User Manual

CP2K

A program package to perform
Molecular Dynamics Simulations

The CP2K developers group

CP2K program release 1.0

September 17, 2002

ETH Zurich – Centro Svizzero di Calcolo Scientifico (CSCS), Switzerland
University of Zurich, Physical Chemistry Institute, Switzerland

Disclaimer

Please note that this manual is not complete. Basically it refers to the CP2K program release 1.0, but the CP2K program package is continuously improved and extended. Therefore the ultimate reference is always the CP2K source code.

Please cite this manual as:

The CP2K developers group, *CP2K User Manual (Release 1.0)*, Zurich, 2002

Copyright

CP2K: A program package to perform molecular dynamics simulations

Copyright © 2002 The CP2K developers group

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version. This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Contents

1 Introduction

The CP2K project was started in 2000 at the Max-Planck institute for solid state research in Stuttgart. Now it is continued at the ETH Zurich (CSCS) and at the University Zurich. The current members of the CP2K developers group are

- Thomas Chassaing (University Zurich)
- Harald Forbert (University Bochum)
- Jürg Hutter (University Zurich)
- Matthias Krack (ETH Zurich/CSCS)
- Fawzi Mohamed (ETH Zurich/CSCS)
- Christopher J. Mundy (LLNL)
- Ari P. Seitsonen (University Zurich)
- Gloria Tabacchi (Università degli studi dell'Insubria, Como)
- Joost VandeVondele (University Zurich)

2 Installation

You can download the current version of the CP2K code from

`http://developer.berlios.de/project/?group_id=129`

using CVS or FTP which also allow to update your current CP2K version. Alternatively, you can directly download the full CP2K tarball which you have to uncompress with

```
$ gunzip cp2k.tar.gz
```

Then extract the archive file with

```
$ tar -xvf cp2k.tar
```

In order to generate an executable change to the directory with the makefile

```
$ cd cp2k/makefiles
```

and run GNU make which is on LINUX systems simply make

```
$ make
```

The default make will create a serial optimized (sopt) executable which is equivalent to

```
make sopt
```

Other choices are

`$ make sdbg` (serial executable for debugging)

`$ make pdbg` (parallel executable for debugging)

`$ make popt` (optimized parallel executable)

You can remove all the stuff generated by `make` with

```
$ make distclean
```

If you want to remove only one version completely use e.g.

```
$ make sopt/realclean
```

or

```
$ make sopt/clean
```

to get rid of the object files only. After a successful compilation you may find the corresponding executable in the directory

`cp2k/exe/`⟨architecture name⟩

There are some test inputs in the directory

`cp2k/tests`

and the next section describes how to run the example inputs.

3 Running CP2K

The CP2K program is started with the command

`$ cp2k.sopt` ⟨input file⟩ `>`⟨output file⟩

The start command for the parallel CP2K version depends on the parallel execution environment of the underlying architecture, e.g. with LINUX/MPICH you may start 4 processes with

`$ mpirun -np 4 cp2k.popt` ⟨input file⟩ `>`⟨output file⟩

whereas on IBMs (AIX) you have to type something like

`$ poe cp2k.popt` ⟨input file⟩ `-procs 4 >`⟨output file⟩

3.1 Input files

The methods implemented in CP2K may require different additional input files.

3.1.1 Fist

3.1.2 Quickstep

- Potential file (default name: `POTENTIAL`)
- Basis set file (default name: `BASIS_SET`)

The Gaussian basis set format and all the atomic potential formats are explained in the corresponding default database files.

