Lab 3: Handout
First principles calculations, part 2.

In this lab, we will be using PWSCF as our first-principles code again. In problem 1, we will once again calculate surface energies, this time with the added complexities of first-principles methods. In problem 2, we will examine different exchange-correlations, and how they affect the energetics and lattice parameters.

Some helpful conversions:

1 **bohr** = 0.529177249 **angstroms**.
1 **Rydberg** (R_∞) = 13.6056981 **eV**
1 **eV** =1.60217733 x 10^{-19} **Joules**

Singapore students:

Your lab instructors will show you the locations of all of the input files.

MIT students:

We will be using faster computers for this lab. We will go over the login information in class

Cambridge students:

Your instructors will go over the login information.

Problem 1

Problem 1 will look at surface energies of the (001) plane in Al. As you may remember, we calculated surface energies in Problem Set 1. In this lab will do another surface energy calculation, with the added complexities of first-principles methods.

In this problem set, we give you the cutoff for Al (14 Ryd). Cutoffs for a particular element are sometimes given to you with the software you use. However, you always have to test energy convergence with respect to $\vec{k}$ -points.

If you remember from Lab 1, the general procedure for surface energy calculations is this. First, we determine energy cutoff and $\vec{k}$ -point mesh required for energy convergence. Next, we calculate the equilibrium lattice constant of the material in question (in this case, Al). Then, we build supercells of Al, removing planes to create a vacuum (we can also separate existing planes instead of removing planes). We have to test for convergence with respect to vacuum size and Al slab size. From these calculations, we can obtain a surface energy.

In this lab, we will give you the energy cutoff (14 Ryd), and lattice constant (7.50 bohr). One of the supercells is built for you already, with a suitable $\vec{k}$ -point mesh chosen for you. For the bulk energy of FCC Al, you will have to find a $\vec{k}$ -point mesh that gives a converged energy.

Calculating energy of the supercell with surfaces

In the empirical energy lab, we could build the supercell without regard to number of atoms. This is because empirical energy methods are fast. In this lab, we build the supercell to minimize the number of atoms. This is because in DFT codes, calculation time scales as NlogN, where N is the number of atoms.

To minimize the number of atoms, we have to perform a coordinate transformation.

A picture is below.

The dark lines are the FCC cell we used before. Our new coordinate system is marked by the lighter lines. The new coordinate system has $a_{new} = \frac{a_{old}}{\sqrt{2}}$, $c_{new} = c_{old}$, and is rotated 45 degrees from the old system (rotated about the *c* axis). Because we have built a supercell in the *z* direction, the new cell will be a primitive tetragonal cell. There will be 2 layers of Al atoms per cell, when you go up in the *z* direction.

An input file for this system is given to you.

`/home/mit3320/HW3/PROBLEM1/al7.norelax.in`

```
1     Al 001 32.5 bohr total, 22.5 taken up by atoms (7 layers), 10 bohr vaccuum
2     Al
3      &input
4       ibrav=  6, celldm(1) =5.3033, celldm(3) =  6.12826,
nat=7,ntyp=1,
5         pseudop(1) = 'Al.vbc',
6         pseudo_dir = '/home/mit3320/HW3/',
7         tmp_dir='.',
8         ecut(1)= 14.0,
9         beta(1) = 0.3,
10        niter=200,iswitch=0, tr2=1.0d-12, upscale=10,
11        degauss=0.05, ngauss=1,
12     &end
13       0.5000000      0.5000000     -2.121320  1
14       0.0000000      0.0000000     -1.414213  1
15       0.5000000      0.5000000     -0.707107  1
16       0.0000000      0.0000000      0.000000  1
17       0.5000000      0.5000000      0.707107  1
18       0.0000000      0.0000000      1.414213  1
19      0.5000000      0.5000000      2.121320  1
20     'Al' 1 1 1.0
21      0
22       4  4  1
23       0  0  0
~
```

