

# FYS4460 Project 1: Introductory Molecular Dynamics Modeling

Linus Ekstrøm

February 2021

## **Abstract**

In this project we tackle the concepts of molecular dynamics simulation by doing a case study of both Argon and Silicon systems using LAMMPS for simulation and OVITO for visualization.

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Theory</b>	<b>3</b>
2.1	Atomic Modeling . . . . .	3
2.2	Interatomic Potentials . . . . .	3
2.3	Lennard-Jones Potential . . . . .	3
2.3.1	Dimensionless Lennard-Jones Potential . . . . .	3
2.4	Velocity-Verlet Integration . . . . .	4
2.5	Equipartition Principle . . . . .	4
2.6	The Virial Equation . . . . .	4
2.7	Diffusion . . . . .	4
2.8	Thermostats . . . . .	5
2.8.1	Berendsen . . . . .	5
2.8.2	Anderson . . . . .	5
2.8.3	Nosé Thermostat . . . . .	5
2.8.4	Nosé-Hoover Thermostat . . . . .	5
2.9	Three Particle Potentials . . . . .	6
<b>3</b>	<b>Method</b>	<b>6</b>
3.1	Studying the time evolution of particle velocities . . . . .	6
3.2	Variation in Total Energy . . . . .	6
3.3	Temperature as a Function of Time . . . . .	7
3.4	Measuring the Pressure of the System . . . . .	7
3.5	Thermostats . . . . .	7
3.6	Mean Square Displacement . . . . .	7
3.7	Silicon System . . . . .	7
<b>4</b>	<b>Results</b>	<b>8</b>
4.1	Velocity Distribution . . . . .	8
4.2	Variation in Total Energy . . . . .	9
4.3	Temperature Comparison . . . . .	9
4.4	Pressure Comparison . . . . .	10
4.5	Nosé-Hoover and Berendsen . . . . .	10
4.6	Mean Square Displacement . . . . .	11
4.7	Radial Distribution Function . . . . .	13
4.8	Silicon Simulation . . . . .	13
<b>5</b>	<b>Discussion and Conclusion</b>	<b>15</b>

# 1 Introduction

In this project we will consider the motion of collections of atoms. We will consider the classical case where Newton's laws of motion suffice.

## 2 Theory

### 2.1 Atomic Modeling

As described in the introduction we will focus on the classical approximation, where we use Newton's laws of motion in addition to a well suited potential to describe the inter-atomic forces in order to simulate the motion of a collection of atoms. In principle, one can, determine the potential energy functions from quantum mechanical principles. However, when dealing with higher temperatures the quantum effects can often be overlooked. Thus, we stick to the classical limit and use simplified models for the interactions that provide the most important features of the interactions.

### 2.2 Interatomic Potentials

The behavior of any system will depend on the potential energy, which depends on every atom's position relative to all the other atoms. In general we model a potential energy as

$$U = U(\mathbf{r}_i)$$

Where we  $U$  denotes the potential energy, and  $\mathbf{r}_i$  denotes the position of particle  $i$ . Of course, this is a simplification because we do not consider the internal states of the particles nor the internal states of their interactions. Both of these will in some cases affect the total energy. If we wish to model this we could use a model such as

$$U = U(\mathbf{r}_i) = \sum_i^j U_{ij}(\mathbf{r}_i, \mathbf{r}_j) + \sum_{ijk} U_{ijk}(\mathbf{r}_i, \mathbf{r}_j, \mathbf{r}_k) + \dots$$

Where we assume that the potential energy can be written as a sum of interactions of higher and higher order. In this project we will consider two types of interactions: *bonded interactions* and *non-bonded interactions*. In the former one the interaction types are prescribed and do not change throughout the simulation, and in the latter we take the view that the potential energy solely depends on the relative positions of all the particles. Non-bonded interactions open for modeling more complicated interactions between atoms, including the formation and breaking of strong bonds between atoms.

### 2.3 Lennard-Jones Potential

One of the simplest models for inter-atomic interactions is the Lennard-Jones potential [2][3]

$$U(r) = 4\epsilon \left( \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^6 \right) \quad (1)$$

Where  $\epsilon$  is the depth of the potential well, commonly referred to as *dispersion energy*, and  $\sigma$  is the distance at which the particle-particle potential energy is zero, often referred to as the *size of the particle*.

It is a representation of the interactions between noble-gas atoms, i.e. the interaction between two Argon atoms. In this interaction we have two main contributions to the potential energy: *dipole-dipole interaction* and *repulsive interaction*. The dipole-dipole interaction, also called the *van der Waals* interaction, is  $\propto (1/r)^6$ . Where  $r$  is the distance between the two atoms. Then there is the repelling force which is a quantum mechanical effect stemming from the *Pauli exclusion principle* [4]. The intuition here is that when atoms are far apart, such that their wavefunction only slightly overlap then they are able to 'have' the same quantum numbers, however when there is significant overlap in the wave functions nature 'does not allow' for fermions to have the same quantum numbers and thus they 'experience' a pressure which repels. We use a power-law of the form  $(1/r)^n$  to represent the interaction. For Argon specifically  $n = 12$  gives a good approximation for the behavior of Argon [1]. The potential has its minimum at a distance of  $r = r_m = 2^{1/6}\sigma$ , where the potential has the value  $U = -\epsilon$ . In addition, the potential has one pole at  $r \rightarrow 0$  i.e. the potential energy diverges, which can cause instabilities in simulations.

The Lennard-Jones potential may be generalized using arbitrary exponents instead of 12 and 6, the resulting potential then is called Mie potential [5]. In this project we will stick to the Lennard-Jones potential. While it is a model for an Argon gas, it is also commonly used as a building block to simulate more complicated systems. However, it is important to remember it is a specific model for a specific gas and to realise when the model might produce unnatural results [3]. Using the Lennard-Jones potential it is possible to simulate systems of  $10^2 - 10^6$  atoms on a common laptop, and as many as  $10^{10} - 10^{11}$  atoms on supercomputers.

#### 2.3.1 Dimensionless Lennard-Jones Potential

As most of the quantities we deal with in molecular simulations have very small absolute values, it is often beneficial to introduce scaling of the units in question to avoid numerical inaccuracies. For the Lennard-Jones we typically use the intrinsic length and energy scales provided by the model:  $\sigma$  and  $\epsilon$ . We thus scale the lengths in the simulation as  $r'_i = r_i/\sigma$ , time as  $\tau = \sigma\sqrt{m/\epsilon}$ , temperature as  $T_0 = \epsilon/k_B$ , where  $k_B$  is the Boltzmann constant. Using these substitutions we can write the *dimensionless* Lennard-Jones potential as

Quantity	Equation	Conversion factor	Specific Argon value
Length	$x' = x/L_0$	$L_0 = \sigma$	$0.3405\mu\text{m}$
Time	$t' = t/\tau$	$\tau = \sigma\sqrt{m/\epsilon}$	$2.1569 \cdot 10^3\text{fs}$
Force	$F' = F/F_0$	$F_0 = \epsilon/\sigma$	$3.0303 \cdot 10^{-1}\text{eV}/\text{\AA}$
Energy	$E' = E/E_0$	$E_0 = \epsilon$	$1.0318 \cdot 10^{-2}\text{eV}$
Temperature	$T' = T/t_0$	$T_0 = \epsilon/k_B$	$119.74\text{K}$

Table 2: Dimensionless scaling for the Lennard-Jones potential. Taken from [1].

