Hartree Centre

Science & Technology Facilities Council

RUNNING CP2K CALCULATIONS

Iain Bethune

iain.bethune@stfc.ac.uk

@iainbethune

Overview

- How to run CP2K
- CP2K Input file
  - The Basics
  - The How – FORCE_EVAL
  - The What – MOTION
- Basis Sets and Pseudopotential libraries
- CP2K Output
  - Controlling what gets written
  - Overview of an output file
- Restarting a calculation

How to run CP2K

- CP2K binaries:
  - `cp2k.version` where `version` is usually one of:
    - `sopt` – Serial, optimised
    - `ssmp` – Single process + symmetric multiprocessor (OpenMP)
    - `popt` – Parallel (MPI), optimised
    - `psmp` – Parallel (MPI) + symmetric multiprocessor (OpenMP)

- Available from http://www.cp2k.org/download
  - Linux binaries (released versions)
  - Also in Linux package managers
  - Source code (released versions and latest trunk), GPL
  - May be pre-installed, e.g. ARCHER …

How to run CP2K

- Basic command line options:
  - `cp2k.sopt -i input_file -o output_file`
    - By default, output goes to the standard output
    - Output to file appends (beware!)
    - Input file is the last argument if not otherwise specified

- Other useful options:
  - `cp2k.sopt --version`
  - `cp2k.sopt --check input_file`
  - `cp2k.sopt --html-manual`
  - `cp2k.sopt --help`

How to run CP2K

- Typical files associated with a CP2K run:
  - Input (required):
    - e.g. `H2O-32.inp` (main input file, name and extension are arbitrary)
  - Optional inputs:
    - `POTENTIAL` (psuedopotential library)
    - `BASIS_SET` (basis set library)
    - Structure file (e.g. psf, xyz, crd …)
    - …
  - Outputs:
    - `PROJECT-1.restart` (input file to restart calculation)
    - `PROJECT-pos-1.xyz` (trajectory for MD or GEO_OPT)
    - `PROJECT-1.ener` (MD energies, temperature, cons. Q …)
    - `PROJECT-1.cell` (cell parameters for NPT MD or CELL_OPT)
    - `PROJECT-RESTART.wfn` (orbitals for restart)
    - …

CP2K Input file: The Basics

- Full documentation available online:
  - http://manual.cp2k.org
  - Or generate with `--html-manual`
- Sections – 13 (optional) top level sections
  - `&BEGIN section_name [params]`
  - …
  - `&END [section_name]`
- Keywords
  - `KEYWORD value`
  - `KEYWORD [ON|OFF] [YES|NO] [TRUE|FALSE] …`
  - `KEYWORD`
- Nesting
  - Sections may others sections and keywords

CP2K Input file: The Basics

- Basic pre-processing syntax

`@INCLUDE 'filename'`	– copy in text from file
`@SET VAR value`	– define a variable
`$VAR`	– replaced with variable value
`@IF / @ENDIF`	– simple logic
`!` or `#`	– comments

- Units
  - Numerical entries have a default unit (see manual)
  - Specify other units by hand e.g.
    `ABC [nm] 100 100 100` (or `bohr`, default is `angstrom`)
    `EMAX_SPLINE [eV] 50` (or `Ry`, `joule`, default is `hartree`)
  - Also combinations e.g. `[hartree*bohr^-2]`

CP2K Input file: The Basics

GLOBAL section (required)

```
&GLOBAL
  PROJECT H2O-32
  RUN_TYPE MD
  PRINT_LEVEL HIGH
  &TIMINGS
     THRESHOLD 0.000001
  &END
  WALLTIME 3600
&END GLOBAL
```

CP2K Input file: The How

FORCE_EVAL section (required)

```
&FORCE_EVAL
  METHOD QS   (or FIST, QMMM ...)
  &DFT
  ...
  &END DFT
  &SUBSYS
  ...
  &END SUBSYS
&END FORCE_EVAL
```

CP2K Input file: The How

```
&DFT
  BASIS_SET_FILE_NAME GTH_BASIS_SETS
  POTENTIAL_FILE_NAME POTENTIAL
  &MGRID
    CUTOFF 280
    REL_CUTOFF 30
  &END MGRID
  &QS
    EPS_DEFAULT 1.0E-12
    WF_INTERPOLATION PS
    EXTRAPOLATION_ORDER 3
  &END QS
  &SCF
    SCF_GUESS ATOMIC
    &OT ON
      MINIMIZER DIIS
    &END OT
    &PRINT
      &RESTART OFF
      &END
    &END
  &END SCF
  &XC
    &XC_FUNCTIONAL Pade
    &END XC_FUNCTIONAL
  &END XC
&END DFT
```

Basis and PP library files

Parameters for the realspace multi-grids

Quickstep options

Control of SCF procedure, including minimisation scheme

Exchange-Correlation Functional (LDA)

