

PHY 905 Project 4: Molecular Dynamics

Thomas Redpath¹

¹*Department of Physics, Michigan State University*

Molecular dynamics simulations provide a means of studying many-body systems through, in this case, numerically solving Newton's equations of motion. We ran a simulation of 500 argon atoms interacting via a Lennard-Jones two-body potential and sampled the system's energy, temperature and diffusion constant. By comparing the behavior of the diffusion constant over time for various equilibrium temperatures, we found the melting point for this system to be roughly 287.5 K.

INTRODUCTION

Molecular dynamics simulations attempt to capture the dynamical evolution of a many body system. These simulations are applied in a wide range of disciplines from studying the internal dynamics of polymer chains [1] to understanding the structure and function of biological macromolecules [3]. The rise of high-performance computing has benefited molecular dynamics simulations through allowing for enhanced methods of calculating the interatomic forces [4]. The increasing scope and complexity of molecular dynamics simulations can provide further insights into the many problems to which this method can be applied.

In this project, we simulate a simple system of argon atoms interacting via a pairwise Lennard-Jones potential [5, 7]. By calculating the diffusion constant, we can estimate the melting point for this system. In the next section, we discuss the statistical thermodynamics background of our project. Thereafter, we describe the simulation code we used and the results we obtained. Finally, we will briefly discuss avenues for further development of the tools developed for this project.

THEORY

Statistical thermodynamics provides the theoretical basis for the type of classical molecular dynamics simulation we employed in this project. The code we developed solves Newton's equations of motion numerically to evolve the system along a deterministic trajectory in its phase space. We can then compute various macroscopic quantities from the system's microscopic properties.

Lennard-Jones Potential

We model the interaction between argon atoms with a Lennard-Jones potential - plotted in FIG. 1. This potential is often used to model interactions between noble gas atoms. It is parameterized by the strength ϵ and range of the interaction σ . Values of $\epsilon/k_B = 119.8$ K and $\sigma = 5.26$ Å are optimal values for argon.

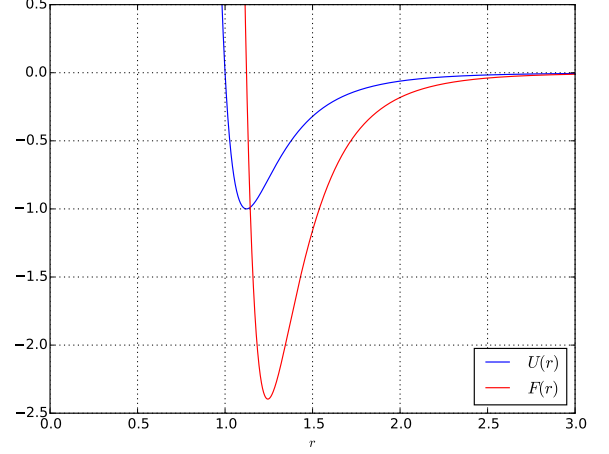


FIG. 1: The Lennard-Jones potential (blue) and derived force (red).

$$U(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right], \quad (1)$$

Thermodynamic quantities

Since our system contains a fixed number of particles, it is contained in a fixed volume and the total energy remains constant, we identify it as a microcanonical ensemble. We can calculate the average energy of the system using the equipartition theorem. Since we are dealing with monoatomic argon, there are only three translational degrees of freedom, so the average energy is given by

$$\langle E_k \rangle = \frac{3}{2} N k_B T$$

Re-writing this relation allows us to compute an estimate of the system's temperature

$$T = \frac{2}{3} \frac{E_k}{N_{\text{atoms}} k_B}. \quad (2)$$

The diffusion constant is another observable that we can extract from our simulation. The Einstein relation between the mean square displacement and the diffusion constant (D) allows us to calculate D from the atoms' average displacements at each time step:

$$D = \frac{\langle r^2(t) \rangle}{6t} \quad (3)$$

where

$$\begin{aligned} \langle r^2(t) \rangle &= \frac{1}{N} \sum_i^N r_i^2(t) \\ &= \frac{1}{N} \sum_i^N |\mathbf{r}_i(t) - \mathbf{r}_i(0)|^2 \end{aligned}$$

The average of all the atomic displacements gives a measure of how mobile the atoms are. For example, in a solid the atoms won't move far from their lattice sites so $\langle r^2(t) \rangle$ will be small. This value will be larger for liquids and larger still for gases. Therefore, we can interpret the diffusion constant as a comparative measure of the state of the system.