## 2.4 Velocity-Verlet Integration

When simulating molecular dynamics the Velocity-Verlet method is commonly used, as it is particularly suited for conservative forces. In opposition to the forward Euler method which has problems with conservation of energy. It is a simple integration algorithm that only uses one sample at the midpoint of the interval in question, opposed to i.e. Runge-Kutta of 4th order which uses four points to approximate higher/lower order slopes. The Velocity-Verlet algorithm is as follows

$$\begin{aligned}\vec{v}_i(t + \frac{1}{2}\Delta t) &= \vec{v}_i(t) + \frac{1}{2}\vec{a}_i(t)\Delta t \quad \text{where} \quad \vec{a}_i(t) = \vec{F}_i/m_i \\ \vec{r}_i(t + \Delta t) &= \vec{r}_i(t) + \vec{v}_i(t + \frac{1}{2}\Delta t) \\ \vec{F}_i(t + \Delta t) &= -\nabla V(\vec{r}_i(t + \Delta t)) \\ \vec{v}_i(t + \Delta t) &= \vec{v}_i(t + \frac{1}{2}\Delta t) + \frac{1}{2}\vec{a}_i(t)\Delta t\end{aligned}$$

This method has good properties when it comes to conservation of energy, and in addition it conserves momentum perfectly [1].

## 2.5 Equipartition Principle

The equipartition principle links the temperature of a system, in thermal equilibrium, to the average kinetic energy of its constituent particles. The idea is that energy is shared equally among its various forms; that is the average kinetic energy per degree of freedom in translational motion of a molecule should equal that in rotation. For the kinetic version of the principle we have

$$\langle E_k \rangle = \frac{3}{2}k_B T \quad (2)$$

where  $k_B$  is the Boltzmann constant, and  $T$  is the temperature after the system has reached equilibrium.

## 2.6 The Virial Equation

In a volume  $V$  with particle density  $\rho = N/V$ , the average pressure is given by

$$P = \rho k_B T + \frac{1}{3V} \sum_{i < j} \vec{F}_{ij} \cdot \vec{r}_{ij}$$

where the sum runs over all particle pairs, this is equivalent to writing  $\sum_i^N \sum_{j=i+1}^N$  and is just a short hand used a lot in physics. The average pressure expression above only applies for the micro-canonical ensemble. [8]

## 2.7 Diffusion

Diffusion is the thermal motion of particles in a fluid or gas. This movement is governed by concentration gradients in the substance in question, and the rate of the movement is proportional to the temperature, viscosity and the mass of the particles. Once the substance has reached concentrational equilibrium the process of molecular diffusion ceases, however another process called *self-diffusion* now takes place. This self-diffusion stems from the random motion of the particles in the substance. Once the self-diffusion stage has been reached we say the system is in *dynamic equilibrium* due to the fact that the particles are still in motion but the distribution of the particles has become uniform. The diffusion equation is a widely studied equation for hundreds of years

$$\frac{\partial \phi(\vec{r}, t)}{\partial t} = \nabla \cdot [D(\phi, \vec{r}) \nabla \phi(\vec{r}, t)]$$

where  $\phi(\vec{r}, t)$  is the density of the diffusing material at location  $\vec{r}$  at time  $t$  and  $D(\phi, \vec{r})$  is the collective diffusion coefficient for density  $\phi$  at location  $\vec{r}$ . If  $D$  is constant then the diffusion equation reduces to the heat equation

$$\frac{\partial \phi(\vec{r}, t)}{\partial t} = D \nabla^2 \phi(\vec{r}, t)$$

For our gas we can find estimate the diffusion constant by considering the self-diffusion of all the particles in our Argon gas. We give each atom a label, and measure its position as a function of time. We can estimate the diffusion constant from the mean square displacement of all the particles

$$\langle r^2(t) \rangle = \frac{1}{N} \sum_{i=1}^N (\vec{r}_i(t) - \vec{r}_{i,initial})^2$$

From theoretical considerations of the diffusion process we can relate the diffusion constant in the liquid to

the mean square displacement through, according to the project text [1]

$$\langle r^2(t) \rangle = 6Dt \quad t \rightarrow \infty$$

## 2.8 Thermostats

In order to simulate canonical ensembles, interactions with an external heat bath must be taken into account. There are plenty of methods for achieving this, chiefly we will concern our discussion on the Berendsen, Andersen and Nose-Hoover thermostats

### 2.8.1 Berendsen

For the Berendsen thermostat we scale all particles velocities with a scalar  $\gamma$  given by

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau} \left( \frac{T_{bath}}{T} - 1 \right)}$$

where  $\tau$  is the so called relaxation time, which couples the system to the external heat bath. The main problem with the Berendsen thermostat is that it does not correctly capture the kinetic energy fluctuations that occur in a realistic canonical ensemble [9].

### 2.8.2 Anderson

Further, the Anderson thermostat simulates *hard* collisions between atoms inside the system and in the heat bath. Atoms which collide will gain a new normally distributed velocity with standard deviation  $\sqrt{k_B T_{bath}/m}$  [1]. In other words, the collisions sample the Maxwell-Boltzmann distribution of the heat bath

$$P(v_i) = \left( \frac{m_i}{2\pi k_B T} \right)^{\frac{1}{2}} e^{-\frac{m_i v_i^2}{2k_B T}}$$

where  $v_i$  represents the  $i$ -th particles individual velocity components. One challenge with using the Anderson thermostats is that the presence of random collisions causes the velocities of the particles to de-correlate, 'lose memory', of their initial value from some previous time [9]. As a result, true molecular kinetics are not preserved using the Anderson thermostat. Even with this result, it can be shown that in the limit over infinitely many heat bath collisions that the Anderson thermostat correctly generates the correct canonical ensemble probabilities. I.E the potential energy distribution, kinetic energy distribution and micro-state probabilities of different configurations of the system all approach their true canonical ensemble values.

### 2.8.3 Nosé Thermostat

Next we will discuss the Nosé-thermostat proposed by Shūichi Nosé in 1984 [10]. We start with the Hamiltonian

$$\begin{aligned} H_{\text{orig}} &= K(\mathbf{p}^N) + U(\mathbf{r}^N) \\ &= \frac{1}{2} \sum m_i |\mathbf{p}_i|^2 + U(\mathbf{r}^N) \end{aligned}$$

