CP2K
http://cp2k.berlios.de/

Basic tutorial to CP2K calculations

Sébastien LE ROUX sebastien.leroux@ipcms.unistra.fr

INSTITUT DE PHYSIQUE ET DE CHIMIE DES MATÉRIAUX DE STRASBOURG,
DÉPARTEMENT DES MATÉRIAUX ORGANIQUES,
23 RUE DU LOESS, BP43,
F-67034 STRASBOURG CEDEX 2, FRANCE

MARCH 23, 2016

Contents

List of Tables

CP2K basics

1.1 General ideas on the CP2K method

For details see references [1–3].

CP2K is a program (written in Fortran95) to perform atomistic and molecular simulations. The CP2K code provides access to a wide number of calculations (MM, QMMM, QM, MC, MD ...), this guide is written to help to set up First Principles calculations (single point, geometry optimization and in particular molecular dynamics) therefore all other aspects of the code will not be discuss in the next pages.
FPMD-CP2K simulations are Born-Oppenheimer molecular dynamics calculations. The DFT approaches uses a mixed Gaussian and plane waves approach (GPW = Gaussian and Plane Waves). The wave-function is described using a combination of Gaussians, but the density is calculated on a grid (PAW like scheme [4]). For large systems the method scales as $O(N^2M)$ with M = number of basis functions and N = number of molecular orbitals.
In CP2K the input format uses a CPMD-like section system:

```
&SECTION
...
&END SECTION
```

And the CP2K input file for a FPMD calculation is decomposed in 3 mandatory sections:

- a `GLOBAL` section
- a `FORCE_EVAL` section
- a `MOTION` section

The order of appearance of these sections does not matter.

1.2 Variables and include files

The CP2K input system is "command interpreter"-like system, thus it allows to define variables, commands and use include files.

Variables

Since input files for CP2K can be huge, it is very interesting to use variables to define the key parameters in the same (beginning) part of the input file to make them easily accessible.

```
@SET MD_STEPS 5000
.
.
.
&MOTION
  &MD
    STEPS ${MD_STEPS}
  &END MD
&END MOTION
```

In this example the variable "MD_STEPS" is declared using the "**@SET**" command, and used further away in the input file to define the number of MD steps. The declaration must be composed of 3 and only 3 instructions, the "**@SET**" command, the name of the variable (ex: MD_STEPS) and the value of the variable (ex: 5000). Any other parameter on the instruction line would be parsed as part of the value of the variable:

```
@SET MD_DT 1.5   ! Integration time step in fs     => Bad
@SET MD_DT 1.5                                     => Good
```

In the CP2K input system this rule is a general rule for the case of text variables, ie. file names, therefore when specifying a file name no commentary should be added on the same line.

Include files

Also it is possible to split an input into several parts, each of the files being inserted at the appropriate place allowing to simplify the reading of the input file.

```
@INCLUDE 'file.inc'
```

In this example the include file "file.inc" is inserted and read using the "**@INCLUDE ' '**" command. Include files can also contains "**@INCLUDE ' '**" command(s), therefore at some point the structure of the input file can become somewhat complicated.
The syntax of the CP2K input file can be tested using:

```
]$ cp2k -c input.inp
```

The command acts recursively on all include files.

1.3 The GLOBAL section

```
&GLOBAL
  PROJECT        My_system
  RUN_TYPE       My_calculation
  PRINT_LEVEL    MEDIUM
  WALLTIME       My_cpu_time
&END GLOBAL
```

- PROJECT = name of the project, here "My_system"
- RUN_TYPE = type of calculation, here "My_calculation", available types of calculation and corresponding keywords are:
  - Molecular dynamics: MD
  - Geometry optimization: GEO_OPT
  - Single point calculation: ENERGY_FORCE
- PRINT_LEVEL = the amount of data outputted by CP2K, use LOW for standard information.
- WALLTIME = internal limit for CPU time in seconds, here "My_cpu_time", for a clean job ending and restart

Defining variables using the "@SET" command [Sec. 1.2] we can write:

```
@SET SYSNAME    liquid-Ge2Se3
@SET RTYPE       MD
@SET CPUTIME    36000

&GLOBAL
  PROJECT        ${SYSNAME}
  RUN_TYPE       ${RTYPE}
  PRINT_LEVEL    MEDIUM
  WALLTIME       ${CPUTIME}
&END GLOBAL
```

The interest in this method is allow to produce more generalized input files, this will be illustrated in the next pages.

