What is CP2K

CP2K is a freely available (GPL) program written in Fortran 95 to perform atomistic and molecular simulations of solid state, liquid, molecular and biological systems. It provides a general framework for different methods: density functional theory (DFT) using a mixed Gaussian and plane waves approach (GPW), classical pair and many-body potentials, semi-empirical (AM1, PM3, MNDO, MNDOd, PM6) Hamiltonians, and Quantum Mechanics/ Molecular Mechanics (QM/MM) hybrid schemes relying on the Gaussian Expansion of the Electrostatic Potential (GEEP).

CP2K Functionality

The key difference between CP2K and other DFT codes is its implementation of the Quickstep algorithm, which uses a dual basis - atom-centred Gaussian functions to represent the wave-functions, and plane waves/regular grids for the electronic density. Functionality includes

- Quickstep algorithms for Hamiltonian construction, integration, collocation, energy minimisation and SCF cycle etc.
- Molecular Dynamics algorithms
- Quantum Monte Carlo algorithms
- FIST classical MD algorithms
- Hybrid OpenMP / MPI parallelism.

Step 1 - Log in

The example used in this guide is configured to run on the Swansea Sandy Bridge cluster. Connect to *login.hpcwales.co.uk* with your HPC Wales user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal), then **`ssh sw-sb-log-001`** to connect to the Swansea system.

The steps below involve typing commands (**`in bold font`**) in the terminal window.

Step 2 - Load a CP2K module

A number of CP2K binary packages are available.

- List pre-installed CP2K versions:
  `module avail cp2k`
- Load the default MPI version (2.4-intel-mpi):
  `module load cp2k`

Ewrop & Chymru: Buddsoddi yn eich dyfodol
Cronfa Datblygu Rhanbarthol Ewrop
Europe & Wales: Investing in your future
European Regional Development Fund
ERDF
Llywodraeth Cymru Welsh Government

- Confirm the loaded modules. All dependencies are handled automatically via the module file:

  ```
  module list
  ```

Step 3 - Create a directory

Create a directory to hold any user data files. For this example, create a directory called cp2k under your home directory:

```
cd ~
mkdir cp2k
```

Step 4 - Obtain a test case

A number of benchmark test cases are provided with the installation at

```
/app/materials/cp2k/2.4/sb/popt/intel-13.1/intel-4.0.3.008/example/
```

- Copy the simplest example to your user space:

  ```
  cd ~/cp2k
  cp /app/materials/cp2k/2.4/sb/popt/intel-13.1/intel-4.0.3.008/example/cp2k_example.tar.gz .
  tar zxf cp2k_example.tar.gz
  cd cp2k_example
  ```

Step 5 - Submit a job

Now you are ready to run these test cases with the supplied job scripts.

From your working directory, submit the job using: `sbatch cp2k.example.SLURM.q`

Check the job queue using: `squeue`

- When the job finishes, it will have created an output file called

  `H2O-tddfpt-saop.out_4.<Job_ID>`

  where <Job_ID> is the ID generated by the system

- Compare your job output with the reference output file

  `H2O-tddfpt-saop.out_4`

Step 6 - More Test Cases

Two additional test cases are provided as tar balls in the directory

/app/materials/cp2k/2.4/sb/popt/intel-13.1/intel-4.0.3.008/example

Ewrop & Chymru: Buddsoddi yn eich dyfodol
Cronfa Datblygu Rhanbarthol Ewrop
Europe & Wales: Investing in your future
European Regional Development Fund
ERDF
Llywodraeth Cymru Welsh Government

both running H_2O clusters of increasing size - *cp2k_H2O-256.tar.gz* and *cp2k_H2O-512.tar.gz*. Repeat the procedure outlined above for these two cases. Note that the $(H_2O)_{256}$ DFT calculation should take under 6 minutes on 64 cores, while the more extensive $(H_2O)_{512}$ calculation requires ca. 27 minutes on the same number of cores. Compare your results with the output files H2O-256.out_64 and H2O-512.out_64 provided in the associated tar files.

References

Documentation and source code is available at the CP2K website: http://www.cp2k.org/

A variety of useful tutorials are also available at: http://www.cp2k.org/tutorials