Physics of Soft Condensed Matter Project
Lennard Jones System

Name: Swetanjal Dutta
Roll Number: 20171077

Negative gradient of potential Energy Function gives the Force on the particle:

$$\circ \quad F_{xij} = -\frac{\partial V_{ij}}{\partial x_i}$$

$$\circ \quad F_{xij} = \left[4\epsilon\left\{6\left(\frac{\sigma}{r}\right)^6 \cdot \frac{1}{r}\right\}\left(\left(\frac{\sigma}{r}\right)^6 - 1\right) + 4\epsilon\left(\frac{\sigma}{r}\right)^6\left(6\left(\frac{\sigma}{r}\right)^6 \cdot \frac{1}{r}\right)\right]\frac{\partial r}{\partial x_i}$$

$$\circ \quad F_{xij} = 4\epsilon\left\{6\left(\frac{\sigma}{r}\right)^6 \cdot \frac{1}{r}\right\}\left(2\left(\frac{\sigma}{r}\right)^6 - 1\right)\frac{\partial r}{\partial x_i}$$

$$\circ \quad r = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2 + (z_j - z_i)^2}$$

$$\circ \quad F_{xij} = 24\epsilon\left(\frac{\sigma}{r}\right)^6 \cdot \frac{1}{r^2}\left(2\left(\frac{\sigma}{r}\right)^6 - 1\right)(x_i - x_j)$$

Velocity Verlet Integration Algorithm to solve the equations of motion from the initial configurations:

Velocity Verlet Integration

Expand $\mathbf{r}(t)$: $\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t)\Delta t + \frac{\mathbf{f}(t)}{2m}\Delta t^2 + \cdots$

$\mathbf{r}(t+\Delta t) + \mathbf{r}(t-\Delta t) = 2\mathbf{r}(t) + \frac{\mathbf{f}(t)}{m}\Delta t^2$

$\mathbf{r}(t+\Delta t) - \mathbf{r}(t-\Delta t) = 2\mathbf{v}(t)\Delta t \quad \leftarrow$ def of discrete derivative

Initialization step: Calculate the initial positions $\mathbf{r}_0$, velocities $\mathbf{v}_0$, and the initial force acting on each particle $\mathbf{f}_0$

Run Algorithm:

(1) calculate all half-step velocities: $\mathbf{v}_{n+1/2} = \mathbf{v}_n + \frac{\mathbf{f}_n}{2m}\Delta t$

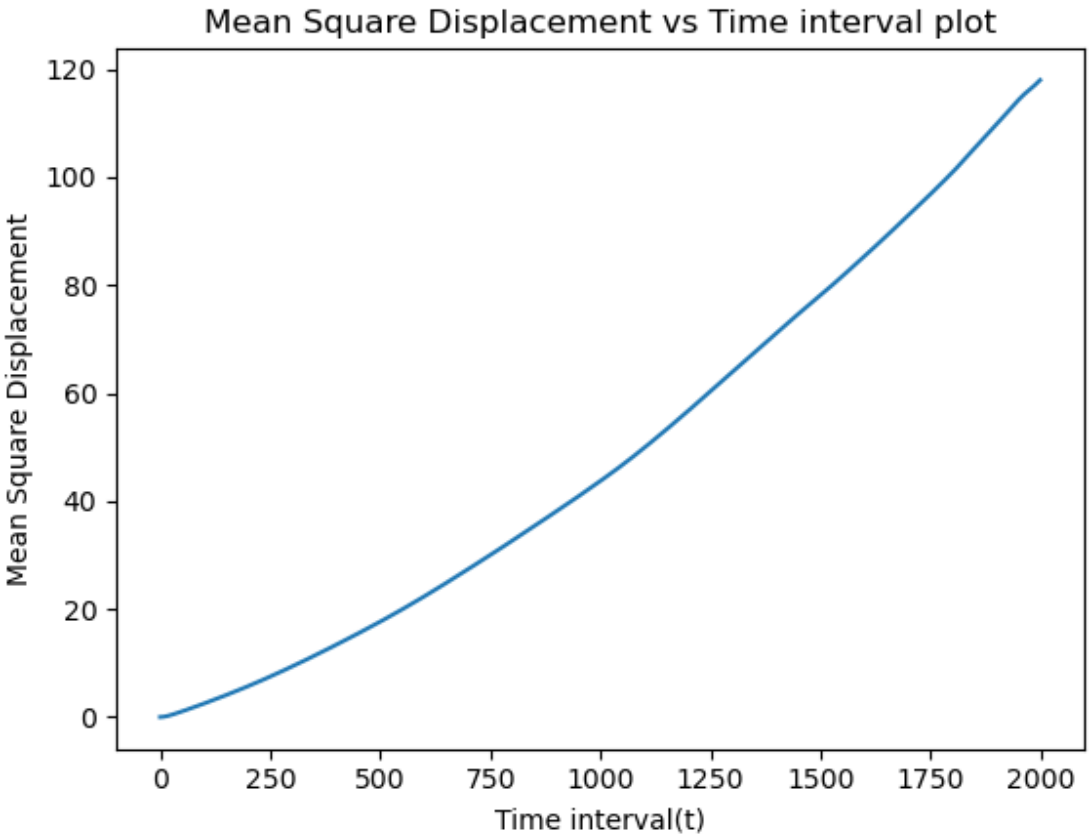
(2) update positions for all particles: $\mathbf{r}_{n+1} = \mathbf{r}_n + \mathbf{v}_{n+1/2}\Delta t$

