DWFWriter Instructions:

The DWFWriter, (Debye Waller Factor Writer) has become the main workhorse of these programs. It takes in the various pieces of information given by Dr. Lindsay, runs them through Xu Ruqing’s equations, and returns scattering intensities with and without a thermal gradient.

First, the wb loads the Excel Workbook that the file will later manipulate. Similarly, ws (worksheet) will select the active worksheet at the time of the Excel Workbooks last session. y is the starting number at which the data points begin, and z is one number beyond the last data point in the sheet (row 12295). This ensures that the following math will occur for all of the data points.

A helper function, mag2, is used to calculate the magnitude squared of a value.

h is set to the reduced Plank’s constant (in units of m2 kg / 2).

w becomes the set of angular frequency values for each of the data points. First, I initialize an array equal to the amount of data points in the sheet (range(y, z)). Then, for each data point, I multiply the given frequency by 2 \* pi \* 1012 in order to obtain the angular frequency of that particular data point in radians (note that the frequency in the excel sheet is in THz, hence the 1012). This is the first instance of storing excel data in a python array, which is used throughout the program. w is the array, which starts at row 8 (w[0] = w[i-y]) and ends at row 12295. In order to retrieve the data from a cell using openpyxl, you use ws[‘Cell number’].value. I wrote this such that all of one particular type of data would be in a specified column (F in the case of frequency) and then I parse together strings that are indexed according to their row in the excel sheet.

From here, I add in additional constants and variables as they are needed, with kb equaling Boltzmann’s constant in units of (m2 kg / s2 K), and set the absolute temperature of the system to 300 K.

At this point, this gives us enough data to determine the Bragg scattering of the crystal (which is not really used in this experiment). This data is then transcribed into column K by using the openpxyl function (ws[‘cell name’] = value).

From here, I set N equal to Avogadro’s number (arbitrarily).

Next, in f, I set the values for the atomic form factor for the two atoms in the unit cell of each material. These values were acquired by online tables. http://lampx.tugraz.at/~hadley/ss1/crystaldiffraction/atomicformfactors/formfactors.php

In order to acquire the Thomson scattering (amount of scattering per incident photon), given by the equation below:

I arbitrarily set I\_incident (the amount of photon flux) to 1 photon per meter (which is likely why my values end up being so small), d (distance to observer) to 1 m, r0, the classical electron radius to the value 2.82 \* 10-15, and both 2θ and φ to 900. For the actual experiment, these values should be adjusted to reflect their real values.

Next, I set a to the lattice constant for the appropriate number for each material (I never changed the Diamond in all of the comments, but I have verified that each of the values are correct for each of the materials).

tau is the set of basis vectors in the unit cells (which is the same for each of the investigated materials).

q is the array of three dimensional vectors set by columns C, D, and E (x, y and z respectively) and represent the vectors in reciprocal space (in units of A-1).

mu is set to the atomic mass of the material in question (since each crystal is monotonic, this can be set to a constant, but for materials with more than one element, this would need to turn into an array quantity).

At first, the temperature gradient (Grad\_T) is set to zero, in order to measure the thermal diffuse scattering without a temperature gradient.

F is set to the normalized distance away from equilibrium for each of the data points, as provided and described by Dr. Lindsay.

These values allow us to calculate the change in phonon populations due to a thermal gradient, which is set to the eta\_1.

Once I obtain eta\_1, I then calculate eta total by adding it to the initial contribution:

eta = eta\_0 + eta\_1.

I then use this value to obtain the Debye Waller factor for each data point, and then sum them all together, as per the equation:

  
which is equation 2.58 in Xu’s paper.

k\_l is set to all of the reciprocal lattice vectors in the unit cell of a diamond cell.

I\_0 is the intensity due to Bragg Scattering

I\_1 is the intensity due to thermal diffuse scattering.

The two aforementioned values are then recalculated for an applied thermal gradient.

Once everything is calculated, in order to save the values, I use the save function:

wb.save(‘name of file you want to save to’)