Units

The small scale of the system we are simulating lends itself to working in a specialized set of units. We used the following four relations to define our system of units

$$\begin{aligned} 1 \text{ unit of mass} &= 1 \text{ a.m.u} = 1.661 \times 10^{-27} \text{ kg}, \\ 1 \text{ unit of length} &= 1.0 \text{ \AA} = 1.0 \times 10^{-10} \text{ m}, \\ 1 \text{ unit of energy} &= 1.651 \times 10^{-21} \text{ J}, \\ 1 \text{ unit of temperature} &= 119.735 \text{ K}. \end{aligned}$$

Converting to this system of units, Boltzmann's constant becomes $k = (1.38 \times 10^{-23})(119.7)(1.65 \times 10^{-21}) = 1$.

ALGORITHMS AND METHODS

The C++ code developed for this project utilizes the velocity Verlet method to solve Newton's equations of motion. In this section we briefly summarize the derivation of this method discussed in Hjorth-Jensen [2]. We then provide a description of the codes which can be found at https://github.com/redpath11/phy905_thr in the `projects/project4/src` directory.

Verlet Method

We numerically solve the equations of motion using the velocity Verlet method. Since we've assumed a Lennard-Jones type pairwise interaction between atoms, the form

of the acceleration is known from the gradient of the interaction potential. The Verlet method derives from a summation of two Taylor expansions. Consider the position in one dimension

$$\begin{aligned} x(t+h) &= x(t) + hx'(t) + \frac{1}{2}h^2x''(t) + O(h^3) \\ x(t-h) &= x(t) - hx'(t) + \frac{1}{2}h^2x''(t) + O(h^3). \end{aligned}$$

Summing these two expansions and switching to the discretized notation gives

$$x_{i+1} = 2x_i - x_{i-1} + h^2x''_i + O(h^4).$$

In general, this algorithm is not self-starting since it requires the value of x at two previous points.

Now, consider a Taylor expansion of the velocity and note the relationship between velocity and acceleration $v' = a$.

$$\begin{aligned} v_{i+t} &= v_i + hv'_i + \frac{1}{2}h^2v''_i + O(h^3) \\ a_{i+1} &= v'_i + hv''_i + O(h^2). \end{aligned}$$

We will assume that the velocity is a linear function of time $hv''_i \approx v'_{i+1} - v'_i$. We can now write the final algorithmic form for the position and velocities.

$$x_{i+1} = x_i + hv_i + \frac{h^2}{2}v'_i + O(h^3) \quad (4)$$

$$v_{i+1} = v_i + \frac{h}{2}(v'_{i+1} + v'_i) + O(h^2) \quad (5)$$

Code

Overview

We were supplied a code skeleton (from <https://github.com/andepiane/molecular-dynamics-fys3150>) outlining the major classes needed to run a molecular dynamics simulation. From this starting point, we filled in details to apply periodic boundary conditions, initialize the system of argon atoms in face-centered cubic (FCC) lattice, calculate the pairwise forces from the Lennard-Jones potential and compute physical properties of the system. Figure 2 gives a general overview of the code structure by identifying key elements and where they reside within the class structure.

Initialization

Initialization of our system takes place in two general steps: (1) setting the size, initial temperature and

