert. *J. Chem. Theory Comput.*, 7:1296, 2011. DOI: 10.1021/ct100607w.

[143] M. A. Rohrdanz and J. M. Herbert. *J. Chem. Phys.*, 129:034107, 2008. DOI: 10.1063/1.2954017.

[144] M. A. Rohrdanz, K. M. Martins, and J. M. Herbert. *J. Chem. Phys.*, 130:054112, 2009. DOI: 10.1063/1.3073302.

[145] A. Ruzsinszky, J. Sun, B. Xiao, and G. Csonka. *J. Chem. Theory Comput.*, 8:2078, 2012. DOI: 10.1021/ct300269u.

[146] R. Sabatini, T. Gorni, and S. de Gironcoli. *Phys. Rev. B*, 87:041108, 2013. DOI: 10.1103/PhysRevB.87.041108.

[147] H. Schröder, A. Creon, and T. Schwabe. *J. Chem. Theory Comput.*, 11:3163, 2015. DOI: 10.1021/acs.jctc.5b00400.

[148] Y. Shao, M. Head-Gordon, and A. I. Krylov. *J. Chem. Phys.*, 118:4807, 2003. DOI: 10.1063/1.1545679.

[149] D. G. Smith, L. A. Burns, K. Patkowski, and C. D. Sherrill. *J. Phys. Chem. Lett.*, 7:2197, 2016. DOI: 10.1021/acs.jpclett.6b00780.

[150] J. W. Song, T. Hirosawa, T. Tsuneda, and K. Hirao. *J. Chem. Phys.*, 126:154105, 2007. DOI: 10.1063/1.2721532.

[151] V. N. Staroverov, G. E. Scuseria, J. Tao, and J. P. Perdew. *J. Chem. Phys.*, 119:12129, 2003. DOI: 10.1063/1.1626543.

[152] P. J. Stephens, F. J. Devlin, C. F. Chabolowski, and M. J. Frisch. *J. Phys. Chem.*, 98:11623, 1994. DOI: 10.1021/j100096a001.

[153] P. A. Stewart and P. M. W. Gill. *J. Chem. Soc. Faraday Trans.*, 91:4337, 1995. DOI: 10.1039/FT9959104337.

[154] J. Sun, B. Xiao, and A. Ruzsinszky. *J. Chem. Phys.*, 137:051101, 2012. DOI: 10.1063/1.4742312.

[155] J. Sun, R. Haunschild, B. Xiao, I. W. Bulik, G. E. Scuseria, and J. P. Perdew. *J. Chem. Phys.*, 138:044113, 2013. DOI: 10.1063/1.4789414.

[156] J. Sun, J. P. Perdew, and A. Ruzsinszky. *Proc. Natl. Acad. Sci. USA*, 112:685, 2015. DOI: 10.1073/pnas.1423145112.

[157] J. Sun, A. Ruzsinszky, and J. P. Perdew. *Phys. Rev. Lett.*, 115:036402, 2015. DOI: 10.1103/PhysRevLett.115.036402.

[158] J. Tao and Y. Mo. *Phys. Rev. Lett.*, 117:073001, 2016. DOI: 10.1103/PhysRevLett.117.073001.

[159] J. Tao, J. P. Perdew, V. N. Staroverov, and G. E. Scuseria. *Phys. Rev. Lett.*, 91:146401, 2003. DOI: 10.1103/PhysRevLett.91.146401.

[160] J. Toulouse, A. Savin, and H.-J. Flad. *Int. J. Quantum Chem.*, 100:1047, 2004. DOI: 10.1002/qua.20259.

[161] T. Tsuneda, T. Suzumura, and K. Hirao. *J. Chem. Phys.*, 110:10664, 1999. DOI: 10.1063/1.479012.

[162] F. Uhlig, J. M. Herbert, M. P. Coons, and P. Jungwirth. *J. Phys. Chem. A*, 118:7507, 2014. DOI: 10.1021/jp5004243.

[163] T. Van Voorhis and G. E. Scuseria. *J. Chem. Phys.*, 109:400, 1998. DOI: 10.1063/1.476577.

[164] P. Verma and R. J. Bartlett. *J. Chem. Phys.*, 140:18A534, 2014. DOI: 10.1063/1.4871409.

[165] P. Verma and D. G. Truhlar. *J. Phys. Chem. Lett.*, 8:380, 2017. DOI: 10.1021/acs.jpclett.6b02757.

[166] P. Verma, Y. Wang, S. Ghost, X. He, and D. G. Truhlar. *J. Phys. Chem. A*, 123:2966, 2019. DOI: 10.1021/acs.jpca.8b11499.

[167] S. H. Vosko, L. Wilk, and M. Nusair. *Can. J. Phys.*, 58:1200, 1980. DOI: 10.1139/p80-159.

[168] O. A. Vydrov and T. Van Voorhis. *Phys. Rev. Lett.*, 103:063004, 2009. DOI: 10.1103/PhysRevLett.103.063004.

[169] O. A. Vydrov and T. Van Voorhis. *J. Chem. Phys.*, 132:164113, 2010. DOI: 10.1063/1.3398840.

[170] O. A. Vydrov and T. Van Voorhis. *J. Chem. Phys.*, 133:244103, 2010. DOI: 10.1063/1.3521275.

[171] O. A. Vydrov, Q. Wu, and T. Van Voorhis. *J. Chem. Phys.*, 129:014106, 2008. DOI: 10.1063/1.2948400.

[172] Y. Wang, X. Jin, H. S. Yu, D. G. Truhlar, and X. He. *Proc. Natl. Acad. Sci. USA*, page 201705670, 2017. DOI: 10.1073/pnas.1705670114.

[173] Y. Wang, P. Verma, X. Jin, D. G. Truhlar, and X. He. *Proc. Natl. Acad. Sci. USA*, 115:10257, 2018. DOI: 10.1073/pnas.1810421115.

[174] Y. Wang, P. Verma, L. Zhang, Y. Li, Z. Liu, D. G. Truhlar, and X. He. *Proc. Natl. Acad. Sci. USA*, 117:2294, 2020. DOI: 10.1073/pnas.1913699117.

[175] E. Weintraub, T. M. Henderson, and G. E. Scuseria. *J. Chem. Theory Comput.*, 5:754, 2009. DOI: 10.1021/ct800530u.

[176] J. Wellendorff, K. T. Lundgaard, A. Møgelhøj, V. Petzold, D. D. Landis, J. K. Nørskov, T. Bligaard, and K. W. Jacobsen. *Phys. Rev. B*, 85:235149, 2012. DOI: 10.1103/PhysRevB.85.235149.

[177] J. Wellendorff, K. T. Lundgaard, K. W. Jacobsen, and T. Bligaard. *J. Chem. Phys.*, 140:144107, 2014. DOI: 10.1063/1.4870397.

[178] S. E. Wheeler and K. N. Houk. *J. Chem. Theory Comput.*, 6:395, 2010. DOI: 10.1021/ct900639j.

[179] E. P. Wigner. *Trans. Faraday Soc.*, 34:678, 1938. DOI: 10.1039/tf9383400678.

[180] K. W. Wiitala, T. R. Hoye, and C. J. Cramer. *J. Chem. Theory Comput.*, 2:1085, 2006. DOI: 10.1021/ct6001016.

[181] P. J. Wilson, T. J. Bradley, and D. J. Tozer. *J. Chem. Phys.*, 115:9233, 2001. DOI: 10.1063/1.1412605.

[182] J. Witte, N. Mardirossian, J. B. Neaton, and M. Head-Gordon. *J. Chem. Theory Comput.*, 13:2043, 2017. DOI: 10.1021/acs.jctc.7b00176.

[183] X. Xu and W. A. Goddard III. *Proc. Natl. Acad. Sci. USA*, 101:2673, 2004. DOI: 10.1073/pnas.0308730100.

[184] T. Yanai, D. P. Tew, and N. C. Handy. *Chem. Phys. Lett.*, 393:51, 2004. DOI: 10.1016/j.cplett.2004.06.011.

[185] H. S. Yu, W. Zhang, P. Verma, X. He, and D. G. Truhlar. *Phys. Chem. Chem. Phys.*, 17:12146, 2015. DOI: 10.1039/C5CP01425E.

[186] H. S. Yu, X. He, S. L. Li, and D. G. Truhlar. *Chem. Sci.*, 7:5032, 2016. DOI: 10.1039/C6SC00705H.

[187] H. S. Yu, X. He, and D. G. Truhlar. *J. Chem. Theory Comput.*, 12:1280, 2016. DOI: 10.1021/acs.jctc.5b01082.

[188] Y. Zhang and W. Yang. *Phys. Rev. Lett.*, 80:890, 1998. DOI: 10.1103/PhysRevLett.80.890.

[189] Y. Zhao and D. G. Truhlar. *J. Phys. Chem. A*, 108:6908, 2004. DOI: 10.1021/jp048147q.

[190] Y. Zhao and D. G. Truhlar. *J. Phys. Chem. A*, 109:5656, 2005. DOI: 10.1021/jp050536c.

[191] Y. Zhao and D. G. Truhlar. *J. Chem. Phys.*, 125:194101, 2006. DOI: 10.1063/1.2370993.

[192] Y. Zhao and D. G. Truhlar. *J. Phys. Chem. A*, 110:13126, 2006. DOI: 10.1021/jp066479k.

[193] Y. Zhao and D. G. Truhlar. *J. Chem. Phys.*, 128:184109, 2006. DOI: 10.1063/1.2912068.

[194] Y. Zhao and D. G. Truhlar. *J. Chem. Theory Comput.*, 4:1849, 2007. DOI: 10.1021/ct800246v.

[195] Y. Zhao and D. G. Truhlar. *Theor. Chem. Acc.*, 120:215, 2008. DOI: 10.1007/s00214-007-0310-x.

[196] Y. Zhao and D. G. Truhlar. *Chem. Phys. Lett.*, 502:1, 2011. DOI: 10.1016/j.cplett.2010.11.060.

[197] Y. Zhao, B. J. Lynch, and D. G. Truhlar. *J. Phys. Chem. A*, 108:2715, 2004. DOI: 10.1021/jp049908s.

[198] Y. Zhao, N. E. Schultz, and D. G. Truhlar. *J. Chem. Phys.*, 123:161103, 2005. DOI: 10.1063/1.2126975.

[199] Y. Zhao, N. E. Schultz, and D. G. Truhlar. *J. Chem. Theory Comput.*, 2:364, 2006. DOI: 10.1021/ct0502763.

[200] T. Ziegler. *Chem. Rev.*, 91:651, 1991. DOI: 10.1021/cr00005a001.

Chapter 6

Wave Function-Based Correlation Methods

6.1 Introduction

The Hartree-Fock procedure, while often qualitatively correct, is frequently quantitatively deficient. The deficiency is due to the underlying assumption of the Hartree-Fock approximation: that electrons move *independently* within molecular orbitals subject to an averaged field imposed by the remaining electrons. The error that this introduces is called the correlation energy and a wide variety of procedures exist for estimating its magnitude. The purpose of this Chapter is to introduce the main wave function-based methods available in QC-PBC to describe electron correlation.

Wave function-based electron correlation methods concentrate on the design of corrections to the wave function beyond the mean-field Hartree-Fock description. This is to be contrasted with the density functional theory methods discussed in the previous Chapter. While density functional methods yield a description of electronic structure that accounts for electron correlation subject only to the limitations of present-day functionals (which, for example, omit dispersion interactions), DFT cannot be systematically improved if the results are deficient. Wave function-based approaches for describing electron correlation[4,5] offer this main advantage. Their main disadvantage is relatively high computational cost, particularly for the higher-level theories.

Currently within QC-PBC, the only class of models for describing electron correlation that are supported is based on perturbative treatment of pair correlations between electrons. Details about these methods are discussed in the following subsections.

In order to carry out a wave function-based electron correlation calculation using QC-PBC, two *$rem* variables need to be set:

- BASIS to specify the basis set (see Chapter 7).
- POSTSCF_METHOD for treating correlation.

The full range of ground and excited state wave function-based correlation methods available (*i.e.* the recognized options to the POSTSCF_METHOD keyword) are as follows. Note that several different variation of the listed methods (ex. spin-component-scaled MP2 or regularized MP2) can be enabled by adding extra *$rem* variables on top of calling the relevant POSTSCF_METHOD keywords, as we will discuss in Section 6.3.

POSTSCF_METHOD

Specifies the level of theory, either DFT or wave function-based.

TYPE:

STRING

DEFAULT:

No default

OPTIONS:

MP2	Section 6.3
LTMP2	Section 6.3
MP3	Section 6.4
dRPA	Section 6.7
BW-s2	Section 6.6
CCSD	
CCSD(T)	
CCSDT	

RECOMMENDATION:

Consult the literature for guidance.

6.2 Møller-Plesset Perturbation Theory

Møller-Plesset Perturbation Theory [19] 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. In the remainder of this section, the theoretical basis of Møller-Plesset theory is reviewed.

The Hartree-Fock wave function Ψ_0 and energy E_0 are *approximate* solutions (eigenfunction and eigenvalue) to the exact Hamiltonian eigenvalue problem or Schrödinger's electronic wave equation, Eq. (4.5). The HF wave function and energy are, however, exact solutions for the Hartree-Fock Hamiltonian H_0 eigenvalue problem. If we assume that the Hartree-Fock wave function Ψ_0 and energy E_0 lie near the exact wave function Ψ and energy E, we can now write the exact Hamiltonian operator as

$$H = H_0 + \lambda V \tag{6.1}$$

where V is the small perturbation and λ is a dimensionless parameter. Expanding the exact wave function and energy in terms of the HF wave function and energy yields

$$E = E^{(0)} + \lambda E^{(1)} + \lambda^2 E^{(2)} + \lambda^3 E^{(3)} + \dots \tag{6.2}$$

and

$$\Psi = \Psi_0 + \lambda \Psi^{(1)} + \lambda^2 \Psi^{(2)} + \lambda^3 \Psi^{(3)} + \dots \tag{6.3}$$

Substituting these expansions into the Schrödinger equation and collecting terms according to powers of λ yields

$$H_0 \Psi_0 = E^{(0)} \Psi_0 \tag{6.4}$$

$$H_0 \Psi^{(1)} + V \Psi_0 = E^{(0)} \Psi^{(1)} + E^{(1)} \Psi_0 \tag{6.5}$$

$$H_0 \Psi^{(2)} + V \Psi^{(1)} = E^{(0)} \Psi^{(2)} + E^{(1)} \Psi^{(1)} + E^{(2)} \Psi_0 \tag{6.6}$$

and so forth. Multiplying each of the above equations by Ψ_0 and integrating over all space yields the following expression for the nth-order (MPn) energy:

$$E^{(0)} = \langle \Psi_0 | H_0 | \Psi_0 \rangle \tag{6.7}$$

$$E^{(1)} = \langle \Psi_0 | V | \Psi_0 \rangle \tag{6.8}$$

$$E^{(2)} = \left\langle \Psi_0 | V | \Psi^{(1)} \right\rangle \tag{6.9}$$

Thus, the Hartree-Fock energy

$$E_0 = \langle \Psi_0 | \, H_0 + V \, | \Psi_0 \rangle \tag{6.10}$$

is simply the sum of the zeroth- and first- order energies

$$E_0 = E^{(0)} + E^{(1)} \tag{6.11}$$

The correlation energy can then be written

$$E_{\text{corr}} = E_0^{(2)} + E_0^{(3)} + E_0^{(4)} + \dots \tag{6.12}$$

of which the first term is the MP2 energy.

In molecular quantum chemistry, the MP2 energy can be written (in terms of spin-orbitals) as

$$E_0^{(2)} = -\sum_{ia} \frac{|F_{ia}|^2}{\varepsilon_a - \varepsilon_i} - \frac{1}{4} \sum_{ij}^{\text{occ}} \sum_{ab}^{\text{virt}} \frac{|\langle ij| \, |ab\rangle|^2}{\varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j} \tag{6.13}$$

where $\mathbf{F}$ is the Fock matrix and

$$\langle ij||ab\rangle = (ia|jb) - (ib|ja) \tag{6.14}$$

In periodic systems, each molecular orbital index is associated with a k-point index. We can write out the MP2 correlation energy expression per unit cell with momentum indices:

$$E_0^{(2)} = \frac{1}{N_k} \sum_{ia} \sum_{\mathbf{k}} \frac{|F_{i_{\mathbf{k}} a_{\mathbf{k}}}|^2}{\Delta_{i_{\mathbf{k}}}^{a_{\mathbf{k}}}} + \frac{1}{4N_k} \sum_{iajb} \sum_{\mathbf{k}_1, \mathbf{k}_2, \mathbf{k}_3, \mathbf{k}_4} \frac{|\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2} || a_{\mathbf{k}_3} b_{\mathbf{k}_4} \rangle|^2}{\Delta_{i_{\mathbf{k}_1} j_{\mathbf{k}_2}}^{a_{\mathbf{k}_3} b_{\mathbf{k}_4}}} \tag{6.15}$$

Here $\Delta_{i_{\mathbf{k}}}^{a_{\mathbf{k}}} = \epsilon_i^{\mathbf{k}} - \epsilon_a^{\mathbf{k}}$ and $\Delta_{i_{\mathbf{k}_1} j_{\mathbf{k}_2}}^{a_{\mathbf{k}_3} b_{\mathbf{k}_4}} = \epsilon_i^{\mathbf{k_1}} + \epsilon_j^{\mathbf{k_2}} - \epsilon_a^{\mathbf{k_3}} - \epsilon_b^{\mathbf{k_4}}$ and

$$\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2} || a_{\mathbf{k}_3} b_{\mathbf{k}_4} \rangle = (i_{\mathbf{k}_1} a_{\mathbf{k}_3} | j_{\mathbf{k}_2} b_{\mathbf{k}_4}) - (i_{\mathbf{k}_1} b_{\mathbf{k}_4} | j_{\mathbf{k}_2} a_{\mathbf{k}_3}) \tag{6.16}$$

$$(i_{\mathbf{k}_1} a_{\mathbf{k}_3} | j_{\mathbf{k}_2} b_{\mathbf{k}_4}) = \int \int \phi_{i,\mathbf{k}_1}^*(\mathbf{r}_1) \phi_{a,\mathbf{k}_3}(\mathbf{r}_1) \frac{1}{|\mathbf{r}_1 - \mathbf{r}_2|} \phi_{j,\mathbf{k}_2}^*(\mathbf{r}_2) \phi_{b,\mathbf{k}_4}(\mathbf{r}_2) d\mathbf{r}_1 d\mathbf{r}_2 \tag{6.17}$$

which can be written in terms of the two-electron repulsion integrals in the AO basis:

$$(i_{\mathbf{k}_1} a_{\mathbf{k}_3} | j_{\mathbf{k}_2} b_{\mathbf{k}_4}) = \sum_{\mu} \sum_{\nu} \sum_{\lambda} \sum_{\sigma} \left(C_{\mu i}^{\mathbf{k}_1} \right)^* C_{\nu a}^{\mathbf{k}_3} \left(C_{\lambda j}^{\mathbf{k}_2} \right)^* C_{\sigma b}^{\mathbf{k}_4} \, (\mu_{\mathbf{k}_1} \nu_{\mathbf{k}_3} | \lambda_{\mathbf{k}_2} \sigma_{\mathbf{k}_4}) \tag{6.18}$$

Similar to molecular quantum chemistry, the number of two-center four-electron integrals grows as N^4 with respect to the number of basis functions N in the system. With respect to number of k-points N_k however, the scaling for number of non zero integrals is reduced to N_k^3 due to the fact that the integral can only be non-zero when momentum conservation is followed: $\mathbf{k_1} - \mathbf{k_2} + \mathbf{k_3} - \mathbf{k_4} = \mathbf{0}$. Therefore, the overall storage cost of 2e4c ERIs scales as $\mathcal{O}(N^4 N_k^3)$. It is also worth noting that contrary to molecular quantum chemistry, ERIs in periodic quantum chemistry are generally complex-valued. The computational cost of the MP2 correlation energy is $\mathcal{O}(N_k^3 N^4)$. The integral transformation is still the bottleneck, which costs $\mathcal{O}(N_k^3 N^5)$.