Lines 1-3

This is the title

Line 4

This is lattice information.

ibrav=6 is the primitive tetragonal lattice. For tetragonal structures a=b and all angles are 90 degrees. celldm(1) is *a*, and celldm(3) is *c*/a. We have picked a_0=7.50 bohr. nat is number of atoms and ntyp the number of types. You have 7 Al atoms (nat=7), all of the same type (ntyp=1). **You will have to change nat as you increase the number of atoms in the cell**

Lines 5-7

This is the pseudopotential file and location, and temporary file location. This is the same as before

Lines 8-9

This means the same as before, and is the cutoff. You will not be changing this value in this lab. Beta sets the mixing factor, **If your run does not converge, you may have to lower beta.**

Lines 10-11

iswitch=1 turns on relaxations (internal relaxations, not volume relaxations). This is like the "opti" switch in gulp. niter is the maximum number of electronic steps before the code quits. tr2 sets the threshold for self-consistency. **You will need to turn iswitch on and off.**

Lines 11

This is parameter which controls smearing of electrons. This matters in metals where you have partial occupancies. This give a finite temperature to the system, which improves convergence rates.

Line 13-19

Same as before. Atom (x, y, z, type) and pseudopotential. All are in units of celldm(1). For example, the atom labelled (0.5 0.5 0.707107) has Cartesian coordinates (2.65165, 3.75 , 3.75). This comes from 2.65=5.3033*.5 and 3.75=5.0333*.707107. The fractional coordinates are then (0.5 0.5 .1153). This comes from 2.65165/5.3033=.5 and 3.75/(celldm(1)*celldm(3))=0.1153. Remember periodic boundary conditions. The positions of all of the atoms are shown in the picture below.

Lines 201.

The name of the atom and which PP it corresponds to

Lines 21-23

This is kpoint information, same as before. We have chosen a 4x4x1 $\vec{k}$ -point mesh. You should think about why we picked 4x4x1 (instead of 2x2x2 or 1x1x4 for example). The mesh given is coarser than is ideal (to save time). **You will have to change this.**

A picture of this cell is below.

A	0.5000000	0.5000000	-2.121320	1
B	0.0000000	0.0000000	-1.414213	1
C	0.5000000	0.5000000	-0.707107	1
D	0.0000000	0.0000000	0.000000	1
E	0.5000000	0.5000000	0.707107	1
F	0.0000000	0.0000000	1.414213	1
G	0.5000000	0.5000000	2.121320	1

The advantage of using this (unintuitive) way of placing your atoms is that you can always increase the size of the vacuum by increasing celldm(3), without changing any other parameters.

To add another layer of atoms:

1. Change the value of nat
2. Change the title (optional)
3. Add another atom 0.70717 higher or lower than the last one (make sure to get 0.0 0.0 or 0.5 0.5 correct)

To see this, remember that the z coordinate is given in units of celldm(1).
So, 0.7071*celldm(1)=0.7071*5.3033=3.75, which is ½ of a_0 for Al.

To get the total cell dimension in the *z* direction : multiply celldm(3) by celldm(1).
Example: 6.12826*5.3033 bohr = 32.5 bohrs.

To get the number of cells: divide cell dimension in z direction by a_0
Example: 32.5 bohrs/(7.5 bohr/cell) = 4.333333 cells

To get size of slab: multiply range of c values by celldm(1)
Example: [2.12132 – (-2.12132)]*5.3033 bohr = 22.5 bohrs

To get size of vacuum : Calculate total cell dimension in *z* direction – slab dimension
Example: 32.5 bohr-22.5bohr =10 bohr.

Another example of a supercell file is in the directory:

```
/home/mit3320/HW3/PROBLEM1/al14.relax.in
```

(Singapore, this will be different for you)

Take a look at this file. Notice **iswitch=1**, which sets the relaxations on. This is like the "opti" switch in gulp.

You will need to perform relaxed and unrelaxed runs.