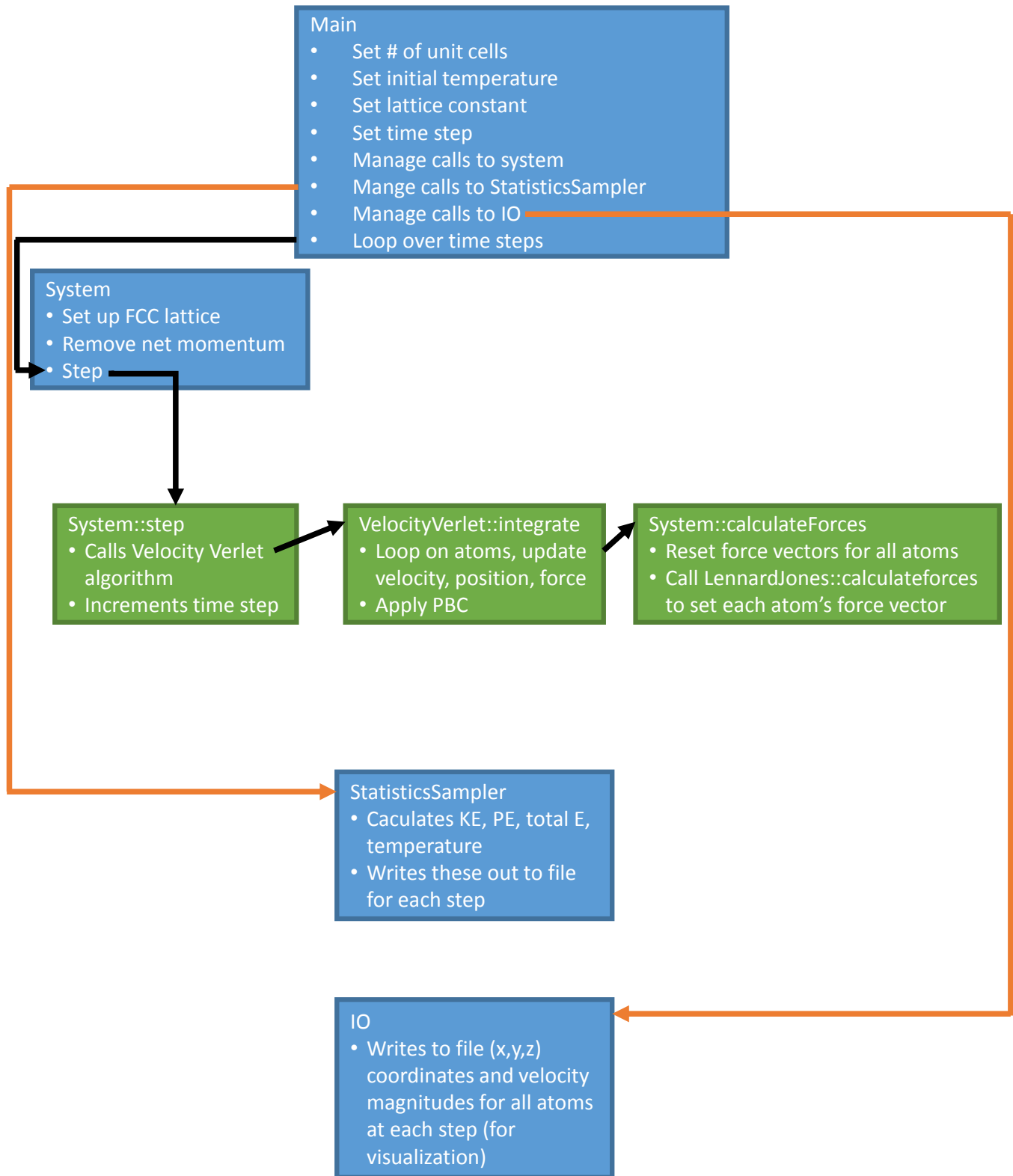


FIG. 2: The flowchart gives a crude overview of how the code operates. Blue boxes represent classes and indicate processes carried out by one or more member functions. Green boxes represent functions crucial to the simulation.

Black arrows indicate the hierarchy of function calls for one timestep. Orange arrows depict function calls that result in writing data to a file.

system to a finite volume. Each time the positions are updated, we run the following routine to ensure that all atoms are contained inside the specified volume. We also need to adjust the initial position vector accordingly in order to properly compute the diffusion constant. Finally, to improve the efficiency of the code, we exploit the $1/r^6$ behavior of the potential and set the contribution to the net force to zero when atoms are more than 2.5 Å apart. To compensate for this, we subtract the constant $V_{LJ}(r = 2.5)$ from the total potential.

```

1  for (Atom *atom : m_atoms)
2  {
3      for (int j=0; j<3; j++)
4      {
5          if (atom->position[j]<0.) {
6              atom->position[j] += m_systemSize[j];
7              atom->initialPosition[j] += m_systemSize[j];
8          }
9          if (atom->position[j]>=m_systemSize[j]) {
10             atom->position[j] -= m_systemSize[j];
11             atom->initialPosition[j] -= m_systemSize[j];
12         }
13     }
14 }

```

Once the positions are updated at each time step, we sample several macroscopic properties of the system. This is handled by the `StatisticsSampler` class. These functions loop over all atoms to calculate the total kinetic energy, the temperature and the diffusion constant for the system. This class also retrieves the potential energy that is calculated when the forces are updated. These quantities are written out to a file. Finally, the positions and velocity magnitudes at each time step for all atoms are also written to a `.xyz` file that can be read into Ovito to produce a movie of the simulation.

