CP2K: INTRODUCTION AND ORIENTATION

4th CECAM CP2K Tutorial, 31 Aug – 4 Sep 2015

Iain Bethune

ibethune@epcc.ed.ac.uk

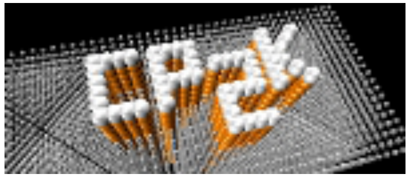
Outline

- CP2K Overview
  - Project History
- CP2K Features
- CP2K Information
- Obtaining CP2K
- CP2K exercises

CP2K Overview

"CP2K is a program to perform atomistic and molecular simulations of solid state, liquid, molecular, and biological systems. It provides a general framework for different methods such as e.g., density functional theory (DFT) using a mixed Gaussian and plane waves approach (GPW) and classical pair and many-body potentials."

From www.cp2k.org (and original home page from 2004!)



CP2K Overview

- Many force models:
  - Classical
  - DFT (GPW, GAPW + vDW)
  - LS-DFT
  - Hybrid Hartree-Fock
  - post-HF (MP2, RPA)
  - Combinations (QM/MM, mixed)

Simulation tools:

  - MD (various ensembles)
  - Monte Carlo
  - Minimisation (GEO/CELL_OPT)
  - Properties (Spectra, excitations …)

Open Source

  - GPL, www.cp2k.org
  - 1m loc, ~2 commits per day
  - ~10 core developers

CP2K History

- 25th June 2001 – CP2K repository online at berliOS.de
  - Merger of Quickstep (DFT) + FIST (MD) codes
  - Jürg Hutter, Matthias Krack, Chris Mundy

- Oct 2011 – First 'official' release
  - CP2K 2.2

- 14 years on…
  - 1m lines of code, ~16k commits
  - 25 developers + many contributors
  - 100s of users
  - Fully open-source (GPL)

CP2K SOURCE CODE DEVELOPMENT

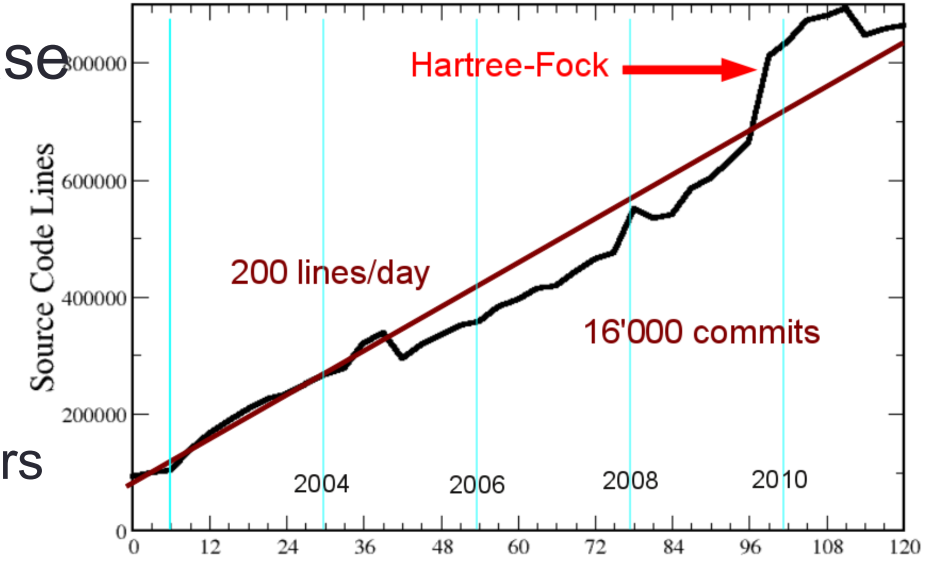
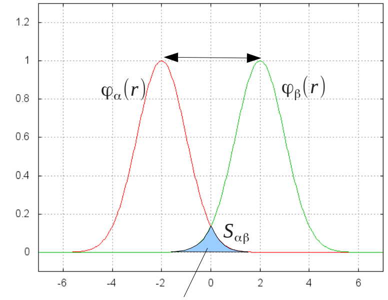
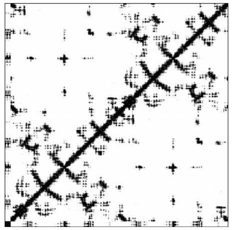


