Molecular Dynamic

Mahyar Albalan

403201223

Argon gas simulation

for simulating each particle movement first i set the initial conditions as was mentioned in the problem. i put every 100 argon atoms in the left side of the chamber in squared crystal structure.i choose initial velocities randomly aswell and make sure that CM's velocity is 0 .one challange i faced was in executing periodic boundry condition, because we have to calculate the shortest distance between each two particle and if this is not considered results will be unexpected.(with a simple trick i calculate this shortest distance, check the code for more details, other than this shortest distance the rest of executing periodic boundry condition was easy). i do the simulation in reduced unit, which means$\sigma = \epsilon = k_B = m = 1$, shortest distance between atoms in this reduced units is between 0.8 to 0.9 , i used 0.85 as shortest distance because it leads to better energy stability.but if we use lower values we can see velocities auto correlation will drope faster.i wrote functions for initial velocities, initial positions, acceleration of each particles based on their distance from lennard-jones intraction,one for calculating total energy of system based on atoms positions(calculating lennard-jones potential) and kenetic energy of particles, then for making a video for this simulation i wrote another function in which i calculated each particles movement change at 0.02 seconds intervals, and each 10 step i use it as a frame in output video, you can check the output of the video in jupyter nootebook file.i could use lower time steps for smoother transitions.the results of this simulation are as follows:

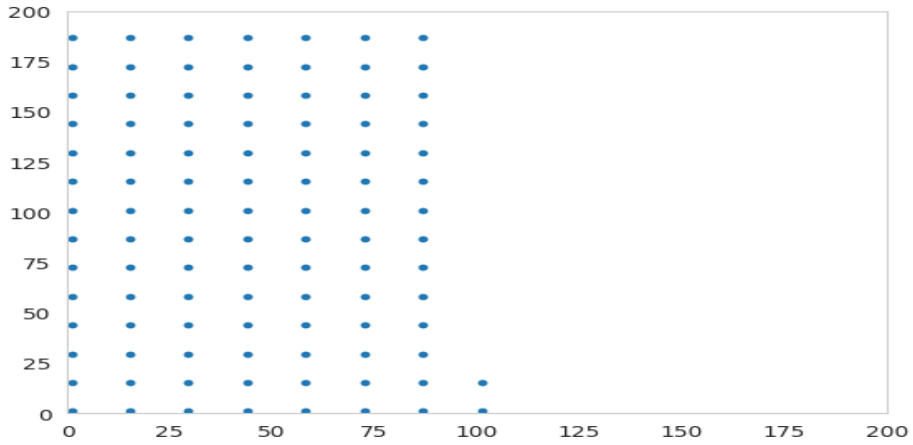

Figure 1: Initial Positions

As intended, all of the 100 particles start from the left. The system's dynamics are as follows. Alongside the system's dynamics, you can see plots for energy and the number of total particles on the left side.

