The Chinese University of Hong Kong

PHYS4061 Computational Physics, 1st semester, 2019-2020

Project A Part II

Reuse your code in part I. Step 3 and 4 require a lattice structure input. Build your part II on top of your part I. If you write a completely independent code, there can be a lot of bug when you combine the two part/read .xyz files from lab1. Also check again does your code submitted for part I satisfy all the requirement.

Remember to write process into small function so they can be reuse easily.

The unit cell is the conventional cell. It is built from cube cells with length a, repeating n_x, n_y and n_z times in the x, y, z direction respectively. And the lattice vectors a_1, a_2 and a_3 are (an_x,0,0), (0, an_y,0), (0,0,an_z). Write your code such that you can change a and n_i easily. This is also a requirement in part I: The simulation cell size should be adjustable in both periodicity n_i and lattice constant a.

1. Reciprocal Lattice Code:

   Calculate a reciprocal lattice vectors b_1, b_2 and b_3 from the lattice vectors of a_1, a_2 and a_3 (unit cell) in real space, calculate and return the volume of the unit cell. (you can calculate the volume first, and reuse it in calculating b_i)

   a. Store the x, y, z coordinates of a_i and b_i in arrays.

   b. Test primitive vectors of SC, BCC, FCC to make sure the code is correct. Check the Lecture Notes for primitive vectors of SC, BCC, FCC.

   c. You do not need to include the 2π factor inside the reciprocal lattice.

2. Periodic Boundary Condition code:

Choose a unit cell and prepare its lattice vectors (a_1, a_2, a_3). Input a coordinate (x_1, y_1, z_1) to apply periodic boundary conditions, return the fractional coordinates (n_1, n_2, n_3) in the range of (-0.5, 0.5) and the (x_2, y_2, z_2).

   a. (x_1, y_1, z_1) is a Cartesian coordinate of your choice. It can be either inside the unit cell or outside of it.

   b. Think before you work on the fractional coordinates: what is the physical meaning of range (-0.5,0.5) and how to get this range.

   c. The (x_2, y_2, z_2) is get from $\sum_{i=1,2,3} n_i \vec{a}_i$.

   d. you can change {a_1, a_2, a_3} and (x_1, y_1, z_1) to test can your code return the (n_1, n_2, n_3) and the (x_2, y_2, z_2) correctly.

(Hint: call the code of reciprocal to have (b_1, b_2, b_3) ready, then take the dot product and mod.)

3. Neighbor List code:

Generate a lattice structure. Prepare lattice vectors a_1, a_2 and a_3 according to the lattice structure generated. Choose a reasonable distance cutoff for first neighbor. Use PBC code to calculate the neighbor list of each atom, which contain the true distance from the neighbor to it.

The true distance is the distance between two atoms after applying periodic boundary condition.

(Hint: Here you need to use your part I. You should have n_i and lattice constant a ready to get the a_i and b_i. Make use of the position coordinates you code generated.)

4. Neighbor List of different structure;

   Use the code in 3. Check the neighbor list and shortest distances of simple cubic, FCC and BCC.

Score will be deducted if the program has bad variable/function naming conventions or without adequate comments. Submit code and output to ylfchong@phy.cuhk.edu.hk

Due date: 1 Oct 2019

Score will be deducted if the submission is late.