Image from Jürg Hutter

CP2K Features

- QUICKSTEP DFT: Gaussian and Plane Waves Method (VandeVondele *et al*, Comp. Phys. Comm., 2005)
  - Advantages of atom-centred basis (primary)
    - Density, KS matrices are sparse
  - Advantages of plane-wave basis (auxiliary)
    - Efficient computation of Hartree potential
  - Efficient mapping between basis sets
    - -> Construction of the KS Matrix is ~O(n)





- Orbital Transformation Method (VandeVondele & Hutter, J. Chem. Phys., 2003)
  - Replacement for traditional diagonalisation to orthogonalise wave functions (non-metallic systems only)
    - Cubic scaling but ~10% cost

CP2K Features

- QM/MM (Laino *et al*, JCTC, 2005, 2006)
  - Fully periodic, linear scaling electrostatic coupling

- Gaussian and Augmented Plane Waves (Iannuzzi *et al*, CHIMIA, 2005)
  - Partitioning the electronic density -> all-electron calculations

- Hartree-Fock Exchange (Guidon *et al*, JCP, 2008)
  - Beyond local DFT (later MP2, RPA…)
  - Auxiliary Density Matrix Method (Guidon *et al*, JCTC, 2010)

- Linear Scaling DFT (VandeVondele, Borstnik & Hutter, JCTC, 2012)
  - Fully linear scaling condensed-phase DFT, up to ~1m atoms

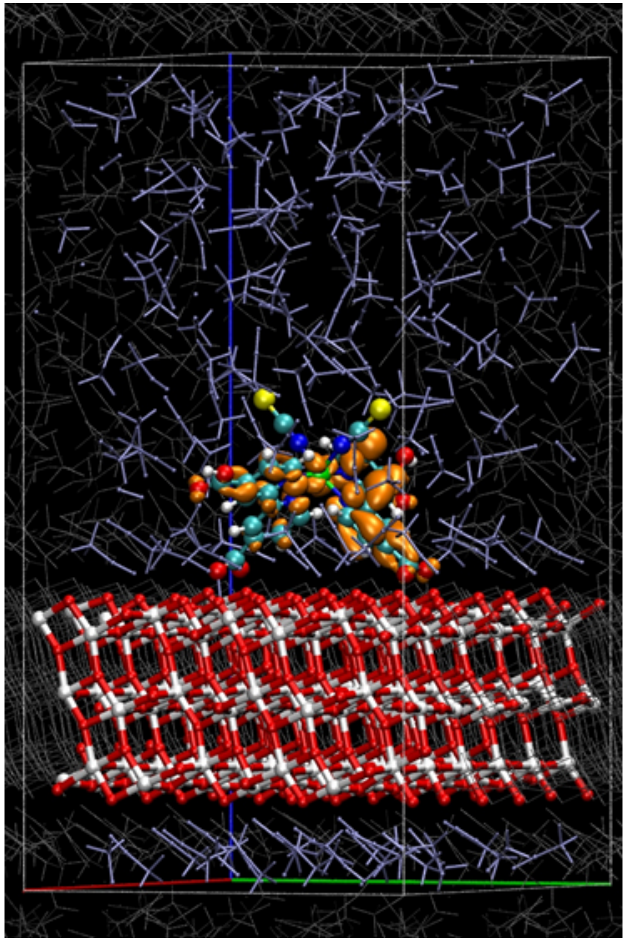
CP2K Features

- And LOTS more…
  - Recent review paper:

    Hutter *et al*, *WIREs Comput Mol Sci* 2014, 4:15–25
    http://dx.doi.org/10.1002/wcms.1159

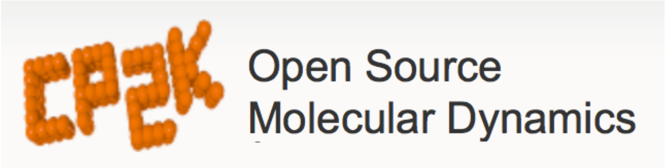
- Some highlight applications:
  - http://www.cp2k.org/science

- All for free!
  - Please cite the references
  - Please give feedback / patches / feature requests
  - Please spread the word about CP2K!



