

# An introduction to density functional theory for experimentalists

## Tutorial 2.1

As usual we create a new folder on the HPC cluster:

```
$ cd ~/scratch/summerschool ; mkdir tutorial-2.1 ; cd tutorial-2.1
```

### Equilibrium structure of a diatomic molecule

In this tutorial we are going to learn how to calculate the equilibrium structures of simple systems. The formal theory required for these calculations will be discussed in Lecture 3.1, for now we can just use the following rule of thumb:

*Among all possible structures, the equilibrium structure at zero temperature and zero pressure is found by minimizing the DFT total energy.*

The total potential energy is the same quantity that we have been using during Tutorial 1.1 and Tutorial 1.2 (eg when we did `grep "\!" silicon-1.out`). This quantity includes all terms of the electron-ion Hamiltonian, except the kinetic energy of the ions.

Let us calculate the equilibrium structure of the  $\text{Cl}_2$  molecule.

The  $\text{Cl}_2$  molecule has only 2 atoms, and the structure is completely determined by the Cl–Cl bond length. Therefore we can determine the equilibrium structure by calculating the total energy as a function of the Cl–Cl distance.

The first step is to find a suitable pseudopotential for Cl. As in Tutorial 1.1 we go to <http://www.quantum-espresso.org/pseudopotentials> and look for Cl. We recognize LDA pseudopotential by the label pz in the filename. Let us go for the following:

```
$ wget http://www.quantum-espresso.org/wp-content/uploads/upf_files/Cl.pz-bhs.UPF
```

We also copy the executable, job submission script, and input file from the previous tutorial:

```
$ cp ../tutorial-1.1/pw.x ./
$ cp ../tutorial-1.1/job-1.pbs ./
$ cp ../tutorial-1.1/silicon-1.in ./cl2.in
```

Now we can modify the input file in order to consider the  $\text{Cl}_2$  molecule:

```
$ more cl2.in

&control
  calculation = 'scf'
  prefix = 'Cl2',
  pseudo_dir = './',
  outdir = './'
/
```

---

```

&system
 ibrav = 1,
celldm(1) = 20.0,
nat = 2,
ntyp = 1,
ecutwfc = 100,
/
&electrons
conv_thr = 1.0d-8
/
ATOMIC_SPECIES
Cl 1.0 Cl.pz-bhs.UPF
ATOMIC_POSITIONS bohr
Cl 0.00 0.00 0.00
Cl 2.00 0.00 0.00
K_POINTS gamma

```

Using `ibrav = 1` we select a simulation box which is simple cubic, with lattice parameter `celldm(1)`. Here we are choosing a cubic box of side 20 bohr (1 bohr = 0.529167 Å). The keyword `gamma` means that we will be sampling the Brillouin zone at the  $\Gamma$  point, that is  $\mathbf{k} = 0$ . This is fine since we want to study a molecule, not an extended crystal. Note that we increased the planewaves cutoff to 100 Ry: this number was obtained from separate convergence tests.

We can now submit a job in order to check that everything will run smoothly, `qsub job.pbs`. In this case it is appropriate to use the submission execution flags `-np 8, -npool 1`. Incidentally, from the output file of this run (say `c12.out`) we can see the various steps of the DFT self-consistent cycle (SCF). For example if we look for the total energy:

```

$ grep "total energy" c12.out
total energy = -55.44311765 Ry
total energy = -55.73166227 Ry
total energy = -55.83945219 Ry
total energy = -55.84156819 Ry
total energy = -55.84161956 Ry
total energy = -55.84162104 Ry
total energy = -55.84162117 Ry
! total energy = -55.84162119 Ry
The total energy is the sum of the following terms:

```

Here we see that the energy reaches its minimum in 8 iterations. The iterative procedure stops when the energy difference between two successive iterations is smaller than `conv_thr = 1.0d-8`.