4 Input description

4.1 General rules

- **Warning:** Do not expect the input to be logic. The programmers logic may be different from yours.
- **Warning:** This input description may not refer to the actual version of the program you are using. Therefore the ultimate and authoritative input guide is the source code.
- The input is free format and is not case sensitive except where especially stated.
- Empty lines and white spaces at the beginning of a line are ignored.
- All characters in a line following the comment character are ignored. The default comment character is #.
- The CP2K input file is divided into input sections which are started and terminated with the section keywords listed below.
- Each section keyword starts with the section character. The default section character is &.
- The order of the keywords inside an input section is arbitrary except where especially stated.
- For some keywords there are one or even more alias names which are given below as a comma-separated list.
- Lists enclosed in { } imply that you have to choose **exactly one** of the items.
- Lists enclosed in [ ] imply that you can choose **any number** of items (optional keywords).
- There are several possibilities to define a floating point number ⟨real⟩, e.g. `0.05`, `5.0E-2`, `5.e-2`, `1/20`, or `50/1000`. Also the specification of an integer number ⟨integer⟩ is allowed where a floating point number ⟨real⟩ is requested, but not vice versa.
- Strings ⟨string⟩ with special characters like blanks have to be delimited by `" "`.

4.2 Section &CP2K . . . &END

The global CP2K section.

4.2.1 Required keywords

`PROGRAM {FIST,QUICKSTEP}`
Defines the methods which is used for the calculation.

4.2.2 Optional keywords

`FFTLIB` {`FFTESSL,FFTSG,FFTSGI,FFTW`}
Defines the library which is used for the Fast Fourier Transformations (FFT). The availability of the libraries depends on the architecture and/or installation, but at least the `FFTSG` library is available which is included in the CP2K distribution.
default: `FFTSG`

`IOLEVEL` {⟨integer⟩}
Global print level. See also input section `&PRINT`.
default: 0

`PP_LIBRARY_PATH` {⟨path name⟩}
Path to the directory with the pseudo potential database files.
Default: ⟨the current working directory⟩

`PROJECT` {⟨string⟩}
Name of the actual project.
default: `project`

4.3 Section &IO ... &END

In this section the names of input and/or output files can be modified.

`BASIS_SET_FILE` {⟨file name⟩}
Name of the Gaussian basis set database file.
default: `BASIS_SET`

`POTENTIAL_FILE` {⟨file name⟩}
Name of the potential database file.
default: `POTENTIAL`

`RESTART_FILE` {⟨file name⟩}
Name of the restart file.
default: `RESTART`

4.4 Section &CELL ... &END

This section is always needed to define the simulation cell.

4.4.1 Required keywords

`ABC` {⟨real⟩ ⟨real⟩ ⟨real⟩}
Lengths of the vectors **a**, **b**, and **c** which define the orthorhombic simulation cell. The unit of length is defined by the keyword `UNIT`.

4.4.2 Optional keywords

`SUBCELLS` {⟨real⟩}
: The simulation cell is divided into subcells for the generation of the neighbor lists. The specified value defines the size of the subcells. Values between `1.0` and `2.0` show a good performance.
default: `1.5`

`UNIT` {`ANGSTROM,BOHR,SCALED_ANGSTROM,SCALED_BOHR`}
: Defines the unit of length for the simulation cell and it also applies to the definition of the atomic coordinates in the input section `&COORD`. Moreover, all lengths and distances in the output are printed using this unit.
default: `BOHR`

4.5 Section &COORD ... &END

Each non-empty input line in this section defines an atom of the considered system. The first entry in each line has to correspond to an atomic kind name defined by a `&KIND` section which can be a ⟨string⟩ or an ⟨integer⟩ number. The kind name has to follow a set of three ⟨real⟩ numbers defining the x, y, and z coordinates of the atom.

4.6 Section &KIND ... &END

This section has to be defined for each atomic kind in a QUICKSTEP run. The name of the kind has to be defined right after the `&KIND` section keyword on the same input line. The kind name is referenced by the `&COORD` section. Alternatively, the atomic number of the kind can be defined as an integer number, e.g.

```
&KIND 6
```

for carbon which is equivalent to

```
&KIND C
```

In general, any ⟨string⟩ can be defined for an atomic kind

`&KIND` ⟨string⟩

which allows to define different atomic kinds for the same element e.g. carbon with different orbital basis sets

```
&KIND C-DZVP
```

```
&KIND C-TZVP
```

One or more `&KIND` sections are required for a QUICKSTEP run.

