

INSTRUCTIONS FOR DAY 1, EXERCISE 1

Note that the following color code has been used in this instruction sheet:

Broad headings are in red.

File names are in magenta.

Phrases to be typed into the command line are in blue.

Input parameters are in dark green.

A very brief reminder of useful linux commands:

- `cp A B` copies file `A` to file `B`.
- `mv A B` moves file `A` to file `B`.
- `mkdir C` creates a directory named `C`.
- `ls C` lists the names of the files and sub-directories contained in directory `C`
- `vi A` or `gedit A` are some of the many ways in which you can open the file `A` for reading and/or editing.
- Adding an ampersand `&` at the end of your command will make it run in the background, so that you can continue to issue other commands while the original one is still running.

In this exercise, you will first perform simple scf (self-consistent field) calculations on silicon.

- Go to the directory `day1/exercise1` (if you are not already there!)

You will see the following files:

`day1_exercise1_instructions.html` this file!

`Si.sample.in` this is a sample input file, for a primitive Si cell containing 2 atoms.

`Si.sample.sh` this is a sample shell script that you can use (optionally) .

`Si.supcell1.in` this is an input file for an Si supercell containing 8 atoms.

`Si.supcell2.in` this is an input file for an Si supercell containing 16 atoms.

[Si.pbe-rrkj.UPF](#) this is the pseudopotential file.

For each calculation that you will run, you can use [Si.sample.in](#) as a template that you can copy to another file and then edit. Alternatively, you can use shell scripts to automate this process.

- Open and read the sample file [Si.sample.in](#)
- Note that in the `&control` namelist, we have specified that we want to run an `scf` calculation.
- Silicon has the diamond structure. Note that we are using a (primitive) unit cell that corresponds to an fcc lattice, with 2 Si atoms in the atomic basis. Notice the values given for `ibrav`, `nat`, `ntyp` and `ATOMIC_POSITIONS`.
- In this file, the lattice constant `celldm(1)` has been set equal to $10.26 \text{ bohr} = 5.43 \text{ \AA}$, which is the experimental value.
- A $6 \times 6 \times 6$ shifted Monkhorst-Pack k-point mesh has been used.
- The plane-wave cut-off for wavefunctions, `ecutwfc`, has been set to 20 Ry. Since we are not using an ultrasoft pseudopotential, we are leaving `ecutrho` at the default value.
- Use Xcrysden to view the structure in the sample input file:

`xcrysden --pwi Si.sample.in`

- Play around with xcrysden! Rotate the view, display the boundaries of the conventional cubic cell, etc.
- How many atoms are contained in the conventional cubic cell?
- What is the coordination number of each atom?
- What is the nearest-neighbor distance?
- Run an scf calculation:

The way to do this is:

```
pw.x < input_filename > output_filename
```

So in this particular case, you will type:

```
pw.x < Si.sample.in > Si.sample.out
```

- Read the output, and answer the following questions:
- Was symmetry used to reduce the number of k-points from $6 \times 6 \times 6 = 216$?
- How many scf iterations were performed before convergence was achieved?
- What is the final (converged) value for the total energy? (Note: you can find this by looking for an exclamation mark in the output file). For example, you can type:

```
grep ! Si.sample.out
```

- Note: If you set `verbosity = 'high'` in the `&control` namelist, you will get additional information in your output file, e.g., about symmetries and occupation numbers.
- Now do convergence tests with respect to plane-wave cut-off:
- Find out how the total energy changes when you vary the plane wave cut-off `ecutwfc` between 10 and 30 Ry, in steps of 5 Ry. You can either do this by copying the sample file `Si.sample.in` to other input files and editing them, and running `pw.x` using each of these input files in turn, or by running the sample shell script `Si.sample.sh`

```
./Si.sample.sh
```

- Now make a data file (e.g., you can call it `etotvsecut.dat`) that contains the value of `ecutwfc` in the first column, and total energy in the second column. Remember that you can easily find the values of the total energy by searching for an exclamation mark, e.g., using the `grep` command.
- Plot a graph of this data using your favorite plotting program (e.g., `xmgrace` or `gnuplot`).
- Does the total energy decrease monotonically as `ecutwfc` is increased?
- What value of `ecutwfc` are you 'happy' with?