Now we calculate the total energy as a function of the Cl–Cl bond length. In the reference frame chosen for the input file above we have one Cl atom at (0,0,0) and one at (2,0,0) in units of bohr. Therefore we can vary the bond length by simply displacing the second atom along the  $x$  axis. In order to automate the procedure we can use the following script:

```

sed "s/2.00/NEW/g" c12.in > tmp
foreach DIST ( 2.2 2.4 2.6 2.8 3.0 3.2 3.4 3.6 3.8 4.0 4.2 4.4 4.6 )
  sed "s/NEW/${DIST}/g" tmp > c12_${DIST}.in
end

```

---

If we create this script using copy/paste in a vi window (say vi myscript.tcsh), then we can simply issue tcsh myscript.tcsh in order to generate identical files which will differ only by the Cl-Cl bond length:

```
$ ls cl2_*
cl2_2.2.in  cl2_2.4.in  cl2_2.6.in  cl2_2.8.in  cl2_3.0.in  cl2_3.2.in
cl2_3.4.in  cl2_3.6.in  cl2_3.8.in  cl2_4.0.in  cl2_4.2.in  cl2_4.4.in
cl2_4.6.in
```

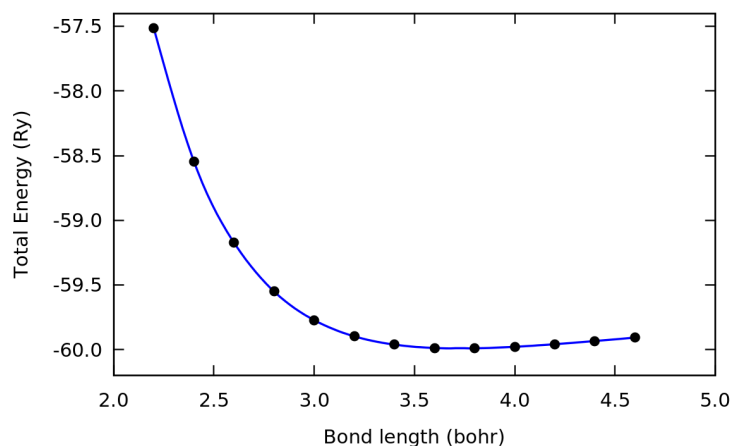
At this point we can execute a batch job which will run pw.x for each of these input files. To this aim we simply duplicate the call to the executable inside our job-1.pbs, and modify this call in order to use the correct input file, eg:

```
mpirun -n 24 pw.x -npool 1 < cl2_2.2.in > cl2_2.2.out
mpirun -n 24 pw.x -npool 1 < cl2_2.4.in > cl2_2.4.out
...
...
mpirun -n 24 pw.x -npool 1 < cl2_4.4.in > cl2_4.4.out
mpirun -n 24 pw.x -npool 1 < cl2_4.6.in > cl2_4.6.out
```

After running these calculations we can look for the total energy as usual:

```
$ grep "\!" cl2_*.out
cl2_2.2.out:!    total energy           =    -57.51390376 Ry
cl2_2.4.out:!    total energy           =    -58.54440037 Ry
cl2_2.6.out:!    total energy           =    -59.17256643 Ry
cl2_2.8.out:!    total energy           =    -59.55021751 Ry
cl2_3.0.out:!    total energy           =    -59.77201221 Ry
cl2_3.2.out:!    total energy           =    -59.89671133 Ry
cl2_3.4.out:!    total energy           =    -59.96077584 Ry
cl2_3.6.out:!    total energy           =    -59.98691109 Ry
cl2_3.8.out:!    total energy           =    -59.98945934 Ry
cl2_4.0.out:!    total energy           =    -59.97764683 Ry
cl2_4.2.out:!    total energy           =    -59.95749243 Ry
cl2_4.4.out:!    total energy           =    -59.93293866 Ry
cl2_4.6.out:!    total energy           =    -59.90658486 Ry
```

By extracting the bond length and the energy from this data we can obtain the plot shown below:



---

In this plot the black dots are the calculated datapoints, and the blue line is a spline interpolation. In `gnuplot` this interpolation is obtained using the flag `'smooth csplines'` at the end of the plot command.

By zooming in the figure we find that the bond length at the minimum is  $3.725 \text{ bohr} = 1.97 \text{ \AA}$ . This value is 1.5% below the measured bond length of  $1.99 \text{ \AA}$ .

## Binding energy of a diatomic molecule

The total energy of  $\text{Cl}_2$  at the equilibrium bond length can be used to calculate the dissociation energy of this molecule.

The dissociation energy is defined as the difference  $E_{\text{diss}} = E_{\text{Cl}_2} - 2E_{\text{Cl}}$ , with  $E_{\text{Cl}}$  the total energy of an isolated Cl atom.

