

Attached written notes are my pseudo code for the problems.

MD

[ same code as the example J ]

def get\_accelerations(x, y):

ax = np.zeros((x.size, x.size))

ay = np.zeros((x.size, x.size))

for i in range(0, x.size - 1):

    for j in range(i + 1, y.size):

        rx = []

        ry = []

$$rmag = \sqrt{r_x^2 + r_y^2}$$

force\_x = []

force\_y = []

We add an extra parameter for y dimension.

ax[i, j] = []

ax[j, i] = []

ay[i, j] = []

ay[j, i] = []

return  $\sum ax, \sum ay$

...

def run\_md(dt, # of steps, T, snaptime, x, y, # of atoms):

Vx = np.zeros((# of steps, # of atoms))

Vy = []

x = []

y = []

Vx\_0 = initial\_velocity()

Vy\_0 = []

ax, ay = get\_acceleration(x, y)

We add extra arrays and # of atoms to expand it to 2D and give control over number of atoms.

for i in range(# of steps):

x = []

y = []

ax = get\_accel()

ay = get\_accel()

Vx = update\_velo()

Vy = update\_velo()

positionsx [i, :] = x

positionsy [i, :] = y

velocityx [i, :] = vx

velocityy [i, :] = vy

return positionsx, positionsy, velocityx, velocityy

plt.plot( ).

Animation using lab computer.

/packages/vmd/bin/vmd

# PHYS 304 AS10

Xiyue Shen\*  
*Haverford College Department of Physics*  
(Dated: April 21, 2024)

Yang and Petra helped me to figure out some problems.  
I have two python files.

*PHYS\_304\_AS2\_MD\_Xiyue\_Shen* shows my molecular dynamics system for position and velocity.

*PHYS\_304\_AS2\_MDCheck\_Xiyue\_Shen* shows my check for the code working in one dimension by setting y dimension equal to zero.

## 1. EXERCISE

### 1.1. Molecular Dynamics

Starting from Newton's second law, we have,

$$F = m \frac{d^2x}{dt^2} \quad (1)$$

Under a molecular dynamic system, all particles move and experience forces from neighboring atoms. Their motion obeys Newton's second law. However, given the large number of atoms, it's impossible to get analytical solutions for all atoms.

In this problem, we use the Lennard-Jones Potential, which is described by,

$$V_{ij}(r_{ij}) = 4\epsilon_{ij}\left(\left(\frac{\sigma_{ij}}{r_{ij}}\right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}}\right)^6\right), \quad (2)$$

where  $i$  and  $j$  represents two neighboring atoms,  $\epsilon$  is the depth of the potential well, and  $\sigma$  is the distance where the potential  $V$  is zero. In the example, we have  $\epsilon = 0.0103$  and  $\sigma = 3.4$ .

Given the interaction potential and Newton's second law, we can modify the given example to generate our 2 dimensional molecular dynamics system.

For this, we aim to change the acceleration functions by adding another degree of freedom, making the forces and acceleration two-dimensional. The main thing I changed is in the function "get\_acceleration" by adding one input "position". Then for each array in x dimension, add another line for y. When calculating the force, we have a parameter "rmag" originally as "np.sqrt(r\_x\*rx)". Since we have two dimensions, the r is the two-dimensional distance between particles as shown in equation 2, which should be,

$$r = \sqrt{r_x^2 + r_y^2} \quad (3)$$

At end of "get\_acceleration", we will have two outputs as np.sum(accel\_x) for x dimension, and np.sum(accel\_y) for y dimension.

Then, the second main thing to modify is "run\_md". We add "snaptimes" to save data, and "number\_of\_atoms" in the input. Similarly, we add an extra array below every x array for y dimension. Then, set a loop to make a plot for positions along x and y and velocity along x and y.

The result is attached below. The curves shown on the plot are labeled by the specific atom and dimension. Figure 1 shows the positions of each atom with stepsizes equal to 10000.