Nosé proposed that adding two additional degrees of freedom to the system stemming from the external heat bath:  $s$  the *position* of the heat reservoir the system is coupled to, and  $p_s$  the *conjugate momentum* of the heat reservoir. In addition from these additional degrees of freedom we define  $p_s = Q ds/dt$ , and we refer to this  $Q$  as the *effective mass*. The expanded Hamiltonian with these additions is then

$$H = \frac{1}{2} \sum m_i |\mathbf{p}_i|^2 + U(\mathbf{r}^N) + \frac{p_s^2}{2Q} + k_B T (3N + 1) \ln(s)$$

where the two added terms represent the potential and kinetic energies for the added degrees of freedom to the heat bath. We now scale the particle momenta according to the relation  $\mathbf{p}_i = m_i \mathbf{v}_i \times s$ , and it is through this scaling that the momenta are coupled to the heat bath's constant temperature. Although some of the intricate details have been skipped in the derivation, the additional degrees of freedom were chosen to obtain a specific form for the micro-canonical partition function

$$\begin{aligned} \Omega &\sim \frac{1}{N!} \int \delta[H - E] s^{3N} d\mathbf{p}^N d\mathbf{r}^N dp_s ds \\ &= \frac{1}{N!} \frac{e^{E/k_B T}}{3N + 1} \left[ \int dp_s e^{-\frac{p_s^2/2Q}{k_B T}} \right] \left[ \int e^{-\frac{H_{\text{orig}}}{k_B T}} d\mathbf{p}^N d\mathbf{r}^N \right] \\ &= \text{const} \times \Omega(T, V, N) \end{aligned}$$

where  $\Omega(T, V, N)$  is the canonical partition function for the original  $N$  particles. This implies that a micro-canonical simulation in the *extended system* will return a canonical simulation for the *original system*. The most important change from the Anderson thermostat is that in the Nosé thermostat there are no stochastic processes, and thus the whole method is deterministic. In other words, it does not de-correlate the particle velocities and thus it can approximate the true molecular kinetics of the original system.

### 2.8.4 Nosé-Hoover Thermostat

The above formulation is not convenient for simulation because of the scaling of the particle momenta by  $s$ , *which is a variable in the simulation*, implies that the time we should use to evaluate time-averages and values such as diffusion constant is also a variable in the simulation. Hoover developed a modification to the above formulation which remedies these problems [11]. He introduced a *friction coefficient*  $\xi$  which in a sense replaces  $p_s$

$$H = \frac{1}{2} \sum m_i |\mathbf{p}_i|^2 + U(\mathbf{r}^N) + \frac{\xi^2 Q}{2} + 3N k_B T \ln(s)$$

This substitution allows for integration schemes such as the velocity-verlet to be used to integrate the following equations of motion

$$\begin{aligned}\frac{d\mathbf{r}_i}{dt} &= \mathbf{v}_i \\ \frac{d\mathbf{v}_i}{dt} &= -\frac{1}{m_i} \frac{\partial U(\mathbf{r}^N)}{\partial \mathbf{r}_i} - \xi \mathbf{v}_i \\ \frac{d\xi}{dt} &= \left( \sum m_i |\mathbf{v}_i|^2 - 3Nk_B T \right) / Q \\ \frac{d \ln(s)}{dt} &= \xi\end{aligned}$$

This form is very commonly used in molecular simulations.

## 2.9 Three Particle Potentials

A natural extension to the relatively simple two particle potentials such as the Lennard-Jones model, which can really only be used for the simplest of cases in its base form, is the three particle potential. Using this model it is possible to simulate a lot of different molecules whose physical macroscopic properties depend on for example the angle between the constituent atoms, which in turn defines the distribution of charge in the molecule.

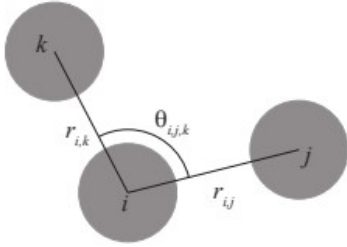


Figure 1: Three Particle Model from [1]

The three particle potential depends, as the name suggest, on the *relative positions* of three particles. In the notation of the above figure we have an atom  $i$  with two neighbors  $j$  and  $k$  with an angle between them  $\theta_{ij,k}$ . A common potential for modelling three particle interactions is the Stillinger-Weber potential

$$\begin{aligned}E &= \sum_i \sum_{j>i} V_2(r_{ij}) + \sum_i \sum_{j \neq i} \sum_{k>j} V_3(r_{ij}, r_{ik}, \theta_{ijk}) \\ V_2 &= A_{ij} \epsilon_{ij} \left( B_{ij} \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{p_{ij}} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{q_{ij}} \right) e^{\frac{\sigma_{ij}}{r_{ij} - a_{ij} \sigma_{ij}}} \\ V_3 &= \lambda_{ijk} \epsilon_{ijk} \left( \cos(\theta_{ijk}) - \cos(\theta_{ijk,0}) \right)^2 \\ &\quad \times \left( e^{\frac{\gamma_{ij} \sigma_{ij}}{r_{ij} - a_{ij} \sigma_{ij}}} e^{\frac{\gamma_{ik} \sigma_{ik}}{r_{ik} - a_{ik} \sigma_{ik}}} \right)\end{aligned}$$

There is a whole lot to unpack here, so we start at the top, with the expression for the energy. In the Stillinger-Weber model we have two contributions to the energy

which we have defined as  $V_2$  and  $V_3$ , the former depends only on the relative positions of the particles and it has the same form as the Lennard-Jones potential (1). Notice, that unlike the Lennard-Jones potential we have exponents in the attraction and repulsion terms  $p_{ij}$  and  $q_{ij}$  that can be varied. We have  $\epsilon$  and  $\sigma$  as described in 2.3.

## 3 Method

### 3.1 Studying the time evolution of particle velocities

Below is the markdown for generating a simulation of an Argon gas used to obtain the results show in the section below. After generating the lammps dump file Ovito GUI-version was used to analyze. The modification/histogram was used to generate the velocity histogram values and then post processed using a simple Python script.

*LAMMPS script task a)*

```
units lj
dimension 3
boundary p p p
atom_style atomic
lattice fcc 0.01
region simbox block 0 10 0 10 0 10
create_box 1 simbox
create_atoms 1 box
mass 1 1.0
velocity all create 2.5 87287 dist uniform
timestep 0.0009
pair_style lj/cut 3.0
pair_coeff 1 1 1.0 1.0 3.0
fix 1 all nve
dump 1 all custom 10 dump.lammpstrj id type x
y z vx vy vz
thermo 100
run 30000
```

### 3.2 Variation in Total Energy

To obtain the total energy plots discussed in the results section below I modified the task a) script using the lammps thermo\_style command [6]. To manipulate the logfiles produced by the script, open-source software called lammps-logfile [7] was used under the GNU general public license. The specific thermo\_style used for the second simulation is the following

Script from a...

```
thermo_style custom time temp pe ke etotal dt
```

Other than this the script is exactly the same as the one above 3.1

### 3.3 Temperature as a Function of Time

We use the equipartition principle, along with our calculated values for the kinetic energy of all our simulated atoms over time, to obtain a measurement of our systems temperature over time. We have (2), which we solve for  $T$  and get

$$T = \frac{2}{3} \frac{\langle E_k(t) \rangle}{k_B}$$

where  $\langle E_k(t) \rangle$  is the average kinetic energy of our particles (*after equilibration*) as a function of time. The actual script used to generate the values is the exact same one as in part b (3.2)

### 3.4 Measuring the Pressure of the System

Again we use the same script as for the two previous tasks, however this time we append a pressure calculation flag to the thermo\_style command.

### 3.5 Thermostats

Both the Berendsen and Nosé-Hoover thermostats are implemented in LAMMPS and can be accessed by the fix # all temp/berendsen and fix # all nvt temp commands respectively, where # represents the particle type you want to affect. It is important to remember that when using the Berendsen thermostats you need to invoke an integrator or else you will get no movement of the particles in the simulation. This was done as follows

#### Berendsen Thermostat

```
fix 1 all nve
fix 2 all temp/berendsen $temperature $temperature $dampening_time
```

### 3.6 Mean Square Displacement

For this part there are some changes as well as auxiliary calculations performed.