4.6.1 Required keywords

`ORBITAL_BASIS_SET,BASIS_SET,BAS` {⟨string⟩}
: Name of the Gaussian orbital basis set which has to be read from the Gaussian basis set database file (see section `&IO`).

`POTENTIAL,POT` {⟨string⟩}
Name of the atomic potential which has to be read from the potential database file (see section `&IO`).

4.6.2 Optional keywords

`ELEMENT_SYMBOL,ELEMENT` {⟨string⟩}
Defines the element to which the atomic kind belongs.

`ATOMIC_MASS,MASS` {⟨real⟩}
Defines an atomic mass different from the default atomic mass, e.g. for the definition of isotopes.

`PAO_MIN_BAS` {list of ⟨integer⟩}
Definition of the projected atomic orbital (PAO) basis. Indices of the PAOs with respect to the full basis set.

4.7 Section &DFT . . . &END

In this section the configuration of a density functional calculation (DFT) can be modified.

`EXCHANGE-CORRELATION-FUNCTIONAL,XC-FUNCTIONAL,FUNCTIONAL` {`BLYP,BP86,LDA,PADE,PBE,NONE`}
Name of the requested exchange-correlation functional for a density functional calculation.
default: `PADE`

`EXCHANGE-FUNCTIONAL,X-FUNCTIONAL` {`B88,SLATER,NONE`}
Name of the requested exchange functional.
default: `NONE`

`CORRELATION-FUNCTIONAL,C-FUNCTIONAL` {`LYP,P86,NONE`}
Name of the requested correlation functional.
default: `PADE`

`KINETIC-ENERGY-FUNCTIONAL,KE-FUNCTIONAL` {`NONE`}
Name of the requested kinetic energy functional.
default: `NONE`

`DENSITY_CUTOFF` {⟨real⟩}
Cutoff for the calculation of the density.
default: `1.0E-10`

`GRADIENT_CUTOFF` {⟨real⟩}
Cutoff for the calculation of the gradients.
default: `1.0E-8`

`GRID` {`CUTOFF,MESH,PLANE_WAVES`}
Definition of the integration grid.

`CUTOFF` {⟨real⟩}
Definition of the plane waves cutoff.

`MESH` {⟨integer⟩ ⟨integer⟩ ⟨integer⟩}
Explicit definition of the grid size.

`PLANE_WAVES,PW`
default

`FORCES`
The calculation of the forces is requested.
default: no force calculation

`CHARGE` {⟨integer⟩}
The total charge of the system.
default: `0`

4.8 Section &QS . . . &END

Program parameters QUICKSTEP

`CUTOFF` {⟨real⟩}
Plane waves cutoff of the largest grid in Rydberg.
default: `320`

`EPS_DEFAULT` {⟨real⟩}
Defines a default threshold value. All threshold values of QUICKSTEP are set to this value.

`EPS_CORE_CHARGE` {⟨real⟩}
Threshold value for the interaction range of the atomic core charge distributions.
default: `1.0E-12`

`EPS_GVG_RSPACE,EPS_GVG` {⟨real⟩}
Threshold value for the integration of the Hartree potential on the real space grid.
default: `1.0E-6`

`EPS_PGF_ORB` {⟨real⟩}
Threshold value for interaction range of the primitive Gaussian-type orbital functions.
default: `1.0E-6`

`EPS_PPL` {⟨real⟩}
Threshold value for the interaction range of the local part of the GTH pseudo potential.
default: `1.0E-12`

EPS_PPNL {⟨real⟩}
Threshold value for the interaction range of the non-local part of the GTH pseudo potential.
default: 1.0E-12