(3) calculate new forces acting on all particles: $\mathbf{f}_{n+1}$

(4) update the velocities for all particles: $\mathbf{v}_{n+1} = \mathbf{v}_{n+1/2} + \frac{\mathbf{f}_n}{2m}\Delta t$

Computing Mean Square Displacement and Diffusion coefficient:

Instructions to run code and generate plot:
Refer Readme



Computing the Diffusion Coefficient:
The slope in the final part(linear part starting around 880 and ending around 1900 along x axis) of the graph gives the diffusion coefficient = 0.07
Algorithm to compute mean square displacement:

We fix t. t is the number by which two frames under consideration is separated by. So for example if t is 1, we consider the frames (1,2); (2, 3); (3, 4); and so on until (n - 1, n). If t is 3, we consider the frame pairs (1,4); (2, 5); (3, 6) and so on until (n - 3, n).

Once t is fixed we consider all pairs of frames separated by t. We loop through all the atoms in the system and compute the square of the displacement for each corresponding atom and then taking the average over all the atoms.

We total this for all possible pairs of frames and finally divide by the number of pairs of frames for that t. This allows us to compute the mean square displacement which is not only averaged over all atoms but also averaged over different instants in time.

Please refer to the code for further details.

Analysis:

$$\langle r^2(t)\rangle = \frac{1}{N_\alpha}\sum_{i=1}^{N_\alpha}\langle |r_i(t) - r_i(0)|^2\rangle$$

The mean square displacement value increases as the number by which two frames are separated increases. This makes sense intuitively because the longer the time of separation, greater the value by which the particles will move away from their starting position. The particles will slowly diffuse from one region to another. Greater the time interval, greater will be the diffusion/displacement.

The mean-squared displacement of particles at an elapsed time ‘t’ is related to the second moment of the self van Hove correlation function. Typically for liquids, particle will show ballistic motions at very short times, followed by particle collisions resulting in a plateau, and eventually undergo random walk.

Diffusion coefficient is a proportionality constant which gives the quantity of a substance that is diffusing from one region to another passes through each unit of cross section per unit of time when the volume-concentration gradient is unity. As the definition suggests, this value can be computed by finding the slope in the msd graph.