RESULTS AND DISCUSSION

Energy Conservation

Our first test of the simulation was to ensure that energy conservation is obeyed. This test revealed a major bug in the way we were applying the periodic boundary conditions. When this is not done correctly, the potential energy and forces are incorrectly calculated because the distances between atoms is wrong. Once this issue was resolved, we obtained the results shown in Table I for a simulation of 1000 time steps and an initial temperature of 300 K.

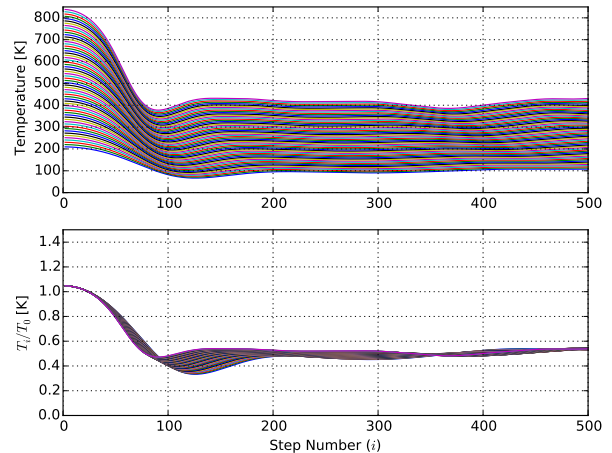


FIG. 4: The top panel plots calculated system temperature (eq. 2) vs. time step number for starting temperatures T_0 ranging from 200 to 800 K (the different lines). The bottom panel presents the same data as the top panel but with the temperature at each step normalized by the T_0 for that simulation. We varied the initial temperature between 200 K and 800 K and used a time step value $dt = 10^{-15}$ s. For all the simulations, the equilibrium temperature is roughly half the initial temperature.

Equilibrium Temperature

In a preceding section about initializing the simulation, we mentioned that the “initial” temperature we set is not the actual temperature of the system but rather a way to set the total energy of the system. When we randomly assign velocities to each atom, we are starting the system in a random one of its virtually uncountable microstates. Through simulating the interactions between atoms, we evolve the system through other microstates at each time step where, according to the ergodic hypothesis, the probability of reaching a given microstate is given by the probability distribution governing the ensemble. Now, the initial state that we populate when we construct the system isn’t likely to be an equilibrium state for the system, but as we run the simulation, the system evolves to reach a region of the phase space corresponding to equilibrium microstates. This process is represented by the peak at 0 and subsequent fall off in FIG. 4. Once the system reaches equilibrium (around step 175), the temperature fluctuates randomly around its equilibrium value (T_{eq}). We found that the equilibrium temperature is roughly half the initial temperature.

TABLE I: Results from a simulation with the initial temperature $T_0 = 300$ K and a time step size $dt = 10^{-14}$ s. Values in the time column are in MD units (1.00224×10^{-13} s).

Timestep	Time	Temperature [K]	KineticEnergy [eV]	PotentialEnergy [eV]	TotalEnergy [eV]
1	0.099777	312.026	20.1662	-38.3532	-18.187
101	10.0774	158.627	10.252	-28.4274	-18.1754
201	20.0551	163.31	10.5547	-28.7268	-18.1721
301	30.0327	171.405	11.0779	-29.252	-18.1741
401	40.0104	160.382	10.3655	-28.5349	-18.1694
501	49.988	169.434	10.9505	-29.1232	-18.1726
601	59.9657	159.62	10.3162	-28.4912	-18.175
701	69.9434	164.443	10.6279	-28.8028	-18.1748
801	79.921	167.812	10.8457	-29.0173	-18.1716
901	89.8987	172.008	11.1168	-29.2901	-18.1732

Diffusion Constant

Next, we investigated the melting point of the system by comparing the diffusion constant from simulations with different starting temperatures T_0 (or equivalently $T_{eq} \approx T_0/2$). In FIG. 5, we show the sampled temperature and diffusion constant vs. step number from simulations with T_i ranging from 100 K to 900 K. These simulations used a step size $dt = 10^{-14}$ s and ran for 1000 steps. We looked at the diffusion coefficient as a function of time to gain insight into the state of the system. Recall that a system of atoms in a solid state will be constrained to the vicinity of their lattice sites and therefore the system will have a smaller average displacement when compared to a system in a liquid state. In this case, the atoms move more freely so the system will exhibit a larger average displacement. Since the diffusion constant (D) is proportional to the average displacement (eq. 3), it provides an indication of the state of the system.