In order to evaluate this quantity we first calculate  $E_{\text{Cl}_2}$  using the equilibrium bond length determined in the previous section. For this we modify the input file `c12.in` as follows:

```
...
ATOMIC_POSITIONS bohr
  Cl 0.00 0.00 0.00
  Cl 3.725 0.00 0.00
...
```

A calculation with this modified input file yields the total energy

$$E_{\text{Cl}_2} = -59.99059545 \text{ Ry}$$

Now we consider the isolated Cl atom.

The only complication in this case is that the outermost ( $3p$ ) electronic shell of Cl has one unpaired electron:  $\uparrow\downarrow \uparrow\downarrow \uparrow$ . In order to take this into account we can perform a **spin-polarized** calculation using the following modification of the previous input file:

```
&control
  calculation = 'scf'
  prefix = 'Cl2',
  pseudo_dir = './',
  outdir = './'
/
&system
 ibrav = 1,
  celldm(1) = 20.0,
  nat = 1,
  ntyp = 1,
  ecutwfc = 100,
  nspin = 2,
  tot_magnetization = 1.0,
  occupations = 'smearing',
  degauss = 0.001,
/
&electrons
  conv_thr = 1.0d-8
/
```

---

```
ATOMIC_SPECIES
Cl 1.0 Cl.pz-bhs.UPF
ATOMIC_POSITIONS
Cl 0.00 0.00 0.00
K_POINTS gamma
```

After running `pw.x` with this input file, we obtain a total energy

$$E_{Cl} = -29.86386108 \text{ Ry}$$

By combining the last two results we find

$$E_{\text{diss}} = 0.262873 \text{ Ry} = 3.58 \text{ eV}$$

This result should be compared to the experimental value of 2.51 eV from [https://en.wikipedia.org/wiki/Bond-dissociation\\_energy](https://en.wikipedia.org/wiki/Bond-dissociation_energy). We can see that DFT/LDA overestimates the dissociation energy of  $\text{Cl}_2$  by about 1 eV: interatomic bonding is slightly too strong in LDA.

## Equilibrium structure of a bulk crystal

In this section we study the equilibrium structure of a bulk crystal. We consider again a silicon crystal, since we already studied the convergence parameters in Tutorial 1.2. We can make a new directory, eg

```
cd ~/scratch/summerschool/tutorial-2.1; mkdir silicon
cd silicon
```

and copy the executable, the submission script, the pseudopotential, and the input file from the folder tutorial-1.2. In this case the input file with the converged parameters for planewaves cutoff and Brillouin-zone sampling is:

```
$ more si.in
&control
  calculation = 'scf'
  prefix = 'silicon',
  pseudo_dir = './',
  outdir = './'
/
&system
 ibrav = 2,
  cellldm(1) = 10.28,
  nat = 2,
  ntyp = 1,
  ecutwfc = 25.0,
/
&electrons
  conv_thr = 1.0d-8
/
ATOMIC_SPECIES
Si 28.086 Si.pz-vbc.UPF
ATOMIC_POSITIONS
Si 0.00 0.00 0.00
Si 0.25 0.25 0.25
K_POINTS automatic
4 4 4 1 1 1
```

---

The **key observation** in the case of bulk crystals is that often we already have information about the structure from XRD measurements. This information simplifies drastically the calculation of the equilibrium structure.

For example, in the case of silicon, the diamond structure is uniquely determined by the lattice parameter, therefore the energy minimization is a one-dimensional optimization problem, precisely as in the case of the  $\text{Cl}_2$  molecule.

In Tutorial 2.2 we will explore the more complicated situation where we want to decide which one among several possible crystal structures is the most stable.

To find the equilibrium lattice parameter of silicon we perform total energy calculations for a series of plausible parameters. We can generate multiple input files at once by using the following script (we can copy/paste this in a vi window: `vi myscript.tcsh` and then execute using `tcsh myscript.tcsh`):

```
sed "s/10.28/NEW/g" si.in > tmp
foreach ALAT ( 10.0 10.1 10.2 10.3 10.4 10.5 10.6 )
sed "s/NEW/${ALAT}/g" tmp > si_${ALAT}.in
end
```

