Module 2: Density functional theory calculation of elastic constants

Project Brief

In this project, we will continue using the PWSCF code and look at convergence with respect to the k-point mesh, how to relax crystal structures, and how to compute elastic constants. Successful completion of this work will demonstrate mastery of Quantum Espresso DFT calculations for condensed systems.

Deliverables

You will produce a short report documenting your findings. Your report should contain a separate section for each of the tasks listed below, and provide the explicit deliverables requested for each task indicated by the **[Report]** below.

Your report should be formatted as a single pdf document comprising your report. You may wish to write your report in latex and convert using `pdflatex`, or in markdown and convert using `pandoc report.text --to latex --out report.pdf`; alternatively, you can put together a clearly formatted jupyter notebook. (`module load pandoc` and `module load texlive` to have the most up-to-date versions of each).

You should creating a subdirectory called
`/class/mse404ela/sp26/<your_net_id>/Project2`
and copying your work into that directory by **11:59pm on 16 February 2026.** *Late submissions will not be accepted*; **let me know in advance if you will have difficulty with completion.**

I will give you feedback on the expectations listed below and for your work.

Convergence in Ecut and k-point mesh

For all first-principles calculations for solids, you must pay attention to two convergence issues. The first is the *cutoff energy*, E_{cut}, which is the cutoff for the wave-function expansion. The second is k-point mesh, which determines how well your discrete grid approximates the continuous integral over the Brillouin zone.

Background for k-point convergence. Remember that we are dealing with infinite systems using periodic boundary conditions. This means that we can use the Bloch theorem to help us solve the

Schrödinger equation. The Bloch theorem says that our wavefunction $\psi_{n\mathbf{k}}(\mathbf{r})$ can be written as

$$\psi_{n\mathbf{k}}(\mathbf{r}) = \exp(i\mathbf{k} \cdot \mathbf{r})u_{n\mathbf{k}}(\mathbf{r})$$

where

$$u_{n\mathbf{k}}(\mathbf{r}) = \sum_{\mathbf{G}} c_{\mathbf{G}} \exp(i\mathbf{G} \cdot \mathbf{r}).$$

The function $u_{n\mathbf{k}}(\mathbf{r})$ has the same periodicity of the lattice; $\mathbf{k}$ is a wavevector in the Brillouin zone, and n is the "band index" (identifying which eigenfunction we are considering). In the walkthrough we studied the plane wave expansion and convergence with respect to the maximum $\mathbf{G}$ vector given by the cutoff energy

$$\frac{\hbar^2 G^2}{2m} \leq E_{\text{cut}}$$

Because of the Bloch theorem, we need to solve a Schrödinger-like Kohn-Sham equation (i.e. iterative diagonalization of a $M \times M$ matrix until self-consistency, where M is the number of planewaves per k-point) everywhere in the Brillouin Zone. In practice, we do it for a finite number of k values, and get a set of values for $\varepsilon_{n\mathbf{k}}$ at each k. The energy of the crystal, E, is computed by integrating over the occupied bands of the first Brillouin zone. Thus, summing over a finite number of k-points is an approximation to performing an integral. You will need to make sure you have enough k-points to have a converged value for the energy computed by this integral.

Input Files

The instructions below specify files on the EWS Linux system. First, let's select our pseudopotential for Al. If you go to www.quantum-espresso.org/pseudopotentials, you can select which pseudopotential you'd like to use. Select the *PBE* (Perdew-Burke-Ernzerhof Generalized Gradient Approximation (GGA)) for the exchange correlation potential, and then select *PAW* for the type. After you filter, you can click on Al, and you should see two options: one that is "scalar relativistic," and the other that is "fully relativistic." The difference has to do with the way core electrons states (such as the $1s$) are treated; for most elements without very large Z, scalar relativistic suffices. Choose, then, the `Al.pbe-n-kjpaw_psl.0.0.UPF` potential. The electron configuration of Al is $[\text{Ne}]3s^2 3p^1 = [1s^2 2s^2 2p^6]3s^2 3p^1$. This particular functional has 3 electrons in the valence, so the filled $[1s^2 2s^2 2p^6]$ shells are in the "core" while the $3s^2 3p^1$ electrons will be considered valence electrons; this is like treating Al as a lumped Al^{3+} ion with $3e^-$. Since this particular pseudopotential does not come prepackaged, we need to download this pseudopotential into a local pseudopotential directory. For example:

```
$ cd ~/QE//pseudo
$ wget https://pseudopotentials.quantum-espresso.org/upf_files/Al.pbe-n-kjp
```

or, alternatively, download it from your browser and move the file to the appropriate place. When we run QE, we will tell the code where to find this pseudopotential on our local disc. You may want to export your `ESPRESSO_PSEUDO` environment variable (see walkthrough); else, you'll need to make sure that the pseudo_dir variable in each input file points to the correct directory.