#### LAMMPS script task f)

```
[H] units lj
dimension 3
boundary p p p
atom_style atomic
lattice fcc 0.01
region simbox block 0 10 0 10 0 10
create_box 1 simbox
create_atoms 1 box
variable myTemp equal 0.0001
mass 1 1.0
velocity all create $myTemp 87287 dist uniform
timestep 0.01
pair_style lj/cut 3.0
pair_coeff 1 1 1.0 1.0 3.0
compute 1 all msd
fix 1 all nvt temp $myTemp $myTemp 1
dump 1 all custom 10 dump1lammprj id type x
y z vx vy vz
thermo 10
thermo_style custom time temp press pe ke etotal
run 10000
```

Here I had to add a thermostat as can be seen from the fix nvt command, *Nosé-Hoover* (2.8.4) in this case. When I did not use a thermostat the result would be a gas in very few time steps regardless of initial temperature. The compute 1 all msd command tells LAMMPS to calculate the mean square displacement, which is then added to the dump output using the dum 1 all custom command. This is then read using a file with the code: [12]. This code produces a text file with the mean square displacement data which can easily be analyzed and graphed using for example Python.

### 3.7 Silicon System

## 4 Results

### 4.1 Velocity Distribution

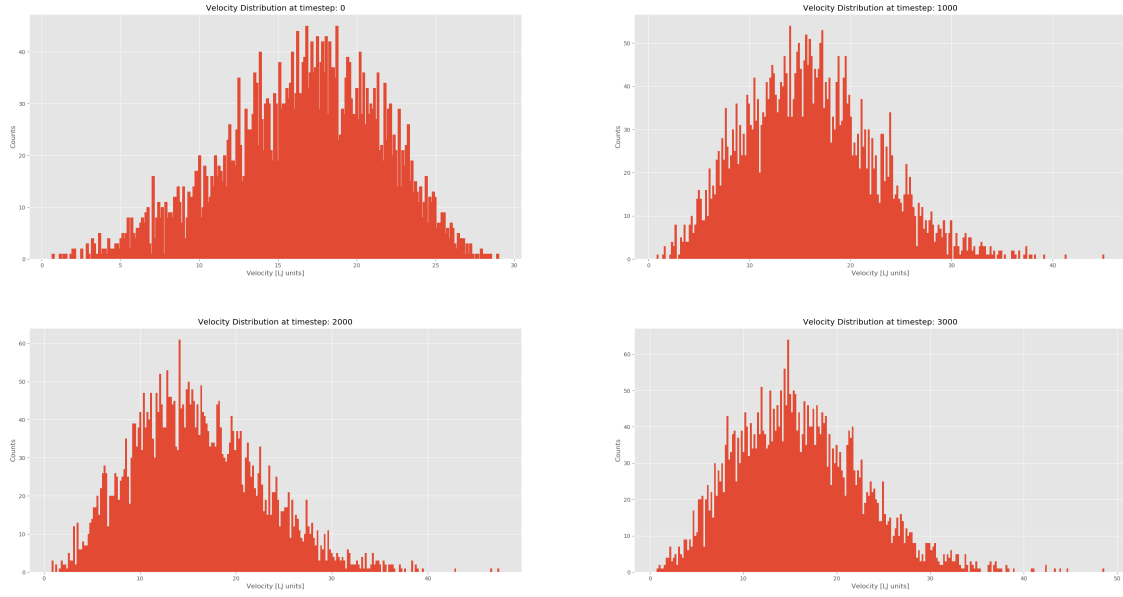


Figure 2: Evolution of velocity distribution for an Argon gas simulated using lammmps. Starting from a uniform distribution at the top left, to something that closely resembles the Maxwell-Boltzmann distribution on the bottom right. Script for generating the values is found in the method section (3.1). We ran the simulation for 30000 time steps, however after the first 3000, in OVITO (i.e  $3/5 \cdot 30000$ ) there is not much difference in the resulting histogram.



## 4.2 Variation in Total Energy



Figure 3: A simulation was ran with  $\Delta t = 1e - 9$  for 100000 time steps and the result was a constant total energy. The end time in this run was  $t = 0.0001$ , and looking at the graphs above we see that the energy stays constant for the time before  $t \approx 0.4$  and then we get this sharp drop. The assumption is that this behavior is due to something inside of LAMMPS's inner workings. After about  $t = 2$  we see the energy is fairly stable with some fluctuations. It would, however, require simulating for a very long time to get to this point with  $\Delta t = 1e - 9$  to check if this behaviour is still present for all time step values.

## 4.3 Temperature Comparison

Below is my result for comparing the value of the temperature measured from LAMMPS and my own calculated temperature from the average kinetic energy. I presume the difference in the two graphs is due to the fact that  $2/3$  cannot be done to infinite precision on a computer, whereas  $3/2$  the value LAMMPS uses, for this simulation, can be exactly represented. From the LAMMPS thermo\_style documentation page [6]: *The kinetic energy of the system  $ke$  is inferred from the temperature of the system with  $1/2k_B T$  of energy for each degree of freedom.*

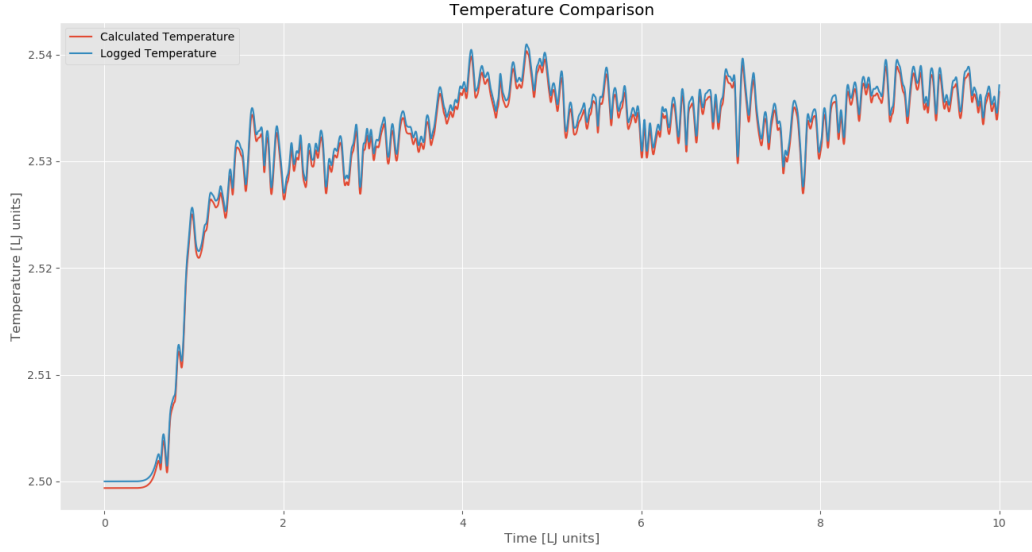


Figure 4: Judging from the OVITO GUI file viewer the system was in equilibrium at around  $t = 5$  for the above plot. The weird startup artifact is present in this simulation as well. During this time all the values for the simulation are completely constant.  $t \approx [0, 0.5]$

#### 4.4 Pressure Comparison

Below is the result for comparing the measured pressure to theoretical predictions given by the ideal gas law.