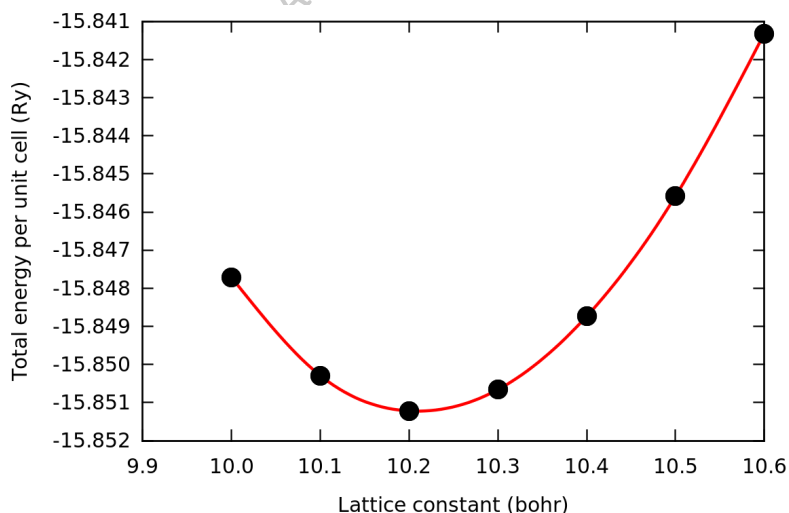
Now we can execute `pw.x` using the generated input files. Once again we can enter all the instances of execution in the same submission script, eg:

```
mpirun -n 12 pw.x -npool 4 < si_10.0.in > si_10.0.out
...
...
mpirun -n 12 pw.x -npool 4 < si_10.6.in > si_10.6.out
```

After running the batch job on the cluster, we should be able to see the output files `si_10.0.out`, `...`, `si_10.6.out`, and extract the total energies as follows:

```
$ grep "\!" si_*.out
si_10.0.out:! total energy = -15.84770898 Ry
si_10.1.out:! total energy = -15.85028964 Ry
si_10.2.out:! total energy = -15.85121715 Ry
si_10.3.out:! total energy = -15.85065982 Ry
si_10.4.out:! total energy = -15.84873489 Ry
si_10.5.out:! total energy = -15.84558108 Ry
si_10.6.out:! total energy = -15.84131402 Ry
```

A plot of the total energy vs. lattice parameter is shown below:



---

Also in this case the black dots are the calculated datapoints, and the red line is a smooth interpolating function (obtained using 'smooth csplines' in gnuplot).

By zooming near the bottom we see that the equilibrium lattice parameter is  $a = 10.2094$  bohr  $= 5.403$  Å. This calculated value is very close to the measured equilibrium parameter of 5.43 Å; DFT/LDA underestimates the measured value by 0.5%.

## Cohesive energy of a bulk crystal

The **cohesive energy** is defined as the heat of sublimation of a solid into its elements.

In practice the calculation is identical to the case of the dissociation energy of the  $\text{Cl}_2$  molecule: we need to take the difference between the total energy at the equilibrium lattice parameter and the total energy of each atom in isolation.

For the energy at equilibrium we just repeat one calculation using the same input files as above, this time by setting

```
...
celldm(1) = 10.2094,
...
```

This calculation yields:

$$E_{\text{bulk}} = -15.85122170 \text{ Ry}$$

(this value is an energy per unit cell, and each unit cell contains 2 Si atoms)

For the isolated atom we need to consider one atom per cell, and spin-polarization as in the case of Cl. In fact the outer valence shell of silicon is  $2p$ :  $\uparrow \uparrow \square$ .

We can modify the input file as follows (this is very similar to what we have done for the Cl atom, but this time the total spin is 2 Bohr magnetons)

```
&control
  calculation = 'scf'
  prefix = 'silicon',
  pseudo_dir = './',
  outdir = './'
/
&system
 ibrav = 1,
celldm(1) = 20,
nat = 1,
ntyp = 1,
ecutwfc = 25.0,
nspin = 2,
tot_magnetization = 2.0,
occupations = 'smearing',
degauss = 0.001,
/
&electrons
  conv_thr = 1.0d-8
/
```

---

```
ATOMIC_SPECIES
Si 28.086 Si.pz-vbc.UPF
ATOMIC_POSITIONS
Si 0.00 0.00 0.00
K_POINTS gamma
```

The calculation for the isolated atom gives:

$$E_{\text{Si}} = -7.53189352 \text{ Ry}$$

By combining the last two results we obtain:

$$E_{\text{cohes}} = E_{\text{bulk}}/2 - E_{\text{Si}} = 0.393717 \text{ Ry} = 5.36 \text{ eV}$$

The measured heat of sublimation of silicon is 4.62 eV (see pag. 71 of the Book), therefore DFT/LDA overestimates the experimental value by 16%.