6.3 MP2 Methods

6.3.1 Introduction

As mentioned in the previous section, for periodic systems the overall storage cost of 2e4c ERIs scales as $\mathcal{O}(N^4 N_k^3)$. To reduce the storage cost of ERIs, we utilize the fact that the ERI tensor is approximately low-rank, so it is possible to compress the tensor by rank-revealing Cholesky Decomposition. In our implementation, we use the modified cholesky decomposition algorithm [7] to generate Cholesky vectors on-the-fly. In the AO basis, the Cholesky vectors can be written as:

$$[\mu(\mathbf{k}_1)\nu(\mathbf{k}_1+\mathbf{q})|\lambda(\mathbf{k}_2+\mathbf{q})\sigma(\mathbf{k}_2)] = \sum_{\gamma}^{N_{Chol}} \left(L_{\mu,\nu}^{\gamma,\mathbf{q}}\right)^* L_{\sigma,\lambda}^{\gamma,\mathbf{q}} \tag{6.19}$$

Note that here the second Cholesky vector has the index ordering σ and λ permuted, which comes from the fact that the periodic ERIs are complex-valued. Furthermore, due to the aforementioned momentum conservation, we know that there are only three possible degrees of freedom for k-point indices, and thus we can do CD on ERI blocks that sahre the same k-point index $\mathbf{q}$. That means overall the storage is reduced to $\mathcal{O}(N^2 N_k^2 N_{Chol})$.

Currently in QC-PBC, all implemented post-HF correlation methods by default use the Cholesky decomposition framework to calculate correlation energies. In the following subsections, we will describe the methods currently available and the relevant *$rem* keywords used to control them.

6.3.2 CD-MP2 Energies

In the CD-MP2 method, we first generate Cholesky vectors of the ERIs in the AO basis, then do AO to MO transformation to generate ov-block MO Cholesky vectors, which are used to evaluate MP2 energies. Writing out the MP2 energy using Cholesky vectors, we have:

$$E_c = \frac{1}{4N_k} \sum_{\mathbf{k}_1,\mathbf{k}_2,\mathbf{q}} \sum_{ij}^{occ} \sum_{ab}^{vir} \frac{|\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2+\mathbf{q}} || a_{\mathbf{k}_1+\mathbf{q}} b_{\mathbf{k}_2}\rangle|^2}{\Delta_{i_{\mathbf{k}_1} j_{\mathbf{k}_2+\mathbf{q}}}^{a_{\mathbf{k}_1+\mathbf{q}} b_{\mathbf{k}_2}}} \tag{6.20}$$

$$(i_{\mathbf{k_1}} a_{\mathbf{k_1}+\mathbf{q}} | j_{\mathbf{k_2}+\mathbf{q}} b_{\mathbf{k_2}}) = \sum_{\mu\nu\lambda\sigma} C_\mu^{i*} C_\nu^a C_\lambda^{j*} C_\sigma^b \cdot \Big[\sum_{J=1}^{N_{Chol}} (\mathbf{L}_{\mu\nu,J}^{\mathbf{q}})^* (\mathbf{L}_{\lambda\sigma,J}^{\mathbf{q}}) \Big] \tag{6.21}$$

$$= \sum_{J=1}^{N_{Chol}} \mathbf{L}_{ia,J}^{L,\mathbf{q}} \mathbf{L}_{jb,J}^{R,\mathbf{q}} \tag{6.22}$$

$$(i_{\mathbf{k_1}} b_{\mathbf{k_2}} | j_{\mathbf{k_2}+\mathbf{q}} a_{\mathbf{k_1}+\mathbf{q}}) = \left(i_{\mathbf{k_1}} b_{\mathbf{k_1}+\mathbf{q}'} | j_{(\mathbf{k_1}+\mathbf{q})+\mathbf{q}'} b_{\mathbf{k_1}+\mathbf{q}}\right) \tag{6.23}$$

$$= \sum_{\mu\nu\lambda\sigma} C_\mu^{i*} C_\nu^b C_\lambda^{j*} C_\sigma^a \cdot \Big[\sum_{J=1}^{N_{Chol}} (\mathbf{L}_{\mu\nu,J}^{\mathbf{q}'})^* (\mathbf{L}_{\lambda\sigma,J}^{\mathbf{q}'}) \Big] \tag{6.24}$$

$$= \sum_{J=1}^{N_{Chol}} \mathbf{L}_{ib,J}^{L,\mathbf{q}'} \mathbf{L}_{ja,J}^{R,\mathbf{q}'} \tag{6.25}$$

Note that because ov block MO ERIs lack four-fold permutational symmetry, in the MO basis the Cholesky vectors for the left hand side and the right hand side are not equivalent. (We denote as $\mathbf{L}^L$ and $\mathbf{L}^R$ respectively.)

The number of Cholesky vectors N_{Chol} for each $\mathbf{q}$ is determined by the threshold for the MCD algorithm. With a threshold of 10^{-3} for the maximum diagonal residual, one may obtain N_{chol} similar to N_{aux} of the density fitting basis set with comparable accuracy. In QC-PBC, this threshold is controlled by the CHOL_THRESH keyword.

CHOL_THRESH

Sets the threshold for modified Cholesky decomposition algorithm to be CHOL_THRESH * max(diagonal of ERI matrix).

TYPE:

DOUBLE

DEFAULT:

1e-5

OPTIONS:

User-defined threshold.

RECOMMENDATION:

Use default if possible; adjust to 1e-4 or 1e-3 if Cholesky Decomposition takes too much time.

Example 6.1 QC-PBC input for an CD-MP2 calculation of SiC with [3,1,1] kpoints.

```
$lattice
3
0.0 4.117713228198666 4.117713228198666
4.117713228198666 0.0 4.117713228198666
4.117713228198666 4.117713228198666 0.0
$end

$unitcell
ABSOLUTE
0 1
Si 0.0 0.0 0.0
C 2.058856614099333 2.058856614099333 2.058856614099333
$end

$rem
jobtype = sp
method = HF
kecut = 800
scf_convergence = 10
scf_algorithm = diis
pseudo = GTH-LDA
basis = SZV-GTH
input_bohr = true
k_alg = mo_k
mp_mesh = [3,1,1]
postscf_method = mp2
$end
```

View output online

6.3.3 CD-MP2 Method with the Laplace Transformation

N_LAPLACE_POINTS

Sets the number of Laplace quadrature points.

TYPE:

INTEGER

DEFAULT:

7

OPTIONS:

Integer from 1 to 10, inclusive.

RECOMMENDATION:

Use default.

Example 6.2 QC-PBC input for an LT-CD-MP2 calculation of SiC with [3,1,1] kpoints.

```
$lattice
3
0.0 4.117713228198666 4.117713228198666
4.117713228198666 0.0 4.117713228198666
4.117713228198666 4.117713228198666 0.0
$end

$unitcell
ABSOLUTE
0 1
Si 0.0 0.0 0.0
C 2.058856614099333 2.058856614099333 2.058856614099333
$end

$rem
jobtype = sp
method = HF
kecut = 800
scf_convergence = 10
scf_algorithm = diis
pseudo = GTH-LDA
basis = SZV-GTH
input_bohr = true
k_alg = mo_k
mp_mesh = [3,1,1]
postscf_method = ltmp2
$end
```

View output online

6.3.4 Spin-Biased CD-MP2 Methods

SPIN_SCALING

Determines the type of spin-scaling used for LT-MP2

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 LT-MP2

1 Spin-Opposite-Scaled LT-MP2

RECOMMENDATION:

NONE

SPIN_SCALING_PARAM

Determines the value of spin-scaling parameter (*1e-6)

TYPE:

INTEGER

DEFAULT:

1000000 Spin-Opposite-Scaled LT-MP2

OPTIONS:

User defined value

RECOMMENDATION:

Consult literature.

6.3.5 Regularized CD-MP2 Methods

Joonho Lee working with Martin Head-Gordon developed a new regularized OOMP2 suite of methods that utilizes energy-dependent regularizers (σ and κ-regularizer).[17] The κ-regularizer modifies the MP2 correlation energy as follows:

$$E_{\kappa-\mathrm{MP2}} = -\frac{1}{4}\sum_{ijab}\frac{|\langle ij||ab\rangle|^2}{\Delta_{ij}^{ab}}\left(1-\exp(-\kappa\Delta_{ij}^{ab})\right)^2 \tag{6.26}$$

MP2_REGULARIZATION

Determines the type of regularization used for MP2

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 No Regularization

1 κ-regularized MP2

2 σ-regularized MP2

RECOMMENDATION:

Up to user's choice.

REG_PARAM

Determines the value of regularization parameter (*1e-3)

TYPE:

INTEGER

DEFAULT:

1450 κ-regularized MP2 (corresponding to $\kappa = 1.45$)

1000 σ-regularized MP2 (corresponding to $\sigma = 1.0$)

OPTIONS:

User defined value

RECOMMENDATION:

Use default.

6.4 MP3

6.4.1 Introduction

Joonho Lee working with Martin Head-Gordon developed and added an implementation of RI-MP3 with only cubic-storage requirement.[18] In molecular quantum chemistry, the MP3 energy can be written (in terms of spin-orbitals) as

$$\begin{aligned} E_0^{(3)} = &\frac{1}{8}\sum_{ij}^{\text{occ}}\sum_{abcd}^{\text{vir}} \frac{\langle ij||ab\rangle^*\langle ab||cd\rangle^*\langle ij||cd\rangle}{(\varepsilon_a+\varepsilon_b-\varepsilon_i-\varepsilon_j)(\varepsilon_c+\varepsilon_d-\varepsilon_i-\varepsilon_j)} \\ &+\frac{1}{8}\sum_{ijkl}^{\text{occ}}\sum_{ab}^{\text{vir}} \frac{\langle ij||ab\rangle^*\langle ij||kl\rangle\langle kl||ab\rangle}{(\varepsilon_a+\varepsilon_b-\varepsilon_i-\varepsilon_j)(\varepsilon_a+\varepsilon_b-\varepsilon_k-\varepsilon_l)} \\ &-\sum_{ijk}^{\text{occ}}\sum_{abc}^{\text{vir}} \frac{\langle ij||ab\rangle^*\langle ic||kb\rangle\langle jk||ac\rangle}{(\varepsilon_a+\varepsilon_b-\varepsilon_i-\varepsilon_j)(\varepsilon_a+\varepsilon_c-\varepsilon_j-\varepsilon_k)} \end{aligned} \tag{6.27}$$

with

$$\langle ij||ab\rangle = (ia|jb) - (ib|ja), \tag{6.28}$$

where the first term represents the $E_{vv}^{(3)}$ energy, the second term the $E_{oo}^{(3)}$ energy, and the third term $E_{ov}^{(3)}$ energy. The evaluation of each term scales as $\mathcal{O}(n_{\text{occ}}^2 n_{\text{vir}}^4)$, $\mathcal{O}(n_{\text{occ}}^4 n_{\text{vir}}^2)$, and $\mathcal{O}(n_{\text{occ}}^3 n_{\text{vir}}^3)$, respectively.

Analogously to the derivation for MP2, we take advantage of momentum conservation to write the MP3 energy per unit cell with momentum indices as

$$\begin{aligned} E_0^{(3)} = &\frac{1}{8N_k}\sum_{ij}^{\text{occ}}\sum_{abcd}^{\text{vir}}\sum_{\mathbf{k}_1\mathbf{k}_2\mathbf{k}_3\mathbf{k}_4} \frac{\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2}||a_{\mathbf{k}_3} b_{\mathbf{k}_{123}}\rangle^*\langle a_{\mathbf{k}_3} b_{\mathbf{k}_{123}}||c_{\mathbf{k}_4} d_{\mathbf{k}_{124}}\rangle^*\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2}||c_{\mathbf{k}_4} d_{\mathbf{k}_{124}}\rangle}{(\varepsilon_{a_{\mathbf{k}_3}}+\varepsilon_{b_{\mathbf{k}_{123}}}-\varepsilon_{i_{\mathbf{k}_1}}-\varepsilon_{j_{\mathbf{k}_2}})(\varepsilon_{c_{\mathbf{k}_4}}+\varepsilon_{d_{\mathbf{k}_{124}}}-\varepsilon_{i_{\mathbf{k}_1}}-\varepsilon_{j_{\mathbf{k}_2}})} \\ &+\frac{1}{8N_k}\sum_{ijkl}^{\text{occ}}\sum_{ab}^{\text{vir}}\sum_{\mathbf{k}_1\mathbf{k}_2\mathbf{k}_3\mathbf{k}_4} \frac{\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2}||a_{\mathbf{k}_3} b_{\mathbf{k}_{123}}\rangle^*\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2}||k_{\mathbf{k}_4} l_{\mathbf{k}_{124}}\rangle\langle k_{\mathbf{k}_4} l_{\mathbf{k}_{124}}||a_{\mathbf{k}_3} b_{\mathbf{k}_{123}}\rangle}{(\varepsilon_{a_{\mathbf{k}_3}}+\varepsilon_{b_{\mathbf{k}_{123}}}-\varepsilon_{i_{\mathbf{k}_1}}-\varepsilon_{j_{\mathbf{k}_2}})(\varepsilon_{a_{\mathbf{k}_3}}+\varepsilon_{b_{\mathbf{k}_{123}}}-\varepsilon_{k_{\mathbf{k}_4}}-\varepsilon_{l_{\mathbf{k}_{124}}})} \\ &-\frac{1}{N_k}\sum_{ijk}^{\text{occ}}\sum_{abc}^{\text{vir}}\sum_{\mathbf{k}_1\mathbf{k}_2\mathbf{k}_3\mathbf{k}_4} \frac{\langle i_{\mathbf{k}_1} j_{\mathbf{k}_2}||a_{\mathbf{k}_3} b_{\mathbf{k}_{123}}\rangle^*\langle i_{\mathbf{k}_1} c_{\mathbf{k}_{234}}||k_{\mathbf{k}_4} b_{\mathbf{k}_{123}}\rangle\langle j_{\mathbf{k}_2} k_{\mathbf{k}_4}||a_{\mathbf{k}_3} c_{\mathbf{k}_{234}}\rangle}{(\varepsilon_{a_{\mathbf{k}_3}}+\varepsilon_{b_{\mathbf{k}_{123}}}-\varepsilon_{i_{\mathbf{k}_1}}-\varepsilon_{j_{\mathbf{k}_2}})(\varepsilon_{a_{\mathbf{k}_3}}+\varepsilon_{c_{\mathbf{k}_{234}}}-\varepsilon_{j_{\mathbf{k}_2}}-\varepsilon_{k_{\mathbf{k}_4}})} \end{aligned} \tag{6.29}$$

A similar strategy as described in Section 6.3.2 can be used to decompose the MP3 energy in terms of MO Cholesky vectors. Doing so brings the storage requirement to $\mathcal{O}(N^2 N_k^2 N_{Chol})$, and the computational cost to $\mathcal{O}(N_k^4 N^6)$.

To run this method, one should set POSTSCF_METHOD to MP3.

6.4.2 Examples

Example 6.3 QC-PBC input for an MP3 calculation of SiC with [3,1,1] kpoints.

```
$lattice
3
0.0 4.117713228198666 4.117713228198666
4.117713228198666 0.0 4.117713228198666
4.117713228198666 4.117713228198666 0.0
$end

$unitcell
ABSOLUTE
0 1
Si 0.0 0.0 0.0
C 2.058856614099333 2.058856614099333 2.058856614099333
$end

$rem
jobtype = sp
method = HF
kecut = 800
scf_convergence = 10
scf_algorithm = diis
pseudo = GTH-LDA
basis = SZV-GTH
input_bohr = true
k_alg = mo_k
mp_mesh = [3,1,1]
postscf_eri_method = ao_chol
chol_thresh = 1e-6
postscf_method = mp3
$end
```

View output online

6.5 Tensor Hypercontraction 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.[11,14,16] 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 QC-PBC.

The auxiliary basis $\{|K\rangle\}$ is used to approximate products of Gaussian basis functions:

$$|\mu\nu\rangle \approx |\widetilde{\mu\nu}\rangle = \sum_K |K\rangle C_{\mu\nu}^K \tag{6.30}$$

In tensor hypercontraction, the two-electron four-center integrals are approximated by

$$(i_{\mathbf{k}_1} a_{\mathbf{k}_1+\mathbf{q}} | j_{\mathbf{k}_2+\mathbf{q}} b_{\mathbf{k}_2}) = \sum_{\mu\nu\lambda\sigma} C_\mu^{i*} C_\nu^a C_\lambda^{j*} C_\sigma^b \cdot \Big[\sum_{P=1}^{N_{isdf}} \sum_{Q=1}^{N_{isdf}} X_{\mu_{\mathbf{k}_1}}^{P*} X_{\nu_{\mathbf{k}_1+\mathbf{q}}}^{P} M_{\mathbf{q}}^{PQ} X_{\lambda_{\mathbf{k}_2+\mathbf{q}}}^{Q*} X_{\sigma_{\mathbf{k}_2}}^{Q} \Big] \tag{6.31}$$

$$= \sum_{P=1}^{N_{isdf}} \sum_{Q=1}^{N_{isdf}} X_{i_{\mathbf{k}_1}}^{P*} X_{a_{\mathbf{k}_1+\mathbf{q}}}^{P} M_{\mathbf{q}}^{PQ} X_{j_{\mathbf{k}_2+\mathbf{q}}}^{Q*} X_{b_{\mathbf{k}_2}}^{Q} \tag{6.32}$$

6.6 Size-Consistent Brillouin-Wigner Perturbation Theory

6.6.1 Introduction

While second-order Møller-Plesset perturbation theory (MP2) offers the simplest and most cost-effective *ab initio* correlation energy, it requires a separate formulation for degenerate states because the MP2 energy diverges in the zero-gap limit. Second-order Brillouin-Wigner perturbation theory (BW2) requires no such reformulation, as the second-order energy given by,

$$E_{\mathrm{c}}^{(2)} = -\frac{1}{4}\sum_{ijab}\frac{|\langle ij||ab\rangle|^2}{\Delta_{ij}^{ab} + E_{\mathrm{c}}^{(2)}} \tag{6.33}$$

is finite when the orbital energy gap $\Delta_{ij}^{ab} = \varepsilon_a + \varepsilon_b - \varepsilon_i - \varepsilon_j$ is zero due to the presence of the correlation energy in the denominator. However, the BW2 correlation energy is not size-consistent in the sense that it does not satisfy $E(A \cup B) = E(A) + E(B)$ for two distant, noninteracting subsystems A and B, severely limiting its applicability to chemistry.