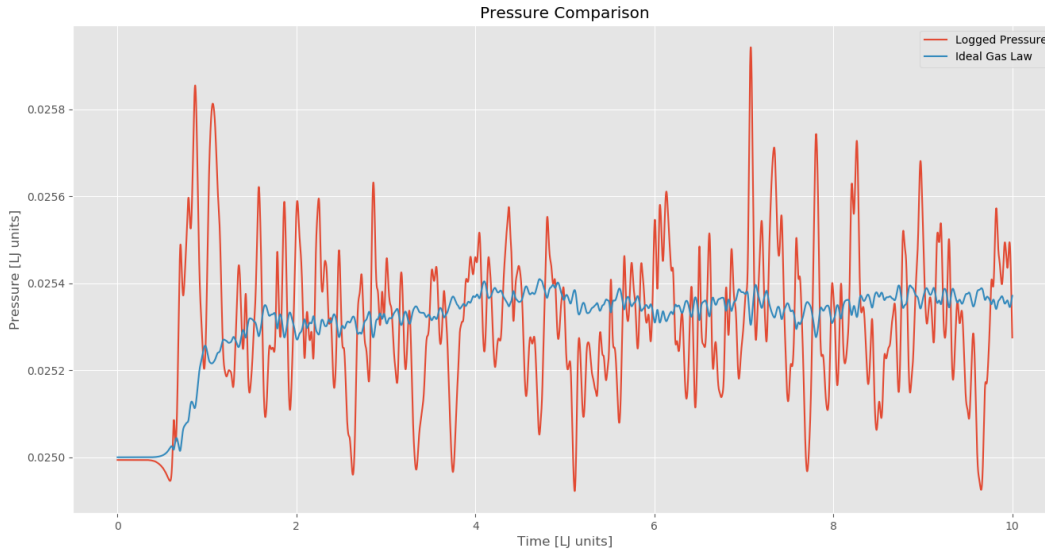


Figure 5: Comparison of the Virial theorem implemented by LAMMPS and the Ideal gas law. Both pressures follow the same trend.

#### 4.5 Nosé-Hoover and Berendsen

Below are figures for the Argon system comparing the action of the Nosé-Hoover and the Berendsen thermostats

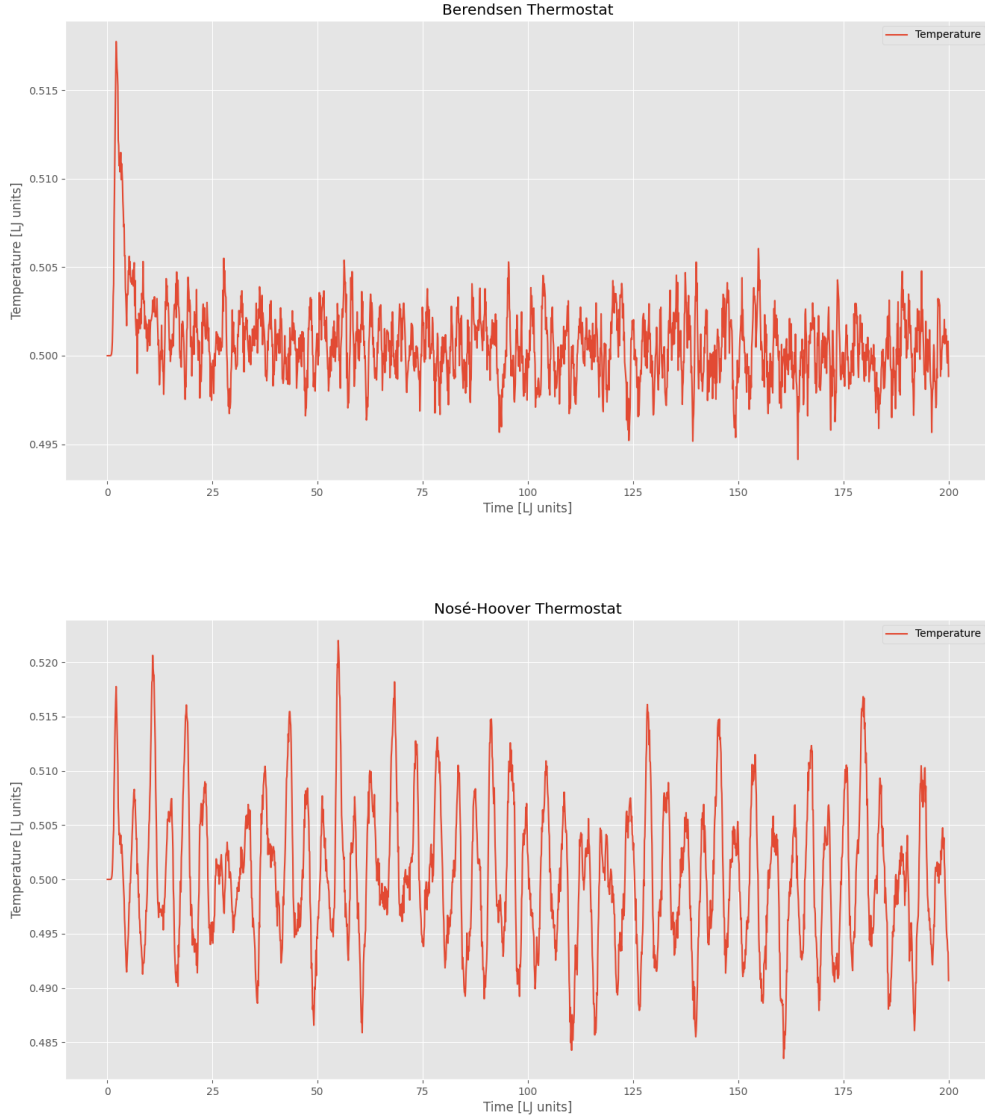


Figure 6: Here the Berendsen thermostat is used on the left and the Nosé-Hoover is used on the right. Both are using the same time step  $\Delta t = 0.01$  and dampening time  $\tau = 1$  as this was recommended in the LAMMPS documentation.

## 4.6 Mean Square Displacement

Below are the results for the mean square displacement measurement using the script described in (3.6)