Computing Velocity Correlation Function:

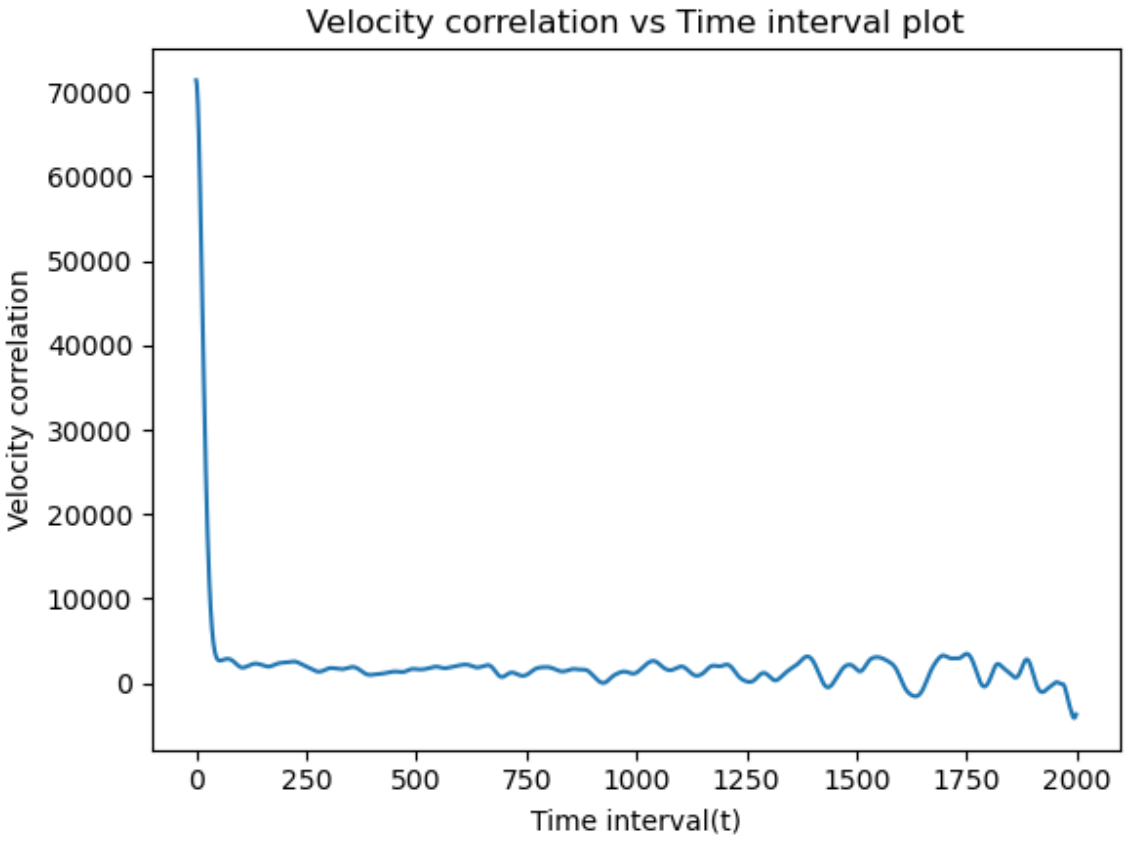
Formula:

$$D = \frac{1}{3}\int_0^\infty \langle \vec{v}(t)\cdot\vec{v}(0)\rangle dt = \frac{1}{3N}\int_0^\infty \sum_{i=1}^{N}(\vec{v}_i(t)\cdot\vec{v}_i(0))dt$$

Instructions to run code and generate plot:
Refer Readme

Algorithm to compute Velocity Correlation Function:

The steps are exactly the same as computing the mean square displacement.
Step 1: Fix t and consider all pairs of frames separated by t
Step 2: Loop through all atoms in the system and compute the dot product of their velocities at the two respective frames of the considered pair.
Step 3: Take the sum of velocity dot products for all atoms and across all frame pairs.
Step 4: Finally take the mean across all atoms, across all frame pairs to get the correlation value as plotted below.