Prof. Feliciano Giustino  
University of Oxford  
PARADIM School · Cornell, July 2016



# An introduction to density functional theory for experimentalists

## Tutorial 2.2

### Hands-on session

We create a new folder as usual:

```
$ cd ~/scratch/summerschool; mkdir tutorial-2.2 ; cd tutorial-2.2
```

In this hands-on session we will study the equilibrium structure of simple crystals, namely **silicon** (as in Tutorial 2.1), **diamond**, and **graphite**.

#### Exercise 1

► Familiarize yourself with the steps of Tutorial 2.1, in particular:

- 1 Calculate the equilibrium lattice parameter of silicon
- 2 Calculate the cohesive energy of silicon

#### Exercise 2

In this exercise we will study the equilibrium structure of diamond.

The crystal structure of diamond is almost identical to the one that we used for silicon in Exercise 1. The two important differences are (i) this time we need a pseudopotential for diamond, and (ii) we expect the equilibrium lattice parameter to be considerably smaller than in silicon.

► After creating a new directory for this exercise, find a suitable pseudopotential for diamond. It is recommended to use the pseudopotential `C.pz-vbc.UPF`.

The link to the pseudopotential library can be found in the PDF document of Tutorial 1.1.

► Download this pseudopotential, copy over all the necessary files from `tutorial-2.1`, and perform a test run to make sure that everything goes smoothly.

For this test run it is sensible to use the experimental lattice parameter of diamond, 3.56 Å.

► Determine the planewaves kinetic energy cutoff `ecutwfc` required for this pseudopotential.

You can generate the input files for various cutoff energies either manually, or by using the script on pag. 3 of Tutorial 1.2.

You should find that the total energy per atom is converged to within 10 meV when using a cutoff `ecutwfc = 100 Ry`.

► Determine the equilibrium lattice parameter of diamond, by performing calculations similar to those for silicon in Exercise 1.

Compare the calculated lattice parameter with the experimental value.

You should find an equilibrium lattice parameter of  $6.66405 \text{ bohr} = 3.5264 \text{ \AA}$ .

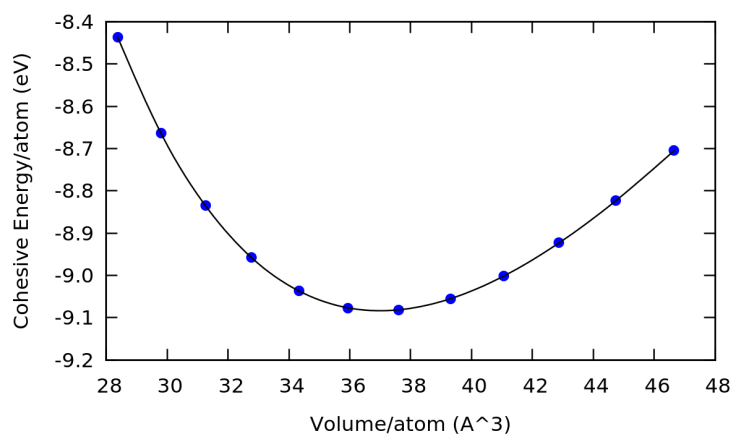
► Using the equilibrium lattice parameter determined in the previous exercise, calculate the cohesive energy of diamond and compare your value with experiments.

For this calculation you can use the same strategy employed in Tutorial 2.1 for the cohesive energy of Si. Note that the C atom in its ground state has a valence electronic configuration  $2s \uparrow\downarrow 2p \uparrow \uparrow \square$

As a reference, the cohesive energy that calculated using these settings should be around 9.08 eV (the experimental value is 7.37 eV).

► Plot the cohesive energy vs. volume/atom for all the lattice parameters that you considered.

The plot should look like the following.