After downloading, you can look at the pseudopotential file using less; the information inside of `PP_HEADER` reiterates much of what is described above.

Next create a directory to store your files. You should create a new subdirectory in your `Runs` directory that you created last time. For example

```
$ cd ~/QE/Runs
$ mkdir Al
$ cd Al
```

From the `/class/mse404ela/QuantumEspresso/Project` directory, copy over the input files `Al.scf.inp` and `Al.relax.inp`.

We will look first at the `Al.scf.inp` input file. This input file is for Al in the fcc crystal phase.

Aluminum exists under standard conditions as a face-centered cubic (FCC) structure with four Al atoms in the conventional unit cell at $[000]$, $[0\frac{1}{2}\frac{1}{2}]$, $[\frac{1}{2}0\frac{1}{2}]$, and $[\frac{1}{2}\frac{1}{2}0]$, in units of the lattice constant, a_0. (Of course, there would be no difference in having these atoms at $[\frac{1}{2}\frac{1}{2}\frac{1}{2}]$, $[\frac{1}{2}11]$, $[1\frac{1}{2}1]$, and $[11\frac{1}{2}]$, why not?)

Wikipedia: Cubic Crystal System

QE operates using not the conventional unit cell, but the primitive unit cell based on the primitive axis vectors as the most economic way to define a unit cell. In the case of FCC lattices, this allows us to define our periodic crystal containing just a single Al atom at the origin [0 0 0].

Look at the input file, which you have copied over to your directory.

```
$ less Al.scf.inp
```

The file will look similar to the one for the H2 molecule in the walkthrough. Here we highlight some key points. First, the differences in the unit cell:

```
...
  ibrav= 2,
...
ATOMIC_POSITIONS alat
 Al 0.00 0.00 0.00
...
```

Here we specify a FCC lattice using `ibrav=2` (c.f. INPUT_PW.html:ibrav) and locate a single Al atom at the origin of our primitive cell.

Next, we use a different smearing:

```
...
  occupations = 'smearing',
  smearing = 'methfessel-paxton',
  degauss = 0.05,
...
```

A metal is a system with unoccupied bands near the Fermi energy (the energy of the highest occupied electron level) meaning that bands may cross at the Fermi surface. The DFT calculation scales quadratically with the number of bands in the calculation. To reduce computational costs, the DFT algorithm only considers the occupied bands, and a few unoccupied bands. At $T = 0$ K,

there is a step discontinuity in the e- occupation numbers of the bands, but at finite temperature (i.e., $T > 0$ K) the function is no longer discontinuous. For example, if you consider the Fermi distribution, the occupancy of an energy level E is given by

$$f(E) = \frac{1}{\exp((E - E_\mathrm{F})/k_\mathrm{B}T) + 1}$$

where E_F is the Fermi energy and T is the temperature. What is the occupancy $f(E_\mathrm{F})$?

When bands are near the Fermi level—as in metals—this means that small changes in the wavefunction can cause large changes in the band occupancy, making the algorithm unstable. To stabilize the algorithm, better reflect the continuous nature of the band occupancies at finite temperature, and reduce the number of k-points required for convergence, we employ a smearing function to the energy bands. *This has the overall effect of producing a more accurate density of states.*

Here we are telling QE to use a smearing potential for the atoms in the metal with "Methfessel-Paxton first-order spreading" (cf. *Phys. Rev. B* **40**, 3616 (1989)) and a 0.05 Ry value for the Gaussian spreading for Brillouin-zone integration. This is a well-tested smearing approach for metallic systems.

Lastly, we also introduce an automatically generated grid of k-points:

```
...
K_POINTS automatic
  4 4 4   0 0 0
```

After the keywork `K_POINTS`, `"automatic"` tells PWSCF to automatically generate a k-point grid. The format of the next line is `nkx nky nkz offx offy offz` where `nk*` is the number of intervals in a direction and `off*` is the offset of the origin of the grid.