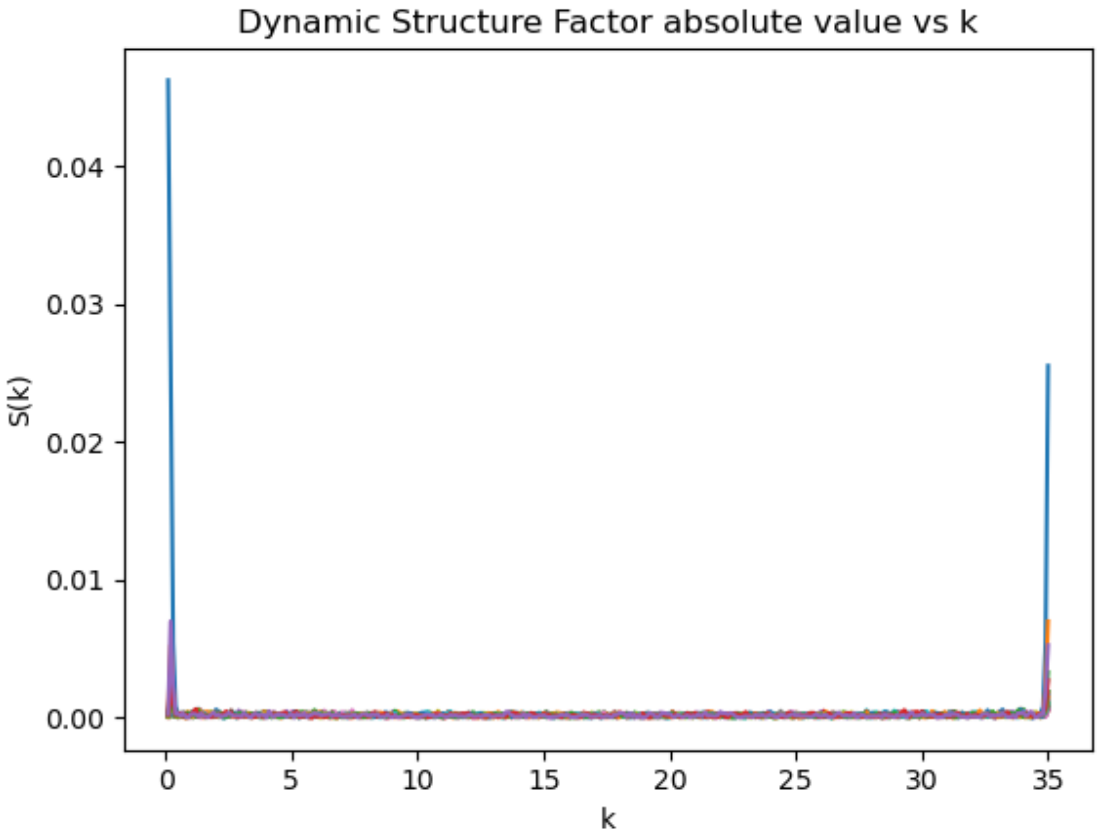
Analysis:
This statistical correlation function can be used to measure the self-diffusion coefficient or the vibrational properties of a given system.
The velocity correlation is maximum when frames are separated by 0. That is the dot products are computed with itself across all frames. This gives us the maxima in the graph. As frame separation increases, the velocity autocorrelation value decreases and for most of the other separation values, the correlation is almost 0.