1.4 Energy and forces calculation - the FORCE_EVAL section

The FORCE_EVAL section describes how forces and energy will be calculated and therefore is the heart of the calculation.
From now on we will use the following syntax in our example CP2K input file:

- Comments will appear in blue font, these comments are in the CP2K input format and can therefore be kept as part of the input file.
- Variables will be declared and appear in green font.
- Include files will be inserted in **bold font**.

First we are going to define variables to generalize and simplify the FORCE_EVAL section for DFT based calculations, this part of the input file is presented in table [Tab. 1.1].

```
! Is the calculation a restart or not ?
@SET RESTART                            FALSE

! Name of the file that contains the basis set
@SET BASISFILE                          file_with_basis_set
! Name of the file that contains the pseudo-potentials
@SET PSEUDOFILE                         file_with_pseudo
! Name of the file that contains the wave-function (for restart)
@SET WAVEFILE                           file_with_wavefunction

! For comments on the following variables see the FORCE_EVAL section
@SET CUTOFF                             300
@SET GRIDS                              5
@SET SCF_NCYCLES                        500
@SET SCF_OCYCLES                        100
@SET SCF_CONV                           1E-6
@IF ( ${RESTART} == TRUE )
  @SET SCF_GUESS                        RESTART
@ENDIF
@IF ( ${RESTART} == FALSE )
  @SET SCF_GUESS                        ATOMIC
@ENDIF
! Minimizer algorithm for SCF: BROYDEN, CG, or DIIS
@SET SCF_MINI                           CG
@SET FUNCTIONAL                         BLYP

! Save results every OUT_STEPS molecular dynamics steps
@SET OUT_STEPS                          1
```

Table 1.1 *Declaration of the variables used in the FORCE_EVAL section of the CP2K input file format.*

A first principles type FORCE_EVAL section is presented in table [Tab. 1.2].

```
!
! This file contains the description the CP2K method to evaluate forces and energy
!
&FORCE_EVAL
! To print the forces
  &PRINT
    &FORCES
    &END
  &END
! For first principles calculation the method is always Quickstep
! which is the name of the method used to expand the wave-function in CP2K
  METHOD Quickstep
  &DFT
! The files that will be used in the calculation
    BASIS_SET_FILE_NAME ${BASISFILE}
    POTENTIAL_FILE_NAME ${PSEUDOFILE}
! If we restart a calculation then we use the previous wave-function
@IF ( ${RESTART} == TRUE )
    WFN_RESTART_FILE_NAME ${WAVEFILE}
@ENDIF
! Setting up of the multigrids calculation
    &MGRID
       CUTOFF ${CUTOFF}             ! => Cutoff of the finest grid level
       NGRIDS ${GRIDS}              ! => Number of multigrids to use, default = 4
    &END MGRID
! Setting up of the Quickstep algorithm
    &QS
       METHOD GPW                   ! => This is the default value
       EPS_DEFAULT 1.0E-12          ! => Default value is 1.0E-10
       MAP_CONSISTENT TRUE          ! => Default value is TRUE, more consistent
       EXTRAPOLATION ASPC           ! => Recommend for MD, PS otherwise
       EXTRAPOLATION_ORDER 3        ! => 4 can be better but increases CPU time
    &END QS
! Setting up the Self Consistent Field calculation parameters
! the SCF section is inserted using an include file:
@INCLUDE 'scf.inc'
! Setting up the exchange-correlation calculation
    &XC
      &XC_FUNCTIONAL ${FUNCTIONAL}
      &END XC_FUNCTIONAL
    &END XC
  &END DFT
! The FORCE_EVAL also requires a SUBSYS section to describe the system
! the SUBSYS section is inserted using an include file:
@INCLUDE 'subsys.inc'
&END FORCE_EVAL
```

Table 1.2 *First principles type FORCE_EVAL section of the CP2K input file format.*