CP2K Input file: The How

```
 &SUBSYS
  &CELL
    ABC 9.8528 9.8528 9.8528
  &END CELL
  # 32 H2O (TIP5P,1bar,300K) a = 9.8528
  &COORD
O       2.280398       9.146539       5.088696
O       1.251703       2.406261       7.769908
O       1.596302       6.920128       0.656695
...
H       0.837635       8.186808       8.987268
H       8.314696      10.115534       2.212519
H       8.687134       8.667252       2.448452
  &END COORD
  &KIND H
    BASIS_SET TZV2P-GTH
    POTENTIAL GTH-PADE-q1
  &END KIND
  &KIND O
    BASIS_SET TZV2P-GTH
    POTENTIAL GTH-PADE-q6
  &END KIND
&END SUBSYS
```

Cell definition

Particle coordinates

Could also @include an external file or parse other formats via

```
&TOPOLOGY
   COORD_FILE_NAME
&END TOPOLOGY
```

Definitions of atomic kinds

Could specify charge, mass …

CP2K Input file: The What

MOTION section

```
&MOTION
  &MD
    ENSEMBLE NVE
    STEPS 10
    TIMESTEP 0.5
    TEMPERATURE 300.0
  &END MD
&END MOTION
```

Also used to control Geometry Optimisation, NEB, Monte Carlo, …

Basis Sets and PP libraries

- CP2K uses Goedecker-Teter-Hutter, seperable Pseudopotentials
  - Several sets of PPs and corresponding optimised basis sets are available
  - See `cp2k/data` or online: http://sourceforge.net/p/cp2k/code/HEAD/tree/trunk/cp2k/data
  - POTENTIAL, GTH_POTENTIALS
    - Wide range of PPs for many elements
    - Optimised with different XC functional: LDA (PADE), PBE, BLYP …
  - BASIS_SET, GTH_BASIS_SET, BASIS_MOLOPT
    - Various qualities / size of basis
    - Make sure Basis and PP match (functional and number of electrons)
    - Some documentation and references at head of each file

CP2K Output: Controlling what gets written

- The `PRINT_LEVEL` keyword in `&GLOBAL`
  - `SILENT`, `LOW`, `MEDIUM` (default), `HIGH`, `DEBUG`
  - `HIGH` can give more information if you are interested
    - Also gives some per-process logging in parallel jobs
  - For long MD runs (e.g. classical), recommend using `LOW`
- Fine grained control is available via print-keys
  - Most input sections contain a `&PRINT` sub-section
  - Each `&PRINT` sub-section has further subsections for each quantity that may be printed

CP2K Output: Controlling what gets written

- For example, the `&PRINT` section in `&MOTION` contains

  `&CELL`

  `&FORCES`

  `&TRAJECTORY`

  `&VELOCITIES`

  …

- Each section has parameters (and defaults) for which print level it is output

  `&TRAJECTORY` defaults to `LOW`

  `&VELOCITIES` defaults to `HIGH`

CP2K Output: Controlling what gets written

- Can also specify frequency of printing via `&EACH` sub-section e.g.

```
&PRINT
   &CELL
      &EACH
         MD 100
      &END EACH
   &END CELL
&END PRINT
```

- Control over filenames, file formats etc. in each `&PRINT` section

CP2K Output: Overview of an output file

...

Restarting a calculation

- If you need to restart your job…
  - Hardware failure
  - Batch system time limit
  - Need more MD sampling
  - …

- CP2K dumps a restart input file which can be directly re-run
  - `cp2k.sopt -i PROJECT-1.restart`
  - Continuous numbering of MD steps
  - Stores all state variables (incl. extended system)
  - Use `SCF_GUESS RESTART`

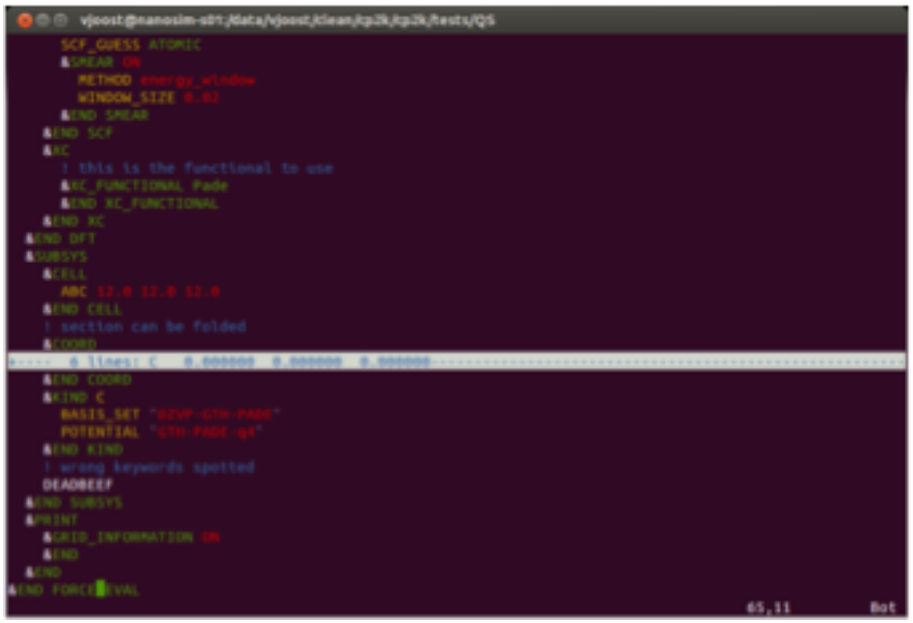
Tools for building CP2K input

Plugins are available for your favourite(!) text editors:

https://www.cp2k.org/tools:vim

https://www.cp2k.org/tools:emacs

Syntax highlighting, indentation, show/hide sections, keywords



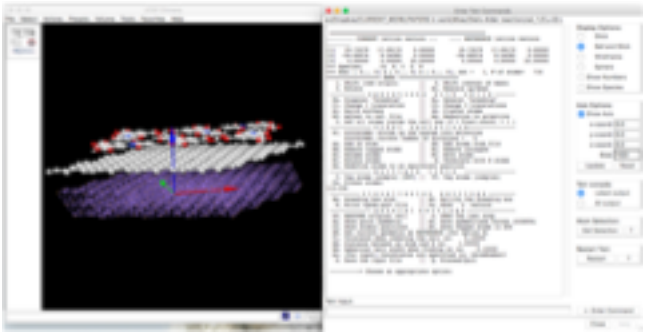
Tools for building CP2K input

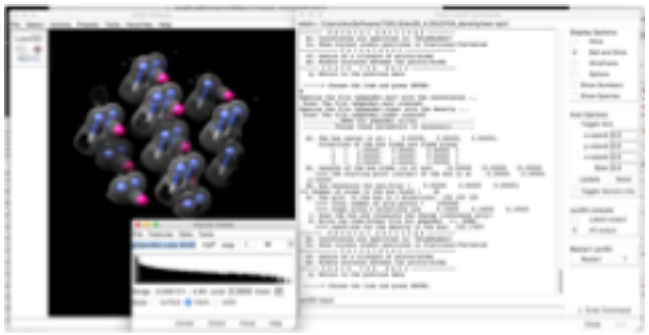
- Python interfaces:
  - Atomic Simulation Environment (ASE, https://wiki.fysik.dtu.dk/ase/)
    - Fully featured Python environment for atomistic simulation
    - System setup, analysis and visualisation
    - Support for many codes – including CP2K
  - PyCP2K (https://github.com/SINGROUP/pycp2k)
    - Object-oriented wrapper following the CP2K input format
    - e.g. `GLOBAL%RUN_TYPE` is `GLOBAL.Run_type`
    - Auto-completion (for e.g. Spyder IDE)
    - May use ASE for execution

```
23
24 #====================== Write the simulation input ==========================
25 GLOBAL.Run_type = "ENERGY_FORCE"
26 GLOBAL.Print_level = "LOW"
27 FORCE_EVAL.Method = "Quickstep"
28 FORCE_EVAL.PRINT.FORCES.Section_parameters = "ON"
29 DFT.Basis_set_file_name = "BASIS_SET"
30 DFT.Potential_file_name = "POTENTIAL"
31 DFT.QS.Eps_default = 1.0E-10
32 SCF.Scf_guess = "ATOMIC"
33 SCF.Eps_scf = 1.0E-7
34 SCF.Max_scf = 300
35 SCF.DIAGONALIZATION.Section_parameters = "ON"
36 SCF.DIAGONALIZATION.Algorithm = "STANDARD"
37 SCF.MIXING.Section_parameters = "T"
38 SCF.MIXING.Method = "BROYDEN_MIXING"
39 SCF.MIXING.Alpha = 0.4
40 SCF.MIXING.Nbroyden = 8
41 KIND = SUBSYS.KIND_add("Si")  # Section_parameters can be provided as argument.
42 KIND.Basis_set = "DZVP-GTH-PADE"
43 KIND.Potential = "GTH-PADE-q4"
44 calc.create_cell(SUBSYS, lattice)
45 calc.create_coord(SUBSYS, lattice)
46
47 #====================== Run the simulation ==========================
48 calc.run()
~
~
3 fewer lines                                                    31,1          Bot
```

Tools for building CP2K input

- GUI setup tools
  - UCSF Chimera plugins (https://github.com/gpsgibb/tetr_lev00_Chimera_plugin)
    - Menu-driven + visualisation
    - TETR: setting up geometry
      - Supercell, surfaces, clusters, …
    - LEV00: analysis
      - Visualising charge/spin densities
      - DOS, phonons, IR spectra, …





  - Avogadro
    - CP2K supported in Avogadro 1
      - https://github.com/brhr-iwao/libavogadro1cp2k
    - Experimental support in Avogadro 2
      - https://github.com/infuniri/avogadrolibs-cp2k

Questions?