EPS_RHO {⟨real⟩}
Threshold values for EPS_RHO_GSPACE and EPS_RHO_GSPACE.
default: 1.0E-8

EPS_RHO_GSPACE {⟨real⟩}
Threshold value for the calculation of the electronic charge density in Fourier space.
default: 1.0E-8

EPS_RHO_RSPACE {⟨real⟩}
Threshold value for the calculation of the electronic charge density in real space.
default: 1.0E-8

PROGRESSION_FACTOR,PROFAC {⟨real⟩}
Progression factor for the generation of the multi-grid levels.
default: 2.0

RELATIVE_CUTOFF,REL_CUTOFF {⟨real⟩}
Relative plane waves cutoff for each multi-grid level. Values less than 20.0 give inaccurate results and values greater than 30.0 are used for reference calculations (save).
default: 25.0

METHOD {GPW}
Method used by QUICKSTEP. GAPW is not available yet.
default: GPW

MULTI_GRID {list of ⟨integer⟩}
The plane waves cutoffs for each multi grid level in Rydberg. The number of grid levels is defined by the keyword NGRID_LEVEL.

NGRID_LEVEL,NGRID {⟨integer⟩}
Number of the multi-grid levels.
default: 4

PAO
Use the projected atomic orbital (PAO) method.
default: no PAO

4.9 Section &SCF ... &END

This section defines the parameters for the configuration of the self consistent field (SCF) procedure which is used for the wavefunction optimization.

`ARPACK_ON`
The ARPACK eigensolver is used in a parallel run which requires a proper installation of the ARPACK library.
default: no ARPACK usage

`CHOLESKY_ON,CHOLESKY_OFF`
Decides whether the Cholesky decomposition is used in the eigensolver or not.
default: `CHOLESKY_ON`

`DENSITY_GUESS,SCF_GUESS,GUESS` {`ATOMIC,CORE,RESTART`}
Defines the type of guess which is employed to generate the first density matrix. For `RESTART` a valid restart has to be supplied.
default: `ATOMIC`

`OT`
An orbital transformation approach instead of a diagonalization is used for the wavefunction optimization during the SCF iteration procedure.

`DENSITY_MIXING,MIXING` {⟨real⟩}
Factor for the mixing of the old and new density matrix during the wavefunction optimization.
default: `0.4` (i.e. 40% of the new and 60% of the old density are used)

`EPS_DIIS` {⟨real⟩}
The DIIS procedure is switched on, if the maximum DIIS error vector element is below this threshold value.
default: `0.1`

`EPS_EIGVAL` {⟨real⟩}
Threshold value for eigenvector quenching when $S^{-1/2}$ is used as the orthogonalization matrix in the eigensolver.
default: `1.0E-5`

`EPS_SCF` {⟨real⟩}
SCF convergence criterion, i.e. the maximum difference between the corresponding density matrix elements of two consecutive SCF iteration steps.
default: `1.0E-5`

`LEVEL_SHIFT` {⟨real⟩}
Shift value for the unoccupied (virtual) molecular orbitals (MOs) in atomic units.
default: `0.0`

`MAX_DIIS` {⟨integer⟩}
Maximum size of the SCF DIIS buffer.
default: `0`

`MAX_SCF` {⟨integer⟩}
Maximum number of SCF iteration steps.
default: `30`

`NREBUILD` {⟨integer⟩}
Number of SCF steps between two full calculations of the electronic charge density.
default: `1`

`SMEAR` {⟨real⟩}
Window size in atomic units with respect to the eigenvalue of the highest occupied molecular orbital (HOMO) for the smearing of the occupation numbers.
default: `0.0`

`WORK_SYEVX` {⟨real⟩}
Defines the amount of additional work space for the PDSYEVX routine from the SCALAPACK library. A value between `0.0` and `1.0` is accepted. (only for parallel runs using SCALAPACK and an eigensolver with diagonalization.
default: `0.0`