If we now do

```
$ less Al.relax.inp,
```

you will see similar changes to what we had to relax the hydrogen molecule; now, however, we use `"vc-relax"` (or variable cell relaxation, to relax the unit cell), and there is also a new card for `&CELL`, which takes the default values.

You can read the documentation for the input file in the `INPUT_PW` file; this is in the Docs subdirectory under the `/class/mse404ela` directory or online at INPUT_PW.html.

A1. SCF energy Calculation

If using the EWS workstations, in order to use the scratch space and not to overwrite each other's files, please edit your input files (using, for example, vi) to set prefix='aluminum-' outdir='/tmp/' where you replace with your netid. Also remember to set pseudo_dir to point to your local pseudopotential directory. Using the Al.scf.inp input file, run the PWSCF code to calculate the energy of the Al phase.

- **[Report]** What is your calculated value of the total energy of the system in

    a) Rydbergs?

    b) eV?

    c) Joules?

- **[Report]** How long did the calculation take in

    a) CPU time?
    b) wall time?

A2. Geometry Relaxation

Using the `Al.relax.inp` input file, run the PWSCF code to relax the lattice parameter. N.B. Remember to make the same modifications to prefix, outdir, and pseudo_dir as you did above.

- **[Report]** What is your calculated value of the total energy of the system in:

    a) Rydbergs?

- **[Report]** How long did the calculation take in

    a) CPU time?
    b) wall time?

- **[Report]** What is the value of the input lattice constant in a.u.?

- **[Report]** Near the top of your output file you will see a block:

```
crystal axes: (cart. coord. in units of alat)
     a(1) = (  -0.500000   0.000000   0.500000 )
     a(2) = (   0.000000   0.500000   0.500000 )
     a(3) = (  -0.500000   0.500000   0.000000 )
```

Towards the end of your output file (just below the line reading "End of BFGS Geometry Optimization") you will see the block that looks something like:

```
CELL_PARAMETERS (alat= 10.20000000)
  -0.509955391   0.000000000   0.509955391
   0.000000000   0.509955391   0.509955391
  -0.509955391   0.509955391   0.000000000
```

In units of the input lattice constant `alat`, the lattice vector of the relaxed crystal geometry has grown—in this case—from 0.5 to 0.509955391. The lattice constant, alat, scales in direct proportion to this change (i.e. by a factor of 0.509955391/0.5).

Using your terminal values for the `CELL_PARAMETERS` block, what is the lattice constant of your final relaxed geometry in a.u.?

A3. Ecut convergence

To speed up the convergence calculations, we use scripts. For the cutoff energy convergence we can use the script from the last lab. Keep the lattice constant fixed at alat = 9.0 a.u.

- **[Report]** Consider a range of energy cutoffs, E_{cut}, and make a plot of your scf energy as a function of E_{cut}.

- **[Report]** Suggest an appropriate value for Ecut for an accuracy of 0.1 mRyd/atom. Support your answer.

Hint: It may not be necessary to fit a decaying exponential if you can adequately substantiate your answer by inspection of your numerical data. For example: "Assuming Etrue = E @ Ecut = XX Ry, then beyond an Ecut of YY Ry, the system energy is converged to within the desired accuracy of ZZ Ry/atom."

A4. k-point convergence

Modify the script from *A3* to loop over *even divisioned* k-point meshes ranging from $2 \times 2 \times 2$ to $20 \times 20 \times 20$ and calculate the energy for the different k-point meshes. Use the optimal E_{cut} value computed in *A3*, and keep the lattice constant fixed at alat = 9.0 a.u.

Hint: To scan over $2 \times 2 \times 2$ to $6 \times 6 \times 6$ k-point meshes, the beginning of the do loop in your modified file might look something like:

```
for kpts in "  2 2 2   0 0 0" "  4 4 4   0 0 0" "  6 6 6   0 0 0" ; do
    sed "/  4 4 4   0 0 0/s/.*/ $kpts/" $INPUTFILE  | $PWSCF > out
```

But you may find other ways to achieve the same result, such as a "here document" (there is an example of how to do this in the `/class/mse404ela/Bash-project` directory).

- **[Report]** Make a plot of your scf energy as a function of the resolution of your k-point mesh.

- **[Report]** Suggest an appropriate k-point mesh for an accuracy of 0.1 mRyd/atom. Support your answer.