The last step is to get the energy of the bulk.

In first-principles calculations, there is a subtle complication in getting the bulk energy. One way (the more obvious way) is to get it from a bulk calculation. The problem with this method is that you will be using different $\vec{k}$ -points for your supercell calculations and bulk calculations. This can lead to inaccurate surface energies. A definition is to take the change in energy when you add a new layer of atoms to your supercell with the vacuum. That is

$$E_{bulk} = \lim_{n \to \infty} E_n - E_{n-1}$$

If everything is converged (slab size, vacuum size) then adding a layer to the supercell is just like adding a layer to the bulk. You should think about this definition to make sure that it makes sense to you.

We will calculate bulk energies both by calculating the energy of FCC Al, and by the "adding a layer" way.

An input file for FCC Al is given to you **al.scf.in**.A few scripts are included to help you calculate the bulk energy of FCC Al. These are **kpts** and **latticeparam**. They work in the same way as the scripts for lab 2. Be sure to look over these scripts before using them.

Running jobs in the background: &

To run a job in the background, use the & symbol at the end of your line.

That is,

```
     mit3320$ kpts &
```

Now you can do other things while your jobs is running.

The nohup command

If you run a job, and log off, your job will die. That is if I typed:

```
     mit3320$ nohup kpts &
```

The script 'kpts' would only work as long as I was logged in.
To solve this, use nohup.

```
  mit3320$ nohup kpts &
```

Now you can log off and go do something else!

With this calculation, we are already doing "real" calculations - this calculation was published recently by a physics group at NUS (National University of Singapore)! The reference is: Zhen, J.C., Wang, H-Q., Huan, C.H.A., Wee, A.T.S. Journal of Electron Spectroscopy and Related Phenomena **114-116** 501 (2001).

Problem 2

Problem 2 will look at calculations involving Iron. We will look at difference exchange-correlations, and the effect they have on energetics and lattice parameter, and spin polarization (magnetism). In particular, we will look at the LDA (local density approximation) and GGA (generalized gradient approximation) functionals. Both are implemented within DFT (Density Functional theory).

In the LDA, the exchange-correlation energy is taken to be that of a homogeneous electron gas at that point. The GGA also takes gradients into account. Thus, in theory, the GGA is better. In practice, the GGA does things better some of the time, but not all of the time. In addition, GGA calculations take a little longer to perform (~50% longer). This means you should always perform tests on your problem of interest. In the literature, LDA is the more-used functional, but there are certain problems (for example magnetic effects in $LiMnO_2$ and CO adsorption on Pt) where the GGA is standard.

Below is a copy of the file Febccldamag.in, along with explanations.

MIT and Cambridge students will find this file in

```
/home/mit3320/HW3/PROBLEM2
```

directory on Armageddon.

Singapore students, your instructors will tell you location of this file.
Febccldamag.in

```
pz
Iron
 &input
    ibrav=  3, celldm(1) =5.20, nat=  1, ntyp= 1,
    pseudop(1) = 'FeUS.RRKJ3',
    pseudo_dir = '/home/mit3320/PP/',
    tmp_dir='.',
    ecut(1) =25.0,
    beta(1) = 0.7,
    tr2 =  1.0d-12,
    lsda=.true.,
    starting_magnetization(1)=0.7,
    ngauss=-1,degauss=0.05,
    lforce=.true., lstres=.true.,
 &end
```

```
16   0.00 0.00 0.00 1
17   'Fe' 1 1 1.0
18   0
19   6 6 6
20   0 0 0
```

Lines 1

This label can denote the exchange correlation (XC). It is purely for your own benefit; you do not have to change it.

Lines 2-3

The title, and start of input. The same meaning as before.

Line 4

This is lattice information, same as before, but **this line will change.** ibrav=3 is the bcc lattice. For FCC you will change this to 2.
Celldm is the a_0, the lattice parameter bohrs, in conventional coordinates. For BCC and FCC, $a=b=c$.