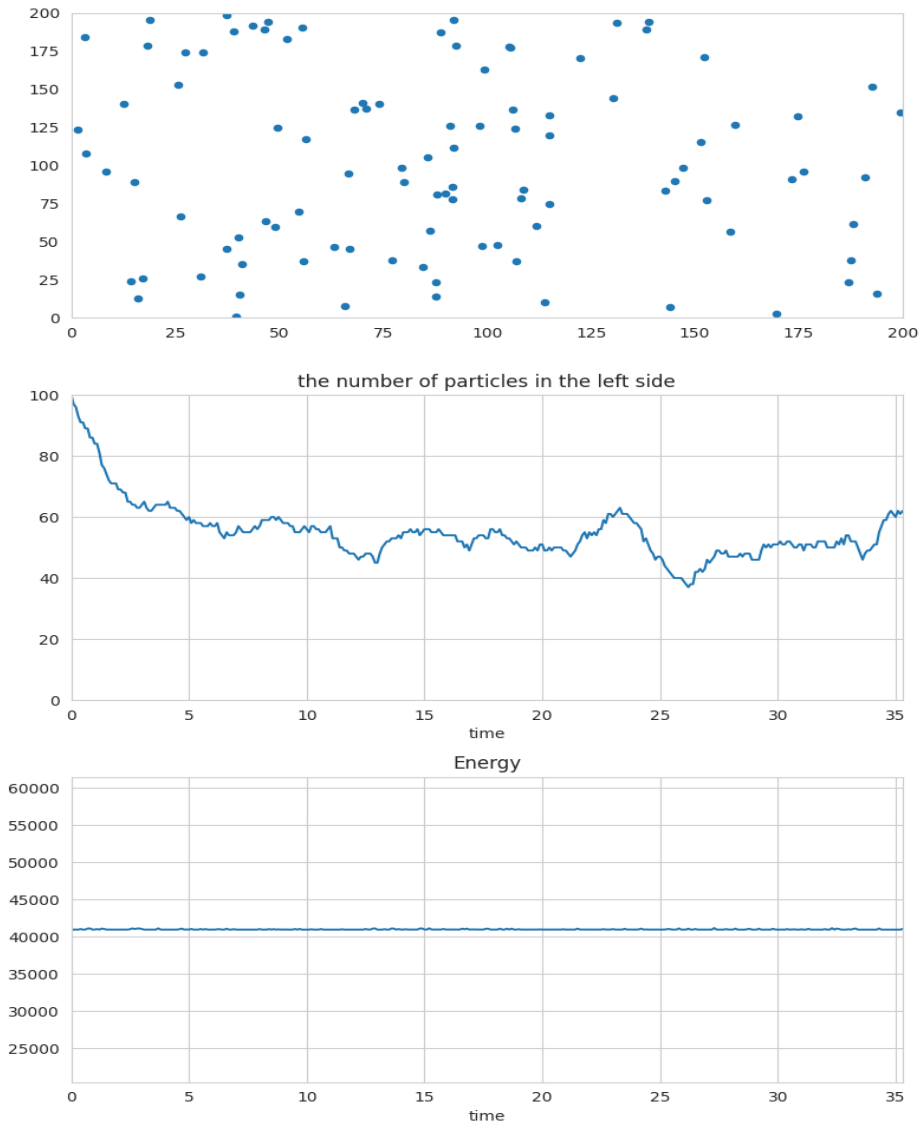

Figure 2: System's Dynamic

As you can see, after around 10 seconds, the number of particles on the left side reaches around 50 (half of the particles), which means it takes around 10 seconds for the system to reach equilibrium. From the energy plot, you can see there are just small fluctuations. I used the following relationship for calculating velocity autocorrelation:

$$\mathrm{VAC}(t) = \langle \mathbf{v}(t) \cdot \mathbf{v}(0) \rangle$$

the result is as follows:

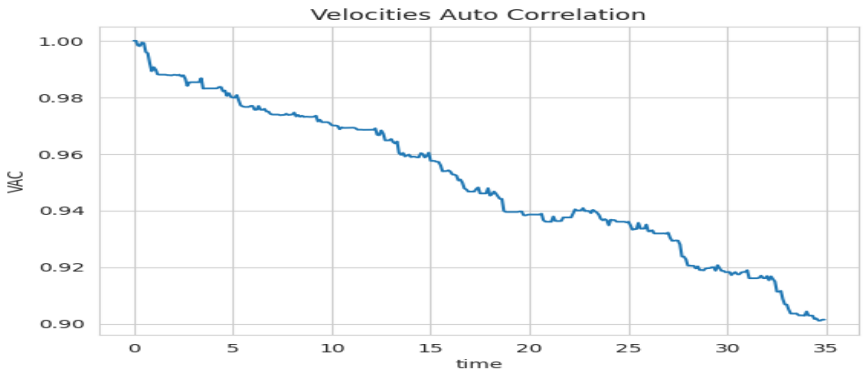

Figure 3: Velocities AutoCorrelation

As you can see, it is dropping. If we run the simulation for more time or use a smaller minimum distance between atoms, this drop will be faster and will reach expected behavior of $1/t$. For calculating temperature and pressure, I used the following formulas:

$$T = \frac{1}{2N-2} \sum_{i=1}^{N} \frac{1}{2} m \left|\mathbf{v}_i\right|^2$$

$$P = \frac{Nk_BT}{V} + \frac{1}{2V} \left\langle \sum_{i<j} \mathbf{F}_{ij} \cdot \mathbf{r}_{ij} \right\rangle$$

the results are as follows:

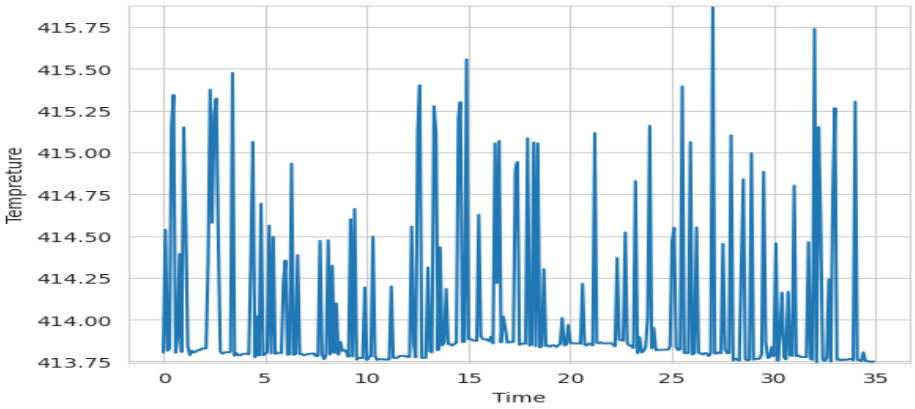

Figure 4: Tempreture vs time

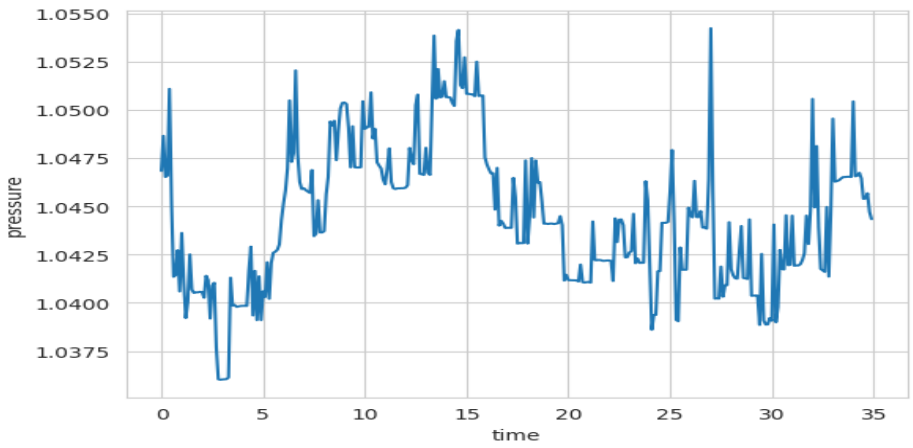

Figure 5: Pressure vs time

For calculating Deffusion coefficient, I used :

$$D = \lim_{t\to\infty} \frac{1}{4t} \left\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \right\rangle$$

I plot D versus time; the value it reaches will be the diffusion coefficient. The result is as follows:

Figure 6: Diffusion Coefficient

as you can see diffusion coefficient is around 0.55 .

We calculated Tempreture, Volume and Pressure, i simulate 10 different system's with different energies and then fit a line to them to calculate a and b using Van-Der-Walls equation,in SI units there were:

$$a = 2.628 \qquad ; \qquad b = 0.0189$$

as you can see values calculated here are far from real values $a = 1.355$ and $b = 3.2 * 10^{-5}$, for better estimation we can run the code for more steps and for more systems so our estimation become much closer to these values.

for showing phase transition i scale velocities at each step, the rest of the process is the same as before, you can see the animation in jupyter file.the results are as follows:

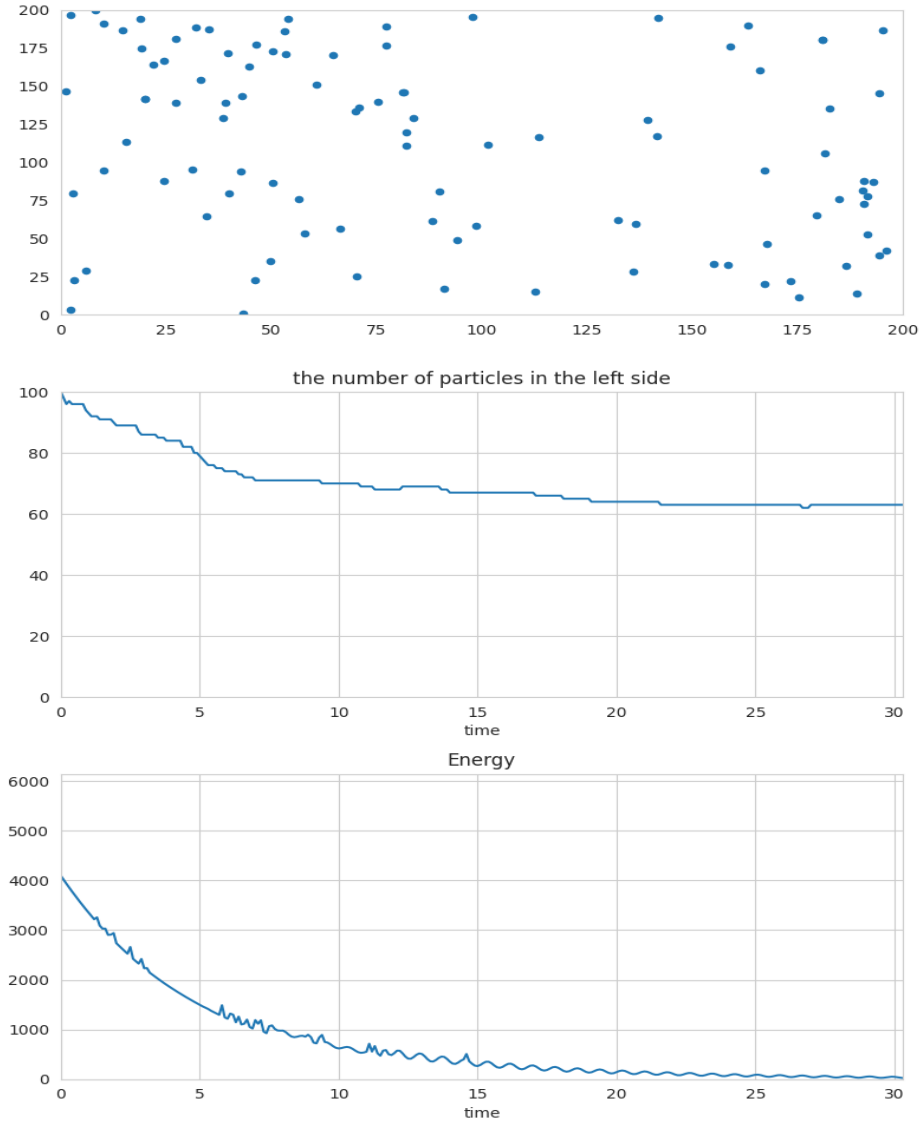

Figure 7: System's Dynamic when changing tempreture.