Recently, Carter-Fenk and Head-Gordon introduced a size-consistent-to-second-order Brillouin-Wigner perturbation theory (BW-s2) that retains this essential property while remaining finite for zero-gap systems.[?] This theory, based on a repartitioning of the zeroth-order Hamiltonian, results in a slightly-modified amplitude equation. Whereas the MP2 amplitudes can be found by solving,

$$\Delta_{ijkl}^{abcd} \cdot t_{kl}^{cd} = -\langle ij||ab\rangle \ , \tag{6.34}$$

the BW-s2 amplitude equation contains a regularizing tensor,

$$\left(\Delta_{ijkl}^{abcd} + R_{ijkl}^{abcd}\right) \cdot t_{kl}^{cd} = -\langle ij||ab\rangle \tag{6.35}$$

In the above equations Δ_{ijkl}^{abcd} is composed of Fock matrix elements and reduces to the familiar Δ_{ij}^{ab} when canonical orbitals are used, and

$$R_{ijkl}^{abcd} = \frac{\alpha}{2}\left(W_{ik}\delta_{jl} + \delta_{ik}W_{jl}\right)\delta_{ac}\delta_{bd} \tag{6.36}$$

where,

$$W_{ij} = \frac{1}{2}\sum_{kab}\left(t_{ik}^{ab}\langle jk||ab\rangle + t_{jk}^{ab}\langle ik||ab\rangle\right) \tag{6.37}$$

This form of $\mathbf{W}$ was chosen because it is size-consistent, leading to a size-consistent BW-s2 correlation energy. Physically, $\mathbf{W}$ represents the correlation energy of a Koopmans' (static orbital) ionization process. Thus, the occupied orbitals in BW-s2 are imbued with correlation such that the occupied/virtual gap increases. After rotating the occupied orbitals into a basis where $\boldsymbol{\Delta} + \boldsymbol{R}$ is diagonal, the BW-s2 working equation looks like that of MP2,

$$E_{\mathrm{c}}^{(2)} = -\frac{1}{4}\sum_{ijab}\frac{|\widetilde{\langle ij||ab\rangle}|^2}{\varepsilon_a + \varepsilon_b - \tilde{\varepsilon}_i - \tilde{\varepsilon}_j} \tag{6.38}$$

but the orbitals and corresponding anti-symmetrized two-electron integrals have been rotated into the new basis.

In equation 6.36 there is an implicit parameter α that controls the regularization strength. Initially, this parameter was set to $\alpha = 1$ to obtain the exact result for the two-electron two-orbital system of minimal-basis H_2 at the dissociation limit, but α was later tuned to achieve much more accurate results for a wide array of chemical problems ranging from thermochemical properties to noncovalent interaction energies and closed-shell transition-metal reaction energies.[?] The optimal α value for the resultant BW-s2(α) approach varies somewhat between chemical problems, but was found to be more flexible than gap-dependent regularizer parameters like κ-, σ-, or σ^2-MP2. A "universal" parameter of $\alpha = 4$ was suggested as a compromise to achieve the best all-around results for a wide array of chemical properties.[?]

6.6.2 BW-s2 Job Control

The BW-s2 approach has been implemented with the resolution-of-the-identity approximation and is invoked with METHOD = RIBWS2. The BW-s2 equation is iterative much like BW2, and keywords that give more precise control over the self-consistent determination of the BW-s2 energy are housed within a separate *$bws2* section. Finally, the underlying algorithm that is used to solve the BW-s2 equation can be used to solve for other forms of the correlation energy such as MP2, BW2, size-extensive xBW2 (which is not size-consistent), or second-order Bethe-Goldstone (BGE2)/independent electron pair approximation (IEPA). While these methods are available for academic purposes, it is advised to only use the BW2, xBW2, and BGE2/IEPA codes for comparison as the first two are not size-consistent and the latter is not orbital invariant. For MP2 calculations, the standard RI-MP2 code should be used.

METHOD

Defines the correlation energy expression to be used.

INPUT SECTION: *$bws2*

TYPE:

STRING

DEFAULT:

BWS2

OPTIONS:

BWS2, BW2, xBW2, BGE2, IEPA, or MP2

RECOMMENDATION:

BW-s2 is the recommended method, but if comparisons with non-size-consistent methods like BW2 or xBW2 or non-orbital-invariant methods like BGE2/IEPA are desired, those options are available.

ALPHA

Controls the extent of BW-s2(α) regularization.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

100

OPTIONS:

n where $n/100$ is the value of α.

RECOMMENDATION:

The default is set to the original BW-s2 theory where $\alpha = 1$ (corresponding to $n = 100$) is exact for a two-electron, two-orbital problem. For general applications $\alpha = 4$ ($n = 400$) is a broadly successful choice. Other values can be set based on the chemical problem at hand.

ENERGY_CONVERGENCE

Defines the convergence threshold for the correlation energy.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

n = SCF_CONVERGENCE

OPTIONS:

n where the convergence threshold is set to 10^{-n}

RECOMMENDATION:

Use the default unless the cost of the calculation becomes prohibitive, then consider reducing to a slightly lower value.

MAX_CYCLES

Defines maximum number of iterations to be used in determining the correlation energy.

INPUT SECTION: *$bws2*

TYPE:

INTEGER

DEFAULT:

n = SCF_MAX_CYCLES

OPTIONS:

$n > 0$

RECOMMENDATION:

Use the default unless convergence is particularly slow, then increase the number of iterations accordingly.

6.7 Direct Random Phase Approximation Methods

6.7.1 Introduction

A useful $\mathcal{O}(N^4)$ approach called the direct random phase approximation (dRPA) based on the RI approximation is available. This particular implementation was added by Joonho Lee working with Martin Head-Gordon.[18] RI-dRPA has been applied to the thermochemistry[10] and non-covalent interaction problems[20] and often demonstrated superior performance over RI-MP2. In terms of the computational cost, RI-dRPA should be compared to the scaled-opposite-spin MP2 while theoretically it involves diagrams far beyond second-order and includes infinite-order diagrams similarly to coupled-cluster theory. In fact, one can view dRPA as a reduced coupled-cluster with doubles approach.[22] In a nutshell, we define the dRPA energy as

$$E = E_{\text{HF}} + E_{\text{c}}^{\text{dRPA}} \tag{6.39}$$

where using the plasmon formula we compute[13]

$$E_{\text{c}}^{\text{dRPA}} = \int_{-\infty}^{\infty} \frac{\mathrm{d}\omega}{4\pi} \text{tr}\left[\ln\left(\mathbf{I} + \mathbf{Q}(\omega)\right) - \mathbf{Q}(\omega)\right] \tag{6.40}$$

where

$$\mathbf{Q}(\omega) = 2\mathbf{B}^T\mathbf{D}(\mathbf{D}^2 + \omega^2\mathbf{I})^{-1}\mathbf{B} \tag{6.41}$$

with

$$B_{ia,P} = \sum_{Q} (ia|Q)(Q|P)^{-1/2} \tag{6.42}$$

$$D_{ia,jb} = \delta_{ij}\delta_{ab}(\epsilon_a - \epsilon_i) \tag{6.43}$$

In this form, the cost of computing the dRPA correlation is quartic-scaling which is comparable to SOS-MP2. To use this method, one must set METHOD = RIDRPA along with AUXBASIS.

6.7.2 dRPA Job Control

CLENSHAW_NGRID

Number of grid points for the Curtis-Clenshaw quadrature.

TYPE:

INTEGER

DEFAULT:

40

OPTIONS:

RECOMMENDATION:

Use default.

6.8 Coupled-Cluster Methods

6.8.1 Introduction

The following sections give short summaries of the various coupled-cluster based methods available in QC-PBC, most of which are variants of coupled-cluster theory. The basic object-oriented tools necessary to permit the implementation of these methods in QC-PBC was accomplished by Anna Krylov and David Sherrill, working at Berkeley with Martin Head-Gordon, and then continuing independently at the University of Southern California and Georgia Tech, respectively. While at Berkeley, Krylov and Sherrill also developed the optimized orbital coupled-cluster method, with additional assistance from Ed Byrd. The extension of this code to MP3, MP4, CCSD and QCISD is the work of Steve Gwaltney at Berkeley, while the extensions to QCCD were implemented by Ed Byrd at Berkeley. More recently, a new code (termed CCMAN2) has been developed in Krylov group by Evgeny Epifanovsky and others; QC-PBC interfaces with CCMAN2 to provide support for several basic flavors of coupled-cluster theory for Γ-point calculations. **k**-point CC calculations are currently too expensive for routine use and thus are not implemented.

POSTSCF_METHOD

Specifies the correlation level of theory handled by CCMAN2.

TYPE:

STRING

DEFAULT:

None No Correlation

OPTIONS:

CCSD Singles and doubles.

CCSD(T) Singles, doubles, and perturbative triples.

CCSDT singles, doubles, and triples

RECOMMENDATION:

Consult the literature for guidance.

6.8.2 Coupled Cluster Singles and Doubles (CCSD)

The standard approach for treating pair correlations self-consistently are coupled-cluster methods where the cluster operator contains all single and double substitutions,[21] abbreviated as CCSD. CCSD yields results that are only slightly superior to MP2 for structures and frequencies of stable closed-shell molecules. However, it is far superior for reactive species, such as transition structures and radicals, for which the performance of MP2 is quite erratic.

A full textbook presentation of CCSD is beyond the scope of this manual, and several comprehensive references are available. However, it may be useful to briefly summarize the main equations. The CCSD wave function is:

$$|\Psi_{\mathrm{CCSD}}\rangle = \exp\left(\hat{T}_1 + \hat{T}_2\right)|\Phi_0\rangle \tag{6.44}$$

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):

$$\hat{T}_1|\Phi_0\rangle = \sum_{i}^{\mathrm{occ}}\sum_{a}^{\mathrm{virt}} t_i^a |\Phi_i^a\rangle \tag{6.45}$$

$$\hat{T}_2|\Phi_0\rangle = \frac{1}{4}\sum_{ij}^{\mathrm{occ}}\sum_{ab}^{\mathrm{virt}} t_{ij}^{ab} |\Phi_{ij}^{ab}\rangle \tag{6.46}$$

It is not feasible to determine the CCSD energy by variational minimization of $\langle E\rangle_{\mathrm{CCSD}}$ 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:

$$
\begin{aligned}
E_{\mathrm{CCSD}} &= \langle\Phi_0|\hat{H}|\Psi_{\mathrm{CCSD}}\rangle \\
&= \left\langle \Phi_0 \left| \hat{H} \right| \left(1 + \hat{T}_1 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2\right)\Phi_0 \right\rangle_C \qquad (6.47) \\
0 &= \left\langle \Phi_i^a \left| \hat{H} - E_{\mathrm{CCSD}} \right| \Psi_{\mathrm{CCSD}} \right\rangle \\
&= \left\langle \Phi_i^a \left| \hat{H} \right| \left(1 + \hat{T}_1 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2 + \hat{T}_1\hat{T}_2 + \frac{1}{3!}\hat{T}_1^3\right)\Phi_0 \right\rangle_C \qquad (6.48) \\
0 &= \left\langle \Phi_{ij}^{ab} \left| \hat{H} - E_{\mathrm{CCSD}} \right| \Psi_{\mathrm{CCSD}} \right\rangle \\
&= \left\langle \Phi_{ij}^{ab} \left| \hat{H} \right| \left(1 + \hat{T}_1 + \frac{1}{2}\hat{T}_1^2 + \hat{T}_2 + \hat{T}_1\hat{T}_2 + \frac{1}{3!}\hat{T}_1^3 \right.\right. \\
&\qquad \left.\left. + \frac{1}{2}\hat{T}_2^2 + \frac{1}{2}\hat{T}_1^2\hat{T}_2 + \frac{1}{4!}\hat{T}_1^4\right)\Phi_0 \right\rangle_C \qquad (6.49)
\end{aligned}
$$

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. QC-PBC supports both energies and analytic gradients for CCSD for RHF and UHF references (including frozen core). For ROHF, only energies and unrelaxed properties are available. Available properties include dipole moments, angular momentum projections, $\langle\hat{S}^2\rangle$, static polarizabilities, and g-tensors (see Section **??** for details).

6.8.3 Coupled Cluster Singles, Doubles and Triples (CCSDT)

The coupled-cluster method with single, double and triple substitutions, abbreviated as CCSDT**?** includes single, double and triple excitation operators in the exponential ansatz. The theory of the method is very similar to that of CCSD – with triple excitations included fully. We only present the basic equations. These can be compared with the CCSD equations presented in the previous section, so as to understand the similarities and differences between CCSD and CCSDT. The CCSDT wave-function defined by

$$|\Psi_{\mathrm{CCSD}}\rangle = \exp\left(\hat{T}_1 + \hat{T}_2 + \hat{T}_3\right)|\Phi_0\rangle \qquad (6.50)$$

where, the operators, $\hat{T}_1$ and $\hat{T}_2$ are defined using Eqs. 6.45 and 6.46. The operator, $\hat{T}_3$ is defined by

$$\hat{T}_3|\Phi_0\rangle = \frac{1}{36}\sum_{ijk}^{\mathrm{occ}}\sum_{abc}^{\mathrm{virt}} t_{ijk}^{abc}\left|\Phi_{ijk}^{abc}\right\rangle \qquad (6.51)$$

The CCSDT equations are coupled non-linear simultaneous equations of the tensors, $\hat{T}_1$, $\hat{T}_2$ and $\hat{T}_3$. However, the correlation energy depends only on $\hat{T}_1$ and $\hat{T}_2$ amplitudes (The energy equations is same as Eq 6.47). The effect of

triples is due to mutual coupling between singles, doubles and triples

$$\begin{aligned}
E_{\mathrm{CCSDT}} &= \langle\Phi_0|\hat{H}|\Psi_{\mathrm{CCSDT}}\rangle \\
&= \left\langle \Phi_0 \left|\hat{H}\right| \left(1+\hat{T}_1+\frac{1}{2}\hat{T}_1^2+\hat{T}_2\right)\Phi_0 \right\rangle_C \qquad (6.52)\\
0 &= \left\langle \Phi_i^a \left|\hat{H}-E_{\mathrm{CCSDT}}\right|\Psi_{\mathrm{CCSDT}}\right\rangle \\
&= \left\langle \Phi_i^a \left|\hat{H}\right| \left(1+\hat{T}_1+\hat{T}_2+\hat{T}_3+\frac{1}{2}\hat{T}_1^2+\hat{T}_1\hat{T}_2+\frac{1}{3!}\hat{T}_1^3\right)\Phi_0 \right\rangle_C \qquad (6.53)\\
0 &= \left\langle \Phi_{ij}^{ab} \left|\hat{H}-E_{\mathrm{CCSDT}}\right|\Psi_{\mathrm{CCSDT}}\right\rangle \\
&= \left\langle \Phi_{ij}^{ab} \left|\hat{H}\right| \left(1+\hat{T}_1+\hat{T}_2+\hat{T}_3+\frac{1}{2}\hat{T}_1^2++\hat{T}_1\hat{T}_2+\hat{T}_1\hat{T}_3+\frac{1}{2}\hat{T}_2^2\right.\right. \\
&\qquad \left.\left. +\frac{1}{3!}\hat{T}_1^3+\frac{1}{2}\hat{T}_1^2\hat{T}_2+\frac{1}{4!}\hat{T}_1^4\right)\Phi_0 \right\rangle_C \qquad (6.54)\\
0 &= \left\langle \Phi_{ijk}^{abc} \left|\hat{H}-E_{\mathrm{CCSDT}}\right|\Psi_{\mathrm{CCSDT}}\right\rangle \\
&= \left\langle \Phi_{ijk}^{abc} \left|\hat{H}\right| \left(\hat{T}_2+\hat{T}_3+\hat{T}_1\hat{T}_2+\hat{T}_1\hat{T}_3+\hat{T}_2\hat{T}_3\right.\right. \\
&\qquad \left.\left. +\frac{1}{2}\hat{T}_2^2+\frac{1}{2}\hat{T}_1^2\hat{T}_2+\frac{1}{2}\hat{T}_1^2\hat{T}_3+\frac{1}{2}\hat{T}_1\hat{T}_2^2+\frac{1}{3!}\hat{T}_1^3\hat{T}_2\right)\Phi_0 \right\rangle_C \qquad (6.55)
\end{aligned}$$

Currently, the CCSDT functionality is available for computation of correlation energy only.

6.8.4 Cholesky Decomposition with CC (CD-CC)

Two-electron integrals can be decomposed using Cholesky decomposition[12] giving rise to the same representation as in RI and substantially reducing the cost of integral transformation, disk storage requirements, and improving parallel performance:

$$(\mu\nu|\lambda\sigma) \approx \sum_{P=1}^{M} B_{\mu\nu}^P B_{\lambda\sigma}^P, \qquad (6.56)$$

The rank of Cholesky decomposition, M, is typically 3-10 times larger than the number of basis functions N (Ref. 6); it depends on the decomposition threshold δ and is considerably smaller than the full rank of the matrix, $N(N+1)/2$ (Refs. 6,9,23). Cholesky decomposition removes linear dependencies in product densities $(\mu\nu|$,[6] allowing one to obtain compact approximation to the original matrix with accuracy, in principle, up to machine precision.

Decomposition threshold δ is the only parameter that controls accuracy and the rank of the decomposition. Cholesky decomposition is invoked by specifying CHOLESKY_TOL that defines the accuracy with which decomposition should be performed. For most calculations tolerance of $\delta = 10^{-3}$ gives a good balance between accuracy and compactness of the rank. Tolerance of $\delta = 10^{-2}$ can be used for exploratory calculations and $\delta = 10^{-4}$ for high-accuracy calculations. Similar to RI, Cholesky-decomposed integrals can be transformed back, into the canonical MO form, using CC_DIRECT_RI keyword.

6.8.5 Job Control Options

There are a large number of options for the coupled-cluster singles and doubles methods. 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 not yet implemented in QC-PBC. Below we list the options that one should be aware of for routine calculations.

CC_CONVERGENCE

Overall convergence criterion for the coupled-cluster codes. This is designed to ensure at least n significant digits in the calculated energy, and automatically sets the other convergence-related variables (CC_E_CONV, CC_T_CONV, CC_THETA_CONV, CC_THETA_GRAD_CONV) [10^{-n}].

TYPE:

INTEGER

DEFAULT:

6 Energies.

7 Gradients.

OPTIONS:

n Corresponding to 10^{-n} convergence criterion. Amplitude convergence is set automatically to match energy convergence.

RECOMMENDATION:

Use the default

Note: For single point calculations, CC_E_CONV = 6 and CC_T_CONV = 4. Tighter amplitude convergence (CC_T_CONV = 5) is used for gradients and EOM calculations.

CC_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 QC-PBC

TYPE:

INTEGER

DEFAULT:

1

OPTIONS:

$0 - 7$ higher values can lead to deforestation...

RECOMMENDATION:

Increase if you need more output and don't like trees

CHOLESKY_TOL

Tolerance of Cholesky decomposition of two-electron integrals

TYPE:

INTEGER

DEFAULT:

3

OPTIONS:

n Corresponds to a tolerance of 10^{-n}

RECOMMENDATION:

2 - qualitative calculations, 3 - appropriate for most cases, 4 - quantitative (error in total energy typically less than 1 μhartree)

CC_DIRECT_RI

Controls use of RI and Cholesky integrals in conventional (undecomposed) form

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

FALSE use all integrals in decomposed format

TRUE transform all RI or Cholesky integral back to conventional format

RECOMMENDATION:

By default all integrals are used in decomposed format allowing significant reduction of memory use. If all integrals are transformed back (TRUE option) no memory reduction is achieved and decomposition error is introduced, however, the integral transformation is performed significantly faster and conventional CC/EOM algorithms are used.