Lines 5-6

This line will change. This is the pseudopotential file and location. Note, that when you change between LDA and GGA exchange correlations, you need to change your pseudopotential.

Lines 8-9

This means the same as before, and is the cutoff. **You may want to increase the cutoff (say to ~40 Ryd) when you are running scripts, but for now keep it at 25 Ryd.** Beta sets the mixing factor, tr2 sets the threshold for self-consistency. Don't worry about these last two.

Lines 11-12

This line will change. Lsda=.true. turns ON the magnetization calculations. Magnetization runs will always take longer, so don't turn this on unless you want to. Starting_magnetization(1) is the starting magnetization for atom 1. Set this to a value between -1 and +1

Lines 13

This is parameter which controls smearing of electrons. This matters in metals where you have partial occupancies. This improves convergence.

Line 14

This has the same meaning as before. This controls the calculation of forces and stresses.

Lines 16-17.

Same as before. Atom (x, y, z, type) and pseudopotential.

Lines 18-20.

This is kpoint information, same as before, but **you will change these lines.** A '0' in line 18 means use a grid. Lines 19-20 control grid size (just change line 19 for k_x, k_y, and k_z)

To summarize:

To change between LDA and GGA:

Change line 1 to pz (LDA) or pbe(GGA), and change the pseudopotential (line 5). You can verify that you are doing an LDA run, because the beginning of the output files will show Exchange-correlation (1100), whereas GGA runs will show Exchange-correlation (1434)

To change between spin-polarized and non-spin polarized:

Change lsda to .true. or .false. (don't forget the periods). Also change the starting magnetization.

To switch between BCC and FCC:

Change ibrav to 3 (BCC) or 2 (FCC)

You should find the correct k-point grid to use yourself. After you have done this, you can set up scripts to find the correct lattice parameter. An example script,

```
/home/mit3320/HW3/PROBLEM2/Fescript
```

is given. You may want to use the higher cutoff given this script, which will give you a smoother E vs. volume curve. For the lab though, use 25 Ryd to save time.

To get energies using this script, use the

```
/home/mit3320/HW3/PROBLEM2/getenergyFe
```

script. This will output your numbers into a file called 'allresults' which you can plot using an appropriate program.

Final notes:

It is a very bad idea to try to do all of these calculations at the last second. You will waste a lot of time waiting for runs to finish. The best strategy is to set up your scripts, and check your runs later in the day, or the next day. If you wait until the last minute, you may not finish your assignment in time.

Remember that "nohup" will let your runs continue, even if you log out. If you do not use nohup, all of your jobs will die when you log out.

FAQ for HW3

How do I kill my script?

Type ps aux. This will tell you which jobs you are running- look at the numbers on the left. Type “kill -9 <number here>” to kill your job. You will need to kill the job AND the script (kill the script first).

My jobs die as soon as I log off.

Use the nohup command, outlined previously.

When I type nohup, it gives me a number. Is this my job number?

Yes. If you want to kill your job later you can typekill -9 <jobnumber>.

I can't log out. I get "there are stopped jobs"

Type 'jobs' <enter>. Find the job you want to kill (it might be a 'top') and type 'kill %<number>'

Can I run more than one job at a time?

Yes. But they must be in different directories. If you try to run 2 jobs in the same directory at the same time, they will both crash. This is because they are both trying to write and read the same wavefunction files (and other files). If you do two runs at the same time, do them in separate directories.

Do I need to test k-point convergence for the supercell (does it have to be more than 4x4x1). What does it mean to explain the rationale behind this choice?

Yes, kpoint convergence for the supercell. Pick a suitable mesh. Explain why we picked 4x4x1 instead of 2x2x2 (for example)

Do I need to test k-point convergence for every single slab size and every single vacuum size?