In the bottom panel of FIG. 5, we see that D , towards the end of the simulation, separates the simulations into two groups. The lower band, indicates T_0 values for which D remains constant over the course of the simulation. In the upper group, D gradually rises over the course of the simulation. For the largest T_0 simulations, D reaches a steady state before the end of the simulation. We interpret this increase in D as a phase transition where the system changes from a solid to a liquid (it melts). Furthermore, the slope indicates how quickly the process happens - for systems initialized with a large amount of kinetic energy (higher T_0), the transition occurs faster than for systems initialized with a lower T_0 . Closer inspection of FIG. 5 reveals a small group of systems with a higher final D than the non melting cases. These systems are likely very close to the melting point so the phase transition takes a long time or they only partially melt.

The information in FIG. 5 can be represented more succinctly by noting that the temperature (T) and D are relatively constant over the last 200 time steps of all the simulations. Therefore, we plot D vs. T both

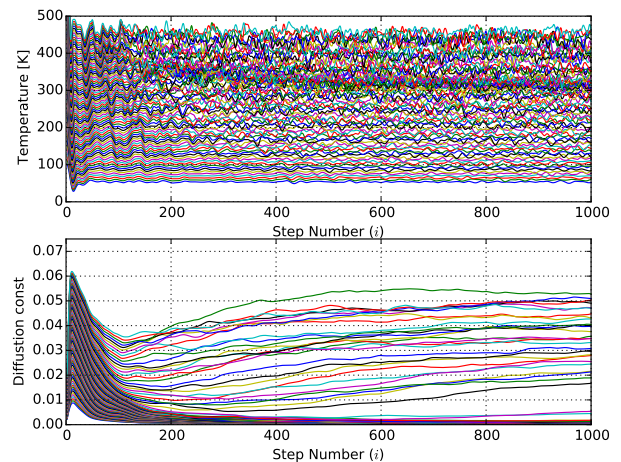


FIG. 5: In the top panel, we show the temperature vs. time step number. In the bottom panel, we plot, as a function of step number, the diffusion constant in units of $\text{\AA}^2/s_{\text{MD}}$ where s_{MD} is the unit of time used in the simulations. The different lines correspond to simulations with different initial temperatures ranging from 100 K to 900 K in steps of 10 K. Each simulation used 1000 steps and a step size of 10^{-14} s.

averaged over the last 200 time steps in FIG. 6. The top panel of FIG. 6 shows that for all the simulations, D remains roughly constant over the last 200 time steps. The bottom panel shows a discontinuity in the diffusion constant as a function of temperature suggesting that the melting point occurs just above 300 K.

To ensure that the simulation time wasn't too short to capture the melting process for systems initialized close to the melting point, we ran two batches of simulations with 10000 time steps each. We simulated systems with T_{eq} ranging from 280 K to 292.5 K and 195 K to 205 K. The D vs. time step index from these simulations are shown in FIG. 7 and FIG. 8, respectively and show the melting point to lie between 285 and 287.5 K.

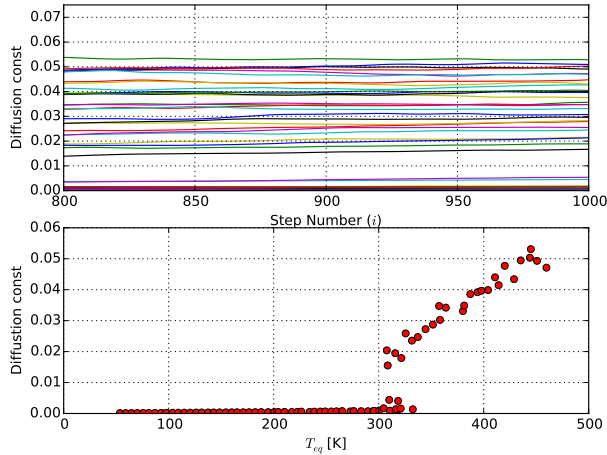


FIG. 6: In the top panel, we plot the diffusion constant in units of $\text{\AA}^2/s_{\text{MD}}$ vs. the time step index. The bottom panel shows the diffusion constant vs. the temperature where both observables have been averaged over the last 200 time steps.