6.8.6 Examples

Example 6.4 Input for a CCSD energy calculation on BN. Note that only Γ-point calculations are available to coupled-cluster methods.

```
$comment
Run Hartree-Fock on BN followed by CCSD.
$end

$lattice
3 ! dimension
2.51242800000000 0.00000000000000 0.00000000000000
-1.25621400000000 2.17582700000000 0.00000000000000
0.00000000000000 0.00000000000000 7.70726500000000
$end

$unitcell
absolute
0 1
B -0.0000012562 1.4505520586 1.9268162500
B 1.2562152562 0.7252749414 5.7804487500
N -0.0000012562 1.4505520586 5.7804487500
N 1.2562152562 0.7252749414 1.9268162500
$end

$rem
    JOBTYPE     sp          ! Energy calculation
    METHOD      hf          ! Hartree-Fock
    BASIS       szv-gth     ! double zeta basis
    KECUT       1000        ! planewave cutoff 1000 eV
    PSEUDO      gth-pbe     ! use GTH pseudopotential fit with PBE
    POSTSCF_METHOD ccsd     ! run CCSD after SCF
$end
```

View output online

References and Further Reading

[1] Self-Consistent Field Methods (Chapter 4).

[2] Excited-State Calculations (Chapter **??**).

[3] Basis Sets (Chapter 7) and Effective Core Potentials (Chapter **??**).

[4] For a general textbook introduction to electron correlation methods and their respective strengths and weaknesses, see Ref. 15.

[5] For a tutorial introduction to electron correlation methods based on wavefunctions, see Ref. 8.

[6] F. Aquilante, T. B. Pedersen, and R. Lindh. *Theor. Chem. Acc.*, 124:1, 2009. DOI: 10.1007/s00214-009-0608-y.

[7] Francesco Aquilante, Linus Boman, Jonas Boström, Henrik Koch, Roland Lindh, Alfredo Sánchez De Merás, and Thomas Bondo Pedersen. Cholesky decomposition techniques in electronic structure theory. In Robert Zalesny, Manthos G. Papadopoulos, Paul G. Mezey, and Jerzy Leszczynski, editors, *Linear-Scaling Techniques in Computational Chemistry and Physics*, volume 13, pages 301–343. Springer Netherlands, Dordrecht, 2011. ISBN 978-90-481-2852-5 978-90-481-2853-2. DOI: 10.1007/978-90-481-2853-2_13.

[8] R. J. Bartlett and J. F. Stanton. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 5, chapter 2, page 65. Wiley-VCH, New York, 1994. DOI: 10.1002/9780470125823.ch2.

[9] N. H. F. Beebe and J. Linderberg. *Int. J. Quantum Chem.*, 12:683, 1977. DOI: 10.1002/qua.560120408.

[10] S. Dohm, A. Hansen, M. Steinmetz, S. Grimme, and M. P. Checinski. *J. Chem. Theory Comput.*, 14:2596, 2018. DOI: 10.1021/acs.jctc.7b01183.

[11] B. I. Dunlap. *Phys. Chem. Chem. Phys.*, 2:2113, 2000. DOI: 10.1039/b000027m.

[12] E. Epifanovsky, D. Zuev, X. Feng, K. Khistyaev, Y. Shao, and A. I. Krylov. *J. Chem. Phys.*, 139:134105, 2013. DOI: 10.1063/1.4820484.

[13] H. Eshuis, J. Yarkony, and F. Furche. *J. Chem. Phys.*, 132:234114, 2010. DOI: 10.1063/1.3442749.

[14] M. Feyereisen, G. Fitzgerald, and A. Komornicki. *Chem. Phys. Lett.*, 208:359, 1993. DOI: 10.1016/0009-2614(93)87156-W.

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

[16] Y. Jung, A. Sodt, P. M. W. Gill, and M. Head-Gordon. *Proc. Natl. Acad. Sci. USA*, 102:6692, 2005. DOI: 0.1073/pnas.0408475102.

[17] J. Lee and M. Head-Gordon. *J. Chem. Theory Comput.*, 14:5203, 2018. DOI: 10.1021/acs.jctc.8b00731.

[18] J. Lee, L. Lin, and M. Head-Gordon. *J. Chem. Theory Comput.*, 16:243, 2020. DOI: 10.1021/acs.jctc.9b00820.

[19] C. Møller and M. S. Plesset. *Phys. Rev.*, 46:618, 1934. DOI: 10.1103/PhysRev.46.618.

[20] B. D. Nguyen, G. P. Chen, M. M. Agee, A. M. Burow, M. P. Tang, and F. Furche. *J. Chem. Theory Comput.*, 16: 2258, 2020. DOI: 10.1021/acs.jctc.9b01176.

[21] G. D. Purvis and R. J. Bartlett. *J. Chem. Phys.*, 76:1910, 1982. DOI: 10.1063/1.443164.

[22] G. E. Scuseria, T. M. Henderson, and D. C. Sorensen. *J. Chem. Phys.*, 129:231101, 2008. DOI: 10.1063/1.3043729.

[23] S. Wilson. *Comput. Phys. Commun.*, 58:71–81, 1990. DOI: 10.1016/0010-4655(90)90136-O.

Chapter 7

Basis Sets and Pseudopotentials

7.1 Introduction to Basis Sets

A basis set is a set of functions combined linearly to model molecular orbitals. Basis functions can be considered as representing the atomic orbitals of the atoms and are introduced in quantum chemical calculations because the equations defining the molecular orbitals are otherwise very difficult to solve.

Many standard basis sets have been carefully optimized and tested over the years. In principle, a user would employ the largest basis set available in order to model molecular orbitals as accurately as possible. In practice, the computational cost grows rapidly with the size of the basis set so a compromise must be sought between accuracy and cost. If this is systematically pursued, it leads to a "theoretical model chemistry",[17] 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 require pseudopotentials and are unable to model core electrons effectively.

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. GTOs as used in molecular calculations do not enforce Bloch's theorem, but this can be remedied through a simple modification to form crystalline GTOs. The QC-PBC program uses the GPW approach by default, which is unable to accurately model core electrons. Therefore, the QC-PBC program has been optimized to exploit crystalline basis sets of the contracted Gaussian function type, with core electrons removed. QC-PBC has a large number of built-in standard basis sets (modifications of the Dunning and Pople all-electron basis sets, among others) which have been developed to model only the valence electrons; the user can access these quickly and easily. Additionally, the QC-PBC program has a large number of all-electron basis sets built in for use in non-GPW calculations.

The selection of a basis set for quantum chemical calculations is very important. It is sometimes possible to use small basis sets to obtain good chemical accuracy, but calculations can often be significantly improved by the addition of diffuse and polarization functions. Consult the literature, including pertinent review articles,[6,13,17,19,20] in order to aid

your selection. See also the "Further Reading" section at the end of this chapter.

7.2 Crystalline Gaussian-Type Orbitals

QC-PBC forms periodic basis sets out of traditionally atomic based GTO basis sets by performing explicit lattice sums to obtain crystalline GTOs (cGTOs). Within this framework, the basis set for **k**-point **k** is given by:

$$\phi_\mu^{\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{R}} e^{i\mathbf{k}\cdot\mathbf{R}} \phi_\mu(\mathbf{r} - \mathbf{R}) \tag{7.1}$$

In the above, $\phi_\mu(\mathbf{r})$ is a typical molecular GTO, centered on an atomic center, $\mathbf{R}$ is a lattice vector - an integer multiple of primitive direct lattice vectors.

cGTOs obey Bloch's theorem and may be used to build the molecular orbitals in much the same way as traditional CGTOs:

$$\phi_i^{\mathbf{k}}(\mathbf{r}) = \sum_{\mu} C_{\mu i}^{\mathbf{k}} \phi_\mu^{\mathbf{k}}(\mathbf{r}) \tag{7.2}$$

In this way, one may simply specify a molecular GTO basis set and QC-PBC will automatically generate the corresponding cGTO basis sets at each **k**-point.

7.3 Built-In Basis Sets

QC-PBC inherits the vast basis set library of Q-CHEM [1,28], but most of these are unsuitable for routine GPW calculations as they are all-electron basis sets containing very compact core orbitals. The Q-CHEM basis set library has therefore been augmented with several basis sets designed for use with GTH pseudopotentials.

The available built-in basis sets include the following types:

- Pseudopotential basis sets
  - Goedecker-Teter-Hutter (GTH) basis sets
  - Correlation-consistent Dunning basis sets for solids,[35] including:
    - Small core standard: GTH-cc-pvXZ-sc (X=D,T,Q) (lc/sc option only changes Li, Be, Na, and Mg)
    - Large core standard: GTH-cc-pvXZ-lc (X=D,T,Q) (lc/sc option only changes Li, Be, Na, and Mg)
    - augmented GTH-aug-cc-pvXZ (X=D,T,Q)
- All-electron basis sets
  - Pople basis sets [3,8–10,14,15]
  - Dunning basis sets [11]
  - Correlation-consistent (cc) Dunning basis sets, including:
    - Standard cc-pVXZ and aug-cc-pVXZ (X = D, T, Q, 5, and 6) [2,12,33,34]
    - Partially-augmented "calendar" versions,[24,36] may-, jun-, and jul-cc-pVXZ (X = D, T, Q)
    - Versions with core–valence polarization functions,[27] cc-pCVXZ and cc-pwCVXZ (X = D, T, Q)
    - Partially-augmented core–valence polarization basis sets jun-, jul-, and aug-cc-pCVXZ (X = D, T, Q)

    - Pseudopotential (PP) basis sets for heavy elements, cc-pVXZ-PP and aug-cc-pVXZ-PP (X = D, T, Q)[18,25,26]
  - Ahlrichs basis sets[31]
  - Karlsruhe "def2" basis sets,[32] including
    - Augmented versions,[29] such as def2-SVPD
    - Partially-augmented versions,[16] such as def2-ha-SVP and def2-ha-SVP
  - Jensen polarization consistent basis sets[21–23]
  - Universal Gaussian basis set (UGBS)[7]

In addition, QC-PBC supports the following features:

- Extra diffuse functions available for high quality excited-state calculations.
- Standard polarization functions.
- s, p, sp, d, f, g and h angular momentum types of basis functions (for energy calculations, up to k are supported).
- Pure and Cartesian basis functions.
- Mixed basis sets (see Section 7.6).
- Basis set superposition error (BSSE) corrections.
- Automatic, on-the-fly generation of a superposition of atomic densities (SAD) guess for any basis set (including general and mixed basis sets) and any SCF level of theory (see Section 4.6.2).

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.

7.4 Basis Set Symbolic Representation

7.4.1 Symbolic Representation Overview

Examples are given in the tables below and follow the standard format generally adopted for specifying basis sets. The single exception applies to additional diffuse functions. These are best inserted in a similar manner to the polarization functions; in parentheses with the light atom designation following heavy atom designation: (*heavy*, *light*), using a period as a placeholder in the unusual case that diffuse functions are to be added to hydrogen atoms but not to heavy

Basis Name[a]	j	m[b]	n[c]
STO-j(k+,l+)G(m,n)	2,3,6	d	p
j-21(k+,l+)G(m,n)	3	2d	2p
j-31(k+,l+)G(m,n)	4,6	3d	3p
j-311(k+,l+)G(m,n)	6	df, 2df, 3df	pd, 2pd, 3pd

[a] k and l denote the number of sets of diffuse functions on heavy atoms and on hydrogen atoms, respectively.
[b] m denotes the number of sets of polarization functions on the heavy atoms.
[c] n denotes the number of sets of polarization functions on the hydrogen atoms.

Table 7.1: Summary of Pople-type basis sets available in QC-PBC.

Symbolic Name	**Atoms Supported**
STO-2G	H, He, Li→Ne, Na→Ar, K, Ca, Sr
STO-3G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→I
STO-6G	H, He, Li→Ne, Na→Ar, K→Kr
3-21G	H, He, Li→Ne, Na→Ar, K→Kr, Rb→Xe, Cs
4-31G	H, He, Li→Ne, P→Cl
6-31G	H, He, Li→Ne, Na→Ar, K→Kr
6-311G	H, He, Li→Ne, Na→Ar, Ga→I
G3LARGE	H, He, Li→Ne, Na→Ar, K→Kr
G3MP2LARGE	H, He, Li→Ne, Na→Ar, Ga→Kr

Table 7.2: Atoms supported for Pople basis sets available in QC-PBC.

Symbolic Name	**Atoms Supported**
3-21G	H, He, Li → Ne, Na → Ar, K →Kr, Rb → Xe, Cs
3-21+G	H, He, Na → Cl, Na → Ar, K, Ca, Ga → Kr
3-21G*	Na → Ar
6-31G	H, He, Li → Ne, Na → Ar, K → Zn, Ga → Kr
6-31+G	H, He, Li → Ne, Na → Ar, Ga → Kr
6-31G*	H, He, Li → Ne, Na → Ar, K → Zn, Ga → Kr
6-31G(d,p)	H, He, Li → Ne, Na → Ar, K → Zn, Ga → Kr
6-31G(.,+)G	H, He, Li → Ne, Na → Ar, Ga → Kr
6-31+G*	H, He, Li → Ne, Na → Ar, Ga → Kr
6-311G	H, He, Li → Ne, Na → Ar, Ga → I
6-311+G	H, He, Li → Ne, Na → Ar
6-311G*	H, He, Li → Ne, Na → Ar, Ga → I
6-311G(d,p)	H, He, Li → Ne, Na → Ar, Ga → I
G3LARGE	H, He, Li → Ne, Na → Ar, K → Kr
G3MP2LARGE	H, He, Li → Ne, Na → Ar, Ga → Kr

Table 7.3: Examples of extended Pople basis sets.

atoms. See Table 7.1 for the general form. This convention can be applied, for example, to the named Pople-style basis sets listed in Table 7.2, resulting in specific examples given in Table 7.3.

Although not widely used in modern quantum chemistry, Dunning[11] introduced an early set of basis sets denoted SV, DZ, and TZ; see Table 7.4. (These are not to be confused with the widely-used "correlation-consistent" basis sets, which are also associated with Dunning's name.) The original Dunning basis sets can be extended with diffuse and polarization functions using a nomenclature similar to that used for Pople basis sets: *name*(k+,l+)(md,np), where k is the number of additional heavy atom diffuse functions, l is the number of additional light atom diffuse functions, m is the number of additional d polarization functions on heavy atoms, and n is the number of additional p polarization functions on light atoms.

Symbolic Name	**Atoms Supported**
SV	H, Li → Ne
SV*	H, B → Ne
SV(d,p)	H, B → Ne
SV(2+,+)(2d,p)	H, B → Ne
DZ	H, Li → Ne, Al → Cl
DZ+	H, B → Ne
DZ++	H, B → Ne
DZ*	H, Li → Ne
DZ**	H, Li → Ne
DZ(d,p)	H, Li → Ne
DZ(2+,+)(2d,p)	H, B → Ne
TZ	H, Li → Ne
TZ+	H, Li → Ne
TZ++	H, Li → Ne
TZ*	H, Li → Ne
TZ**	H, Li → Ne
TZ(d,p)	H, Li → Ne

Table 7.4: Examples of extended Dunning basis sets.

The much more widely-used basis sets that are associated with Dunning are the correlation-consistent ("cc") ones.[12,34] The basic ones and their augmented counterparts are listed in Table 7.5. Those appended with "-PP" are pseudopotential basis sets, defined for heavy elements only and intended to be used in conjunction with effective core potentials (ECPs), which are discussed in Section **??**. Each correlation-consistent basis set (cc-*name* has an "augmented" counterpart (aug-cc-*name*) that includes diffuse functions.

The correlation-consistent paradigm adds additional diffuse functions for each angular momentum class, meaning that for a second-row atom such as carbon, the aug-cc-pVDZ basis set contains diffuse s, p, and d functions (10 diffuse functions per atom), while hydrogen contains diffuse s and p functions. The aug-cc-pVTZ basis set also includes diffuse f functions for carbon (for a total of 20 diffuse functions per atom) and diffuse d functions for hydrogen. As compared to functions with tighter exponents, inclusion of diffuse functions is relatively expensive and prone to incur linear dependencies that hamper SCF convergence, as discussed in Section 7.4.2. At the same time, diffuse functions are often crucial to the description of anions, excited states, and noncovalent interactions but the high angular momentum diffuse functions included in aug-cc-pVXZ are not always necessary. In recognition of this fact. "calendar" versions of the correlation-consistent basis sets have been introduced (jul-, jun-, and may-*name*),[24,36] which systemically remove diffuse basis functions starting from aug-cc-*name*. The jul-cc-pVXZ basis set removes all diffuse functions from hydrogen, and is equivalent to using cc-pVXZ for hydrogen and aug-cc-pVXZ for heavy atoms. The jun-cc-pVXZ

Symbolic Name	**Atoms Supported**
cc-pVDZ	H → Ar, Ca, Ga → Kr
cc-pVDZ-full	H → Ar, Ca → Kr
cc-pVDZ-PP	Cu → Rn
cc-pVTZ	H → Ar, Ca, Ga → Kr
cc-pVTZ-full	H → Ar, Ca → Kr
cc-pVTZ-PP	Cu → Rn
cc-pVQZ	H → Ar, Ca, Ga → Kr
cc-pVQZ-full	H → Ar, Ca → Kr
cc-pVQZ-PP	Cu → Rn
cc-pV5Z	H → Ar, Ca → Kr
cc-pV6Z	H → Ar except Li, Na, Mg
cc-pCVDZ	H → Ar, Ca (H and He use cc-pVDZ)
cc-pCVTZ	H → Ar, Ca (H and He use cc-pVTZ)
cc-pCVQZ	H → Ar, Ca (H and He use cc-pVQZ)
cc-pCV5Z	H, He, B → Ar, Ca (H and He use cc-pV5Z)
cc-pwCVDZ	B → Ne, Al → Ar
cc-pwCVTZ	B → Ne, Al → Ar, Sc → Zn
cc-pwCVQZ	B → Ne, Al → Ar, Sc → Zn, Br
cc-pwCVDZ-PP	Cu → Rn
cc-pwCVTZ-PP	Cu → Rn
cc-pwCVQZ-PP	Cu → Rn
aug-cc-pVDZ[a]	H → Kr
aug-cc-pVDZ-PP[b]	Cu → Rn
aug-cc-pVTZ[a]	H → Kr
aug-cc-pVTZ-PP[b]	Cu → Rn
aug-cc-pVQZ[a]	H → Kr
aug-cc-pVQZ-PP[b]	Cu → Rn
aug-cc-pV5Z	H → Ar, Sc → Kr
aug-cc-pV6Z	H → Ar except Li, Be, Na, Mg
aug-cc-pCVDZ[a]	H → Ar (H and He use aug-cc-pVDZ)
aug-cc-pCVTZ[a]	H → Ar (H and He use aug-cc-pVTZ)
aug-cc-pCVQZ[a]	H → Ar (H and He use aug-cc-pVQZ)
aug-cc-pCV5Z	H, He, B → Ar (H and He use aug-cc-pV5Z)
aug-cc-pwCVDZ[c]	B → Ne, Al → Ar
aug-cc-pwCVTZ[c]	B → Ne, Al → Ar, Sc → Zn
aug-cc-pwCVQZ[c]	B → Ne, Al → Ar, Sc → Zn, Br
aug-cc-pwCVDZ-PP[c]	Cu → Rn
aug-cc-pwCVTZ-PP[c]	Cu → Rn
aug-cc-pwCVQZ-PP[c]	Cu → Rn

[a] may-, jun-, and jul-cc-p(C)VXZ variants are also available
[b] jun-cc-pVXZ-PP variant is also available
[c] jun-cc-p(w)VXZ(-PP) variant is also available

Table 7.5: Atoms supported in QC-PBC for correlation-consistent basis sets. For cc-pVXZ (X = D, T, Q), those names which do not end in "-full" correspond to the definitions with a segmented contraction scheme,[5] and those that do end in "-full" correspond to the original optimized generally-contracted definitions. For all other basis sets, where there is no distinction, the only definition is from optimized general contraction. For the augmented basis sets, footnotes indicate the availability of "calendar" variants.[24,36]

Symbolic Name	Atoms Supported
TZV	H → Kr
VDZ	H → Kr
VTZ	H → Kr

Table 7.6: Atoms supported in QC-PBC for the original Ahlrichs basis sets.[31] (Note that these are different from the more modern Karlsruhe "def2" basis sets, which are described in Table 7.7.)