No. A general rule of thumb is that kpoints/atom is constant (but that the distribution of kpoints along k_x, k_y, and k_z will depend on crystal geometry!!). A kpoint mesh that is suitable for a 7 layer supercell will be more than adequate for a 14 layer supercell, assuming geometry stays relatively constant. Your runs will take slightly longer, but this is more than offset by the fact that you don’t have to retest kpoint convergence for each cell. In practice, we use automatic routines to find suitable kpoint meshes for supercells so that we can use fewer kpoints as supercells get bigger. But here, you will find a kpoint mesh for smaller supercell, and keep it constant as you increase supercell size.

You do not need to test for each vacuum size for kpoint convergence.

On page 5, on the picture of the cell, it looks as if there are 4 'D' atoms, 4 'F' atoms, etc... and 1 'E' and 'G' atom. But D, E, F, and G, are all only listed once in the input.

Remember periodic boundary conditions. They are all the same atom. To build a layer of Al, you only need to add one atom.

My runs aren't converging

In a long slab, sometimes it is harder for the runs to converge. There are a number of steps you can take. First, make sure all of the output files from the previous run are deleted (filewfc, etc..) between each run. Second, try lowering the beta parameter. The beta parameter sets the charge mixing between different iterations. Third set the "niter" parameter to a higher number (it defaults to 50). So a line such as "niter=200". Fourth, check the convergence of the surface energy (do the actual calculation for the full surface energy, don't just look at energies of slabs). Do you really need that big of a vacuum size or that long of a cell? The bigger your vacuum and slab, the more problems you will have with convergence.

My surface energies aren't converging

Look at the surface energies, and not the direct energy (see the solutions for HW 1). Remember that since you are taking differences and dividing by area, the final answer will converge faster than the individual energy for the slab.

I don't get the definition of bulk energies of adding a layer to a supercell

Think about it this way. Assume you have an enormous vacuum and an enormous slab in a 1x1x100000 cell. Before you add an atom (which is the SAME as adding a layer given this cell), you have 2 surfaces, a vacuum, and a bulk. After you add an atom (add a layer) you still have 2 surfaces, a vacuum, and a bulk, but the bulk has one more atom (layer). In other words, it is the same as adding an atom to the bulk, from which you can obtain the bulk energy

Why do we do this? Remember that first-principles calculations are all about canceling errors. When you calculate the bulk directly, you use different k-points than when you calculate the slab. By calculating the energy of 1 Al atom by adding a layer to a supercell (shrinking the vacuum size by layer, and keeping the total cell geometry the same) the errors arising from incomplete kpoint meshes will cancel. In the limit of infinite k-points the answers should be the same. But nobody has time to calculate energies of infinite kpoints.

Errors in the runs.

I get an error :

```
RECOVER: atomic pos. and cell dim. read from file
   Restarting in ELECTRONS
 wrong nat        13        14 or ibrav        6        6
<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

This means you have halted the run (probably with Ctrl-C), changed something, then restarted it. You should delete the filewfc, savepot, and restart files.

More errors in the runs.

I get an error:

```
warning: symmetry operation #  5 not allowed.   fractionary translation:
     0.2500000  0.2500000  0.2500000  in crystal coordinates
   warning: symmetry operation #  6 not allowed.   fractionary translation:
     0.2500000  0.2500000  0.2500000  in crystal coordinates
etc….
```

This is another "feature" of the code. It has to do with the FFT grid which depends on the geometry and cutoff. It does not change you answer. Don't worry about it.

My E vs. lattice constant plot is jagged.

There are a number of solutions to this; the easiest is to raise the energy cutoff, and to remove the output files (which give wavefunction information, etc..) between runs.

What is the difference between absolute and total magnetization?

In Fe, most of the magnetization arises from the d orbitals, and is positive (if you set starting magnetization positive). But there is also a small contribution from the s orbitals, that is negative. Total magnetization is the sum; absolute takes the absolute value. See the work of Matteo Cococcioni for more details.