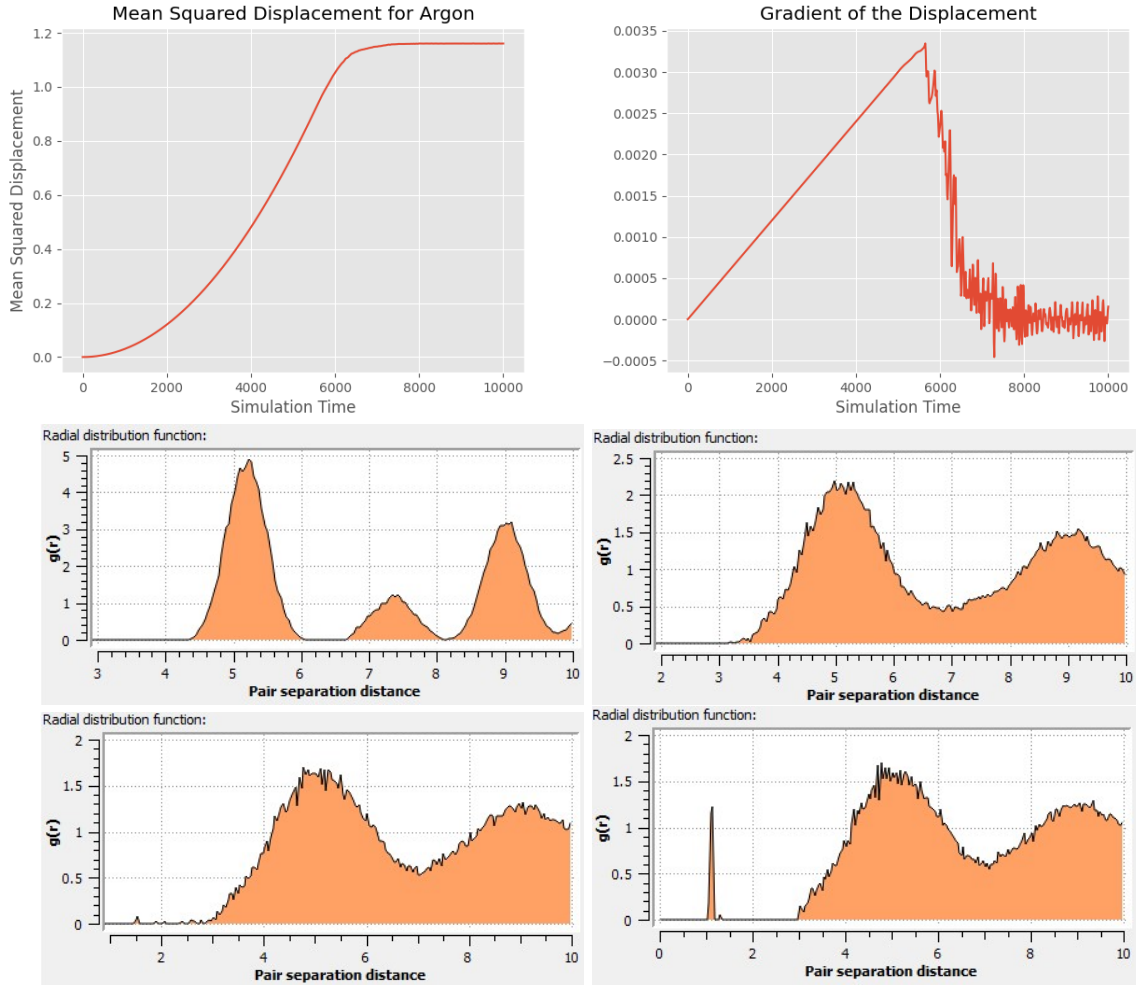


Figure 7: Upper left we have the the graph of the mean squared displacement with the accompanying derivative to the upper right (calculated using numpy's gradient function), the following four plots are the radial distribution function for timesteps: 2000, 4500, 6000, 8000 moving from left to right and from top to bottom.

## 4.7 Radial Distribution Function

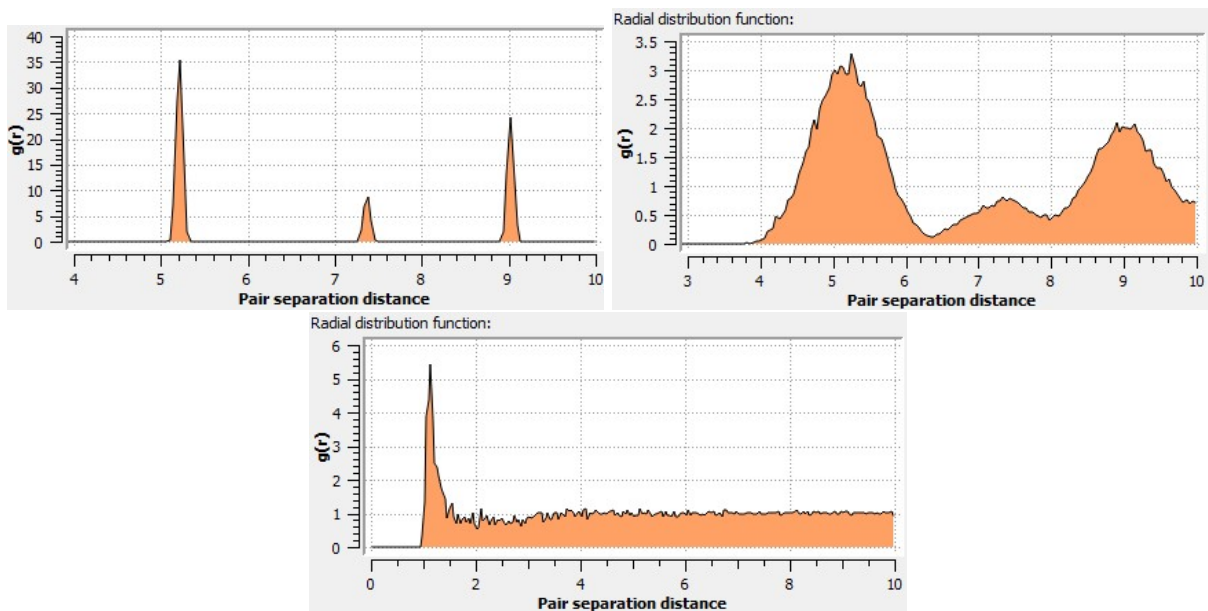


Figure 8: Radial distribution functions for the different phases of Argon, top left we have solid Argon, top right we have liquid Argon, and on the bottom we have gaseous Argon