### Exercise 3

In this exercise we study the equilibrium structure of graphite. A search for carbon allotropes in the Inorganic Crystal Structure Database (ICSD) yields the following structural information:

Summary		Collection Code 76767	
Struct. formula	C	Author	Trucano, P.; Chen, R.
Space Group	P 63/m m c(194)	Title of Article	Structure of graphite by neutron diffraction
Unit Cell	2.464(2) 2.464 6.711(4) 90. 90. 120.	Reference	Nature (London) (1975) 258, p136-p137
Cell Volume	35.29 Å³	Formula Units per Cell	4
Temperature	room temperature	Pressure	atmospheric
PDF-numbers	01-089-7213 41-1487	R-value	0.042
Remark	High Quality Data		Warnings & Comments
			0 Warnings / 3 Comments

From the data in this page we know that the unit cell of graphite is hexagonal, with lattice vectors

$$\begin{aligned} \mathbf{a}_1 &= a \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \\ \mathbf{a}_2 &= a \begin{pmatrix} -1/2 & \sqrt{3}/2 & 0 \end{pmatrix} \\ \mathbf{a}_3 &= a \begin{pmatrix} 0 & 0 & c/a \end{pmatrix} \end{aligned}$$

( $a = 2.464 \text{ \AA}$  and  $c/a = 2.724$ ), and with 4 C atoms per primitive unit cell, with fractional coordinates:

$$\begin{aligned}
C_1 &: ( 0 & 0 & 1/4 ) \\
C_2 &: ( 0 & 0 & 3/4 ) \\
C_3 &: ( 1/3 & 2/3 & 1/4 ) \\
C_4 &: ( 2/3 & 1/3 & 3/4 )
\end{aligned}$$

► Starting from the input file that you used for diamond in Exercise 2, build an input file for calculating the total energy of graphite, using the experimental crystal structure given above.

Here you will need to pay attention to the entries `ibrav` and `cellldm()` in the input. Search for these entries in the documentation page:

[http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT\\_PW.html](http://www.quantum-espresso.org/wp-content/uploads/Doc/INPUT_PW.html)

Here you should find the following:

**Namelist: SYSTEM**

ibrav	INTEGER
Status:	REQUIRED

Bravais-lattice index. If `ibrav`  $\neq 0$ , specify EITHER [ `cellldm(1)`-`cellldm(6)` ] OR [ `A,B,C,cosAB,cosAC,cosBC` ] but NOT both. The lattice parameter "alat" is set to `alat = cellldm(1)` (in a.u.) or `alat = A` (in Angstrom); see below for the other parameters.  
For `ibrav=0` specify the lattice vectors in `CELL_PARAMETER`, optionally the lattice parameter `alat = cellldm(1)` (in a.u.) or `= A` (in Angstrom), or else it is taken from `CELL_PARAMETERS`

ibrav	structure	cellldm(2)-cellldm(6) or: b,c,cosab,cosac,cosbc
0	free	crystal axis provided in input: see card <code>CELL_PARAMETERS</code>
1	cubic P (sc)	$v_1 = a(1,0,0), v_2 = a(0,1,0), v_3 = a(0,0,1)$
2	cubic F (fcc)	$v_1 = (a/2)(-1,0,1), v_2 = (a/2)(0,1,1), v_3 = (a/2)(-1,1,0)$
3	cubic I (bcc)	$v_1 = (a/2)(1,1,1), v_2 = (a/2)(-1,1,1), v_3 = (a/2)(-1,-1,1)$
4	Hexagonal and Trigonal P	<code>cellldm(3)=c/a</code> $v_1 = a(1,0,0), v_2 = a(-1/2,\sqrt{3}/2,0), v_3 = a(0,0,c/a)$

Based on this information we must use `ibrav = 4` and `cellldm(1)` and `cellldm(3)`.

As a sanity check, if you run a calculation with `ecutwfc = 100` and `K_POINTS gamma` you should obtain a total energy of -44.581847 Ry.

► A convergence test with respect to the number of **k**-points indicates that the total energy is converged to 4 meV/atom when using a shifted  $6 \times 6 \times 2$  grid (6 6 2 1 1 1 with `K_POINTS automatic`). Using this setup for the Brillouin-zone sampling, calculate the lattice parameters of graphite *a* and *c/a* at equilibrium. Note that this will require a minimization of the total energy in a **two-dimensional** parameter space.