Hint: It may not be necessary to fit a decaying exponential if you can adequately substantiate your answer by inspection of your numerical data. For example: "Assuming Etrue = E @ k-point mesh with resolution X-by-X-by-X, then for k-point meshes with resolution higher than Y-by-Y-by-Y, the system energy is converged to within the desired accuracy of ZZ Ry/atom."

A5. Converged geometry relaxation

Once you have selected your cutoff energy and k-point mesh size to be within 0.1 mRyd/atom, modify `Al.relax.inp` to reflect these values and re-run the relaxation.

Hint: If you get the error `Error in routine scale_h (1): Not enough space allocated for radial FFT: try restarting with a larger cell_factor` try a better initial guess for your lattice parameter.

- **[Report]** What is your calculated value for the relaxed lattice constant in a.u. using your converged Ecut and k-point mesh?

- **[Report]** Find (from an appropriate reference) the lattice constant for aluminum at room temperature. How does your value compare with the experimental value?

A6. Bulk modulus

You've computed the lattice constant for Al with density functional theory; now, let's compute some elastic constants. As we talked about earlier, it's better to separate the strains that change volume from those that conserve volume. The bulk modulus is defined to be the (linear) change in pressure with a small change in volume per atom. We will use a finite difference approximation to extract this. Edit your `Al.scf.inp` script to use your converged values of Ecut, kpoint mesh, and your calculated lattice constant. Run this calculation and extract the computed pressure (derivative of total energy with respect to volume) using

```
$ grep 'P=' Al.scf.out
```

you will see a line that looks something like

```
total   stress   (Ry/bohr**3)                    (kbar)     P=    -0.73
```

though your number may be different.

- **[Report]** Now, create two new input files, one with with a unit-cell volume that is 1% smaller than that corresponding to your calculated relaxed lattice constant, and one with a unit-cell volume 1% larger. After you run each script, you will now have the pressure at three different volumes. Use this data to compute the bulk modulus, defined as,

$$K = -V\frac{dP}{dV}$$

at the volume, V, corresponding to the equilibrium volume (where $P = 0$).

Hint: The unit-cell volume, V, and a_0 are related as: $V = (a_0^3)/4$. N.B. Astute users will note that this data can be used to improve your estimate of the equilibrium lattice constant.

- **[Report]** Find (from an appropriate reference) the bulk modulus for aluminum at room temperature. How does your value compare with the experimental value?

A7. Shear modulus: C_{44}

The next step is to compute the two shear moduli. Aluminum is cubic, and so there are three unique elastic constants: $C_{11} = C_{22} = C_{33}$, $C_{12} = C_{13} = C_{23}$, and $C_{44} = C_{55} = C_{66}$, while all others are zero. This is standard (Voigt) notation, where "1"=*xx*, "2"=*yy*, "3"=*zz*, "4"=*yz=zy*, "5"=*xz=zx*, "6"=*xy=yx*. Unlike an isotropic crystal that has only two unique elastic constants, cubic materials have three; this means

- The Young's modulus of a single crystal depends on the *orientation* of the tensile axis; the stiffness along $[100]$ is different than $[110]$ and $[111]$.

- The shear modulus depends on the *orientation* of the shear; a $[100](010)$ shear has a different stiffness than $[1\bar{1}0](110)$ shear.

We've already calculated one elastic constant, the bulk modulus K. In a cubic material,

$$K = \frac{C_{11} + 2C_{12}}{3}$$

So, what remains is to determine $C' = (C_{11} - C_{12})/2$ and C_{44}; these are both shear moduli. We need to systematically apply strain, and compute the stress, to determine the elastic constant. We'll start with the simpler case of C_{44}; this is the linear relationship between the shear strain e_4 and shear stress σ_4. Recall that if $e_4 = \gamma$, then the strain tensor is

$$\underline{\varepsilon} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \gamma/2 \\ 0 & \gamma/2 & 0 \end{pmatrix}$$

while the stress tensor will be

$$\underline{\sigma} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & C_{44}\gamma \\ 0 & C_{44}\gamma & 0 \end{pmatrix}$$