## 4.8 Silicon Simulation

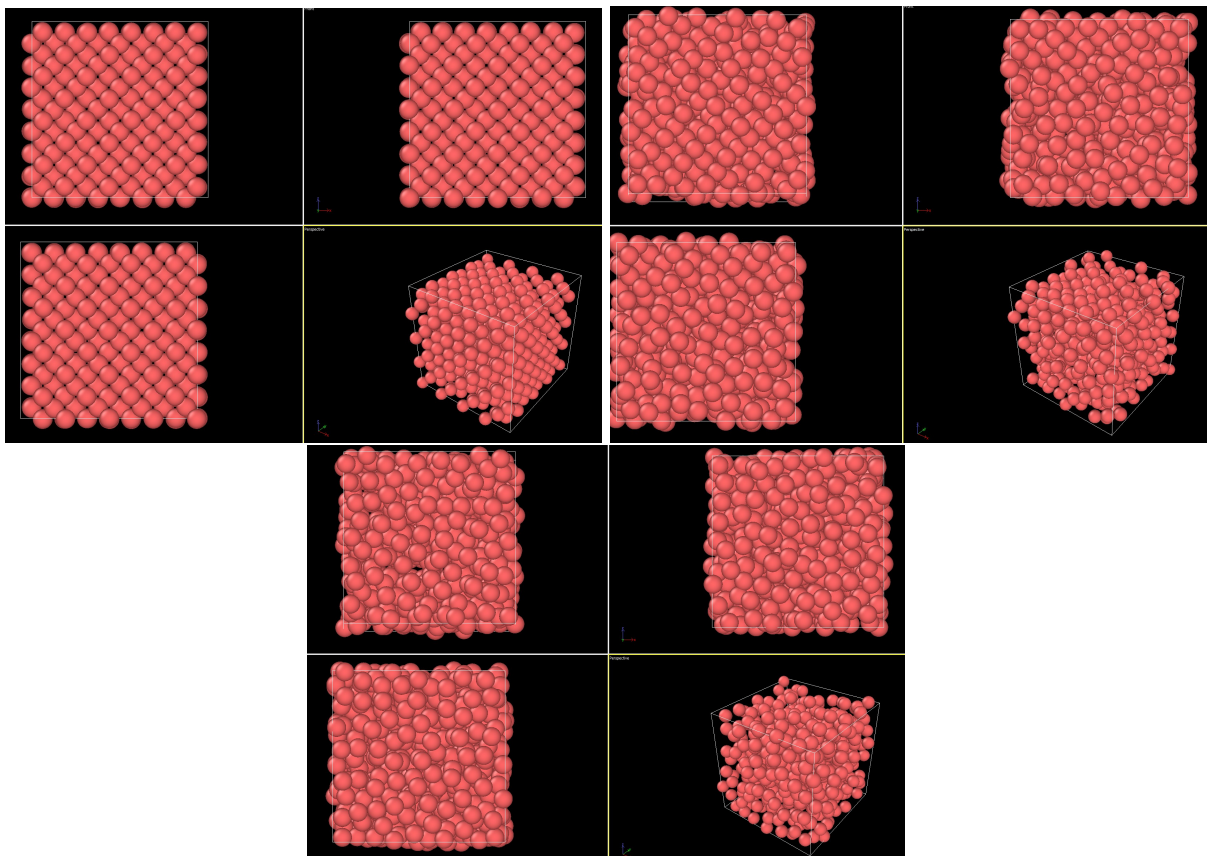


Figure 9: Silicon system phases visualized

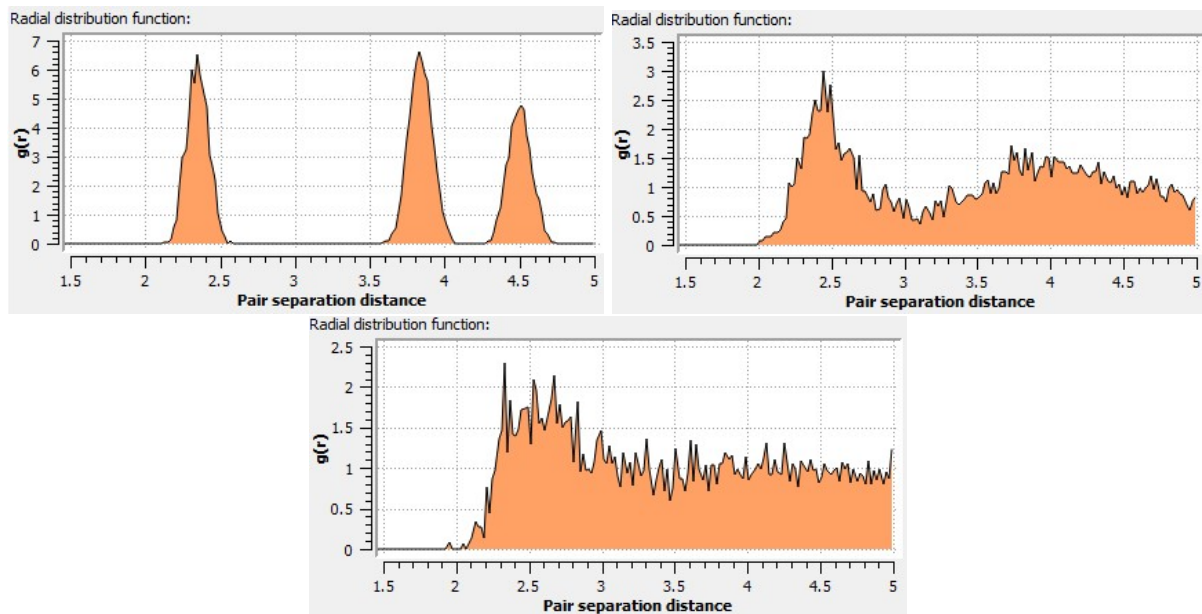


Figure 10: Radial Distribution Function for Silicon system, note that these three are within the first 50 frames of my simulation regardless of initial temperature. This was using the example code from the project text.