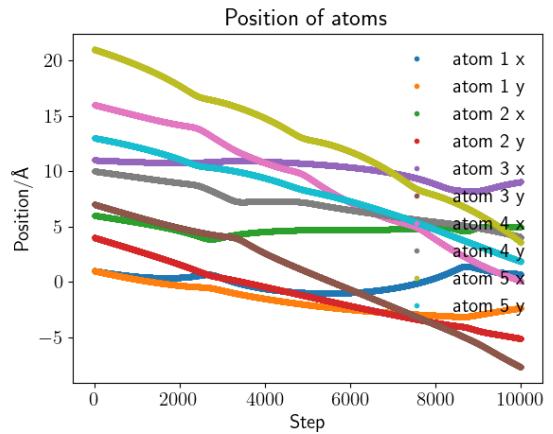


FIG. 1: [The position of atoms with 1000 stepsizes

Figure 2 shows the velocity of each atom with stepsizes equal to 10000.

One thing worth checking is to set the velocity and position along the y dimension as zero and use a non-random distribution for x. We should get the same thing every time we run. Figure 3 and figure 4 show the positions and velocities we run for twice. They look exactly what we want.

For the last part, we apply the ".xyz" to VMD to make an animation. The main command is "/packages/vmd/bin/vmd." I took 8 screenshots to show the movement. There are 50 frames in total. Figure 5 shows the 0th, 7th, 14th, and 21st frames, and figure 6 shows the 28th, 35th, 42nd, and 49th frames.

\*Electronic address: [xshen2@brynmawr.edu](mailto:xshen2@brynmawr.edu);  
URL: [Optional homepage](#)

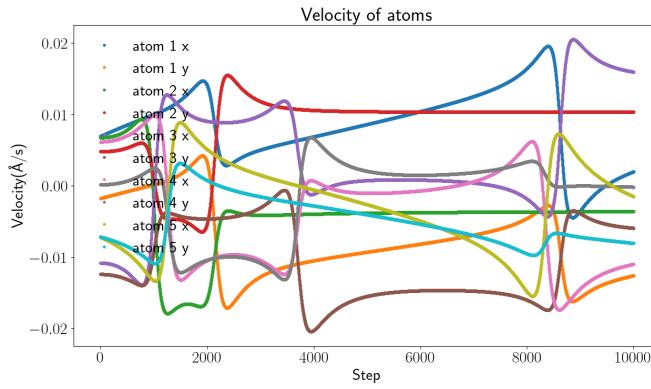


FIG. 2: [The velocity of atoms with 1000 stepsize

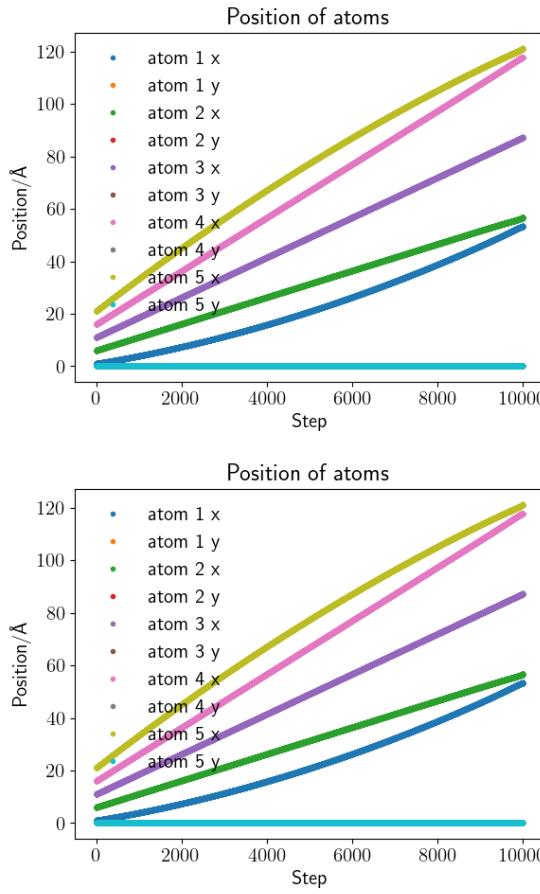


FIG. 3: Set initial position and velocity along y dimension equal to 0. We get the same x trajectory for x dimension position for every run.

