**Frequency Calculation Method**

1. Copy final structure into a folder on pypc-noether and name it POSCAR
2. Generate displaced structures with Phonopy: phonopy -d --dim="1 1 1"

This will produce a set of structures labelled POSCAR-001 … POSCAR-NNN

1. Copy POSCAR-\* to a new folder on BALENA
2. Run a single-point force calculation on each structure in VASP

Attached INCAR-frequencies file and Frequencies.slm script. Check that the correct modules are being loaded and the mpirun line is the same as for your optimisation calculations. I have run these calculations 2 different ways (from code on BALENA and in my local folders).

In Frequencies.slm change the counter to how many structures can run in one free queue, eg if 12 can run then the line will read:

if [ $counter -gt 11 ]; then

It is worth running the calculation once with

if [ $counter -gt 0 ]; then

to find out how fast a single structure calculation can run and therefore how many can run in a single 6hr limit on the free queue. Then you can change the number and run the remaining number of required jobs using Frequency.slm

1. Check all calculations are complete: grep Elapsed \*/OUTCAR | wc -l
2. Copy vasprun files to your folder on pypc-noether
3. Collect forces: phonopy -f vasprun-{001..NNN}.xml

This should produce a "FORCE\_SETS" file

1. Calculate frequencies and eigenvectors: phonopy --dim="1 1 1" --eigenvectors --mesh="1 1 1" --fc-symmetry

This will produce a mesh.yaml file containing the frequencies in THz and the eigenvectors

1. Calculate the irreducible representations: phonopy --dim="1 1 1" --irreps="0 0 0"

**Raman Intensity Calculation**

1. Add Jonathan Skelton’s files to the python path:

In Phonopy-Spectroscopy-master:

export PYTHONPATH=${PYTHONPATH}:’pwd’

cd Scripts/

export PATH=${PATH}:’pwd’

1. From the irreducible representations find which bands are Raman active- you can find this out from the character table for the symmetry group (Phonopy can give you the symmetry group: phonopy --symmetry)
2. Generate displaced structures: phonopy-raman -d –bands=”1 2 3 4 5 …..”

You can do the calculation for all the bands, but for the size of your system I would only run the calculations on bands you expect to be Raman active otherwise you will be waiting a long long time for the calculations to complete

1. Copy Raman-POSCAR files to a new folder on BALENA
2. Run dielectric constant calculations on the displaced structures

Attached INCAR-Raman file and Raman.slm script. Again check the correct modules are loaded and the mpirun is correct

Adjust the counter as before (line: if [ $counter -gt 5 ]; then)

1. Check all the calculations are complete: grep Elapsed \*/OUTCAR | wc -l
2. Copy OUTCAR files to pypc-noether
3. Extract dielectric tensors: phonopy-raman -r OUTCAR.\*
4. Calculate spectra phonopy-raman -p

**Phonon dispersion calculations method**

(instructions from Jonathan Skelton)

1. Generate supercells with Phonopy: phonopy -d --dim="X Y Z" (you already did this).

2. Run a single-point force calculation on each structure in VASP (as in Frequency calculation); once done, collect the vasprun.xml files from each run and make sure you name them consistently, e.g. vasprun-001.xml, vasprun-002.xml, ..., vasprun-NNN.xml

3. Collect forces: phonopy -f vasprun-{001..NNN}.xml (this should produce a "FORCE\_SETS" file).

----------

4. You're now ready to do some post-processing. First, it's worth checking the frequencies at the commensurate points - the wavevectors at which your chosen supercell allows you to calculate the frequencies exactly:

phonopy --dim="X Y Z" --fc-symmetry --mesh="X Y Z" --gamma-center

This should produce a "mesh.yaml" file; open it in a text editor (the format is fairly "human readable") and check the lists of frequencies at each wavevector for imaginary modes (-ve values) - ideally there shouldn't be any.

----------

5. Generate a phonon density of states:

phonopy --dim="X Y Z" --fc-symmetry --mesh="16 16 16" --gamma-center --thm --fpitch=0.1 --nowritemesh -p -s

(In rough order, this command is asking Phonopy to: symmetrise the force constants, interpolate the frequencies onto a uniform 16x16x16 grid centred on the \Gamma point at q = [0, 0, 0], do the integration using the tetrahedron method, generate the DoS with a pitch between data points of 0.1 THz, not write a gigantic and not-terribly-useful output file, and plot and save the result.)

Annoyingly, different versions of Phonopy use slightly different command-line options, so depending on which version you have you might get an error message. If you do, e-mail it to me and I'll correct it.

----------

6. Draw a dispersion. You'll need to know the wavevector path that you want to follow. If you're trying to compare your dispersion to the literature, you might be able to take one from there; if not, and you need to generate your own, you can use the "KVEC" program on the Bilbao crystallographic server to look at the Brillouin zone:

<http://www.cryst.ehu.es/cryst/get_kvec.html>

You'll need to know the spacegroup of your system - if you don't, Phonopy can tell you:

phonopy --symmetry

Once done, you can use for example:

phonopy --dim="X Y Z" --fc-symmetry --band="1/2 0 0  0 0 0  1/2 1/2 1/2" --band-points=101 --band-labels="X G R" -p -s

(This asks Phonopy to draw a dispersion between three vectors with 101 points per segment and label them X, G and R in the plot it generates.)

Some of the commands I was using:

phonopy --dim="2 4 1" --fc-symmetry --mesh="8 16 4" --gamma-center --fpitch=0.01 --nowritemesh -p -s

phonopy --dim="2 4 1" --fc-symmetry --band="0 0 1/2 -1/2 0 1/2 -1/2 1/2 1/2 0 1/2 1/2 0 0 0 0 1/2 0 1/2 1/2 0 1/2 0 0 0 0 0" --band-points=101 --band-labels="Y X D A GM B L Z GM" -p –s

phonopy --dim="2 4 1" --fc-symmetry --mesh="8 16 4" --gamma-center --fpitch=0.01 --nowritemesh -p -s -t --tmax=500 --tstep=1

Points in the Brillouin zone for coronene system

|  |  |  |  |
| --- | --- | --- | --- |
| Y | 0 | 0 | ½ |
| X | -½ | 0 | ½ |
| D | -½ | ½ | ½ |
| A | 0 | ½ | ½ |
| GM | 0 | 0 | 0 |
| B | 0 | ½ | 0 |
| L | ½ | ½ | 0 |
| Z | ½ | 0 | 0 |
| GM | 0 | 0 | 0 |

**Pseudo-Gibbs calculation**

https://atztogo.github.io/phonopy/qha.html#phonopy-qha

1. Adjust the volume of the unit cell +/- 2.5% and +/-5%. Suggest do this manually by changing unit cell parameters, then for each structure:
2. Run structural optimisation with ISIF = 4
3. Calculate the phonons as before
4. Calculate the thermal properties as before
5. Plot Helmholtz energy (T=0K) vs volume using the equation in the link to make sure the results make sense

If you then plot the total energy versus volume, it should fit an equation of state like the Birch-Murnaghan equation: https://en.wikipedia.org/wiki/Birch%E2%80%93Murnaghan\_equation\_of\_state.

1. Create the e-v file and name the thermal property files appropriately
2. Run the phonopy-qha script

The analysis is done with the phonopy-qha script: https://atztogo.github.io/phonopy/qha.html