Symbolic Name	Atoms Supported
def2-mSVP	H–Kr,[a] Rb–Rn (with def2-ECP)
def2-SV(P), def2-SVP	H–Kr; Rb–Rn (with def2-ECP)
def2-ma-SVP, def2-ha-SVP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-SVPD	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-TZVP, def2-TZVPP	H–Kr; Rb–Rn (with def2-ECP)
def2-ma-TZVP, def2-ma-TZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-ha-TZVP, def2-ha-TZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-TZVPD, def2-TZVPPD	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-QZVP, def2-QZVPP	H–Kr; Rb–Rn (with def2-ECP)
def2-ma-QZVP, def2-ma-QZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-ha-QZVP, def2-ha-QZVPP	H–Kr; Rb–La, Hf–Rn (with def2-ECP)
def2-QZVPD, def2-QZVPPD	H–Kr; Rb–La, Hf–Rn (with def2-ECP)

[a] Na–Kr are identical to def2-SV(P)

Table 7.7: Atoms supported in QC-PBC for the Karlsruhe "def2" basis sets.[16,29,32]

basis set additionally removes the highest angular momentum diffuse functions from each heavy atom, *e.g.*, for a carbon atom the diffuse d functions are removed to make jun-cc-pVDZ and the diffuse f functions are removed to make jun-cc-pVTZ. The may-cc-pVXZ basis sets then remove the highest angular momentum diffuse functions that remain in jun-cc-pVXZ, so that for a carbon atom, may-cc-pVDZ is minimally augmented with only a single diffuse s function. QC-PBC includes may-, jun-, and jul-cc-pVXZ and similarly may-, jun-, and jul-cc-pCVXZ (for X = D, T, and Q in both cases). Also available are the jun-cc-pVXZ-PP parings of aug-cc-pVXZ-PP and the jun-cc-pwCVXZ(-PP) parings of aug-cc-pwCVXZ(-PP), again for X = D, T, or Q. If the user has questions as to what functions are included in any of these basis sets, simply set PRINT_GENERAL_BASIS = TRUE in the *$rem* section (as described in Section 7.4.2) to get a printout of the basis function information.

The name Ahlrichs is also associated with two different collections of basis sets. The older set (TZV, VDZ, and VTZ) is listed in Table 7.6;[31] these basis sets are available but are no longer in common use. Much more widely used are the second-generation "def2" basis sets that are listed in Table 7.7,[29,32] which are sometimes called "Karlsruhe" basis sets to distinguish them from the older basis sets developed by Ahlrichs and co-workers at the University of Karlsruhe. These basis sets were originally designed for SCF calculations although more recently they have seen some use in correlated wave function calculations. Diffuse functions were added later,[29] and are stipulated with a name ending in "D", *e.g.*, def2-SVP does not contain diffuse functions but def2-SVPD does. The def2-ha and def2-ma variants (*e.g.*, def2-ha-SVP) include partial augmentation.[16] The def2-ha basis sets are "heavy-augmented", eliminating all diffuse functions on the hydrogen atoms, so that def2-ha-SVP consists of def2-SVP for hydrogen and def2-SVPD for other atoms. The def2-ma basis sets are "minimally-augmented", and are constructed from def2-ha-SVP by removing the highest angular moment diffuse function on each heavy atom, similar to the jun-cc-pVXZ prescription.

Finally, there is a set of basis sets associated with the name of Jensen[21,22] (see Table 7.8), which were developed

Symbolic Name[a]	**Atoms Supported**
pcseg-n	H $\rightarrow$ Kr
pc-n	H $\rightarrow$ Kr
pcJ-n	H $\rightarrow$ Ar
psS-n	H $\rightarrow$ Ar
aug-pcseg-n	H $\rightarrow$ Kr
aug-pc-n	H $\rightarrow$ Kr
aug-pcJ-n	H $\rightarrow$ Ar
aug-psS-n	H $\rightarrow$ Ar

[a]For $n = 0, 1, 2, 3, 4$ in each case

Table 7.8: Atoms supported for Jensen polarization consistent basis sets available in QC-PBC. The pcseg-n sets should be preferred instead of pc-n, as they are more efficient in QC-PBC. The pcJ-n[22] and pcS-n[21] basis sets are optimized for NMR spin-spin couplings and chemical shieldings, respectively.

primarily for NMR calculations. There is also a "universal" Gaussian basis set,[7] which is supported for elements H–Lr except for Pa–Np and Cm–Bk.

7.4.2 Customization

QC-PBC 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. This effect is especially pronounced in periodic calculations where the basis set is repeated in every unitcell. Furthermore, when evaluating periodic two electron integrals in real-space via lattice sums, diffuse basis functions can dramatically increase the computational cost of a calculation. For this reason, periodic basis sets generally do not include diffuse basis functions.

QC-PBC 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. QC-PBC checks for linear dependence by considering the eigenvalues of the overlap matrix. Very small eigenvalues are an indication that the basis set is close to being linearly dependent. The size at which the eigenvalues are considered to be too small is governed by the *$rem* variable BASIS_LIN_DEP_THRESH. By default this is set to 6, corresponding to a threshold of 10^{-6}. This has been found to give reliable results, however SCF convergence failure (especially for large molecules or those with highly diffuse basis sets) may be a symptom of linear dependencies. The smallest overlap matrix eigenvalue is printed in the QC-PBC output file, and usually when this number goes below 10^{-5}, numerical issues caused by basis function linear dependence may occur and the SCF calculation may not give reasonable solutions. If the smallest overlap matrix eigenvalue is less than the square root of the integral threshold, a warning message urging to tighten the integral threshold (*e.g.*, setting THRESH = 14) will be printed out. In any case, when a linear dependence issue is suspected, tightening the integral threshold should be tried first. Especially for larger molecules in basis sets that contain diffuse functions, tightening the integral threshold sometimes has the nonintuitive effect of *decreasing* the

time-to-solution, by significantly reducing the number of SCF cycles at only a modest per-cycle increase in cost.

PRINT_GENERAL_BASIS

Controls print out of built in basis sets in input format

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Print out standard basis set information

FALSE Do not print out standard basis set information

RECOMMENDATION:

Useful for modification of standard basis sets.

BASIS_LIN_DEP_THRESH

Sets the threshold for determining linear dependence in the basis set

TYPE:

INTEGER

DEFAULT:

6 Corresponding to a threshold of 10^{-6}

OPTIONS:

n Sets the threshold to 10^{-n}

RECOMMENDATION:

Set to 5 or smaller if you have a poorly behaved SCF and you suspect linear dependence in you basis set. Lower values (larger thresholds) may affect the accuracy of the calculation.

7.5 User-Defined Basis Sets (*$basis*)

7.5.1 Introduction

Users may, on occasion, prefer to use non-standard basis, and it is possible to declare user-defined basis sets in QC-PBC input (see Chapter 3 on QC-PBC inputs). The format for inserting a non-standard user-defined basis set is both logical and flexible, and is described in detail in the job control section below.

Note that the SAD guess is not currently supported with non-standard or user-defined basis sets. The current default is to use SCF_GUESS = AUTOSAD, unless a mixed basis is specified. When using a mixed basis, the simplest alternative is to specify the GWH or CORE options for SCF_GUESS, but these are relatively ineffective other than for small basis sets. The recommended alternative is to employ basis set projection by specifying a standard basis set for the BASIS2 keyword. See Section 4.6 on initial guesses for more information.

7.5.2 Job Control

In order to use a user-defined basis set, the BASIS *$rem* must be set to GENERAL or GEN.

When using a non-standard basis set which incorporates d or higher angular momentum basis functions, the *$rem* variable PURECART needs to be set. This *$rem* variable indicates to the QC-PBC program how to handle the angular form of the basis functions. As indicated above, each integer represents an angular momentum type which can be defined as either pure (1) or Cartesian (2). For example, 1111 specifies that d, f, g and h basis functions should have

pure form whereas 1121 indicates that d- g- and h-functions are pure but f functions are Cartesian. These four-digit codes can be used even if the basis does not contain g or h functions; PURECART = 1111 (or 2222) specifies to use pure (or Cartesian) functions for all angular momentum types.

PURECART
INTEGER
TYPE:
Controls the use of pure (spherical harmonic) or Cartesian angular forms
DEFAULT:
1111 Pure h, g, f, d functions
OPTIONS:
$hgfd$ Use 1 for pure and 2 for Cartesian.
RECOMMENDATION:
This is pre-defined for all standard basis sets

In standard basis sets all functions are pure, except for the d functions in n-21G–type bases (*e.g.*, 3-21G) and n-31G bases (*e.g.*, 6-31G, 6-31G*,6-31+G*, ...). In particular, the 6-311G series uses pure functions for both d and f.[??] Note that other electronic structure codes may use Cartesian functions for basis sets of the same name, in which case absolute energies will differ. Set PURECART = 2222 to request Cartesian Gaussian functions, if desired.

7.5.3 Format for User-Defined Basis Sets

The format for the user-defined basis section is as follows:

```
$basis
   X    0
   L    K      scale
   α1   C1^Lmin  C1^(Lmin+1)  ...  C1^Lmax
   α2   C2^Lmin  C2^(Lmin+1)  ...  C2^Lmax
   ⋮    ⋮      ⋮             ⋱    ⋮
   αK   CK^Lmin  CK^(Lmin+1)  ...  CK^Lmax
****
$end
```

where

X	Atomic symbol of the atom (atomic number not accepted)
L	Angular momentum symbol (S, P, SP, D, F, G)
K	Degree of contraction of the shell (integer)
$scale$	Scaling to be applied to exponents (default is 1.00)
α_i	Gaussian primitive exponent (positive real number)
C_i^L	Contraction coefficient for each angular momentum (non-zero real numbers).

Atoms are terminated with $\star\star\star\star$ and the complete basis set is terminated with the *$end* keyword terminator. No blank lines can be incorporated within the general basis set input. Note that more than one contraction coefficient per line is required for compound shells like SP. As with all QC-PBC input deck information, all input is case-insensitive.

7.6 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.

7.7 Ghost Atoms and Basis Set Superposition Error

When calculating intermolecular interaction energies, a naïve calculation of the energy difference

$$\Delta E_{AB} = E_{AB} - E_A - E_B \tag{7.3}$$

usually results in severe overestimation of the interaction energy, even if all three energies in Eq. (7.3) are computed at a good level of theory. This phenomenon, known as *basis set superposition error* (BSSE), is an artifact of an unbalanced approximation, namely, that the dimer energy E_{AB} is computed in a more flexible basis set as compared to the two monomer energies. Although BSSE disappears in the complete basis-set limit, it does so extremely slowly: in $(H_2O)_6$, for example, an MP2/aug-cc-pVQZ calculation of the interaction energy is still a bit more than 1 kcal/mol away from the MP2 complete-basis limit.[30] Short of computing all energies in very large basis sets and extrapolating to the complete-basis limit, the conventional solution to the BSSE problem is the *counterpoise correction*, originally proposed by Boys and Bernardi.[4] Here, one corrects for BSSE by computing the monomer energies E_A and E_B in the dimer basis set, with the idea being that this results in a more balanced treatment of ΔE_{AB}.

In truth the average of the counterpoise-corrected and uncorrected results is often a better approximation than either of them individually, but in any case one needs the counterpoise-corrected result. This requires basis functions to be placed at arbitrary points in space, not just those defined by the nuclear centers; these are usually termed "floating centers" or "ghost atoms". Ghost atoms have zero nuclear charge but can support a user-defined basis set. Their positions are specified in the *$unitcell* section alongside all the other atoms (atomic symbol: Gh), and their intended basis functions are specified in one of two ways:

1. Via a user-defined *$basis* section, using BASIS = MIXED.

2. Placing "@" next to an atomic symbol in the *$unitcell* section designates it as a ghost atom supporting the same basis functions as the corresponding atom, so that a *$basis* section is not required.

Examples of either procedure appear below.

Example 7.1 Input for the calculation of cohesive energy of CO_2. The first job computes the energy of the molecular crystal. The second job computes the energy of a single molecule with the basis set of the surrounding molecules included (counterpoise correction).

```
$comment
Compute cohesive energy of a co2 molecular crystal using counterpoise correction.
Job 1 - compute energy of the molecular crystal.
$end

$lattice
3 ! dimension
5.49707328000000 -0.00000000000000 -0.00000000000000
0.00000000000000 5.49707328000000 0.00000000000000
0.00000000000000 -0.00000000000000 5.49707328000000
$end

$unitcell
absolute
0 1
C 0.0000000000 0.0000000000 0.0000000000
C 2.7485366400 0.0000000000 2.7485366400
C 2.7485366400 2.7485366400 0.0000000000
C 0.0000000000 2.7485366400 2.7485366400
O 0.6735947586 0.6735947586 0.6735947586
O 2.0749418814 4.8234785214 3.4221313986
O 3.4221313986 2.0749418814 4.8234785214
O 4.8234785214 3.4221313986 2.0749418814
O 4.8234785214 4.8234785214 4.8234785214
O 3.4221313986 0.6735947586 2.0749418814
O 2.0749418814 3.4221313986 0.6735947586
O 0.6735947586 2.0749418814 3.4221313986
$end


$rem
    JOBTYPE     sp          ! Energy calculation
    METHOD      pbe         ! PBE functional
    BASIS       szv-gth     ! double zeta basis
    KECUT       1000        ! planewave cutoff 1000 eV
    PSEUDO      gth-pbe     ! use pbe gth pseudopotential
$end


@@@

$comment
Compute cohesive energy of a co2 molecular crystal using counterpoise correction.
Job 2 - compute energy of the CO2 molecule.
NOTE: in reality this should be done with a molecular job.
$end

$lattice
3 ! dimension
5.49707328000000 -0.00000000000000 -0.00000000000000
0.00000000000000 5.49707328000000 0.00000000000000
0.00000000000000 -0.00000000000000 5.49707328000000
$end

$unitcell
absolute
0 1
C 0.0000000000 0.0000000000 0.0000000000
@C 2.7485366400 0.0000000000 2.7485366400
@C 2.7485366400 2.7485366400 0.0000000000
@C 0.0000000000 2.7485366400 2.7485366400
O 0.6735947586 0.6735947586 0.6735947586
@O 2.0749418814 4.8234785214 3.4221313986
```

7.8 Pseudopotentials

7.8.1 Introduction & Overview

The GPW-based approach used by QC-PBC uses a set of plane waves as an auxiliary basis set to expand the atomic orbitals via FFT. An important consequence of this approach is that core orbitals (which are highly localized) must be removed, as they would require a prohibitively large number of plane waves to accurately reproduce. To address this, GPW-based calculations are run with pseudopotentials.

Core electrons are removed from the calculation, their effect on additional electrons replaced by an empirically fit one electron potential. Running a pseudopotential calculation therefore requires two ingredients:

1. An empirically fit function representing the effect of core electrons on the valence electrons.
2. A basis set where the core basis functions are removed.

A successful pseudopotential therefore requires a combination of a basis set and a pseudopotential that accurately represents the effect of the missing orbitals.

QC-PBC currently supports the use of Goedecker, Tetter, and Hutter (GTH) pseudopotentials. Commonly used GTH pseudopotentials are built-in, while custom pseudopotentials may be parameterized in the input file.

7.8.2 GTH Pseudopotentials

Goedecker, Tetter, and Hutter (GTH) pseudopotentials are widely used among periodic quantum chemistry codes due to their combination of reasonable accuracy and high speed. In the GTH framework, the effect of the core electrons is decomposed into a local and a nonlocal component:

$$\hat{V}_{PP} = V_{loc}(r) + \hat{V}_{\text{nl}} \tag{7.4}$$

The local part of the pseudopotential is given in terms of the effective nuclear charge Z_{eff} (The nuclear charge minus the number of removed core electrons), a parameter specifying the range of the local pseudopotential r_{loc}, and a number of coefficients C_i^{loc} specifying the contraction of the local components.

$$V_{\text{loc}}(r) = -\frac{Z_{\text{eff}}}{r}\text{erf}\left[\frac{r}{\sqrt{2}r_{\text{loc}}}\right] + \exp\left[-\frac{r^2}{2r_{\text{loc}}^2}\right]\sum_i^{N_{\text{loc}}} C_i^{loc}\left(\frac{r}{r_{\text{loc}}}\right)^{2i-2} \tag{7.5}$$

The nonlocal portion of the pseudopotential is a sum over projectors onto gaussian-type orbital functions $p_{lm}^i(\mathbf{r})$, parameterized by a set of coefficients h_{ij}^l.

$$\hat{V}_{\text{nl}} = \sum_{lm}^{N_l^{\text{nl}}}\sum_{ij}^{N_{\text{nl}}(l)} |p_{lm}^i\rangle h_{ij}^l \langle p_{lm}^j| \tag{7.6}$$

The guassian-type projectors are parameterized via another radius r_l^{nl}.

$$p_{lm}^i(\mathbf{r}) = N_i^l Y_{lm}(\hat{r}) r^{l+2i-2}\exp\left[\frac{1}{2}\left(\frac{r}{r_l^{\text{nl}}}\right)^2\right] \tag{7.7}$$

A GTH-style pseudopotential may therefore be specificied via the following parameters:

- Local
  1. $\mathbf{Z}_{\text{eff}}$ - the effective nuclear charge determined by the number of core electrons replaced.
  2. $\mathbf{r}_{\text{loc}}$ - the range of the local pseudopotential.
  3. $\mathbf{N}_{\text{loc}}$ - the number of components in the local pseudopotential.
  4. $\mathbf{C}_{\mathbf{i}}^{\text{loc}}$ - contraction coefficients for each component of the local pseudopotential.
- Nonlocal
  1. $\mathbf{N}_{\mathbf{l}}^{\text{nl}}$ - The maximum angular momentum to use in the gaussian-type projectors
  2. $\mathbf{N}_{\text{nl}}(l)$ - The number of gaussian-type projectors to use for each angular momentum
  3. $\mathbf{h}_{\mathbf{ij}}^{\mathbf{l}}$ - The contraction coefficients for each component of the nonlocal pseudopotential
  4. $\mathbf{r}_{\mathbf{l}}^{\text{nl}}$ - The radius of the components with angular momentum l of the nonlocal pseudopotential.

QC-PBC contains several standard sets of these parameters for many atoms which may be specified via a simple rem variable. Additionally, QC-PBC allows the user to specify a set of these parameters in the input file.

7.8.3 Built-In GTH Pseudopotentials

7.8.3.1 List of Available Pseudopotentials.

QC-PBC is equipped with several standard GTH pseudopotential that may be used by simply specifying their name with the PSEUDO keyword in the *$rem* block. The built-in pseudopotentials and their available elements are listed below:

Pseudopotential	**Core**	**Elements**
GTH-LDA	Small	H–Rn
GTH-BLYP	Small	H–Kr, Sr–Zr, Mo, Ru–Ag, In, Sb–I, Ba, Ce, Gd, W, Au, Bi
GTH-OLYP	Small	H, O
GTH-PADE	Small	H–Rn
GTH-BP	Small	H–Kr, Zr, Ru, Te, Cs
GTH-PBE	Small	H–Rn
GTH-HF-REV	Small	H–Kr
GTH-HCTH120	Small	H, C–F, P, Ar
GTH-HCTH407	Small	H,C–O

Table 7.9: Supported elements for the GTH pseudopotentials.