DSSC: see Shiffmann *et al*, PNAS, 2010

CP2K Information



Open Source
Molecular Dynamics

- CP2K Website (http://www.cp2k.org)
  - Everything else is linked from here
  - Now a wiki – so feel free to contribute!

- CP2K Sourceforge site (http://sf.net/p/cp2k) :
  - Contains source code repository (SVN)
    - public read-only, read-write access to developers
  - Bug reporting
  - Source tarball / binary downloads

CP2K Information

- CP2K Discussion Group (http://groups.google.com/group/cp2k)
  - Email / web forum
  - Users and developers
  - Searchable history

- CP2K Input reference manual (http://manual.cp2k.org)
  - Documents *every* possible CP2K input keyword
  - Mostly with helpful descriptions
  - More later…

Obtaining CP2K

- Which version?
  - Current release 2.6 (Dec 2014) / 2.6.1 (May 2015)

  + stable, major bug-fixes are back-ported

  + source and binaries available from http://www.cp2k.org/download

  + available for Ubuntu / Debian / Fedora via package managers

  - missing latest features, minor bugs are not always fixed

  http://www.cp2k.org/version_history

  - SVN trunk version 2.7

  + latest features, fixes, performance improvements

  + actively developed

  - bugs may exist (see dashboard.cp2k.org)

  - must be obtained from SVN and compiled from source

Obtaining CP2K

- CP2K download contents:
  - `README, COPYRIGHT, INSTALL`
  - `src` – source code (mostly Fortran 03, a little C++)
  - `makefiles` – To build CP2K
  - `arch` – machine-specific options files
  - `data` – standard data files (basis sets, PPs …)
  - `tests` – over 2700 input files!
  - `tools` – mostly for developers + cubecruncher
- After building:
  - `lib` – CP2K internal libraries
  - `obj` – compiled object files
  - `exe` – CP2K binaries

CP2K Exercises

- Various exercises are available from:
  - http://www.cp2k.org/exercises
  - See “CECAM 4th CP2K Tutorial” for this week
  - Also older exercises
  - Mostly ‘worked examples’ from system setup and calculations to analysis / visualisation of results

- For specific ‘HowTo’ guides see:
  - http://www.cp2k.org/tutorials
  - Guides to basic (and some advanced!) CP2K skills
    - e.g. converging `CUTOFF` for QS calculations

CP2K Exercises

- The CP2K `tests` directory
  - Great source for example input files for all kinds of calculations
  - Grouped (mostly) logically:
    - `QS/regtest-gpw-1` – Quickstep GPW calculations
    - `QS/regtest-dm-ls-scf` – Quickstep using linear scaling SCF
    - `Fist/regtest-opt` – Geometry and Cell optimisations using classical potentials
    - `SE/regtest-*` - various semi-empirical calculations

  - WARNING:
    - Tests are designed to run quickly so may not produce converged or accurate outputs! Check parameters for your system…

CP2K: Introduction and Orientation

Questions?

BASICS OF CP2K CALCULATIONS

Iain Bethune (ibethune@epcc.ed.ac.uk)

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. NSCCS, 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 at many elements - 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. at 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)
  - May want to use `SCF_GUESS RESTART`

Basics of CP2K Calculations

Questions?