The SCF section, in the 'scf.inc' include file is presented in table [Tab. 1.3].

```
!
! This file contains the CP2K parameters for the SCF calculation
!
&SCF
  MAX_SCF   ${SCF_NCYCLES}    ! => Maximum number of SCF cycles
  EPS_SCF   ${SCF_CONV}       ! => Threshold for the SCF convergence
  SCF_GUESS ${SCF_GUESS}      ! => Initial guess for the wave-function
                              !    Default value of initial configuration,
                              !    Initial calculation: ATOMIC
                              !    Restart calculation: RESTART
! Setting up the Orbital Transformation to speed up the calculation
! WARNING - This requires a gap HOMO-LUMO
! WARNING - therefore it works only for insulators
  &OT ON
    MINIMIZER ${SCF_MINI}     ! => Which algorithm for the minimizer DIIS or CG
    PRECONDITIONER FULL_ALL   ! => Preconditioner for the minimization scheme,
                              !    FULL_ALL is the most effective state selective
                              !    preconditioner and is based on diagonalization
    ENERGY_GAP 0.001          ! => Underestimate value of the Gap (HOMO-LUMO) in a.u.
                              !    to be used with the FULL_ALL preconditioner
  &END OT
! If after the ${SCF_NCYCLES} first SCF steps no convergence has been reach
! more SCF cycles can be done updating the preconditioner.
! detailed information is then specified in the 'OUTER_SCF' section
  &OUTER_SCF
    MAX_SCF ${SCF_OCYCLES}    ! => We update the preconditioner and start a new cycle
                              !    up to ${SCF_OCYCLES} times, thus in this case:
                              !    500 x 100 cycles can be computed
    EPS_SCF ${SCF_CONV}       ! => Convergence threshold for the extra cycles
                              !    it is safer to keep the same value
  &END OUTER_SCF
  &PRINT
! To tune the printing of the restart file
    &RESTART
      LOG_PRINT_KEY T         ! => Printing on screen when restart file is written
      &EACH
        QS_SCF 0              ! => Never write restart file(s) during the SCF cycle
        MD     ${OUT_STEPS}   ! => Write restart file every ${OUT_STEPS} MD steps
      &END EACH
      ADD_LAST NUMERIC
    &END RESTART
  &END PRINT
&END SCF
```

Table 1.3 *SCF subsection of the FORCE_EVAL section of the CP2K input file format.*

The SUBSYS section, in the 'subsys.inc' include file is presented in table [Tab. 1.4].

```
!
! This file contains the description of the system in the CP2K input format
!
&SUBSYS
  &CELL
    ABC [angstrom] 15.28 15.28 15.28     ! => Simulation box parameters
    PERIODIC XYZ                         ! => Periodicity of the system
  &END CELL
  &TOPOLOGY
! From XYZ file coordinates are always Cartesian and in angstrom
    COORDINATE XYZ
! Name of the file that contains the atomic coordinates
    COORD_FILE_NAME GeSe.xyz
  &END TOPOLOGY
  &KIND Ge                           ! => For each species we create a KIND section
    BASIS_SET SZV-MOLOPT-SR-GTH      ! => Very tricky section, thorough tests
                                     !    to ensure that proper basis set is used
    POTENTIAL GTH-BLYP-q4            ! => CP2K pseudo-potential: GTH, ALL or KG
                                     !    the exact sequence 'GTH-BLYP-q4' appears
                                     !    in the file that contains the pseudo-pot.
                                     !    in front of the name of the element
  &END KIND
  &KIND Se
    BASIS_SET SZV-MOLOPT-SR-GTH
    POTENTIAL GTH-BLYP-q6
  &END KIND
&END SUBSYS
```

Table 1.4 *SUBSYS subsection of the FORCE_EVAL section of the CP2K input file format.*

1.5 The MOTION section

If atoms are moved during the calculation in a geometry optimization or a molecular dynamics simulation it is required to create a MOTION section that contains the information related to either of both.

1.5.1 MOTION section for geometry optimization

For geometry optimization it is required to define a GEO_OPT subsection in the MOTION section. As done previously we define variables to generalize and simplify the MOTION section for geometry optimization calculations, this part of the input file is presented in table [Tab. 1.5]. The MOTION section for geometry optimization is presented in table [Tab. 1.6].

```
! Type of calculation (see section [Sec. 1.3] for details)
@SET RTYPE                              GEO_OPT

! Minimizer algorithm for geometry optimization
@SET GEO_MINI                           CG
! Maximum number of geometry optimization steps
@SET GEO_MAXS                           10000

! Output format
@SET OUT_FORM                           XYZ
! Output unit
@SET OUT_UNIT                           angstrom
! Save results every OUT_STEPS steps of geometry optimization
@SET OUT_STEPS                          1
```

