Modeling and Simulation - Hand in 1

Georgios Smyridis – 1840118

March 1, 2023

Exercise 4

4.1

The code for the generation of the primitive cubic lattice is written in the file PC_Generator.c.

4.2

The primitive cubit lattice has a unit cell containing exactly one particle. In our case, the lattice spacing is l, and therefore, the unit cell has volume $V = l^3$. The diameter of the hard sphere cannot be bigger than the lattice spacing, because otherwise, there will be overlaps. Another way to see it, is that the one sphere contained in the unit cell would exceed its limits. Thus, the maximum radius of the particle is $r_{max} = l/2$ and the maximum volume is:

$$V_{sphere} = \frac{4}{3}\pi r^3 = \frac{\pi}{6}l^3 \tag{1}$$

It is straightforward to see that the fraction between the volume of the sphere and the volume of the cell is

$$f_{PC} = \frac{V_{sphere}}{V_{cell}} = \frac{\pi}{6} \tag{2}$$

4.3

The code for the generation of the FCC lattice is written in FCC_generator.c.

4.4

Similarly to question 4.2, we want to calculate the maximum . In the face-centered cubic lattice with spacing l, the balls will barely touch on each face when the diagonal is equal to two diameters, which would mean that the maximum radius the spheres can acquire is $r_{max} = l/2\sqrt{2}$. We have to note that in an FCC lattice, the unit cell contains 4 spheres in total. Then, it is straightforward to calculate that the maximum volume is

$$V_{sphere} = \frac{\pi l^3}{3\sqrt{2}} \tag{3}$$

and the fraction is

$$f_{FCC} = \frac{\pi}{3\sqrt{2}} \tag{4}$$

It is thus obvious, that the maximum volume fraction one can obtain with an FCC lattice without overlaps is larger than the one obtained for a cubic lattice.

4.5 & 4.6 & 4.7

In HardSphereMonteCarlo.c there are the two main subroutines that are required. The read_data(), which imports the data from the dat file that has been generated, and the move_particle() which attempts to displace a random sphere, with δ being the maximum displacement of the spheres in each direction. The displacement is chosen randomly from a homogeneous distribution between $-\delta$ and δ for all three directions. A reasonable value for δ is the lattice spacing l, since if it was significantly larger than it, it's most probable that the spheres would have collided before this displacement was completed. If δ was much smaller, then nothing interesting would have happened. In theory, this δ would depend on the temperature. If the displacement leads to an overlap, it is detected by the overlap() function and it's rejected.

4.8

After defining all the functions needed, the Monte Carlo Simulation was executed with packing fraction 0.6. All the snapshots of the states of the crystal were collected into a gif named MonteCarloSimulation.gif.

4.9

Finally, the Monte Carlo Simulation was ran for different parameters, namely, 0.4, 0.47, 0.5 and 0.6. One can observe that, at a packing fraction of 0.4, the spheres can escape their starting position after some time. For packing fraction 0.6, the movement is more constrained and the particles remain close to their initial position. Finally, after reviewing visually the simulations, a rough estimate of the packing parameter for which the crystal of hard sphere melts is 0.47.