The following *$rem* variable controls which pseudopotential is used:

PSEUDO

Defines the pseudopotential to be used

TYPE:

STRING

DEFAULT:

No pseudo

OPTIONS:

General, Gen	User defined. (*$pseudo* section required)
Symbol	Use standard pseudopotentials discussed above.

RECOMMENDATION:

Pseudopotentials should be used whenever a GPW calculation is performed.

7.8.3.2 Examples

Example 7.2 Input for a PBE calculation on SiC using a non-included pseudopotential - in this case a newly parameterized GTH-HF pseudopotential.

```
$comment
use newly parameterized GTH-HF pseudopotentials from https://github.com/juerghutter/GTH
$end

$lattice
3 ! dimension
2.66626621000000 0.00000000000000 1.53936850000000
0.88875474000000 2.51377858000000 1.53936850000000
0.00000000000000 0.00000000000000 3.07873800000000
$end

$unitcell
absolute
0 1
Si 0.8887552375 0.6284446450 1.5393687500
C 0.0000000000 0.0000000000 0.0000000000
$end

$rem
    JOBTYPE     sp          ! Energy calculation
    METHOD      pbe         ! PBE functional
    BASIS       szv-gth     ! double zeta basis
    KECUT       1000        ! planewave cutoff 1000 eV
    PSEUDO      gen         ! use custom pseudopotential
$end

$pseudo
Si
    2    2    0    0
    0.44576081273929       2   -6.12039571332320    0.03404437454348
       2
    0.43461677430244       2    8.96541315822458   -2.70628585008445
                                                     3.49727517789453
    0.49929239145080       1    2.43776178627916
****
C
    2    2    0    0
    0.34816792458406       2   -8.54312820557867    1.33276540541946
       1
    0.30230247000627       1    9.59710582360109
$end
```

View output online

7.8.4 User-Defined Pseudopotentials

While QC-PBC contains several standard pseudopotentials that cover most general use cases, some users may require the use of a non-included pseudopotential, or even a custom pseudopotential. In this case, users may input the pseudopotential in the input file in much the same way as a basis set (see section 7.4.2.

To use a user-defined PP, you must set the PSEUDO to GEN, then add a *$pseudo* block that defines the pseudopotential for each element desired, separated by asterisks. The syntax of the *$pseudo* block is defined below:

$pseudo

```
****
```

E_0

N_{val}^{s} $(N_{\text{val}}^{p}$ N_{val}^{d} $\dots)$

r_{loc} N_{loc} C_0^{loc} C_1^{loc} $C_{N_{\text{loc}}-1}^{\text{loc}}$

N_l^{nl}

r_s^{nl} $N_{\text{nl}}(s)$ h_{00} h_{01} $\dots$ $h_{0(N_{\text{nl}}(s)-1)}$

h_{11} $\dots$

$\ddots$

$h_{(N_{\text{nl}}(s)-1),(N_{\text{nl}}(s)-1)}$

r_p^{nl} $N_{\text{nl}}(p)$ h_{00} h_{01} $\dots$ $h_{0(N_{\text{nl}}(s)-1)}$

$\vdots$

$****$

E_1

$\vdots$

$end

Note:

1. All of the information in the *$pseudo* block is case-insensitive.
2. The power of r (which includes the Jacobian r^2 factor) must be 0, 1, or 2.
3. If an r^0 or r^1 term is included you must include the rem keyword "ECP_FIT = TRUE".

7.8.5 Forces and Vibrational Frequencies with Pseudopotentials

QC-PBC contains analytical first and second derivatives of GTH pseudopotentials, allowing the calculation of forces/geometry optimization as well as vibrational frequencies using pseudopotentials.

References and Further Reading

[1] Basis sets were obtained from the Extensible Computational Chemistry Environment Basis Set Database, Version 1.0, as developed and distributed by the Molecular Science Computing Facility, Environmental and Molecular Sciences Laboratory which is part of the Pacific Northwest Laboratory, P.O. Box 999, Richland, Washington 99352, USA, and funded by the U.S. Department of Energy. The Pacific Northwest Laboratory is a multiprogram laboratory operated by Battelle Memorial Institute for the U.S. Department of Energy under contract DE-AC06-76RLO 1830. Contact David Feller, Karen Schuchardt or Don Jones for further information.

[2] N. B. Balabanov and K. A. Peterson. *J. Chem. Phys.*, 123:064107, 2005. DOI: 10.1063/1.1998907.

[3] J. S. Binkley, J. A. Pople, and W. J. Hehre. *J. Am. Chem. Soc.*, 102:939, 1980. DOI: 10.1021/ja00523a008.

[4] S. F. Boys and F. Bernardi. *Mol. Phys.*, 19:553, 1970. DOI: 10.1080/00268977000101561.

[5] E. R. Davidson. *Chem. Phys. Lett.*, 260:514, 1996. DOI: 10.1016/0009-2614(96)00917-7.

[6] E. R. Davidson and D. Feller. *Chem. Rev.*, 86:681, 1986. DOI: 10.1021/cr00074a002.

[7] E. V. R. de Castro and F. E. Jorge. *J. Chem. Phys.*, 108:5225, 1998. DOI: 10.1063/1.475959.

[8] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 7:359, 1986. DOI: 10.1002/jcc.540070216.

[9] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 8:861, 1987. DOI: 10.1002/jcc.540080614.

[10] K. D. Dobbs and W. J. Hehre. *J. Comput. Chem.*, 8:880, 1987. DOI: 10.1002/jcc.540080615.

[11] T. H. Dunning Jr. *J. Chem. Phys.*, 55:716, 1971. DOI: 10.1063/1.1676139.

[12] T. H. Dunning Jr. *J. Chem. Phys.*, 90:1007, 1989. DOI: 10.1063/1.456153.

[13] D. Feller and E. R. Davidson. In K. B. Lipkowitz and D. B. Boyd, editors, *Reviews in Computational Chemistry*, volume 1, page 1. Wiley-VCH, New York, 1990. DOI: 10.1002/9780470125786.ch1.

[14] E. D. Glendening and D. Feller. *J. Phys. Chem.*, 99:3060, 1995. DOI: 10.1021/j100010a015.

[15] M. S. Gordon, J. S. Binkley, J. A. Pople, W. J. Pietro, and W. J. Hehre. *J. Am. Chem. Soc.*, 104:2797, 1982. DOI: 10.1021/ja00374a017.

[16] M. Gray and J. M. Herbert. *J. Chem. Theory Comput.*, 18:2308, 2022. DOI: 10.1021/acs.jctc.1c01302.

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

[18] J. G. Hill and K. A. Peterson. *J. Chem. Phys.*, 147:244106, 2017. DOI: 10.1063/1.5010587.

[19] S. Huzinaga. *Comp. Phys. Rep.*, 2:281, 1985. DOI: 10.1016/0167-7977(85)90003-6.

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

[21] F. Jensen. *J. Chem. Theory Comput.*, 4:719, 2008. DOI: 10.1021/ct800013z.

[22] F. Jensen. *Theor. Chem. Acc.*, 126:371, 2010. DOI: 10.1007/s00214-009-0699-5.

[23] F. Jensen. *J. Chem. Theory Comput.*, 10:1074, 2014. DOI: 10.1021/ct401026a.

[24] E. Papajak and D. G. Truhlar. *J. Chem. Theory Comput.*, 7:10, 2011. DOI: 10.1021/ct1005533.

[25] K. A. Peterson and C. Puzzarini. *Theor. Chem. Acc.*, 114:283, 2005. DOI: 10.1007/s00214-005-0681-9.

[26] K. A. Peterson, D. Figgen, E. Goll, H. Stoll, and M. Dolg. *J. Chem. Phys.*, 119:11113, 2003. DOI: 10.1063/1.1622924.

[27] B. P. Prascher, D. E. Woon, K. A. Peterson, T. H. Dunning Jr., and A. K. Wilson. *Theor. Chem. Acc.*, 128:69, 2011. DOI: 10.1007/s00214-010-0764-0.

[28] B. P. Pritchard, D. Altarawy, B. Didier, T. D. Gibson, and T. L. Windus. *J. Chem. Inf. Model*, 59:4814, 2019. DOI: 10.1021/acs.jcim.9b00725.

[29] D. Rappoport and F. Furche. *J. Chem. Phys.*, 133:134105, 2010. DOI: 10.1063/1.3484283.

[30] R. M. Richard, K. U. Lao, and J. M. Herbert. *J. Phys. Chem. Lett.*, 4:2674, 2013. DOI: 10.1021/jz401368u.

[31] A. Schäfer, H. Horn, and R. Ahlrichs. *J. Chem. Phys.*, 97:2571, 1992. DOI: 10.1063/1.463096.

[32] F. Weigend and R. Ahlrichs. *Phys. Chem. Chem. Phys.*, 7:3297, 2005. DOI: 10.1039/b508541a.

[33] A. K. Wilson, D. E. Woon, K. A. Peterson, and T. H. Dunning Jr. *J. Chem. Phys.*, 110:7667, 1999. DOI: 10.1063/1.478678.

[34] D. E. Woon and T. H. Dunning Jr. *J. Chem. Phys.*, 98:1358, 1993. DOI: 10.1063/1.464303.

[35] Hong-Zhou Ye and Timothy C Berkelbach. Correlation-consistent gaussian basis sets for solids made simple. *J. Chem. Theory Comput.*, 18(3):1595–1606, 2022. DOI: 10.1021/acs.jctc.1c01245.

[36] E. Papajak J. Zheng, X. Xu, H. R. Leverentz, and D. G. Truhlar. *J. Chem. Theory Comput.*, 7:3027, 2011. DOI: 10.1021/ct200106a.

Chapter 8

Equilibrium Geometry Optimization

8.1 Equilibrium Geometries with QC-PBC

8.1.1 Introduction

Potential energy surfaces rely on the Born-Oppenheimer separation of nuclear and electronic motion. Of particular interest are the critical points on these surfaces, *i.e.* where the gradient of the energy vanishes. Characterization of a critical point requires consideration of the eigenvalues of the Hessian (second derivative matrix) calculated at that point. An equilibrium geometry corresponds to a critical point where the eigenvalues of the Hessian are all positive, whereas a transition-state structure is defined as a first-order saddle point, and therefore has a Hessian with precisely one negative eigenvalue. The latter is a local maximum along the reaction path between the minima corresponding to the reactants and products, and a minimum in all directions perpendicular to this reaction path.

The quality of a geometry optimization algorithm is of major importance; even the fastest integral code in the world will be useless if combined with an inefficient optimization algorithm that requires excessive numbers of steps to converge. QC-PBC utilizes Q-Chem's new library, LIBOPT3, which improves on the older OPTIMIZE driver.

The key to optimizing a 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 (including lattice vectors)
- coordinate system
- optimization algorithm
- quality of the Hessian (and gradient)

QC-PBC controls the last three of these, but the starting geometry is solely determined by the user, and the closer it is to the converged geometry, the fewer optimization steps will be required. Decisions regarding the optimization algorithm and the coordinate system are generally made by the LIBOPT3 drivers (*i.e.*, internally, within QC-PBC) to maximize the rate of convergence. Although users may override these choices, caution should be exercised when doing so as changes may significantly impact the computational cost.

QC-PBC provides the capability to optimize a structure using Cartesian or delocalized periodic internal coordinates. Delocalized periodic internal coordinates are generated automatically from the Cartesian coordinates, but do not necessarily offer speedup over Cartesian coordinates. QC-PBC's internal coordinate implementation follows the work of Kudin et al.[8]

The heart of the geometry optimization in QC-PBC is Baker's eigenvector-following (EF) algorithm.[1] This was developed following the work of Cerjan and Miller,[6] and of Simons and co-workers.[5,13] The Hessian mode-following option incorporated into this algorithm is capable of locating a transition state by walking uphill from the associated minimum. By following the lowest Hessian mode, the EF algorithm can locate a transition state starting from any reasonable input geometry and Hessian.

Another consideration when trying to minimize the total optimization time concerns the quality of the gradient and Hessian. A higher-quality Hessian (*i.e.*, analytical versus approximate) will in many cases lead to faster convergence, in the sense of requiring fewer optimization steps. However, the construction of an analytical Hessian (especially for periodic systems) requires significant computational effort and may outweigh the advantage of fewer optimization cycles.

Features of QC-PBC's geometry optimization capabilities include:

- Cartesian or internal coordinate systems
- Lattice vector optimization
- Eigenvector Following (EF) algorithms
- Equilibrium structure searches
- Hessian-free characterization of stationary points
- Initial Hessian and Hessian update options

8.1.2 Theoretical Background

Geometry optimization refers to the determination of stationary points, principally minima and transition states, on molecular potential energy surfaces. It is an iterative process, requiring the calculation of the energy, gradient and (possibly) Hessian at each optimization cycle. The energy, gradient and Hessian information is used to compute an optimization step, $\mathbf{h}$, which displaces the geometry to be closer to the target stationary point. This process is repeated until convergence is attained. The art of geometry optimization lies in calculating the step, $\mathbf{h}$, so as to converge in as few cycles as possible.

Consider the energy, $E(\mathbf{x}_0)$ at some point $\mathbf{x}_0$ on a potential energy surface. We can express the energy at a nearby point $\mathbf{x} = \mathbf{x}_0 + \mathbf{h}$ by means of the Taylor series

$$E(\mathbf{x}_0 + \mathbf{h}) = E(\mathbf{x}_0) + \mathbf{h}^t \left(\frac{dE(\mathbf{x}_0)}{d\mathbf{x}} \right) + \frac{1}{2}\mathbf{h}^t \left(\frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2} \right) \mathbf{h} + \cdots \tag{8.1}$$

If we knew the exact form of the energy functional $E(\mathbf{x})$ and all its derivatives, we could move from the current point $\mathbf{x}_0$ directly to a stationary point, (*i.e.*, we would know exactly what the step $\mathbf{h}$ ought to be). Since we typically know only the lower derivatives of $E(\mathbf{x})$ at best, then we can estimate the step $\mathbf{h}$ by differentiating the Taylor series with respect to $\mathbf{h}$, keeping only the first few terms on the right hand side, and setting the left hand side, $dE(\mathbf{x}_0 + \mathbf{h})/d\mathbf{h}$, to zero, which is the value it would have at a genuine stationary point. Thus

$$\frac{dE(\mathbf{x}_0 + \mathbf{h})}{d\mathbf{h}} = \frac{dE(\mathbf{x}_0)}{d\mathbf{x}} + \left(\frac{d^2E(\mathbf{x}_0)}{d\mathbf{x}_1 d\mathbf{x}_2} \right) \mathbf{h} + \text{higher terms (ignored)} \tag{8.2}$$

from which

$$\mathbf{h} = -\mathbf{H}^{-1}\mathbf{g} \tag{8.3}$$

where

$$\frac{dE}{d\mathbf{x}} \equiv \mathbf{g} \text{ (gradient vector)}, \qquad \frac{d^2E}{d\mathbf{x}_1 d\mathbf{x}_2} \equiv \mathbf{H} \text{ (Hessian matrix)} \tag{8.4}$$

Equation (8.3) is known as the Newton-Raphson step. It is the major component of almost all geometry optimization algorithms in quantum chemistry.

The above derivation assumed exact first (gradient) and second (Hessian) derivative information. Analytical gradients are available for most SCF methodologies supported in Q-CHEM, however, analytical second derivatives are less likely to be available. Furthermore, even if they were available, it would not necessarily be advantageous to use them as their evaluation is computationally demanding, and efficient optimizations can be performed without an exact Hessian. An excellent compromise in practice is to begin with an approximate Hessian matrix, and update this using gradient and displacement information generated as the optimization progresses. In this way the starting Hessian can be "improved" at essentially no cost. Using Eq. (8.3) with an approximate Hessian is called the quasi Newton-Raphson step.

The nature of the Hessian matrix (in particular its eigenvalue structure) plays a crucial role in a successful optimization. All stationary points on a potential energy surface have a zero gradient vector, however, the character of the stationary point (*i.e.*, what type of structure it corresponds to) is determined by the Hessian matrix. Diagonalization of the Hessian yields a set of mutually orthogonal directions on the energy surface (the eigenvectors) together with the curvature along these directions (the eigenvalues). At a local minimum (corresponding to a well in the potential energy surface) the curvature along all of these directions must be positive, reflecting the fact that a small displacement along any of these directions causes the energy to rise. At a transition state, the curvature is negative (*i.e.*, the energy is a maximum) along one direction, but positive along all other directions. Thus, for a stationary point to be a transition state, the Hessian matrix at that point must have one, and only one, negative eigenvalue, while for a minimum the Hessian must have all positive eigenvalues. In the latter case the Hessian is called *positive definite*. If searching for a minimum it is important that the Hessian matrix be positive definite; in fact, unless the Hessian is positive definite there is no guarantee that the step predicted by Eq. (8.3) is will actually lower the energy. Similarly, for a transition state search, the Hessian must have one negative eigenvalue. Maintaining the correct Hessian eigenvalue structure is not difficult for minimization, but can be problematic when searching for a transition state.

In a diagonal Hessian representation the Newton-Raphson step can be written

$$\mathbf{h} = -\sum_i \left(\frac{F_i}{b_i}\right) \mathbf{u}_i \tag{8.5}$$

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.*,[13] the Newton-Raphson step can be considered as minimizing along directions $\mathbf{u}_i$, which have positive eigenvalues, and maximizing along directions with negative eigenvalues. Thus, if the user is searching for a minimum and the Hessian matrix is positive definite, then the Newton-Raphson step is appropriate since it is attempting to minimize along all directions simultaneously. However, if the Hessian has one or more negative eigenvalues, then the basic Newton-Raphson step is not appropriate for a minimum search, since it will be maximizing and not minimizing along one or more directions. Exactly the same arguments apply during a transition state search except that the Hessian must have one negative eigenvalue, because the user has to maximize along one direction. However, there must be *only* one negative eigenvalue. A positive definite Hessian is a disaster for a transition state search because the Newton-Raphson step will head towards a minimum.

If firmly in a region of the potential energy surface with the right Hessian character, then a careful search (based on the Newton-Raphson step) will almost always lead to a stationary point of the correct type. However, this is only true if the Hessian is exact. If the Hessian is being updated approximately, then there is no guarantee that the Hessian eigenvalue structure will be preserved from one cycle to the next unless one is very careful during the update. Updating procedures that "guarantee" conservation of a positive definite Hessian do exist (or at least warn the user if the update is likely to introduce negative eigenvalues). This can be very useful during a minimum search; but there are no such guarantees for preserving the Hessian character required for a transition state.

In addition to the difficulties in retaining the correct Hessian character, there is the matter of obtaining a "correct" Hessian in the first instance. This is particularly acute for a transition state search. For a minimum search it is possible to "guess" a reasonable, positive-definite starting Hessian (for example, by carrying out a molecular mechanics

minimization and calculating the Hessian, also using molecular mechanics, at the minimum) but this option is usually not available for transition states. Even if the user calculates the Hessian exactly at the starting geometry, the guess for the structure may not be sufficiently accurate, and the expensive, exact Hessian may not have the desired eigenvalue structure.

Consequently, particularly for a transition state search, an alternative to the basic Newton-Raphson step is clearly needed, especially when the Hessian matrix is inappropriate for the stationary point being sought.

One of the first algorithms that was capable of taking corrective action during a transition state search if the Hessian had the wrong eigenvalue structure, was developed by Poppinger,[9] 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.

