This code is divided into three main parts implemented as classes: MolDyn, RDF, and Plots.

For the MolDyn class, this will implement the core of the molecular dynamics simulations. The positions, velocities, and accelerations at each time steps are stored in the 3-dimensional arrays *pos_config*, *vel_config*, and *acc_config*. The first index is the timestep, the second is the particle number, and the third are the coordinates. For instance, *pos_config[120,64,1]* returns the position(0->x,1->y,2->z coordinate) of the 64th particle for the 120th timestep.

For the RDF class, this will implement on how to get radial distribution function and the cumulative distribution function (i.e. the coordination number). All the $N(N-1)/2$ distances computed from unique pairs of particles were used to create the histogram for a specific timestep. In other words, all the particles were considered as reference particles for computing the RDF. The obtained RDF should be then divided by N reference particles. The final RDF was taken to be the average of 8 RDFs from equally spaced time intervals.

Lastly, the Plot class will implement procedures for plotting different graphs of interest.

How to run the code?

Use the **instances.py** file to run the code. This file will create a instance of the classes used in the code. You can then change the parameters such as temp, box_scale, rcut_scale, thermo_rate, equilibration, Niter (see the MolDyn class for the meanings of these parameters).