For this calculation it is convenient to automatically generate input files as follows, assuming that your input file is called `graph.in`:

- Replace the values of `cellldm(1)` and `cellldm(3)` by the placeholders `ALAT` and `RATIO`, respectively;
- Create a script `myscript.tcsh` with the following content:

```

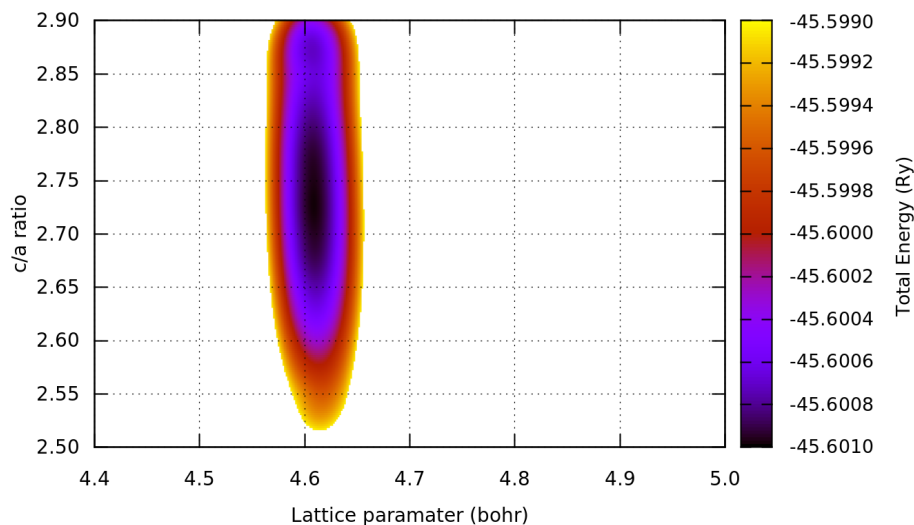
rm tmp.pbs
foreach A ( 4.4 4.5 4.6 4.7 4.8 4.9 5.0 )
  foreach CA ( 2.50 2.55 2.60 2.65 2.70 2.75 2.80 2.85 2.90 )
    sed "s/ALAT/${A}/g" graph.in > tmp
    sed "s/RATIO/${CA}/g" tmp > graph_${A}_${CA}.in
    echo "mpirun -n 12 pw.x -npool 12 < graph_${A}_${CA}.in > graph_${A}_${CA}.out" >> tmp.pbs
  end
end

```

- By running `tcsh myscript.tcsh` you will be able to generate input files for all these combinations of  $a$  and  $c/a$ . The file `tmp.pbs` will contain all the correct execution commands, that you can copy/paste directly inside your submission script.
- Note that this will produce  $7 \times 9 = 63$  input files, but the total execution time on 12 cores should be around 1 min.
- At the end you will be able to extract the total energies by using `grep` as usual

```
grep "\!" graph_*.out > mydata.txt
```

If you plot the total energies that you obtained as a function of  $a$  and  $c/a$  you should be able to get something like the following:



This plot was generated using the following commands in `gnuplot` (the file `mydata.txt` must first be cleaned up in order to obtain only three columns with the values of  $a$ ,  $c/a$ , and energy):

```

set dgrid3d splines 100,100
set pm3d map
splot [] [] [:-45.599] "mydata.txt"

```

The 'splines' keyword provides a smooth interpolation between our discrete set of datapoints. The plotting range along the energy axis is restricted in order to highlight the location of the energy minimum.

Here we see that the energy minimum is very shallow along the direction of the  $c/a$  ratio, while it is very deep along the direction of the lattice parameter  $a$ . This corresponds to the intuitive notion that the bonding in graphite is very strong within the carbon planes, and very weak in between planes.

By zooming in a plot like the one above you should be able to find the following equilibrium lattice parameters:

---

$$a = 2.439 \text{ \AA}, c/a = 2.729$$

From these calculations we can see that the agreement between DFT/LDA and experiments for the structure of graphite is excellent. This result is somewhat an artifact: most DFT functionals cannot correctly predict the interlayer binding in graphite due to the lack of van der Waals corrections. Since LDA generally tends to overbind (as we have seen in all examples studied so far), but it does not contain van der Waals corrections, this functional works well for graphite owing to a cancellation of errors.

For future reference let us just note that the total energy calculated using these optimized lattice parameters is  $-45.60104552$  Ry.

