User Manual

CP2K

A program package to perform
Molecular Dynamics Simulations

The CP2K developers group

CP2K program release 1.0

September 3, 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

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 `_id=129`
using CVS or FTP which also allow to update your current CP2K version. Alternatively,
`http://developer.berlios.de/project/?group`
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 xvfto generate an e

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⟩
```

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.

FFTLIB *f*FFTESSL,FFTSG,FFTSGI,FFTW*g*

Defines the library which is used for the Fast Fourier Transformations (FFT). The

4.4.2   Optional keywords

`UNIT` *f*`ANGSTROM,BOHR,SCALED_ANGSTROM,SCALED_BOHR`*g*
> 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

4.6.2   Optional keywords

`ELEMENT_SYMBOL,ELEMENT` *fh*string*ig*
   Defines the element to which the atomic kind belongs.

`ATOMIC_MASS,MASS` *fh*real*ig*

MESH *fh*integer*i h*integer*i h*integer*ig*
    Explicit definition of the grid size.

PLANE_WAVES,PW
    default

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

CHARGE *fh*integer*ig*
The total charge of the system.
default: 0

4.8   Section &QS ... &END

Program parameters QUICKSTEP

CUTOFF *fh*real*ig*
Plane waves cutoff of the largest grid in Rydberg.
default: 320

EPS_DEFAULT *fh*real*ig*
Defines a default threshold v2 ETh
default: 3201.0E-]TJ/F57.173 -23.398 Td[(CHA)]TJ6611.532.17 0.3-52.29587 0.

EPS_RHO_GSPACE *fh*real*ig*
  Threshold value for the calculation of the electronic charge density in Fourier space.
  default: `1.0E-8`

EPS_RHO_RSPACE *fh*real*ig*
  Threshold value for the calculation of the electronic charge density in real space.
  default: `1.0E-8`

PROGRESSION_FACTOR,GREFAC *fh*real*ig*
  Progression factor for the generation of the multi-grid levels.
  default: `2.0`

RELATIVE_CUTOFF,REL_CUTOFF *fh*real*ig*
  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 e(ellw)10(a)20A1.2970.91

DENSITY_GUESS,SCF_GUESS,GUESS *f*ATOMIC,CORE*g*
  Defines the type of guess which is employed to generate the first density matrix.
  default: ATOMIC

OT
  An orbital transformation approach instead of a diagonalization is used for the wave-
  function optimization during the SCF iteration procedure.

DENSITY_MIXING,MIXING *f*real*ig*
  Factor for the mixing of the old and new density matrix during the wavefunction opti-
  mization.
  default: 0.4 (i.e. 40% of the new and 60% of the old density are used)

EPS_DIIS *f*real*ig*
  The DIIS procedure is switched on, if the maximum DIIS error vector element is below
  this threshold value.
  default: 0.1

EPS_

`WORK_SYEVX` *fh*real*ig*
   Defines the amount of additional work space for the PDSYEVX routine from the

`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_`

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
&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
```

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 $V_{\text{loc}}^{\text{PP}}(r)$ and a non-local part $V_{\text{nl}}^{\text{PP}}(\mathbf{r}; \mathbf{r}')$ as shown in Eq. 1.

Index