4.10 Section &PRINT . . . &END

This sections allows for detailed output control when running QUICKSTEP. There are 5 predefined print levels: 0, 1, 2, 3, and 4 which correspond to the keywords `NO`, `LOW`, `MEDIUM`, `HIGH`, and `DEBUG` or `FULL`. The print level has to be defined right after the `&PRINT` section keyword on the same input line, e.g.

```
&PRINT LOW
```

which is equivalent to

```
&PRINT 1
```

The following keywords may be used based on the selected print level to requested an additional output or to suppress an output selectively by using the prefix `NO_` for the keyword, e.g. at print level `LOW` the atomic coordinates are listed in the output which may be inconvenient for large sytems, thus simply request `NO_COORDINATES` in the `&PRINT` section. The default print level is `LOW`.

`ANGLES`
Print the angles between all atom triples in the simulation cell.
Warning: That is much output for large systems.

`ATOMIC_COORDINATES,COORDINATES,COORD`
Print all atomic coordinates together with the some atomic kind information.

`BASIC_DATA_TYPES`
Print informations about the basic data types like `REAL`, `INTEGER`, or `LOGICAL`.

`BASIS_SETS,BASIS_SET,BASIS`
Print the Gaussian basis set information, i.e. all Gaussian function exponents and the corresponding contraction coefficients as read from the Gaussian basis set database file. Furthermore, the normalized contraction coefficients are printed.

BLACS_INFO
Print the process grid information of BLACS (Basic linear algebra subprograms)

CARTESIAN_MATRICES
Print all operator matrices in the Cartesian instead of the spherical representation.

CELL_PARAMETERS,CELL
Print the simulation cell data like the cell vectors, cell volume etc.

CORE_HAMILTONIAN_MATRIX,H_MATRIX
Print the core Hamiltonian matrix.

CORE_CHARGE_RADII,CORE_RADII
Print the radius of the core charge distribution for each atomic kind.

DENSITY_MATRIX,P_MATRIX
Print the density matrix.

DERIVATIVES
Print the first derivatives of the operator matrices.

DFT_CONTROL_PARAMETERS
Print the DFT control parameters as defined in the &DFT section.

DIIS_INFORMATION
Print information about the SCF DIIS procedure.

DISTRIBUTION
Print the distribution and the sparsity of the overlap matrix (only parallel version).

EACH_SCF_STEP
Print the requested energies, densities, or matrices for each SCF iteration step.

E_DENSITY_CUBE
Print the electronic charge density as a cube file.

FORCES
Print the atomic force contributions for all atoms.

HOMO
Print the highest occupied molecular orbital (HOMO) as a cube file.

INTERATOMIC_DISTANCES,DISTANCES
Print a matrix with the interatomic distances.
Warning: That is much output for large systems.

KIND_RADII
Print the maximum interaction radius of each atomic kind.

KINETIC_ENERGY_MATRIX,T_MATRIX
Print the kinetic energy integral matrix.

KOHN_SHAM_MATRIX
Print the Kohn-Sham matrix.

LUMO
Print the lowest unoccupied molecular orbital (LUMO) as a cube file.

MEMORY
Print informations about the memory usage of the CP2K program.

MO_EIGENVALUES
Print the eigenvalues of the molecular orbitals (MOs).

MO_EIGENVECTORS,MOS
Print the eigenvectors, eigenvalues, and the occupation numbers of the molecular orbitals (MOs).

MO_OCCUPATION_NUMBERS
Print the occupation numbers and the eigenvalues of the molecular orbitals (MOs).

NEIGHBOR_LISTS
Print all neighbor lists.
Warning: That is much output for large systems.

ORTHO_MATRIX
Print the orthogonalisation matrix used to transform the Kohn-Sham matrix.

OVERLAP_MATRIX
Print the overlap matrix.

PGF_RADII
Print the interaction radii of all primitive Gaussian-type functions.