Table 1.5 *Declaration of the variables used in the `GEO_OPT` section of the CP2K input file format.*

```
!
! This file contains the CP2K parameters for the geometry optimization
!
&MOTION
  &GEO_OPT
    MINIMIZER ${GEO_MINI}
    MAX_ITER  ${GEO_MAXS}
  &END GEO_OPT
  &PRINT
    &RESTART
      LOG_PRINT_KEY T
      &EACH
        GEO_OPT ${OUT_STEPS}
      &END EACH
      ADD_LAST NUMERIC
    &END RESTART
    &TRAJECTORY
      LOG_PRINT_KEY T
      FORMAT ${OUT_FORM}
      UNIT   ${OUT_UNIT}
      &EACH
        GEO_OPT ${OUT_STEPS}    ! Trajectory files will be printed every ${OUT_STEPS} steps
      &END EACH
      ADD_LAST NUMERIC
    &END TRAJECTORY
  &END PRINT
&END MOTION
```

Table 1.6 *`MOTION` section for geometry optimizations in the CP2K input file format.*

1.5.2 MOTION section for molecular dynamics

For molecular dynamics it is required to define a MD subsection in the MOTION section. As done previously we define variables to generalize and simplify the MOTION section for molecular dynamics calculations, this part of the input file is presented in table [Tab. 1.7].

```
! Type of calculation (see section [Sec. 1.3] for details)
@SET RTYPE                              MD

! Thermodynamical ensemble: NVT or NVE
@SET MD_ENS                             NVT
! Number of MD steps
@SET MD_STEPS                           10000
! Integration time step of the Newton's equation of motion (in fs)
@SET MD_DT                              2.0
! Target temperature (in K)
@SET MD_TEMP                            300

! Output format
@SET OUT_FORM                           XYZ
! Output unit
@SET OUT_UNIT                           angstrom
! Save results every OUT_STEPS steps of molecular dynamics
@SET OUT_STEPS                          1
```

Table 1.7 *Declaration of the variables used in the MD section of the CP2K input file format.*

The MOTION section for molecular dynamics calculations is presented in table [Tab. 1.8].

```
!
! This file contains the CP2K parameters for the molecular dynamics calculation
!
&MOTION
  &MD
    ENSEMBLE    ${MD_ENS}
    STEPS       ${MD_STEPS}
    TIMESTEP    ${MD_DT}
    TEMPERATURE ${MD_TEMP}
    @IF ( ${MD_ENS} == NVT )
      &THERMOSTAT
        TYPE   NOSE
        REGION GLOBAL
! Parameters of the Nosé-Hoover chain thermostat
        &NOSE
          TIMECON 50.
          LENGTH  3
          YOSHIDA 3
          MTS     2
        &END
      &END
    @ENDIF
  &END MD
! Setting up how to print results
  &PRINT
    &RESTART
      LOG_PRINT_KEY T
      &EACH
        MD   ${OUT_STEPS}   ! Restart files will be printed every ${OUT_STEPS} steps
      &END EACH
      ADD_LAST NUMERIC
    &END RESTART

    &TRAJECTORY
      LOG_PRINT_KEY T
      FORMAT ${OUT_FORM}
      UNIT   ${OUT_UNIT}
      &EACH
        MD   ${OUT_STEPS}   ! Trajectory files will be printed every ${OUT_STEPS} steps
      &END EACH
      ADD_LAST NUMERIC
    &END TRAJECTORY

    &VELOCITIES
      LOG_PRINT_KEY T
      FORMAT ${OUT_FORM}
      UNIT   ${OUT_UNIT}
      &EACH
        MD   ${OUT_STEPS}   ! Velocity files will be printed every ${OUT_STEPS} steps
      &END EACH
      ADD_LAST NUMERIC
    &END VELOCITIES
  &END PRINT
&END MOTION
```

Table 1.8 *MOTION section for MD calculations in the CP2K input file format.*

1.5.3 Generalized MOTION section

The CP2K input format allows to go further in the writing of the MOTION section with a general syntax for first principles calculations. Again we declared variables to generalize and simplify the MOTION section, this part of the input file is presented in table [Tab. 1.9].

```
! Type of calculation (see section [Sec. 1.3] for details)
@SET RTYPE                              MD

!
! Geometry optimization options
!
! Minimizer algorithm for geometry optimization
@SET GEO_MINI                           CG
! Maximum number of geometry optimization steps
@SET GEO_MAXS                           10000

!
! Molecular dynamics options
!
! Thermodynamical ensemble: NVT or NVE
@SET MD_ENS                             NVT
! Number of MD steps
@SET MD_STEPS                           10000
! Integration time step of the Newton's equation of motion (in fs)
@SET MD_DT                              2.0
! Target temperature (in K)
@SET MD_TEMP                            300

! Output format
@SET OUT_FORM                           XYZ
! Output unit
@SET OUT_UNIT                           angstrom
! Save results every OUT_STEPS steps of MD or geometry optimization
@SET OUT_STEPS                          1
```