BUILDING CP2K

Iain Bethune (ibethune@epcc.ed.ac.uk)

Overview

- Machine Access
- Prerequisites
  - Environment
  - Libraries
- Optional Libraries
  - Functionality
  - Performance
- Arch files and compilation
- Running example input files
- Testing CP2K

Machine Access

- Where can you run CP2K?
  - Own Laptop
    - Serial / OpenMP build
  - Institute workstation / cluster
  - UZH Guest logins
    - CP2K 2.6.0 pre-installed
  - ARCHER Guest accounts
    - Cray XC30 @ EPCC
    - CP2K 2.7 psmp pre-installed, massively parallel calculations

Prerequisites - Environment

- POSIX-compliant OS
  - Linux, UNIX (e.g. AIX) …
  - Cygwin, Mac OS X also possible
- Build tools
  - GNU Make, Python 2.x (or later)
- Compilers
  - GNU gcc / gfortran 4.6 (or later)
  - Intel ifort 15.x
  - IBM XLF 14.1

Prerequisites - Libraries

- BLAS & LAPACK (required)
  - Vendor-tuned libraries preferred (MKL, ACML, ESSL)
  - Free auto-tuned libraries (GotoBLAS, ATLAS)
  - Reference BLAS + LAPACK from Netlib (last resort, very slow!)

- MPI & ScaLAPACK (required for MPI parallel build)
  - Usually provided by your cluster / HPC
  - Require MPI 2.x (3.x optional)
  - OpenMPI. MPICH, Intel MPI, Cray MPT all tested
  - ScaLAPACK provided by vendor maths libraries…
    - … or download from Netlib
  - `-D__parallel -D__SCALAPACK`

Prerequisites - Libraries

- FFTW3 (Recommended)
  - CP2K has an inbuilt FFT implementation
  - FFTW3 will give much better performance
    - + freely available
    - + easy to compile / install
  - Enable using `-D__FFTW3`

Optional Libraries

- Libxc
  - CP2K has various common XC functionals e.g. PBE, LDA, BLYP…
  - Many more available via libxc
  - Version 2.0.1 or later
  - `-D__LIBXC2` or `-D__LIBXC3`

- Libint
  - Required for all Hartree-Fock Exchange calculations
  - Version 1.1.4 only
  - `-D__LIBINT`

Optional Libraries

- ELPA
  - Optimised diagonalisation routines
  - Build process optimises for specific architecture
  - < June 2014 version : `-D__ELPA`
  - >= June 2014 version : `-D__ELPA2`

- All other libraries / options / flags
  - See http://www.cp2k.org/howto:compile
  - and `cp2k/INSTALL`

- Auto-tuned performance libraries (libsmm, libgrid)
  - More on Friday…

Arch files and compilation

- Compiler and architecture-specific options are given in an ‘arch file’
  - Examples in `cp2k/arch`
  - e.g. `Linux_x86-64-gfortran.popt`
  - Copy/customise for your environment

- To build CP2K
  - in the `cp2k/makefiles` directory:

corresponding to arch file

```
make -j 4 ARCH=Linux-x86-64-gfortran VERSION=popt
```

parallel build

Errors? Ask us!

Arch files and compilation

- CP2K binary should be built in
  - `cp2k/exe/<ARCH>/cp2k.<VERSION>`

- Very quick test:

```
cp2k.sopt --version
```

  - MPI binaries (`popt`) should be run with `mpirun`
  - Maybe within a batch script?

- Quick test
  - in the `cp2k/tests/QS` directory:

```
../../exe/ARCH/cp2k.sopt C.inp
```

Testing CP2K

- CP2K comes with a suite of >2600 test input files

- Good for checking you have correctly compiled CP2K
  - Tests that all enabled features of CP2K run
  - Most tests compare against a reference result

- To execute regression tests:
  - Instructions in `cp2k/tools/regtesting`
  - Also online: http://cp2k.org/dev:regtesting

Testing CP2K

- `do_regtest` script
  - SVN update, builds CP2K (`--nosvn -nobuild` to skip)
  - Runs all tests (in parallel, if possible)
  - Takes ~10 mins – a few hours
  - Summary of results and details of any failing tests

```
--------------------------------- Summary --------------------------------
Number of COMPILE warns 0
Number of FAILED  tests 2
Number of WRONG   tests 51
Number of CORRECT tests 2589
Number of NEW     tests 0
Total number of   tests 2642
```

Test failed to complete

Test completed, but does not match reference

Test completed for first time (and no reference result available)

Testing CP2K

- Automatic testing on 30+ different platforms
  - Test failures automatically reported to developers

- Results available online at http://dashboard.cp2k.org

- Check here when using an SVN trunk version

Building CP2K

Questions?