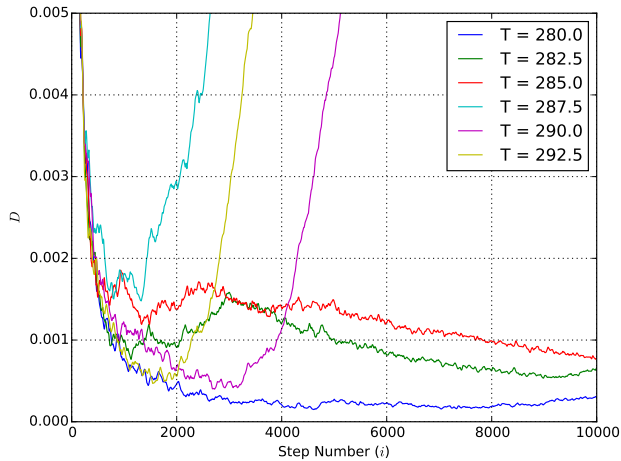


FIG. 7: Diffusion constant vs. time step for systems with 5 different T_{eq} ranging from 280 K to 292.5 K in steps of 2.5 K. Each simulation was run with 10000 time steps.

CONCLUSIONS

We have developed a simple molecular dynamics simulation for argon atoms interacting via a simple two-body force. We extracted the system energy, temperature and diffusion constant and used these to determine the melting point as roughly 287.5 K. In order to compare this result to the actual argon melting point, we need to calculate the pressure of our simulated system. This is a

logical next step that will involve additions to the sim-

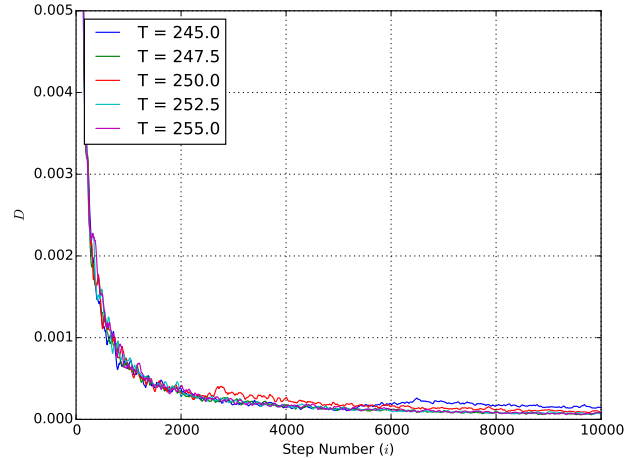


FIG. 8: Diffusion constant vs. time step for systems with 5 different T_{eq} ranging from 245 K to 255 K in steps of 2.5 K. Each simulation was run with 10000 time steps.

ulation package. Another avenue for further study, is to vary the size of the system and explore how this impacts the random fluctuations of the macroscopic observables.

-
- [1] K. Binder, J. Horbach, W. Kob, W. Paul, and F. Varnik. Molecular dynamics simulations. *Journal of Physics Condensed Matter*, 16:S429–S453, February 2004. doi: 10.1088/0953-8984/16/5/006.
 - [2] M. Hjorth-Jensen. Computational physics lecture notes fall 2015. url = <http://github.com/CompPhysics/ComputationalPhysicsMSU>, Aug 2015.
 - [3] Martin Karplus. Molecular dynamics simulations of biomolecules. *Accounts of Chemical Research*, 35(6):321–323, 2002. doi:10.1021/ar020082r. URL <http://dx.doi.org/10.1021/ar020082r>. PMID: 12069615.
 - [4] T. D. Kühne. Ab-Initio Molecular Dynamics. *ArXiv e-prints*, January 2012.
 - [5] A. Rahman. Correlations in the motion of atoms in liquid argon. *Phys. Rev.*, 136:A405–A411, Oct 1964. doi:10.1103/PhysRev.136.A405. URL <https://link.aps.org/doi/10.1103/PhysRev.136.A405>.
 - [6] Alexander Stukowski. Visualization and analysis of atomistic simulation data with ovito the open visualization tool. *Modelling and Simulation in Materials Science and Engineering*, 18(1):015012, 2010. URL <http://stacks.iop.org/0965-0393/18/i=1/a=015012>.
 - [7] Loup Verlet. Computer “experiments” on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Phys. Rev.*, 159:98–103, Jul 1967. doi: 10.1103/PhysRev.159.98. URL <https://link.aps.org/doi/10.1103/PhysRev.159.98>.