The final consideration is to construct the strain tensor to *conserve volume*. We can compute the fractional change in volume as

$$\frac{\Delta V}{V} = \det(\underline{1} + \underline{\varepsilon}) - 1 = -\frac{\gamma^2}{4}$$

which is a small change for small γ values; we'll be investigating values as large as $\gamma = 0.01$. A simple fix is to add in a *quadratic* normal strain in a perpendicular direction to correct this:

$$e_1 = \frac{4}{4 - \gamma^2} - 1 \approx +\frac{\gamma^2}{4}$$

We need to strain the crystal by changing the lattice vectors; in FCC, they are

$$\mathbf{a}_1 = \frac{a_0}{2}\mathbf{y} + \frac{a_0}{2}\mathbf{z}$$

$$\mathbf{a}_2 = \frac{a_0}{2}\mathbf{x} + \frac{a_0}{2}\mathbf{z}$$

$$\mathbf{a}_3 = \frac{a_0}{2}\mathbf{x} + \frac{a_0}{2}\mathbf{y}$$

To be able to easily enter arbitrary lattice vectors, use the `ibrav = 0` option, and then you can use the `CELL_PARAMETERS` card as such (c.f. INPUT_PW.html:CELL_PARAMETERS)

```
CELL_PARAMETERS alat
  0.0 0.5 0.5
  0.5 0.0 0.5
  0.5 0.5 0.0
```

would give us an FCC crystal. **Be aware:** the cell parameters card is *only* used if `ibrav=0`. The first line is the first lattice vector, the second line is the second lattice vector, and the third line is the third lattice vector. In this case, the units of the lattice vectors are `alat` (as specified by `celldm(1)`).

You will need to apply the strains to the lattice constants. **Keep in mind:** each vector is strained individually. So, for example, if we use the strain matrix above, the first lattice vector $\mathbf{a}_1 = \frac{1}{2}\mathbf{y} + \frac{1}{2}\mathbf{z}$ will become, under the strain above,

$$
\mathbf{a}_1' = (\underline{1} + \underline{\varepsilon})\mathbf{a}_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \gamma/2 \\ 0 & \gamma/2 & 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} = \left(\frac{1}{2} + \frac{\gamma}{4} \right) \mathbf{y} + \left(\frac{1}{2} + \frac{\gamma}{4} \right) \mathbf{z}
$$

and similar for the other two lattice vectors.

- **[Report]** Create two input files, one with a shear strain of 0.5% and another with 1%, to compute the shear stresses (you may find a script useful for doing the math). Check that your stresses are linearly related to your strains, and determine C_{44}.

- **[Report]** Find (from an appropriate reference) the value of C_{44} at room temperature. How does your value compare with the experimental value?

A8. Shear modulus: $C' = (C_{11} - C_{12})/2$

For C', we'll need a positive normal strain and a negative normal strain; for example, if

$$
e_1 = \gamma/2, \qquad e_2 = -\gamma/2
$$

while all other strains are zero, the normal stresses are

$$
\sigma_1 = \sum_{j=1}^{6} C_{1j} e_j = C_{11} \frac{\gamma}{2} - C_{12} \frac{\gamma}{2} = \frac{C_{11} - C_{12}}{2} \gamma
$$

and

$$
\sigma_2 = \sum_{j=1}^{6} C_{2j} e_j = C_{21} \frac{\gamma}{2} - C_{22} \frac{\gamma}{2} = -\frac{C_{11} - C_{12}}{2} \gamma
$$

9

so that the stresses should be equal and opposite. All other stresses should be zero, *as long as we're in the linear limit*. This also assumes that the cell initially has zero stress: we need to be at equilibrium. You can eliminate small nonzero stress by computing

$$\frac{\sigma_1 - \sigma_2}{2} = \frac{C_{11} - C_{12}}{2}\gamma$$

The final consideration is to construct the strain tensor to *conserve volume*. We can compute the fractional change in volume as

$$\frac{\Delta V}{V} = \det(\underline{1} + \underline{\varepsilon}) - 1 = -\frac{\gamma^2}{4}$$

which is a small change for small γ values; we'll be investigating values as large as $\gamma = 0.01$. A simple fix is to add in a *quadratic* normal strain in a perpendicular direction to correct this:

$$e_3 = \frac{4}{4 - \gamma^2} - 1 \approx +\frac{\gamma^2}{4}$$

- **[Report]** Create two input files, one with a shear strain of 0.5% and another with 1%, to compute the shear stresses (you may find a script useful for doing the math). Check that your stresses are linearly related to your strains, and determine $C' = (C_{11} - C_{12})/2$.

- **[Report]** Combine all of your data to determine C_{11} and C_{12}. Find (from an appropriate reference) the value of the elastic constants at room temperature. How does your value compare with the experimental value?