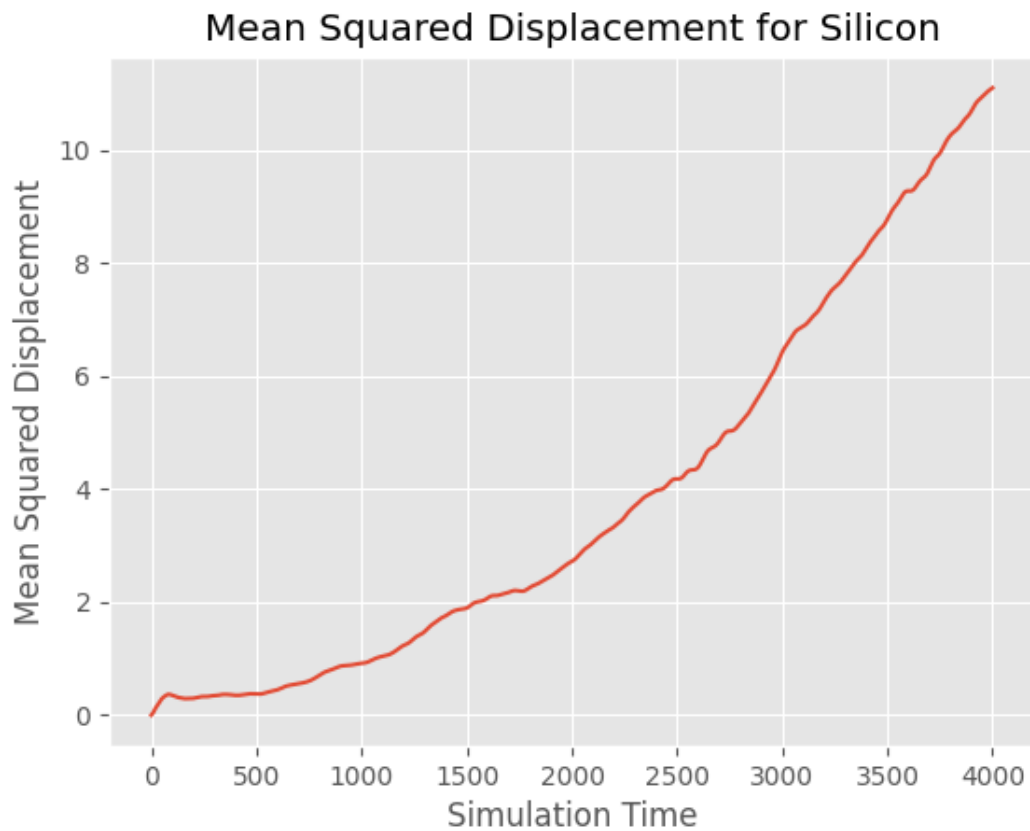


Figure 11: Mean Squared Displacement for liquid Silicon system

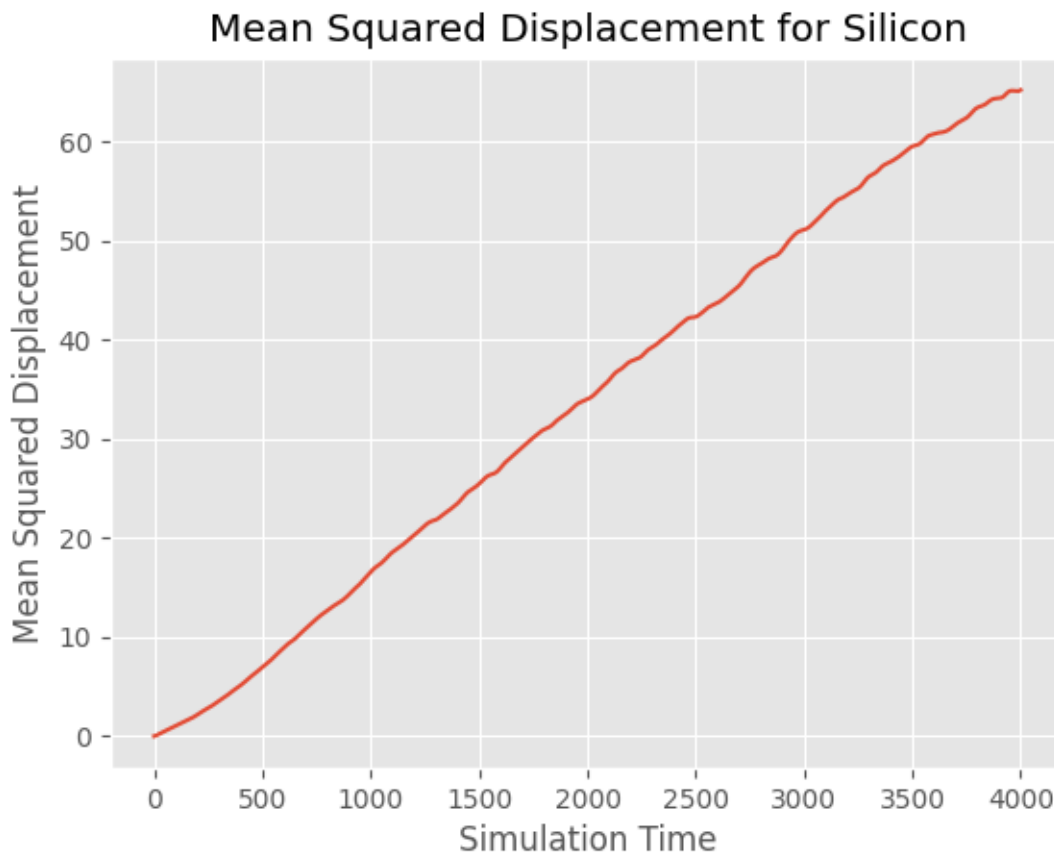


Figure 12: This gives closer to the expected result, what needed to change was one variable `myScale` in the input script from the project text. Happy it works but a lot of hours of frustration were wasted looking for solutions in the documentation.

## 5 Discussion and Conclusion

Throughout this project I have had some weird effects show up in my simulations. I don't know if this is common or what. First of all I there seems to be some weird behaviour with the Argon system that we can see in (??, 4 and 5). All of these graphs exhibit the same weird "constancy" in the first  $\approx 0.4t$  in the simulation time. Secondly, my mean squared displacement does not look like it should, in addition the lammps script (3.6) needed a temperature of  $T = 0.00001$  in order to obtain a system which did not immediately boil and become gaseous. This would usually take about 60 time steps. I checked for a lot of different time step sizes and temperatures. This did not fit with how I expected the code to work. Keeping in mind I am literally using example code, in which really the only alteration was the thermostats, initial temperature used to initialize the velocities, and the time steps. I would have very much liked to be able to resolve these issues but given time constraints and the

size of the project there is no more time and I will have to continue onto the second project straight away. I also apologize for how ugly this report is, easily the ugliest I have made during my time at the university, and something I would definitely want to fix given more time, however, there is not enough time to problem solve any longer. For the silicon simulation, also using the example script in the project text, the temperature jumps to around 8000 regardless of the initial temperature I set for the system. In my experience the LAMMPS documentation is very dense, and often even finding out that something is possible in the language takes 10 minutes, then you would have to look up a YouTube video to actually get a decent explanation of what the code actually does. With a project this large that quickly becomes unfeasible, at least in my opinion. I hope this does not come of as me just complaining because I have not done the work. I have had to sacrifice my other subjects for a whole week in order to get to this point. I really really hope that this project does not fail me from the course.

## References

- [1] Project text; *Introductory molecular modeling*, Anders Malthe Sørenssen, <https://www.uio.no/studier/emner/matnat/fys/FYS4460/v21/notes/project01-part1to3-v1.pdf>
- [2] Jones J. E. 1924 On the determination of molecular fields.—I. From the variation of the viscosity of a gas with temperature Proc. R. Soc. Lond. A106441–462
- [3] Wang, Xipeng & Ramírez-Hinestrosa, Simón & Dobnikar, Jure & Frenkel, Daan. (2019). The Lennard-Jones potential: when (not) to use it.
- [4] Pauli Exclusion Principle <http://hyperphysics.phy-astr.gsu.edu/hbase/molecule/Paulirep.html>
- [5] Mie potential: [http://www.sklogwiki.org/SklogWiki/index.php/Mie\\_potential](http://www.sklogwiki.org/SklogWiki/index.php/Mie_potential)
- [6] LAMMPS software documentation [https://lammps.sandia.gov/doc/thermo\\_style.html](https://lammps.sandia.gov/doc/thermo_style.html)
- [7] Henrik Andersen Sveinsson, University of Oslo, <https://github.com/henriassv/lammps-logfile>
- [8] Mark Tuckerman (New York University), [https://chem.libretexts.org/Bookshelves/Physical\\_and\\_Theoretical\\_Chemistry\\_Textbook\\_Maps/Supplemental\\_Modules\\_\(Physical\\_and\\_Theoretical\\_Chemistry\)/Statistical\\_Mechanics/Advanced\\_Statistical\\_Mechanics/Liouville's\\_Theorem\%2C\\_non-Hamiltonian\\_systems\%2C\\_the\\_microcanonical\\_ensemble/Equilibrium\\_ensembles/The\\_Microcanonical\\_Ensemble](https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Supplemental_Modules_(Physical_and_Theoretical_Chemistry)/Statistical_Mechanics/Advanced_Statistical_Mechanics/Liouville's_Theorem\%2C_non-Hamiltonian_systems\%2C_the_microcanonical_ensemble/Equilibrium_ensembles/The_Microcanonical_Ensemble)
- [9] Advanced Molecular Dynamics, M.S. Shell 2009, last visited 2/23/2021, [https://sites.engineering.ucsb.edu/~shell/che210d/Advanced\\_molecular\\_dynamics.pdf](https://sites.engineering.ucsb.edu/~shell/che210d/Advanced_molecular_dynamics.pdf)
- [10] Shūichi Nosé, Molecular Physics vol. 52, 1984, <https://doi.org/10.1080/00268978400101201>
- [11] William Hoover, Computational Statistical Mechanics <http://williamhoover.info/book.pdf>
- [12] Ovito Documentation, Alexander Stukowski, [https://ovito.org/manual\\_testing/python/introduction/examples/modifiers/msd\\_calculation.html](https://ovito.org/manual_testing/python/introduction/examples/modifiers/msd_calculation.html)