Table 1.9 *Declaration of the variables used in the generalized MOTION section of the CP2K input file format.*

The generalized MOTION section that can be used for both molecular dynamics and geometry optimization calculations is presented in table [Tab. 1.10].

```
!
! This file contains the CP2K parameters for geometry optimization and MD calculation
!
&MOTION
  &${RTYPE}
    @IF ( ${RTYPE} == MD )         ! Running molecular dynamics
      ENSEMBLE    ${MD_ENS}
      STEPS       ${MD_STEPS}
      TIMESTEP    ${MD_DT}
      TEMPERATURE ${MD_TEMP}
      @IF ( ${MD_ENS} == NVT )
        &THERMOSTAT
          TYPE   NOSE
          REGION GLOBAL
! Parameters of the Nosé-Hoover chain thermostat
          &NOSE
            TIMECON 50.
            LENGTH  3
            YOSHIDA 3
            MTS     2
          &END
        &END
      @ENDIF
    @ENDIF
    @IF ( ${RTYPE} == GEO_OPT )    ! Running geometry optimization
      MINIMIZER ${GEO_MINI}
      MAX_ITER  ${GEO_MAXS}
    @ENDIF
  &END ${RTYPE}
! Setting up how to print results
  &PRINT
    &RESTART
      LOG_PRINT_KEY T
      &EACH
        ${RTYPE} ${OUT_STEPS}     ! Restart files will be printed every ${OUT_STEPS} steps
      &END EACH
      ADD_LAST NUMERIC
    &END RESTART
    &TRAJECTORY
      LOG_PRINT_KEY T
      FORMAT ${OUT_FORM}
      UNIT   ${OUT_UNIT}
      &EACH
        ${RTYPE} ${OUT_STEPS}     ! Trajectory files will be printed every ${OUT_STEPS} steps
      &END EACH
      ADD_LAST NUMERIC
    &END TRAJECTORY
    @IF ( ${RTYPE} == MD )
      &VELOCITIES
        LOG_PRINT_KEY T
        FORMAT ${OUT_FORM}
        UNIT   ${OUT_UNIT}
        &EACH
          ${RTYPE} ${OUT_STEPS}   ! Velocity files will be printed every ${OUT_STEPS} steps
        &END EACH
        ADD_LAST NUMERIC
      &END VELOCITIES
    @ENDIF
  &END PRINT
&END MOTION
```

Table 1.10 *Generalized* `MOTION` *section for both MD and geometry optimizations in the CP2K input file format.*

1.6 Restarts

To restart a calculation it is required to insert an EXT_RESTART section in the CP2K input file. Variables to generalize and simplify the EXT_RESTART section are presented in table [Tab. 1.11].

```
! Type of calculation see section [Sec. 1.3]
@SET RTYPE                              MD
@SET RESTART                            TRUE

! Name of the restart file from the previous run
@SET RESTARTFILE                        My_restart_file

! Thermodynamical ensemble: NVT or NVE
@SET MD_ENS                             NVT
```

Table 1.11 *Declaration of the variables used in the generalized MOTION section of the CP2K input file format.*

The generalized RESTART_RES section for first principles calculations in CP2K is presented in table [Tab. 1.12].

```
&EXT_RESTART ON
  RESTART_DEFAULT F
  RESTART_FILE_NAME ${RESTARTFILE}
  RESTART_POS T
  RESTART_COUNTERS T
  @IF ( ${RTYPE} == MD )
    RESTART_VEL T
    @IF ( ${MD_ENS} == NVT )
      RESTART_THERMOSTAT T
    @ENDIF
  @ENDIF
&END EXT_RESTART
```

Table 1.12 *EXT_RESTART section in the CP2K input file format.*

CP2K input structure setup for first principle calculations

2.1 Organization

A typical set up for an initial calculation would be to prepare the following files:

- `System.inp` → Main input file including the `GLOBAL` and `EXT_RESTART` sections.
- `forces.inc` → `FORCE_EVAL` section include file.
- `scf.inc` → `SCF` section include file.
- `subsys.inc` → `SUBSYS` section include file.
- `motion.inc` → `MOTION` section include file.
- `restart.inc` → `EXT_RESTART` section include file.
- `system.xyz` → Atomic positions in XYZ angstrom Cartesian coordinates format.
- `BASIS_SETS` → File that contains the basis sets.
- `PSEUDO_POT` → File that contains the pseudo-potentials.