## 2. SURVEY

I spent roughly 6 hours on this homework. I'm now more familiar with molecular dynamics problems and how to expand them to higher dimensions. I like the final animation the most. I practiced my troubleshooting

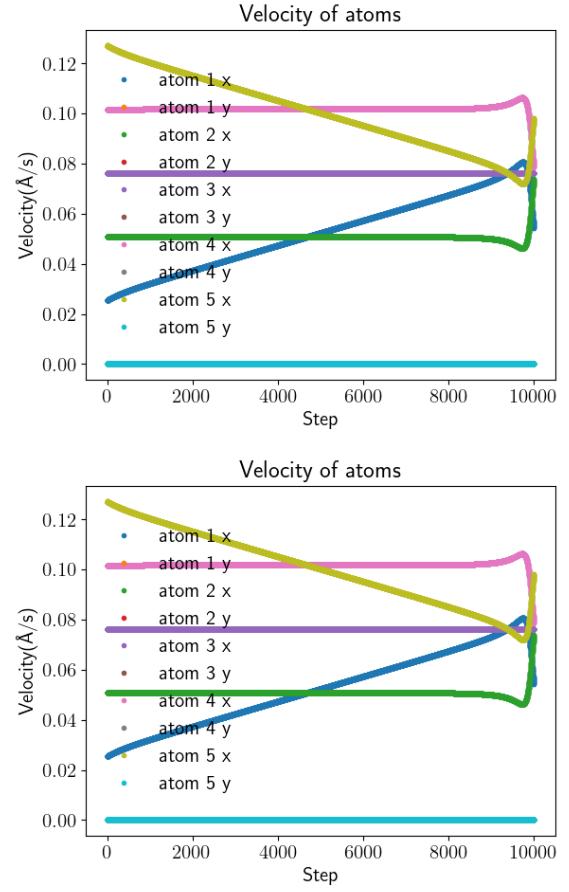


FIG. 4: Set initial position and velocity along y dimension equal to 0. We get the same x velocity for x dimension position for every run.

skills. I think this set is about the right length.

## 3. UNGRADED PART

I have done all of the required and ungraded work.