8.1.3 Eigenvector-Following (EF) Algorithm

The work of Cerjan and Miller,[6] and later Simons and co-workers,[5,13] showed that there was a better step than simply directly following one of the Hessian eigenvectors. A simple modification to the Newton-Raphson step is capable of guiding the search away from the current region towards a stationary point with the required characteristics. This is

$$\mathbf{h} = -\sum_i \left(\frac{F_i}{b_i - \lambda} \right) \mathbf{u}_i \tag{8.6}$$

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[9]), along one of the local eigenmodes, depending on the value chosen for λ. References 5,6,13 all use the same basic approach of Eq. (8.6) but differ in the means of determining the value of λ.

The EF algorithm[1] uses the rational function approach presented in Refs. 5, yielding an eigenvalue equation of the form

$$\begin{pmatrix} \mathbf{H} & \mathbf{g} \\ \mathbf{g}^t & 0 \end{pmatrix} \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} = \lambda \begin{pmatrix} \mathbf{h} \\ 1 \end{pmatrix} \tag{8.7}$$

from which a suitable λ can be obtained. Expanding Eq. (8.7) yields

$$(\mathbf{H} - \lambda)\mathbf{h} + \mathbf{g} = 0 \tag{8.8}$$

and

$$\mathbf{g}^t \mathbf{h} = \lambda \tag{8.9}$$

In terms of a diagonal Hessian representation, Eq. (8.8) rearranges to Eq. (8.6), and substitution of Eq. (8.6) into the diagonal form of Eq. (8.9) gives

$$\lambda = -\sum_i \left(\frac{-F_i^2}{b_i - \lambda} \right) \tag{8.10}$$

which can be used to evaluate λ iteratively.

The eigenvalues, λ, of the RFO equation Eq. (8.7) have the following important properties:[5]

- 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. (8.7) 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

$$\begin{pmatrix} b_1 & F_1 \\ F_1 & 0 \end{pmatrix} \begin{pmatrix} h_1 \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_1 \\ 1 \end{pmatrix} \tag{8.11}$$

$$\begin{pmatrix} b_2 & & & F_2 \\ & \ddots & \mathbf{0} & \vdots \\ & \mathbf{0} & b_n & F_n \\ F_2 & \cdots & F_n & 0 \end{pmatrix} \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} = \lambda_n \begin{pmatrix} h_2 \\ \vdots \\ h_n \\ 1 \end{pmatrix} \tag{8.12}$$

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. (8.11), which is always positive and approaches zero at convergence, and λ_n is the lowest eigenvalue of Eq. (8.12), 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. (8.6) to give the final step

$$\mathbf{h} = -\left(\frac{F_1}{b_1 - \lambda_p}\right)\mathbf{u}_1 + \sum_{i=2}^{n}\left(\frac{F_i}{b_i - \lambda_n}\right)\mathbf{u}_i \tag{8.13}$$

If this step is greater than the maximum allowed, it is scaled down. For minimization only one shift parameter, λ_n, is used which acts on all modes.

In Eq. (8.11)) and Eq. (8.12) it was assumed that the step would maximize along the lowest Hessian mode, b_1, and minimize along all the higher modes. However, it is possible to maximize along modes other than the lowest, and in this way potentially locate transition states for alternative rearrangements/dissociations from the same initial starting point. For maximization along the kth mode (instead of the lowest mode), Eq. (8.11) is replaced by

$$\begin{pmatrix} b_k & F_k \\ F_k & 0 \end{pmatrix} \begin{pmatrix} h_k \\ 1 \end{pmatrix} = \lambda_p \begin{pmatrix} h_k \\ 1 \end{pmatrix} \tag{8.14}$$

and Eq. (8.12) now excludes the kth mode, but includes the lowest mode. Since what was originally the kth mode is the mode along which the negative eigenvalue is required, then this mode will eventually become the lowest mode at some stage of the optimization. To ensure that the original mode is being followed smoothly from one cycle to the next, the mode that is actually followed is the one with the greatest overlap with the mode followed on the previous cycle. This procedure is known as *mode following*. For more details and some examples, see Ref. 1.

8.1.4 Delocalized Internal Coordinates

The choice of coordinate system can have a major influence on the rate of convergence during a geometry optimization. For complex potential energy surfaces with many stationary points, a different choice of coordinates can even result in convergence to a different final structure.

The key attribute of a good set of coordinates for geometry optimization is the degree of coupling between the individual coordinates. In general, the less coupling the better, as variation of one particular coordinate will then have minimal

impact on the other coordinates. Coupling manifests itself primarily as relatively large partial derivative terms between different coordinates. For example, a strong harmonic coupling between two different coordinates, i and j, results in a large off-diagonal element, H_{ij}, in the Hessian matrix. Normally this is the only type of coupling that can be directly "observed" during an optimization, as third and higher derivatives are ignored in almost all optimization algorithms.

In the early days of computational quantum chemistry geometry optimizations were carried out in Cartesian coordinates. They are an obvious choice as they can be defined for all systems and gradients and second derivatives are calculated directly in Cartesian coordinates. Unfortunately, Cartesian coordinates are often heavily coupled, making them a poor choice for optimizations. Despite this, Cartesians have recently returned to favor because of their generality, and because it has been clearly demonstrated that if reliable second derivative information is available (*i.e.*, a good starting Hessian) and the initial geometry is reasonable, then Cartesians can be as efficient as any other coordinate set for small to medium-sized molecules.[2,3] Without good Hessian data, however, Cartesians are inefficient, especially for long chain acyclic systems.

In the 1970s Cartesians were replaced by *Z*-matrix coordinates. Initially the *Z*-matrix was used simply as a means of geometry input; it is far easier to describe a molecule in terms of bond lengths, bond angles and dihedral angles (the natural coordinates for a chemist) than to develop a suitable set of Cartesian coordinates. It was subsequently found that optimization was generally more efficient in *Z*-matrix coordinates than in Cartesians, especially for acyclic systems. This is not always the case, and care must be taken in constructing a suitable *Z*-matrix. A good general rule is ensure that each variable is defined in such a way that changing its value will not change the values of any of the other variables. A brief discussion concerning good *Z*-matrix construction strategy is given by Schlegel.[12]

In 1979 Pulay *et al.* published a key paper introducing what were termed natural internal coordinates into geometry optimization.[11] These coordinates involve the use of individual bond displacements as stretching coordinates, but linear combinations of bond angles and torsions as deformational coordinates. Suitable linear combinations of bends and torsions (the two are considered separately) are selected using group theoretical arguments based on local pseudo-symmetry. For example, bond angles around an sp^3 hybridized carbon atom are all approximately tetrahedral, regardless of the groups attached, and idealized tetrahedral symmetry can be used to generate deformational coordinates around the central carbon atom.

The major advantage of natural internal coordinates in geometry optimization is their ability to significantly reduce the coupling, both harmonic and anharmonic, between the various coordinates. Compared to natural internals, *Z*-matrix coordinates arbitrarily omit some angles and torsions (to prevent redundancy), and this can induce strong anharmonic coupling between the coordinates, especially with a poorly constructed *Z*-matrix. Another advantage of the reduced coupling is that successful minimizations can be carried out in natural internals with only an approximate (*e.g.*, diagonal) Hessian provided at the starting geometry. A good starting Hessian is still needed for a transition state search.

Despite their clear advantages, natural internals have only become widely used more recently. This is because, when used in the early programs, it was necessary for the user to define them. This situation changed in 1992 with the development of computational algorithms capable of automatically generating natural internals from input Cartesians.[7] For minimization, natural internals have become the coordinates of first choice.[2,7]

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.*[4] developed a set of delocalized internal coordinates that eliminate all of the above-mentioned difficulties. Building on some of the ideas in the redundant optimization scheme of Pulay and Fogarasi,[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 Wilson $\mathbf{B}$-matrix,[14]

$$\Delta\mathbf{q} = \mathbf{B}\Delta\mathbf{x} \tag{8.15}$$

If any of the internal coordinates $\mathbf{q}$ are redundant, then the rows of the $\mathbf{B}$-matrix will be linearly dependent.

Delocalized internal coordinates are obtained by constructing and diagonalizing the matrix $\mathbf{G} = \mathbf{B}\mathbf{B}^t$. Diagonalization of $\mathbf{G}$ results in two sets of eigenvectors; a set of m (typically $3N - 6$, where N is the number of atoms) eigenvectors with eigenvalues $\lambda > 0$, and a set of nm eigenvectors with eigenvalues $\lambda = 0$ (to numerical precision). In this way, any redundancies present in the original coordinate set $\mathbf{q}$ are isolated (they correspond to those eigenvectors with zero eigenvalues). The eigenvalue equation of $\mathbf{G}$ can thus be written

$$\mathbf{G}(\mathbf{UR}) = (\mathbf{UR}) \begin{pmatrix} \Lambda & 0 \\ 0 & 0 \end{pmatrix} \tag{8.16}$$

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. (8.16) 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.[4]

8.2 Geometry Optimization Job Controls

8.2.1 Job Control Overview

Obviously a level of theory, basis set, and starting molecular geometry must be specified to begin a geometry optimization or transition-structure search. These aspects are described elsewhere in this manual, and this section describes job-control variables specific to optimizations.

The job controls for geometry optimization have the general rem variable pattern, GEOM_OPT. LIBOPT3 also offers more fine-grained controls via the *$geom_opt* section, see Section 8.2.3. The *$geom_opt* section is not required however, as the GEOM_OPT variables can be read within the LIBOPT3 driver if they have similar matching variables/controls. The decreasing order of precedence for setting of rem variables are as follows: 1) *$geom_opt* variables, 2) GEOM_OPT variables, 3) Default.

JOBTYPE

Specifies the calculation.

TYPE:

STRING

DEFAULT:

Default is single-point, which should be changed to one of the following options.

OPTIONS:

OPT Equilibrium structure optimization.

RECOMMENDATION:

Application-dependent.

8.2.2 *$rem* Job Control

GEOM_OPT_HESSIAN

Determines the initial Hessian status.

TYPE:

STRING/INTEGER

DEFAULT:

-1

OPTIONS:

-1 Approximate Hessian based on the force constant matrix (same as the "Model" option in LIBOPT3)

READ Have exact or initial Hessian. Use as is if Cartesian, or transform if internals.

RECOMMENDATION:

An accurate initial Hessian will improve the performance of the optimizer, but it may be expensive to compute. Note that more options for the initial Hessian are available in LIBOPT3, controlled by INITIAL_HESSIAN under the *$geom_opt* section (see Sec. 8.2.3).

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.

RECOMMENDATION:

Use cartesian, as these are currently more robust.

GEOM_OPT_TOL_GRADIENT

Convergence on maximum gradient component.

TYPE:

INTEGER

DEFAULT:

300 $\equiv 300 \times 10^{-6}$ tolerance on maximum gradient component.

OPTIONS:

n Integer value (tolerance = $n \times 10^{-6}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_TOL_DISPLACEMENT

Convergence on maximum atomic displacement.

TYPE:

INTEGER

DEFAULT:

$1200 \equiv 1200 \times 10^{-6}$ tolerance on maximum atomic displacement.

OPTIONS:

n Integer value (tolerance = $n \times 10^{-6}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_TOL_ENERGY

Convergence on energy change of successive optimization cycles.

TYPE:

INTEGER

DEFAULT:

$100 \equiv 100 \times 10^{-8}$ tolerance on maximum (absolute) energy change.

OPTIONS:

n Integer value (tolerance = value $n \times 10^{-8}$).

RECOMMENDATION:

Use the default. To converge GEOM_OPT_TOL_GRADIENT and one of GEOM_OPT_TOL_DISPLACEMENT and GEOM_OPT_TOL_ENERGY must be satisfied.

GEOM_OPT_MAX_CYCLES

Maximum number of optimization cycles.

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

n User defined positive integer.

RECOMMENDATION:

The default should be sufficient for most cases. Increase if the initial guess geometry is poor, or for systems with shallow potential wells.

GEOM_OPT_PRINT

Controls the amount of OPTIMIZE print output.

TYPE:

INTEGER

DEFAULT:

3 Error messages, summary, warning, standard information and gradient print out.

OPTIONS:

0 Error messages only.

1 Level 0 plus summary and warning print out.

2 Level 1 plus standard information.

3 Level 2 plus gradient print out.

4 Level 3 plus Hessian print out.

5 Level 4 plus iterative print out.

6 Level 5 plus internal generation print out.

7 Debug print out.

RECOMMENDATION:

Use the default.

GEOM_OPT_MODE

Determines Hessian mode followed during a transition state search.

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 Mode following off.

n Maximize along mode n.

RECOMMENDATION:

Use the default, for geometry optimizations.

GEOM_OPT_DMAX

Maximum allowed step size. Value supplied is multiplied by 10^{-3}.

TYPE:

INTEGER

DEFAULT:

300 = 0.3

OPTIONS:

n User-defined cutoff.

RECOMMENDATION:

Use the default.

GEOM_OPT_UPDATE

Controls the Hessian update algorithm.

TYPE:

INTEGER

DEFAULT:

-1

OPTIONS:

-1 Use the default update algorithm.
0 Do not update the Hessian (not recommended).
1 Murtagh-Sargent update.
2 Powell update.
3 Powell/Murtagh-Sargent update (TS default).
4 BFGS update (OPT default).
5 BFGS with safeguards to ensure retention of positive definiteness (GDIIS default).

RECOMMENDATION:

Use the default.

GEOM_OPT_LINEAR_ANGLE

Threshold for near linear bond angles (degrees).

TYPE:

INTEGER

DEFAULT:

165 degrees.

OPTIONS:

n User-defined level.

RECOMMENDATION:

Use the default.

FINITEDIFFSZ_POWER

The exponent controlling the order of magnitude of the step size used for calculating derivatives by finite difference.

TYPE:

INTEGER

DEFAULT:

5 Corresponding to 10^{-5}.

OPTIONS:

n Use an exponent of 10^{-n}.

RECOMMENDATION:

None.

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 $n \times 10^{-m}$, where m is FINITEDIFFSZ_POWER.

RECOMMENDATION:

Use the default except in cases where the potential surface is very flat in which case a larger value should be used. See FDIFF_STEPSIZE_QFF for third and fourth derivatives.

8.2.3 LIBOPT3 Job Control

Within LIBOPT3 there is a separate *$geom_opt* section for geometry optimization job controls, which provide greater control then the GEOM_OPT variables. This new section contains all the following input variables for the LIBOPT3 driver.

ALGORITHM

Specifies which type of minimization algorithm to use.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

BFGS (OPT) / BOFILL (TS)

OPTIONS:

SD	Steepest Descent?
CG	Conjugate Gradient?
Newton	Exact Newton's Method?
BFGS	Broyden-Fletcher-Goldfarb-Shanno????
LBFGS	Limited-memory BFGS?
SR1	Symmetric-Rank One (Murtagh-Sargent)?
PSB	Powell symmetric Broyden?
BOFILL	Bofill combination of PSB and SR1?
FS	Farkas and Schlegel combination of SR1 and BFGS?

RECOMMENDATION:

Steepest descent and conjugate gradient methods are slow to converge in general but are useful when the near the minimum. Best to start with other algorithms and finalize with these two methods if a tighter converged minimum is needed. Newton's will be efficient but requires a Hessian evaluation at each step, so the cost of the Hessian calculation must be accounted for when using exact Newton's method. BFGS is a default algorithm for jobtype = OPT for its speed and efficiency for finding the minimum. L-BFGS is recommended when dealing with very large systems when memory is of concern. The default algorithm for jobtype = TS is Bofill.

COORDINATES

Specifies which type of coordinate system to use for optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Delocalized

OPTIONS:

Cartesian	Cartesian Coordiantes
Delocalized	Delocalized Natural Internal Coordinates

RECOMMENDATION:

Cartesian can be more stable than internal coordinates but can be slower than internal coordinates. If there are problems with internal coordinate optimization restart with Cartesian coordinates at the last known internal coordinate geometry can be controlled with OPTIMIZATION_RESTART.

MAXITER

Maximum number of geometry optimization cycles.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

50

OPTIONS:

Integer	Any positive integer

RECOMMENDATION:

None

INITIAL_HESSIAN

Specifies the type of initial Hessian to use.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Model (OPT) / EXACT (TS)

OPTIONS:

Identity	Identity Matrix
Simple	Simple Approximate Guess Hessian
Model	Model Approximate Guess Hessian
Exact	Analytical Hessian
Read	Read-in Hessian

RECOMMENDATION:

The initial guess Hessian for Cartesian coordinates is a unit matrix where for internal coordinates it is an approximate Hessian based on the internal coordinates.**?** The model approximate Hessian is based on a force constant matrix.[12]**?** The exact Hessian is always a quality initial Hessian but could be costly but can be calculated at the start of the optimization or read in from scratch. The default for OPT is the model approximate guess Hessian but for TS the Exact analytical Hessian will be used.

CONVERGENCE_CHECK

Specifies the type of convergence check during geometry optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

DEFAULT

OPTIONS:

Default	Check max absolute gradient component and (maximum absolute displacement or change in energy)
Energy	Change in energy
Gradient	Check norm of gradient

RECOMMENDATION:

None.

GRADIENT_CONVERGENCE

The value of maximum absolute gradient or norm of gradient for convergence check.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

3e-4

OPTIONS:

Float	Any positive float

RECOMMENDATION:

This variable is used as the comparison value for the gradient checked with CONVERGENCE_CHECK.

DISPLACEMENT_CONVERGENCE

The value of maximum absolute displacement for convergence check.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

1.2e-3

OPTIONS:

Float	Any positive float

RECOMMENDATION:

None.

ENERGY_CONVERGENCE

The value of maximum absolute energy difference for convergence check.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

1e-6

OPTIONS:

Float Any positive float

RECOMMENDATION:

None.

STEP_LIMITER

Specifies the type of limiter to use for adjustment of the step during geometry optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

RMS (OPT) / NORM (TS)

OPTIONS:

RMS Root mean square of the step

NORM Norm of the step

RECOMMENDATION:

Adjust the type of condition used for MAX_DISPLACEMENT.

MAX_DISPLACEMENT

The value of maximum for the STEP_LIMITER of the Eigenvector following algorithm step.

INPUT SECTION: *$geom_opt*

TYPE:

FLOAT

DEFAULT:

3e-1

OPTIONS:

Float Any positive float

RECOMMENDATION:

If this value is too large there may be trouble with the optimization and a small value could require additional optimization cycles.

RECOMPUTE_HESSIAN

Recompute the exact Hessian during optimization algorithms

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

False

OPTIONS:

None Do not recompute of Hessian

Recompute Compute Hessian during optimization

RECOMMENDATION:

Recompute the exact Hessian during optimization during BFGS, SR1, PSB, BOFILL, and FS Quasi-Newton Hessian update algorithms.

RECOMPUTE_HESSIAN_CYCLES

The number of cycles before recomputing the Hessian during optimization algorithms.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

5

OPTIONS:

Integer Any positive integer

RECOMMENDATION:

None.

STEP_SEARCH_ALGORITHM

Specifies the type of algorithm for geometry step generation.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

EFA

OPTIONS:

EFA Eigenvector Following Algorithm

LS Simple Line Search

RECOMMENDATION:

For Quasi-Newton methods the default step generation is Eigenvector following algorithm, but line search can be used if desired.

EIGENVECTOR_ALGORITHM

Specifies the type of Eigenvector algorithm.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

RFO

OPTIONS:

RFO Rational Function Optimization

PRFO Partitioned Rational Function Optimization

RECOMMENDATION:

This allows the switching of the Eigenvector following algorithm for state specific searches with PRFO.