For the purpose of this manual it is not necessary to emphasizes the content of `system.xyz` (user defined), as well as the content of `BASIS_SETS` and `PSEUDO_POT` that are both distributed with `CP2K`. The `forces.inc`, `scf.inc`, `subsys.inc` (partially user defined), `motion.inc` (generalized `MOTION` section) and `restart.inc` files are strictly identical to the one presented respectively in tables [Tab. 1.2, 1.3, 1.4, 1.10 and 1.12].
The main input file will contains all the variable definitions as well as the small `GLOBAL` and `EXT_RESTART` sections.

2.2 The main input file: `System.inp`

```
!     /* This is my CP2K input file */
!
! General options
!
@SET SYSNAME                            My_system
@SET RTYPE                              MD
! Internal CPU time limit (in s) for a clean job ending
@SET CPUTIME                            36000
! Is the calculation a restart or not ?
@SET RESTART                            FALSE
!
! Files options
!
! Name of the file that contains the basis set
@SET BASISFILE                          BASIS_SETS
! Name of the file that contains the pseudo-potentials
@SET PSEUDOFILE                         PSEUDO_POT
! Name of the file that contains the wave-function (for restart)
@SET WAVEFILE                           file_with_wavefunction
! Name of the restart file from the previous run (for restart)
@SET RESTARTFILE                        file_with_restart
!
! DFT calculation options
!
@SET CUTOFF                             300
@SET GRIDS                              5
@SET SCF_NCYCLES                        500
@SET SCF_OCYCLES                        100
@SET SCF_CONV                           1E-6
@IF ( ${RESTART} == TRUE )
  @SET SCF_GUESS                        RESTART
@ENDIF
@IF ( ${RESTART} == FALSE )
  @SET SCF_GUESS                        ATOMIC
@ENDIF
! Minimizer algorithm for SCF: BROYDEN, CG, or DIIS
@SET SCF_MINI                           CG
@SET FUNCTIONAL                         BLYP
!
! Geometry optimization options
!
! Minimizer algorithm for geometry optimization
@SET GEO_MINI                           CG
! Maximum number of geometry optimization steps
@SET GEO_MAXS                           10000
!
! Molecular dynamics options
!
! Thermodynamical ensemble: NVT or NVE
@SET MD_ENS                             NVT
! Number of MD steps
@SET MD_STEPS                           10000
! Integration time step of the Newton's equation of motion (in fs)
@SET MD_DT                              2.0
! Target temperature (in K)
@SET MD_TEMP                            300
!
! Output options
!
@SET OUT_FORM                           XYZ
@SET OUT_UNIT                           angstrom
! Save results every OUT_STEPS steps for SCF, MD or GEO_OPT calculations
@SET OUT_STEPS                          1
```

```
!
! The mandatory GLOBAL section
!
&GLOBAL
  PROJECT     ${SYSNAME}
  RUN_TYPE    ${RTYPE}
  PRINT_LEVEL MEDIUM
  WALLTIME    ${CPUTIME}
&END GLOBAL
!
! The optional EXT_RESTART section include file
!
@IF ( ${RESTART} == TRUE )
  @INCLUDE 'restart.inc'
@ENDIF
!
! The mandatory FORCE_EVAL section include file
!
@INCLUDE 'forces.inc'
!
! The optional MOTION section include file
!
@IF ( ${RTYPE} /= ENERGY_FORCE )
  @INCLUDE 'motion.inc'
@ENDIF
```

Table 2.1 *The main input file for first principles CP2K calculations.*

It has to be stressed that many other keywords are available for an advanced parametrization of the calculation, detailed can be found in the CP2K user manual [1].

Bibliography

[1] http://cp2k.berlios.de.

[2] G. Lippert, J. Hutter, and M. Parrinello. *Mol. Phys.*, **92**(3):477–487 (1997).

[3] J. VandeVondele and J. Hutter. *J. Chem. Phys.*, **118**(10):4365–4369 (2003).

[4] P.E. Blöchl. *Phys. Rev. B*, **50**(24):17953–17979 (1994).

This document has been prepared using the Linux operating system and free softwares:

The text editor	"gVim"
The GNU image manipulation program	"The Gimp"
The WYSIWYG plotting tool	"Grace"
And the document preparation system	"LaTeX 2ε".