ONETEP workshop: Compiling and running a simple job

Peter Haynes
Imperial College London

Aim: To compile ONETEP and run some simple test jobs.

Compiling ONETEP

Log onto PWF Linux and open a terminal window.
The ONETEP distribution is available from the URL
http://www.tcm.phy.cam.ac.uk/onetep/workshop/
for which you will the user name and password were distributed at the workshop.
Point a web browser at this address and save the file `onetep-2.2.0.tar.gz` in your home directory.

Unpack the distribution:

```
> gunzip -c onetep-2.2.0.tar.gz | tar xf -
> cd ONETEP-2.2.0
```

The configuration files that specify which compiler and libraries to use are found in the `config` directory:

```
> cd config
```

On the PWF Linux system the best compiler is gfortran, and the BLAS, LAPACK and FFTw libraries are already installed. We will build a serial version of the code and run some simple tests.
Have a look at the file `conf.i386_gfortran`:

```
> cat conf.i386_gfortran
```

In fact this file will suffice to build ONETEP on PWF Linux, although it will not be well optimized. Make a copy of the file with an easier name e.g.

```
> cp conf.i386_gfortran conf.pwf
```

Now return to the `ONETEP-2.2.0` directory:

```
> cd ..
```

The ONETEP build system relies on GNU make, which is usually available as `gmake`. You can get some basic help on the build system by typing `gmake` on its own. More information about the build system may be found in the `INSTALL` file.

```
> gmake
```

Build ONETEP using the `conf.pwf` file by typing

```
> gmake onetep ARCH=pwf
```

The compilation will take a few minutes. Once finished, the executable `onetep.pwf` should have been created in the bin directory.
> ls bin

It is now necessary to test that the compiled executable runs correctly. Return to your home directory and create a new `test` directory.

```
> cd
> mkdir test
> cd test
```

It is better to create a symbolic link to the ONETEP executable rather than copying. Do this by typing:

```
> ln -s ../ONETEP-2.2.0/bin/onetep.pwf .
```

A simple input file for the ethene molecule can be found on the workshop web page. Right-click on the link, select "Save link as..." and save into the `test` directory.

The pseudopotential files may be copied from the `quality_control/PSEUDOS` directory of the ONETEP distribution:

```
> cp ../ONETEP-2.2.0/quality_control/PSEUDOS/hydrogen.recpot .
> cp ../ONETEP-2.2.0/quality_control/PSEUDOS/carbon.recpot .
```

You are now ready to run your first ONETEP job:

```
> ./onetep.pwf ethene
```

(If you wanted to run the job in the background and redirect the output into a file you would have typed:

```
> ./onetep.pwf ethene >ethene.out 2>&1 &
```

)

This test should take less than ten minutes to run. When it has finished, you can compare the results with the output file on the workshop web page.

Running a simple job

We will run a simple job on a silane molecule.
Create a working directory in which to run ONETEP:

```
> cd
> mkdir silane
> cd silane
```

Create a new input file called silane.dat in your favourite text editor e.g.

```
> nedit silane.dat &
```

(If using nedit it will complain that the file does not exist – click on "New file" to create it.)

You might like to put a comment at the top explaining what this input file is for e.g.

```
# Simple ONETEP input file for a silane molecule
```

The first thing is to specify the simulation cell. The simplest choice is a cubic shape with sides of about 40 bohr. Enter the cell vectors between the `%block lattice_cart` and `%endblock lattice_cart` keywords.

Second, the atomic species need to be specified, in this case silicon and hydrogen. This information needs to be provided between the `%block species` and `%endblock species` keywords. Remember to specify (i) your symbol for the atomic species, (ii) the element symbol, (iii) the atomic number Z, (iv) the number of NGWFs per atom (usually four for silicon and one for hydrogen) and (v) the NGWF radius – 6 bohr should be a reasonable starting point.

Next specify the atomic positions, in the `positions_abs` block. There is one line per atom. Remember to use your symbol for the atomic species as defined in the `species` block and to give the Cartesian coordinates in bohr. It is currently a requirement in ONETEP that all the atoms should lie within the simulation cell so it is best to start by placing the silicon atom at the centre of the cell. The Si-H bond length is about 2.76 bohr and silane is a tetrahedral molecule. The simplest way to work out the coordinates is to note that tetrahedral bonds can be chosen to lie along unit vectors (a,b,0), (-a,b,0), (0,-b,a) and (0,-b,-a) where $a = \sqrt{2/3}$ and $b = 1/\sqrt{3}$. For example, the vector for the first Si-H bond is (2.2535,1.5935) bohr.

The pseudopotential files are specified in the `species_pot` block. You can use the `hydrogen.recpot` and `silicon.recpot` files from the `quality_control/PSEUDOS` directory of the ONETEP distribution.

The last essential parameter to specify is the kinetic energy cutoff parameter for the psinc basis set. A reasonable value to start with is 300 eV. Use the `cutoff_energy` keyword and remember to specify the energy unit as well as the value.

Save the changes to the `silane.dat` file. If you want to check it, there is an example on the workshop web page.
Create a link to the ONETEP executable as before:

```
> ln –s ../ONETEP-2.2.0/bin/onetep.pwf .
```

Copy the pseudopotential files:

```
> cp ../ONETEP-2.2.0/quality_control/PSEUDOS/hydrogen.recpot .
> cp ../ONETEP-2.2.0/quality_control/PSEUDOS/silicon.recpot .
```

Run the job:

```
> ./onetep.pwf silane >silane.out 2>&1 &
```

Compare the output file to the one on the workshop web page.

Here are some suggestions for further investigation:

- Converge the total energy with respect to:
    - NGWF radius
    - Kinetic energy cutoff
    - Number of NGWFs per atom
- Different exchange-correlation functionals may be specified using the `xc_functional` keyword – see the online documentation for allowed values
- Forces may be calculated at the end of the calculation with the `write_forces` keyword

You may now wish to set up a larger run for Darwin, the Cambridge High-Performance computer. One computational unit (65 quad-core nodes with 8 GB memory each) is available to us for the duration of the workshop. You should be able to log in to Darwin without a password from your visitor account on the PWF. A compiled ONETEP executable is already available. Please talk to a demonstrator about running jobs on Darwin.