PHYSICAL_CONSTANTS,PHYSCON
Print the values of all physical constants used in the program.

POTENTIALS
Print a detailed atomic potential information for each atomic kind.

PPL_RADII
Print the interaction radii of the local part of the Goedecker-Teter-Hutter (GTH) pseudo potential [1, 2].

PPNL_RADII
Print the interaction radii of the non-local projector functions of the Goedecker-Teter-Hutter (GTH) pseudo potential [1, 2].

PROGRAM_BANNER
Print a program banner.

`PROGRAM_RUN_INFORMATION`
Print informations about the current program run.

`PW_GRID_INFORMATION`
Print detailed informations about the used plane waves grid.

`RADII`
Print all interaction radii for each atomic kinds.

`SCF`
Print the SCF iteration.

`SCF_ENERGIES`
Print all contributions to the total SCF energy.

`SET_RADII`
Print the interaction radii of all Gaussian orbital sets.

`SPHERICAL_HARMONICS`
Print the transformation matrices between Cartesian and spherical function.

`TIMING_INFORMATION`
Print timing information depending on the `IOLEVEL` defined in the `&CP2K` section.

`TITLE`
Print the title.

`TOTAL_DENSITIES`
Print

`TOTAL_NUMBERS`
Print the total number of atoms, shell sets, basis functions, projectors etc.

`V_HARTREE_CUBE`
Print the Hartree potential as a cube file.

`W_MATRIX`
Print the energy weighted density matrix used for the force calculation.

5 Input examples

5.1 Argon atom

```
&CP2K
 PROGRAM      Quickstep
 IOLEVEL      10
 FFTLIB       FFTSG
&END

&DFT
 FUNCTIONAL   PADE
&END

&QS
 CUTOFF       300
 EPS_DEFAULT  1.0E-12
 EPS_RHO      1.0E-8
 EPS_GVG      1.0E-6
 REL_CUTOFF   30
&END

&SCF
 GUESS        ATOMIC
 EPS_DIIS     0.1
 MAX_DIIS     4
 EPS_SCF      1.0E-6
 MAX_SCF      30
 MIXING       0.4
&END

&PRINT medium
NO_BLACS_INFO
&END

&KIND Ar
 BASIS_SET    DZVP-GTH-PADE
 POTENTIAL    GTH
&END

&CELL
 UNIT         ANGSTROM
 ABC          12.0  12.0  12.0
&END

&COORD
 18     0.000000  0.000000  0.000000
&END
```

5.2 Water molecule

```
&CP2K
 PROGRAM      Quickstep
 IOLEVEL      10
 FFTLIB       FFTSG
&END

&DFT
 FUNCTIONAL   Pade
 FORCES
&END

&QS
 CUTOFF       200
&END

&SCF
 GUESS        ATOMIC
 MIXING       0.4
 EPS_SCF      1.0E-5
&END

&PRINT medium
&END

&KIND H
 BASIS_SET    DZV-GTH-PADE
 POTENTIAL    GTH
&END

&KIND O
 BASIS_SET    DZVP-GTH-PADE
 POTENTIAL    GTH
&END

&CELL
 UNIT         ANGSTROM
 ABC          10.0  10.0  10.0
&END

&COORD
 H   0.000000   -0.757136    0.520545
 O   0.000000    0.000000   -0.065587
 H   0.000000    0.757136    0.520545
&END
```