#### Exercise 4

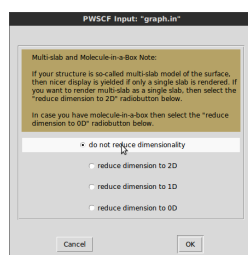
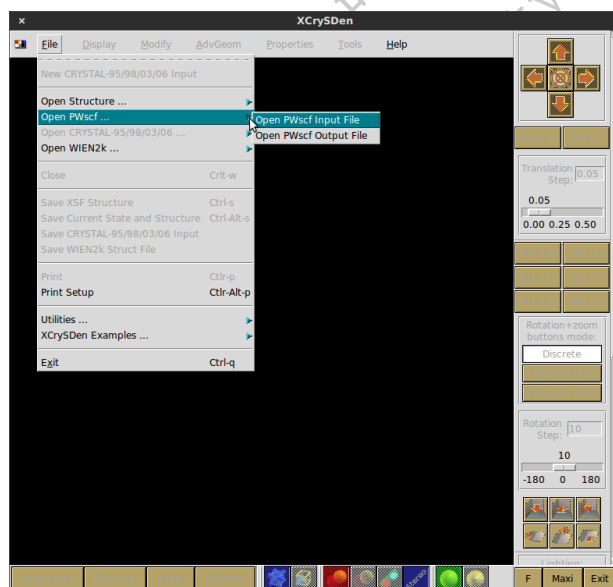
In this exercise we want to see how the structure of graphite that we are using in our input file looks like in a ball-and-stick model.

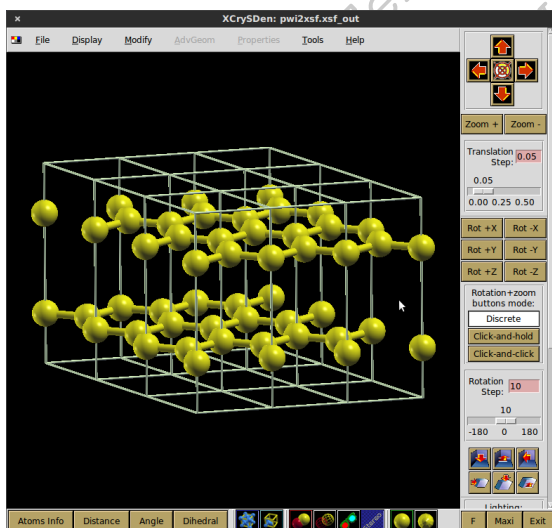
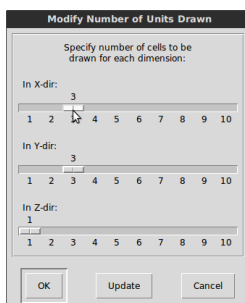
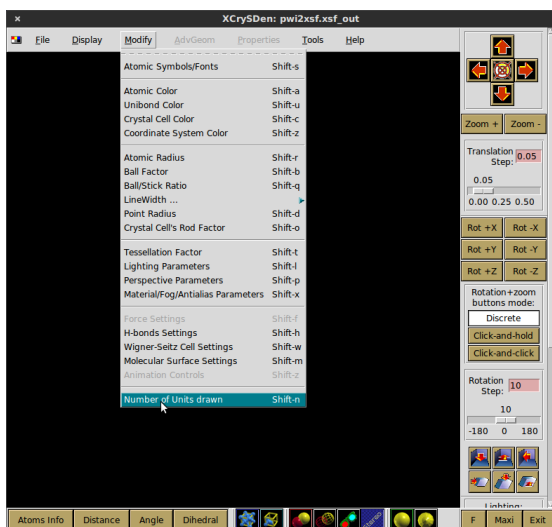
The software [xcryden](http://www.xcryden.org) can import QE input files and visualize the atomistic structures. General info about this project can be found at <http://www.xcryden.org>.

We launch xcryden by typing:

```
$ xcryden
```

The user interface is very simple and intuitive. The following snapshots may be helpful to get started with the visualization.





## Exercise 5

► Which carbon allotrope is more stable at ambient conditions, diamond or graphite?

Note: The answer to this question is rather delicate. In nature graphite is more stable than diamond by 40 meV/atom at ambient pressure and low temperature.

Using the calculations of Exercises 2 and 3 we find that the cohesive energy of diamond is 8 meV lower than in graphite. Therefore DFT/LDA would predict diamond to be more stable, contrary to experiments. This is in agreement with the following study by Janotti et al, <http://dx.doi.org/10.1103/PhysRevB.64.174107>.