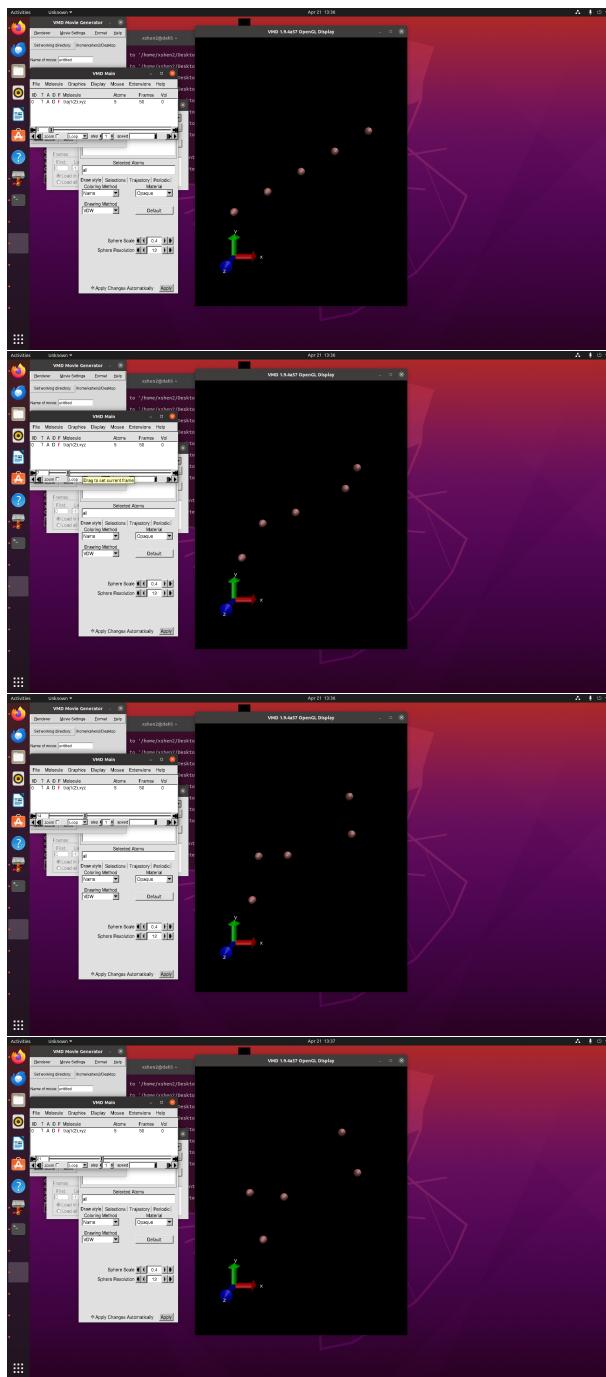


FIG. 5: A series of screenshots showing the particle moving in sequence. It shows the 0th, 7th, 14th, and 21st frame

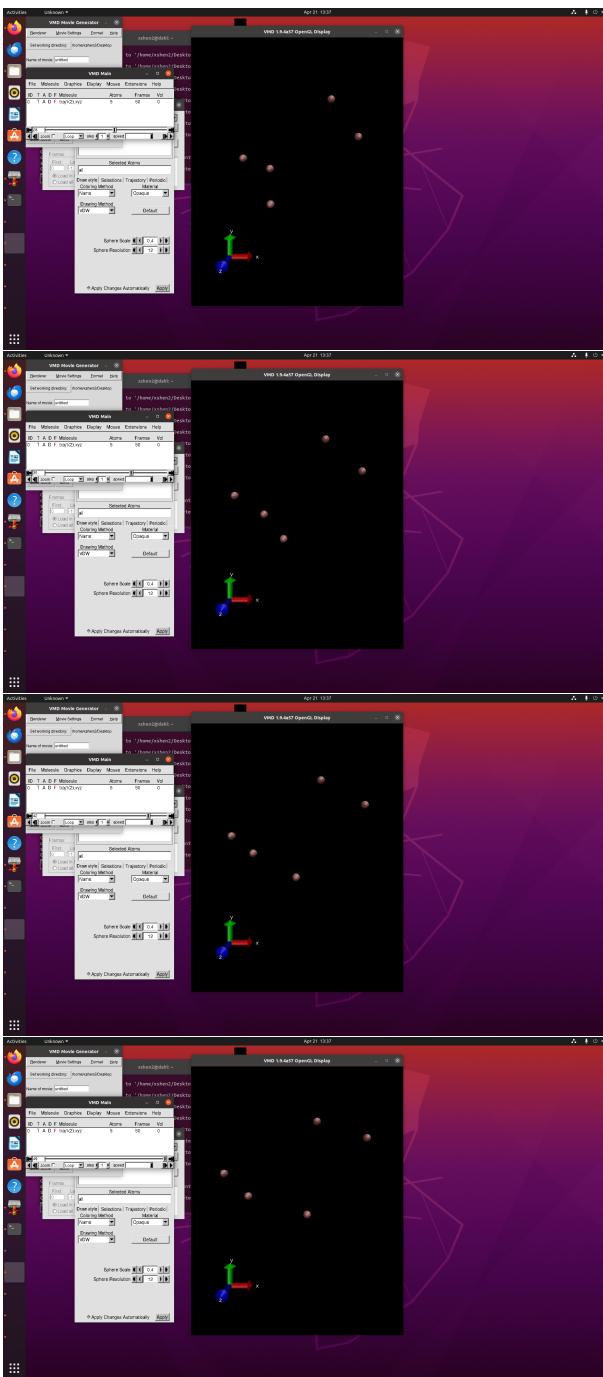


FIG. 6: A series of screenshots showing the particle moving in sequence. It shows the 28th, 35th, 42nd, and 49th frame