PRINT_LEVEL

Specifies the printing verbosity of the optimizer.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

0

OPTIONS:

0 General Print

10 Verbose

RECOMMENDATION:

None.

MAX_LBGFGS_HISTORY

Specifies the number of cycles to retain for L-BFGS history.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

Integer Any positive integer

RECOMMENDATION:

It is recommended to keep this number small.

LS_PARAM

Specifies the type of line search algorithm to use.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Quasi-Newton

OPTIONS:

Quasi-Newton	Quasi-Newton
Strict	Strict line search parameters
Very_Strict	Very strict line search parameters

RECOMMENDATION:

None

LS_MAXITER

Specifies the number of maximum iterations to perform during line search step calculation.

INPUT SECTION: *$geom_opt*

TYPE:

INTEGER

DEFAULT:

10

OPTIONS:

Integer	Any positive integer

RECOMMENDATION:

None

LS_PRINT

Specifies the verbosity of printing for the line search algorithm.

INPUT SECTION: *$geom_opt*

TYPE:

String

DEFAULT:

Minimal

OPTIONS:

Minimal	Minimal printing
Verbose	Verbose printing

RECOMMENDATION:

None

HESSIAN_VERIFY

Specifies the type of verification with the Hessian after geometry optimization

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

RESULT

OPTIONS:

No_Verification	Do no verification of optimization
Result	Verify with final Hessian obtained during optimization
Without	Verify without a Hessian (Only convergence criteria)
Recomputed	Verify with recomputed exact Hessian

RECOMMENDATION:

None.

OPTIMIZATION_RESTART

Specifies if optimization should restart in Cartesian coordinates after back-transformation failure with internal coordinate optimizations.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

TRUE

OPTIONS:

True	Restart with Cartesian Coordinates
False	Do not restart

RECOMMENDATION:

Restart a failed back-transformation internal coordinate optimization job in Cartesian coordinates. This will use the current retained updated Internal coordinate Hessian transform it to Cartesian coordinates and continue the optimization from the last known position.

FINAL_ZMAT_PRINT

Controls if a *Z*-matrix is printed at the end of the job.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

TRUE

OPTIONS:

True	Construct *Z*-matrix and print
False	Do calculate or print *Z*-matrix

RECOMMENDATION:

After optimization the final structure can be used to compute and print the final *Z*-matrix. This can be turned off if molecule contains many atoms and Z-Matrix is not needed.

FINAL_VIBRATIONAL_ANALYSIS

Run Vibrational Analysis after geometry optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

False

OPTIONS:

True Perform Vibrational Analysis

False Do not compute vibrational analysis

RECOMMENDATION:

Vibrational analysis can be performed only if the final Hessian for verification was re-computed, HESSIAN_VERIFY

PRINT_TOPOLOGY

Print the topology for optimization.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

FALSE

OPTIONS:

True Print the topology

False Do not print the topology

Terminate Print the topology and terminate job

RECOMMENDATION:

Print the topology used in optimization. In addition the topology can be printed then the job terminated.

USER_TOPOLOGY

Specifies if a user provided topology is to be read.

INPUT SECTION: *$geom_opt*

TYPE:

STRING

DEFAULT:

Generated

OPTIONS:

Generated Generate the topology

Read Read a user provided topology.

RECOMMENDATION:

A user can provide a topology for a given molecule in the *$geom_opt_topology* section in the input.

A user defined topology can be read in using the USER_TOPOLOGY keyword and providing the topology in *$geom_opt_topology* section. The available topology definitions that can be used are:

1 (Bond) A-B

2 (Angle) A-C-B (Apex is B)

4 (Torsion) Torsion A-B-C-D

5 (Co-Linear Angle) use with 6 D A-B-C (ABC is Linear)

6 (Co-Linear Angle) use with 5 D A-B-C (ABC is Linear)

Note: The first line is the total number of coordinates to read. Then the following lines are the coordinate definition to be used, following the definition above: coordinate type, atom number 1, ...

```
$geom_opt_topology
Total Number of Coordinates
1   atomA  atomB !Bonds
...
2   atomA  atomC  atomB !Angles
...
4   atomA  atomB  atomC atomD !Torsions
...
5   atomD  atomA  atomB atomC !Co-linear Torsion
...
6   atomD  atomA  atomB atomC !Co-linear Torsion
$end
```

8.2.4 LIBOPT3 Job Examples

References and Further Reading

[1] J. Baker. *J. Comput. Chem.*, 7:385, 1986. DOI: 10.1002/jcc.540070402.

[2] J. Baker and D. Bergeron. *J. Comput. Chem.*, 14:1339, 1993. DOI: 10.1002/jcc.540141111.

[3] J. Baker and W. J. Hehre. *J. Comput. Chem.*, 12:606, 1991. DOI: 10.1002/jcc.540120510.

[4] J. Baker, A. Kessi, and B. Delley. *J. Chem. Phys.*, 105:192, 1996. DOI: 10.1063/1.471864.

[5] A. Banerjee, N. Adams, J. Simons, and R. Shepard. *J. Phys. Chem.*, 89:52, 1985. DOI: 10.1021/j100247a015.

[6] C. J. Cerjan and W. H. Miller. *J. Chem. Phys.*, 75:2800, 1981. DOI: 10.1063/1.442352.

[7] G. Fogarasi, X. Zhou, P. W. Taylor, and P. Pulay. *J. Am. Chem. Soc.*, 114:8191, 1992. DOI: 10.1021/ja00047a032.

[8] Konstantin N. Kudin, Gustavo E. Scuseria, and H. Bernhard Schlegel. A redundant internal coordinate algorithm for optimization of periodic systems. *The Journal of Chemical Physics*, 114(7):2919–2923, 02 2001. ISSN 0021-9606. DOI: 10.1063/1.1340578. URL https://doi.org/10.1063/1.1340578.

[9] D. Poppinger. *Chem. Phys. Lett.*, 35:550, 1975. DOI: 10.1016/0009-2614(75)85665-X.

[10] P. Pulay and G. Fogarasi. *J. Chem. Phys.*, 96:2856, 1992. DOI: 10.1063/1.462844.

[11] P. Pulay, G. Fogarasi, F. Pang, and J. E. Boggs. *J. Am. Chem. Soc.*, 101:2550, 1979. DOI: 10.1021/ja00504a009.

[12] H. B. Schlegel. *Theor. Chem. Acc.*, 66:333, 1984. DOI: 10.1007/BF00554788.

[13] J. Simons, P. Jørgensen, H. Taylor, and J. Ozment. *J. Phys. Chem.*, 87:2745, 1983. DOI: 10.1021/j100238a013.

[14] E. B. Wilson, J. C. Decius, and P. C. Cross. *Molecular Vibrations*. McGraw-Hill, New York, 1955.

Chapter 9

Fragment-based approaches, calculation of non-covalent interactions, and energy decomposition analysis.

9.1 Introduction

9.2 Fragment Input

In defining fragment inputs, QC-PBC follows the convention established in Q-CHEM closely.

Within the *$unitcell* section, the input is specified identically a Q-CHEM calculation. For example, the following example describes two fragments (in this case each of one atom). The Li is given a charge of 0 and a multiplicity of 2 while the F is given a charge of 0 and a multiplicity of -2. The inclusion of a minus-sign in F multiplicity allows our total calculation to remain a singlet.

```
$unitcell
absolute
0 1
---
0 2
Li 0.0000000000 0.0000000000 0.0000000000
---
0 -2
F 1.6670520000 1.1787840000 2.8874195000
$end
```

The total charge and multiplicity (here 0 charge and a multiplicity of 1) must represent the composite of all the fragments.

For fragment jobs with periodic boundary conditions (as seen also in the AUTOSAD section Sec 4.6.5), we have the option to consider our fragments either as periodic or as molecular. The QC-PBC default is to run fragments as isolated molecules. Periodic fragments must be run either at the Γ-point or with the same MP_MESH as the full calculation.

The options in Sec. 9.2.2 below apply to all fragment based calculations in QC-PBC and control both the printing and whether the fragments are molecular or periodic.

Note, for running jobs with molecular fragments the

9.2.1 Job Control

SCF_PRINT_FRGM

Controls whether the individual fragment jobs are printed to standard out or redirected to scratch.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE The individual fragment jobs are printed to standard out.

FALSE The individual fragment jobs are redirected to the scratch.

RECOMMENDATION:

Generally FALSE is recommended.

TRUE recommended for troubleshooting a fragment calculation or printing more information

ISOLATED_FRGM

Controls whether fragments are run through molecular code (miniqc). TRUE also requires GAMMA_FRGM = TRUE.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE The individual fragment jobs are run with molecular code if GAMMA_FRGM = TRUE.

FALSE The individual fragment jobs are run with the periodic code.

RECOMMENDATION:

Determined by system being studied. TRUE is recommended for molecular crystals and FALSE is recommended for periodic fragments.

GAMMA_FRGM

Controls whether the fragments are run with Γ-point code or k-points code.

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE The fragments are run with the Γ-point PBC code if ISOLATED_FRGM = FALSE.
The fragments are run with the molecular miniqc code if ISOLATED_FRGM = TRUE.

FALSE The fragments are run with the value of MP_MESH specified in the input.

RECOMMENDATION:

Determined by system being studied. TRUE is recommended for molecular crystals and FALSE is recommended for periodic fragments.

9.2.2 Additional Options

For further control over the fragment jobs, the *$rem_frgm* input section may be used identically to in Q-CHEM. Options specified in *$rem_frgm* are overwritten for the fragment part of the calculation. In the example input below, the fragment jobs are run with SCF_ALGORITHM set to DIIS and PRINT_LEVEL set to 2, while the full calculation is run with the SCF_ALGORITHM set to GDM algorithm and the default PRINT_LEVEL. Please note that care should be used when using this input section particularly when running EDA because EDA requires that the methods be identical between fragment and total systems for the results to be meaningful.

```
$rem
   jobtype = sp
   method = pbe
   kecut = 800
   scf_convergence = 10
   scf_algorithm = gdm
   pseudo = GTH-LDA
   basis = SZV-GTH
$end

$rem_frgm
   scf_algorithm = diis
   print_level = 2
$end
```

In the case that the fragments are periodic, the user may desire to have each fragment run with a separate set of lattice vectors. While the default for periodic fragments is for the same set of lattice vectors to be used for all of the fragments, the following input structure may be used to specify each fragment. Please note that if any fragment lattice vectors are specified then all fragments must have specified lattice vectors. This is not generally necessary or recommended; however, the functionality is available.

```
$lattice
3 ! Total system
2.50057800000000 0.00000000000000 1.44371000000000
0.83352600000000 2.35756800000000 1.44371000000000
0.00000000000000 0.00000000000000 2.88741900000000
---
3 ! frgmt 1 isolated
10.0000000000000 1.00000000000000 1.00000000000000
0.00000000000000 10.0000000000000 1.00000000000000
0.00000000000000 0.00000000000000 10.0000000000000
---
3 ! frgmt 2 isolated
10.0000000000000 1.00000000000000 2.00000000000000
0.00000000000000 11.0000000000000 3.00000000000000
0.00000000000000 0.00000000000000 12.0000000000000
$end
```

9.2.3 FRAGMO SCF Guess

9.2.3.1 Introduction

The FRAGMO SCF guess computes the molecular orbitals of each specified fragment and uses a block diagonal molecular orbital coefficient matrix as the initial guess for the following SCF calculation. FRAGMO is the first step in an EDA calculation, but it can also be helpful when dealing with weakly bonded fragments, or when the user wants the initial guess to respect a specific charge or spin state of the material. The following example demonstrates a standard

usage of FRAGMO. Note that the inclusion of the *$rem_frgm* section is optional.

Example 9.1 Input for a PBE calculation of ice with a FRAGMO guess. The fragments are run with LDA and a looser convergence criteria.

```
$lattice
3 ! dimension
  7.8200001717         0.0000000000         0.0000000000
 -3.9100000858         6.7723188063         0.0000000000
  0.0000000000         0.0000000000         7.3600001335
$end

$unitcell
ABSOLUTE
0 1
---
0 1
 O    5.213334    0.000000    6.900000
 H    5.626490    0.711093    7.176000
 H    5.626490   -0.711093    7.176000
---
0 1
 O    2.606667    4.514879    0.460000
 H    2.606667    4.514879    1.280640
 H    2.197420    3.806043    0.191360
---
0 1
 O   -1.303333    2.257440    0.460000
 H   -1.303333    2.257440    1.280640
 H   -1.712580    2.966276    0.191360
---
0 1
 O   -1.303333    2.257439    3.220000
 H   -0.480930    2.255182    3.496000
 H   -1.712580    1.544089    3.496000
---
0 1
 O    1.303333    2.257439    6.900000
 H    1.712580    1.544089    7.176000
 H    0.480930    2.255182    7.176000
---
0 1
 O   -2.606667    4.514879    6.900000
 H   -2.197420    5.228230    7.176000
 H   -3.429070    4.517137    7.176000
---
0 1
 O    5.213333    0.000000    4.140000
 H    5.213333    0.000000    4.960640
 H    4.394840    0.000000    3.871360
---
0 1
 O   -2.606667    4.514879    4.140000
 H   -2.606667    4.514879    4.960640
 H   -2.197420    3.806043    3.871360
---
0 1
 O    1.303333    2.257440    4.140000
 H    1.303333    2.257440    4.960640
 H    1.712580    2.966276    3.871360
---
0 1
 O    2.606667    0.000000    3.220000
 H    2.193510    0.711093    3.496000
 H    2.193510   -0.711093    3.496000
---
0 1
 O    2.606667    0.000000    0.460000
 H    2.606667    0.000000    1.280640
```

9.3 Energy Decomposition Analys

9.3.1 Introduction

Energy decomposition analysis (EDA) enables the analysis of the interaction energy between multiple fragments.

$$\Delta E_{\text{int}} = E_{AB} - E_A - E_B. \tag{9.1}$$

In this equation we consider two fragments A and B and the interaction energy is the difference between the interacting system's energy (E_{AB}) and the two isolated fragments energies ($E_{A/B}$). It is important to note that we only use two fragments for notational convenience and the fragment and EDA functionality in QC-PBC extends to any number of fragments.

The interaction energy can be partitioned into chemically intuitive contributions

$$\Delta E_{\text{int}} \equiv E_{\text{frz}} + E_{\text{pol}} + E_{\text{ct}} \tag{9.2}$$

The first term is the frozen energy which consists of the electrostatic and dispersion interactions as well as the Pauli repulsion. It is calculated by evaluating the energy of unoptimized isolated fragment orbitals within the interacting geometry. Second is the polarization term which corresponds to electronic reorganization within the fragments. It is calculated through the SCF-MI solver and corresponds to SCF with a constraint that the orbitals be block diagonal. Finally, the charge transfer term captures the electronic reorganization between fragments and is calculated with a standard SCF calculation. The values of E_{frz}, E_{pol}, and E_{ct} are determined via successive differences starting with the isolated fragment energies E_A and E_B.

QC-PBC support running both stand alone SCF-MI calculations (see FRGM_METHOD below), and automated energy decomposition analysis jobs.

QC-PBC supports gamma-point EDA as well as **k**-point EDA.

Example 9.2 Γ-point energy decomposition analysis job for two water molecules in a large box.

```
$lattice
3
10.0 0.0 0.0
0.0 10.0 0.0
0.0 0.0 10.0
$end

$unitcell
ABSOLUTE
0 1
--
0 1
O    1.303333    2.257440    4.140000
H    1.303333    2.257440    4.960640
H    1.712580    2.966276    3.871360
--
0 1
O    2.606667    4.514879    3.220000
H    2.197420    5.228230    3.496000
H    3.429070    4.517137    3.496000
$end

$rem
jobtype = eda
method = b3lyp
scf_guess = core
kecut = 500
scf_convergence = 6
scf_algorithm = diis
pseudo = gth-lda
basis = dzvp-GTH
eda_bsse = true
scf_print_frgm = false
scf_print_scfmi = false
scf_print_ct = false
include_virs 0
correct_k_vir true
$end
```

View output online

The following example demonstrates how to use the **k**-point resolved version of EDA.

Example 9.3 **k**-point energy decomposition analysis job for LiF with ionic fragments.

```
$lattice
3 ! dimension
2.50057800000000 0.00000000000000 1.44371000000000
0.83352600000000 2.35756800000000 1.44371000000000
0.00000000000000 0.00000000000000 2.88741900000000
$end

$unitcell
absolute
0 1
---
1 1
Li 0.0000000000 0.0000000000 0.0000000000
---
-1 1
F 1.6670520000 1.1787840000 2.8874195000
$end

$rem
   jobtype = eda
```

9.3.2 Job Control

Self-contained EDA calculations can be requested with the rem command JOBTYPE = EDA. This will automatically run a FRAGMO guess, and SCF-MI single point, an SCF single point, and, if requested, a counterpoise correction. It will summarize the results after each step and print out the full EDA results at the end.

EDA_BSSE

Controls whether the automated counterpoise correction is run (Only for JOBTYPE = EDA). Currently only supported for molecular fragments.

TYPE:

LOGICAL

DEFAULT:

FALSE

OPTIONS:

TRUE Counterpoise correction is computed.

FALSE Counterpoise correction is not computed.

RECOMMENDATION:

User should consult relevant literature to determine whether a counterpoise correction is desired for their system of interest.

BSSE_PAD

Controls the distance cutoff (in input units) from non-ghost atoms for inclusion of ghost atoms in the BSSE calculations.

TYPE:

FLOAT

DEFAULT:

5.0

OPTIONS:

$x > 0.0$ Distance to include ghost orbitals in input units

RECOMMENDATION:

User should determine a cutoff distance which provides a sufficiently converged BSSE correction for their system of interest.

FRGM_METHOD

Controls the distance cutoff (in input units) from non-ghost atoms for inclusion of ghost atoms in the BSSE calculations.

TYPE:

STRING

DEFAULT:

NONE

OPTIONS:

NONE standard SCF methods are run.

ISOLATED Fragments are run individually and then job terminates.

STOLL SCF-MI is run with the Stoll projection scheme.

RECOMMENDATION:

This is not generally necessary to set for an EDA run. The user should set this if they want to run a stand-alone SCF-MI single point, or stand-alone fragment calculations.

9.3.3 Additional Options

SCF_PRINT_SCFMI

Directs the SCF-MI calculation output to the standard out or the scratch directory (Only for JOBTYPE = EDA).

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE SCF-MI calculation printed to standard out.

FALSE SCF-MI calculation redirected to scratch folder.

RECOMMENDATION:

Use default of TRUE unless a less verbose output file is desired.

SCF_PRINT_CT

Directs the charge transfer step in EDA (SCF calculation) to the standard out or the scratch directory (Only for JOBTYPE = EDA).

TYPE:

LOGICAL

DEFAULT:

TRUE

OPTIONS:

TRUE Charge transfer SCF calculation printed to standard out.

FALSE Charge transfer SCF calculation redirected to scratch folder.

RECOMMENDATION:

Use default of TRUE unless a less verbose output file is desired.

SCFMI_DO_FROZEN

Controls whether the first SCFMI iteration (frozen energy) is saved. Warning: This only has the meaning of frozen energy in the context of a FRAGMO guess.

TYPE:

LOGICAL

DEFAULT:

FALSE if JOBTYPE $\neq$ EDA

TRUE if JOBTYPE = EDA

OPTIONS:

TRUE Frozen energy is evaluated and saved.

FALSE Frozen energy is not evaluated and saved

RECOMMENDATION:

User should generally not need to set this variable, as it is automatically set to TRUE by JOBTYPE = EDA

References and Further Reading