56·5·61

61/61

# HW9 Write-up

Nina Martinez Diers*
*Bryn Mawr College Department of Physics*
(Dated: April 23, 2024)

An algorithm produced a simulation of the trajectories of 10 particles subject to the Lennard-Jones potential.

## 1. INTRODUCTION

In this assignment, we simulated the motion of 10 identical particles subject only to the Lennard-Jones force exerted by neighboring particles. The Lennard-Jones force exerted on a particle by another particle is given by

$$\vec{F} = 24\frac{\epsilon}{r}\left[2\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right]\frac{\vec{r}}{r} \tag{1}$$

where $\epsilon$ is the potential energy at the equilibrium bond length, $\sigma$ is the distance at which the potential energy is zero, $\vec{r}$ is the displacement vector between particles, and $r$ is the distance between particles. The value used for $\epsilon$ was 0.0103eV and the value used for $\sigma$ was 3.4Å. The distance between two particles was calculated using the equation

$$r = \sqrt{r_x^2 + r_y^2} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \tag{2}$$

where $i$ and $j$ denote the particles and $x$ and $y$ denote the components of the position vectors of the particles. Using the Lennard-Jones potential allows for the simulation of collisions because the potential increases drastically when particles get too close to each other, producing a repulsive force.[1] However, it also accounts for the attractive effects of Van der Waals interactions of induced dipole moments of particles.[1] These characteristics make studying the effects of systems subject only to a Lennard-Jones potential interesting.

## 2. EXPERIMENT

As particles move in a system, the Lennard-Jones force exerted on each particle by all the other particles in the system changes, influencing the evolution of the system in time. Two-dimensional kinematics equations were used to calculate the trajectories of the particles in two dimensions. Figure 1 shows the plan for adding motion in a second dimension to molecular dynamics simulation.

In this experiment, 10 particles were created. They were given identical initial x- and y- coordinates of that

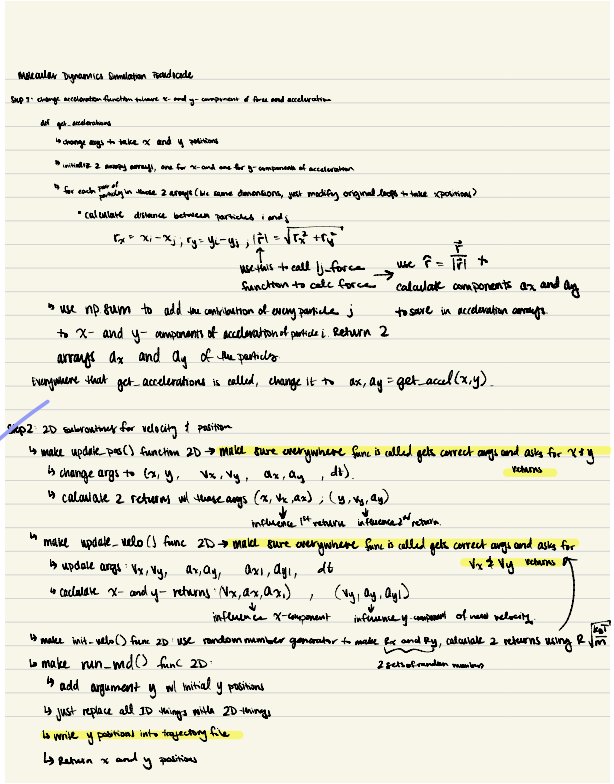


FIG. 1: Pseudocode to convert the algorithm for a molecular dynamics simulation in one-dimension into a simulation in two-dimensions.

were evenly spaced. Initial velocities were assigned according to

$$v = n\sqrt{\frac{k_B T}{m}} \tag{3}$$

with Boltzmann constant $k_B$, temperature of the system $T$ and mass $m$. **Rand**, a random number generator in the **Numpy** package, produced an array of 10 random numbers $n$ that were scaled according to Equation 3 to produce a system at the desired temperature of 300K. The mass of argon was used as the mass $m$ of the simulation particles. This process was done twice to produce arrays with the initial component of the velocities in the x- and y-dimension.

*Electronic address: nmartinezd@brynmawr.edu

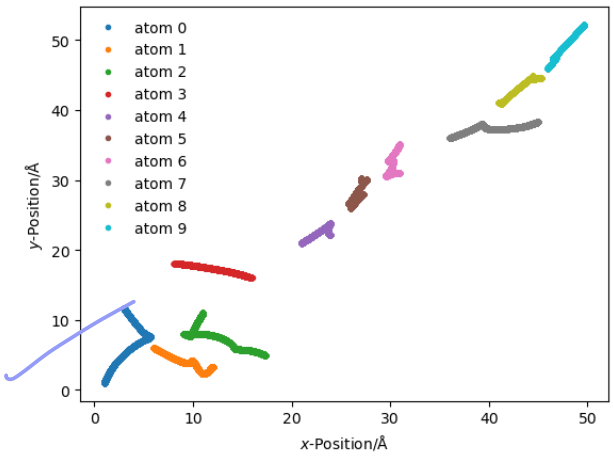


FIG. 2: Trajectories of simulation particles in two dimensions at 300K.