6 Methods

6.1 GPW method

The electronic energy functional for a molecular or crystalline system in the framework of the Gaussian Plane Waves (GPW) method is defined as [3]

$$\begin{aligned}
E^{\mathrm{el}}[n] &= E^{\mathrm{T}}[n]+E^{\mathrm{V}}[n]+E^{\mathrm{H}}[n]+E^{\mathrm{XC}}[n] \\
&= \sum_{\mu\nu} P_{\mu\nu}\langle\varphi_\mu(\mathbf{r})\mid -\tfrac{1}{2}\nabla^2 \mid \varphi_\nu(\mathbf{r})\rangle + \\
&\quad \sum_{\mu\nu} P_{\mu\nu}\langle\varphi_\mu(\mathbf{r})\mid V_{\mathrm{loc}}^{\mathrm{PP}}(r)\mid \varphi_\nu(\mathbf{r})\rangle - \qquad (1)\\
&\quad \sum_{\mu\nu} P_{\mu\nu}\langle\varphi_\mu(\mathbf{r})\mid V_{\mathrm{nl}}^{\mathrm{PP}}(\mathbf{r},\mathbf{r}')\mid \varphi_\nu(\mathbf{r}')\rangle + \\
&\quad 4\pi\Omega \sum_{|\mathbf{G}|<G_{\mathrm{C}}} \frac{\tilde{n}^*(\mathbf{G})\,\tilde{n}(\mathbf{G})}{\mathbf{G}^2} + \\
&\quad \int \tilde{n}(\mathbf{r})\,\varepsilon_{\mathrm{XC}}[\tilde{n}]\,d\mathbf{r} \qquad (2)
\end{aligned}$$

where $E^{\mathrm{T}}[n]$ is the kinetic energy, $E^{\mathrm{V}}[n]$ is the electronic interaction with the ionic cores, $E^{\mathrm{H}}[n]$ is the electronic Hartree (Coulomb) energy and $E^{\mathrm{XC}}[n]$ is the exchange-correlation energy.

The electronic density

$$n(\mathbf{r}) = \sum_{\mu\nu} P_{\mu\nu}\varphi_\mu(\mathbf{r})\varphi_\nu(\mathbf{r})$$

is expanded in a set of contracted Gaussian functions

$$\varphi_\mu(\mathbf{r}) = \sum_i C_{i\mu} g_i(\mathbf{r})$$

$P_{\mu\nu}$ is a density matrix element, $g_i(\mathbf{r})$ is a primitive Gaussian function and $C_{i\mu}$ is the corresponding contraction coefficient.

An auxiliary basis set of plane waves is used as an intermediate basis set to describe the electronic charge density

$$\tilde{n}(\mathbf{r}) = \frac{1}{\Omega}\sum_{|\mathbf{G}|<G_{\mathrm{C}}} n(\mathbf{G})\mathrm{e}^{i\mathbf{G}\mathbf{r}}$$

which is used for the calculation of the density dependent contributions $E^{\mathrm{H}}[n]$ and $E^{\mathrm{XC}}[n]$. Ω is the volume of the periodic cell. The plane wave expansion is truncated by the specification of a cutoff value for the kinetic energy

$$E_{\mathrm{C}} = \frac{1}{2}G_{\mathrm{C}}^2$$

of the plane waves. Since the $G=0$ term of the Hartree energy is treated with the Ewald method, the nuclear charges are represented by a Gaussian charge distribution and not by point charges.

The GPW method works like pure plane waves methods with atomic pseudo potentials, since an expansion of Gaussian functions with large exponents is numerically not efficient or even not feasible. The current implemention of the GPW method uses only the pseudo potentials of Goedecker, Teter and Hutter (GTH) [1, 2] which consist of a local part $V_{\mathrm{loc}}^{\mathrm{PP}}(r)$ and a non-local part $V_{\mathrm{nl}}^{\mathrm{PP}}(\mathbf{r},\mathbf{r}')$ as shown in Eq. 1.

References

[1] S. Goedecker, M. Teter, and J. Hutter. *Phys. Rev. B*, 54:1703, 1996.

[2] C. Hartwigsen, S. Goedecker, and J. Hutter. *Phys. Rev. B*, 58:3641, 1998.

[3] G. Lippert, J. Hutter, and M. Parrinello. *Mol. Phys.*, 92:477, 1997.

Index