- Now examine convergence with respect to Brillouin zone sampling:

- Fix the value of `ecutwfc` at whatever value you have decided is good enough. (Remember: differences in energy will converge at lower values of `ecutwfc` than the absolute value of the total energy!) I chose `ecutwfc` to be 20 Ry, but you can make whatever choice you want!
- Let the lattice constant `cellldm(1)` remain at the experimental value.

Vary the Monkhorst-Pack grid parameters `nk1`, `nk2` and `nk3`.

You can do this either by manually editing `input files`, or by modifying the script `Si.sample.sh`

Use `nk1=nk2=nk3= 1, 2, 3, 4, 5, 6`.

Leave the offset at `k1=k2=k3 = 1`.

- Run `pw.x` for each of the six `input files` you have made (either one by one, or using the shellscript).

Remember: don't overwrite `output files` before you extract data from them!

- Assemble a `data file` that contains how the total energy varies with `nk1=nk2=nk3`. The first column should contain the value of `nk1` (or, if you prefer, the number of k-points generated and used), and the second column should contain the value of the total energy.
- Plot a graph of this data. Do you think that the total energy is converging as you increase the number of k-points?
- Do you think it was a good choice to take `nk1=nk2=nk3`? Why?
- Is the convergence wrt the number of k-points monotonic? (Note that it need not

necessarily be so!)

- Now obtain the equilibrium lattice constant of Si:
- Fix values of `ecutwfc` and `nk1=nk2=nk3`. I chose `ecutwfc = 20` and `nk1=nk2=nk3=6`, but you can choose other values if you feel that they are more appropriate.
- Make input files in which you vary the lattice constant, i.e., `celldm(1)`, from 9.8 to 11.0 bohr, in steps of 0.2 bohr. (Again, you can do this by hand or modify the `shell script`).
- Run these seven scf calculations.

Assemble the results for the dependence of total energy on lattice constant in a data file, e.g., one called `etotvsalat.dat` (2 columns, the first containing the lattice constant in bohr, and the second containing total energy in Ry).

- Plot this data.
- By looking at this data, do you think that your result for the equilibrium lattice constant is larger or smaller than experiment? Why is there a discrepancy between the theoretical and experimental values?
- Is the energy symmetric about the minimum value? Would you expect it to be so?
- Fit this data to an equation of state by running

`ev.x`

When you run this, you will be prompted to supply:

- the type of Bravais lattice (note: using the name and not the `ibrav` number)
- which particular equation of state you want to use
- (I usually use the 2nd Order Birch eq., but you can experiment with other choices too.)
- the name of the `input file` containing the data of lattice constant and total energy
- the name of the `output file` where you want the results of the fitting procedure
- Look at the `output file` (whose name you supplied in the previous step):
- From the fit, what are the values of the equilibrium lattice constant a_0 and bulk modulus k_0 ? (The experimental value for the bulk modulus is 101.97 GPa = 1019.7

kbar).

- If there is a discrepancy between experimental and theoretical values, can you understand where this arises from?
- Can you tell whether the quality of fit is good or poor?

The equivalence between supercell size and BZ sampling:

Let us now do a couple of calculations using supercells, i.e., unit cells that are larger than the primitive one.

- We will now use two **input files**:
- **Si.supcell1.in** makes use of the conventional cubic cell, containing 8 atoms. Read this file, and notice the ways in which it differs from the input files you have been using up to now. Especially note the values used for **ibrav**, **nat** and the atomic coordinates.
- The lattice constant has been set equal to the computed value of 10.348 Bohr
- It uses an unshifted k-point mesh of 6×6×6.
- **Si.supcell2.in** makes use of a tetragonal unit cell, which is twice as long in the z-direction as it is in the other two directions. It contains 16 atoms. Read it carefully!
- Since it is twice as long along z, we will halve the number of k-points along k_z .
- So we will use an unshifted k-point mesh of 6×6×3
- Run **pw.x** using both **input files** (remember to use different names for the **output files** in the two cases!)
- Is the total energy per atom the same in the two cases or different? What is the expected answer?
- Which calculation took longer? Note that in the first case we had fewer atoms but more k-points, while in the second case, we had more atoms but fewer k-points. The cells and k-points were chosen such that the second calculation 'folds' exactly into the first one.

(D) Some optional extra stuff :

If you finished all of the above stuff, and the instructor still hasn't moved on to the next exercise, you can (if you choose) do other things, e.g.:

- Obtain graphs of total energy vs. lattice constant at some very low values of ϵ_{cutwfc} .
- Do you notice any change in the shape or nature of the curve? If yes, do you understand why?