The accelerations of the particle were found using the Lennard-Jones force (Eq. 1 on each particle with Equation 6 at each time-step, which are then used to update the velocity and position vectors:

$$\vec{v} = \vec{v_0} + (\vec{a_0} + \vec{a})\, dt \tag{4}$$

$$\vec{x} = \vec{x_0} + \vec{v_0}dt + 0.5\vec{a_0}dt^2 \tag{5}$$

$\vec{a_0}$ is the acceleration of the previous time-step and $\vec{a}$ is the acceleration of the current time-step.

$$\vec{a} = \frac{\vec{F}}{m} \tag{6}$$

The simulation ran for 10000 time-steps of 0.1 seconds, and the positions of the particles were saved every 100 time-steps to generate the frames of the video simulation. Visual Molecular Dynamics (VMD)[2] was used to generate an animation of the particles trajectories in time. In this video, collisions of the particles could observed as the two-dimensional positions of particles naturally evolved in time.

## 3. RESULTS

The resulting simulation produced trajectories (Fig. 2) that did not overlap. This is important because it means that the Lennard-Jones forces did not allow particles to occupy the same space, which would be an unphysical representation. Jagged edges in trajectories indicate the collisions are taking place.

## 4. CONCLUSIONS

A simulation was produced that showed two-dimensional motion of 10 particles.

[1] C. Kittel, *Introduction to Solid State Physics* (2005), eighth ed., ISBN 0-4-41526-X.
[2] W. Humphrey, A. Dalke, and K. Schulten, Journal of Molecular Graphics **14**, 33 (1996).

### Appendix A: Statement of Collaborators

I collaborated with Melanie on this assignment.

### Appendix B: Survey Question

I spent approximately 4 hours coding for this assignment, 2 hours visualizing using VMD and making the video, and 5 hours on the write-up. I most enjoyed making the visualization, but it was frustrating to try to figure out a way to save the visualization that I had made because the Make Movie feature required an external download that I was struggling to find. Because of this I ended up not using VMD to do so, and instead I took screenshots of the frames and compiled them in iMovie. I thought writing the code for this project was pretty simple but helped me to engage with the theory a little bit more, and the most valuable thing I learned from this assignment was probably gaining some familiarity with VMD. The length of this assignment was just right for me.

56/56

# Computational Physics/Astrophysics, Winter 2024:

# Grading Rubrics [1]

## Haverford College, Prof. Daniel Grin

For coding assignments, roughly 56 points will be available per problem. Partial credit available on all non-1 items.

4 1. Does the program complete without crashing in a reasonable time frame? (+4 points)

2 2. Does the program use the exact program files given (if given), and produce an answer in the specified format? (+2 points)

3 3. Does the code follow the problem specifications (i.e numerical method; output requested etc.) (+3 points)

5 4. Is the algorithm appropriate for the problem? If a specific algorithm was requested in the prompt, was it used? (+5 points)

4 5. If relevant, were proper parameters/choices made for a numerically converged answer? (+4 points)

4 6. Is the output answer correct? (+4 points).

3 7. Is the code readable? (+3 points)

. 5.1. Are variables named reasonably?

. 5.2. Are the user-functions and imports used?

[1] Inspired by rubric of D. Narayanan, U. Florida, and C. Cooksey, U. Hawaii

. 5.3. Are units explained (if necessary)?

. 5.4. Are algorithms found on the internet/book/etc. properly attributed?

3 8. Is the code well documented? (+3 points )

. 6.1. Is the code author named?

. 6.2. Are the functions described and ambiguous variables defined?

. 6.3. Is the code functionality (i.e. can I run it easily enough?) documented?

9. Write-up (up to 28 points)

5 . Is the problem-solving approach clearly indicated through a flow-chart, pseudo-code, or other appropriate schematic? (+5 points)

✓ . Is a clear, legible LaTeX type-set write up handed in?

3 . Are key figures and numbers from the problem given? (+ 3 points)

4 . Do figures and or tables have captions/legends/units clearly indicated. (+ 4 points)

3 . Do figures have a sufficient number of points to infer the claimed/desired trends? (+ 3 points)

2 . Is a brief explanation of physical context given? (+2 points)

1 . If relevant, are helpful analytic scalings or known solutions given? (+1 point)

3 . Is the algorithm used explicitly stated and justified? (+3 points)

2 . When relevant, are numerical errors/convergence justified/shown/explained? (+2 points)

2. Are 3-4 key equations listed (preferably the ones solved in the programming assignment) and algorithms named? (+2 points)
1. Are collaborators clearly acknowledged? (+1 point)
2. Are any outside references appropriately cited? (+2 point)