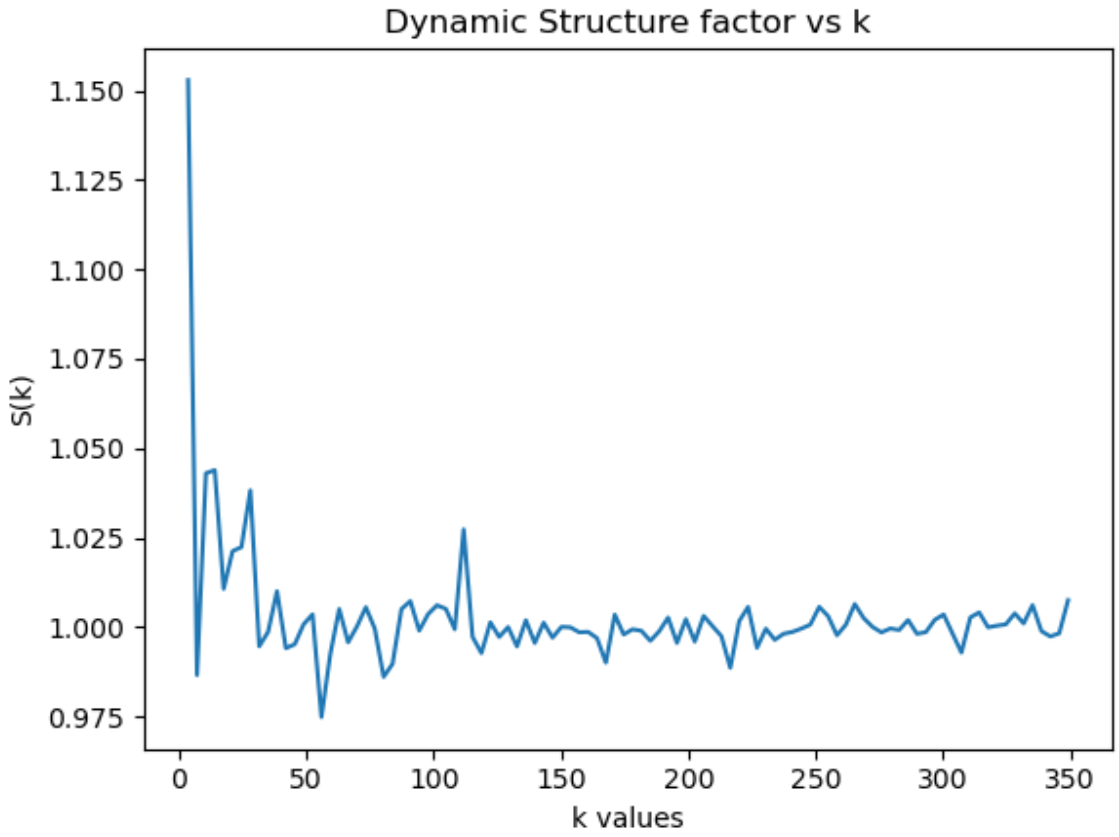
This time correlation function gives us hidden information regarding the memory of the system under study. For example, a sharp fall in the value indicates that the system

behaves like a memoryless Markovian System.

Computing Dynamic Structure Factor

Dynamic Structure Factor S(k) is simply a double Fourier transformation of the van Hove correlation function.





Computing van Hove correlation function

Instructions to run code:
Refer Readme

Algorithm to compute van Hove correlation function:

We have to plot the graphs for different values of t. That is different values of frame separation.

We fix a distance, say r. Now we compute how many pairs of atoms are separated by a distance r at frame 0 and frame t.
We take the average of this value across all pairs and finally do a division by N according to the definition.
Mathematically,

G(r, t) = $(1 \div N) * < \sum_{i,j} \delta(|r - r_j(t) + r_i(0)|) >$

where symbols have meanings as discussed in class.

Analysis:

$$G(r,t) = \frac{1}{N} \left\langle \sum_{i=1}^{N} \sum_{j=1}^{N} \delta(\mathbf{r} + \mathbf{r}_j(0) - \mathbf{r}_i(t)) \right\rangle$$

This function captures the time dependent variation of the radial distribution function or the structure factor.

No matter by how much the frames are separated by, the van Hove Correlation plots are almost the same. Only for separation of 0, we get a very noisy plot.

The Van Hove function is the probability density of finding a particle i in the vicinity of r at time t, knowing that a particle j is in the vicinity of the origin at time t=0.

The self part G_s(r,t) is the probability density of finding a particle i at time t knowing that this particle was at the origin at time 0. Probability that a particle has moved a distance r in time t (dynamics).

Self part is given by:

$$G_s(\mathbf{r},t) = \frac{1}{N} \langle \sum_{i=1}^{N} \delta((\mathbf{r} + \mathbf{r}_i(0) - \mathbf{r}_i(t)) \rangle$$

The distinct part G_d(r,t) is the probability density of finding a particle j different from i at time t knowing that the particle i was at the origin at time t=0. Probability to find at time t a different particle at a distance r from a place at which at time t=0 there was a particle. And G_d(r,0)=g(r) where g(r) is the radial distribution function

Distinct part is given by:

$$G_d(\mathbf{r},t) = \frac{1}{N}\langle \sum_{i\neq j}^{N} \delta(\mathbf{r} + \mathbf{r}_j(0) - \mathbf{r}_j(t))\rangle$$

In the long-time limit, the system loses memory of the initial configuration and the correlation functions become independent of the distance