

1 Theory

1.1 Interacting Argon Atoms

The goal of this project is to simulate the dynamics of a system of N argon atoms. Interactions between atoms are approximated by the Lennard-Jones potential energy function:

$$V(\vec{\mathbf{r}}_i, \vec{\mathbf{r}}_j) = 4\varepsilon \left[\left(\frac{\sigma}{R} \right)^{12} - \left(\frac{\sigma}{R} \right)^6 \right] \quad (1)$$

where $R = |\vec{\mathbf{r}}_i - \vec{\mathbf{r}}_j|$ is the distance between the centers of atom i and atom j . The constants ε and σ , along with the mass m , are set to unity for this project. The force on atom i due to all other atoms in the system is:

$$\vec{\mathbf{F}}(\vec{\mathbf{r}}_i(t)) = \sum_{j \neq i} 24 \left[2 \left(\frac{1}{R} \right)^{14} - \left(\frac{1}{R} \right)^8 \right] (\vec{\mathbf{r}}_i(t) - \vec{\mathbf{r}}_j(t)) \quad (2)$$

An equation of motion is now written using Newton's classical laws for the i^{th} atom:

$$\frac{d^2 \vec{\mathbf{r}}_i(t)}{dt^2} = \frac{\vec{\mathbf{F}}(\vec{\mathbf{r}}_i(t))}{m} = \vec{\mathbf{F}}(\vec{\mathbf{r}}_i(t)) \quad (3)$$

Given proper initial conditions, these $3N$ ordinary differential equations are integrated in time to simulate the argon atoms dynamics.

1.2 Velocity Verlet Algorithm

Eqn. (3) is integrated in time using the well known Velocity Verlet Algorithm, which is derived from Taylor expansions. This algorithm updates both particle position and velocities in a three step process:

1. Define $\vec{\mathbf{a}}_i(t) = \vec{\mathbf{F}}(\vec{\mathbf{r}}_i(t))$
2. $\vec{\mathbf{r}}_i(t + \Delta t) = \vec{\mathbf{r}}_i(t) + \vec{\mathbf{v}}_i(t)\Delta t + \frac{1}{2}\vec{\mathbf{a}}_i(t)\Delta t^2$
3. $\vec{\mathbf{a}}_i(t + \Delta t) = \vec{\mathbf{F}}(\vec{\mathbf{r}}_i(t + \Delta t))$
4. $\vec{\mathbf{v}}_i(t + \Delta t) = \vec{\mathbf{v}}_i(t) + \frac{1}{2}[\vec{\mathbf{a}}_i(t + \Delta t) + \vec{\mathbf{a}}_i(t)]$

Here, Δt is the time step size of the time integration. The Velocity Verlet algorithm requires initial conditions for both particle positions and velocities, and will compute updated positions, velocities, and accelerations at each time step.

1.3 Initial Conditions

Initial conditions must be properly prescribed in order to integrate the equations of motion. For the positions, the argon particles are initially placed on a face centered cubic (FCC) lattice, shown in Figure 1. A subset of the FCC lattice positions, shown in red, can then be used as a template and iterated in all spatial directions until the desired amount of atoms are present in the system.

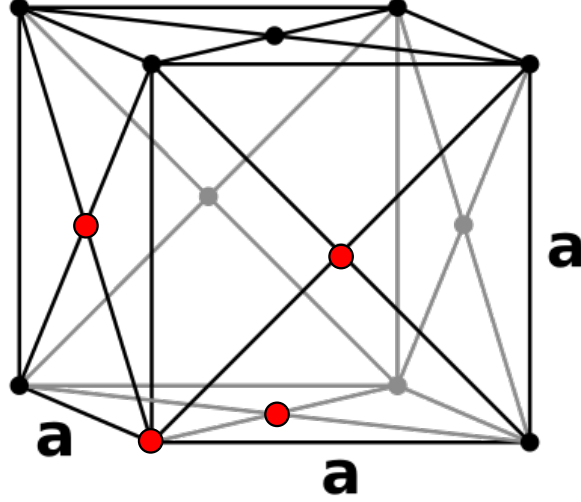


Figure 1: The face centered cubic (FCC) initial particle positions for the Velocity Verlet algorithm.

Next, the initial velocities must be set. A good initial velocity is generated from a Gaussian probability distribution:

$$P(x) = \frac{\exp [-(x - x_0)^2 / (2\sigma)^2]}{\sqrt{2\pi\sigma^2}} \quad (4)$$

For this code, the center and variance were set to $x_0 = 0$ and $\sigma^2 = 1$, respectively. Since the random initial velocities are distributed around zero, the total momentum will not be exactly zero. To prevent the system drifting in space, the random initial velocities are transformed to a center of mass reference frame:

$$\vec{v}_0 \rightarrow \vec{v}_0 - \frac{1}{N} \sum_{i=1}^N \vec{v}_i \quad (5)$$

The final part of initializing the velocities is to then scale them in order to get the desired temperature:

$$\vec{v}_0 \rightarrow \sqrt{\frac{3(N-1)T_{\text{desired}}}{\sum_{i=1}^N \vec{v}_i \cdot \vec{v}_i}} \vec{v}_0 \quad (6)$$

1.4 Periodic Boundary Conditions

Now that the integration method and initial conditions are set, boundary conditions on the system of argon atoms are the final piece of the puzzle in order to simulate the dynamics. Periodic boundary conditions (PCB) were chosen here, but a hard “box-like” boundary condition could also be imposed. The PCBs are needed in two aspects of the Velocity Verlet algorithm. The first is computing the accelerations between particles, and the second is when a particle leaves the computational domain created by the initial FCC lattice positions.

The *minimum image convention* is used to compute the accelerations between particles in the periodic FCC lattice. An illustration of the method is shown in Figure 2. If $R_{igkr} = |\vec{r}_{ig} - \vec{r}_{kr}|$ is less than $R_{irkr} = |\vec{r}_{ir} - \vec{r}_{kr}|$, then the stronger force with distance

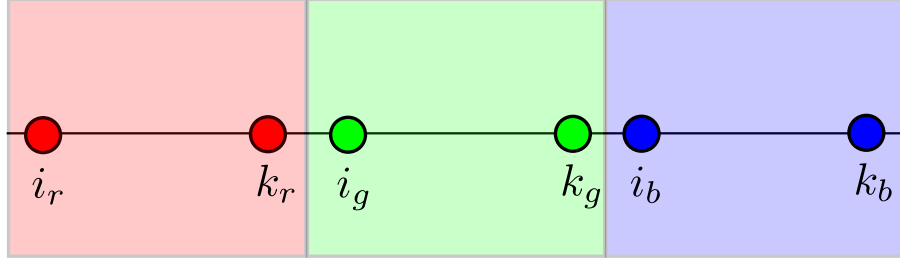


Figure 2: The minimum image convention illustration in one dimension.

$R_{i_g k_r}$ is used for computing the acceleration and the weaker force with distance $R_{i_r k_r}$ is neglected. Since the lattice is periodic, then the minimum image convention can be written in the succinct algorithm:

- If $R_{ik} = |\vec{\mathbf{r}}_i - \vec{\mathbf{r}}_k| > \frac{L}{2}$
 - If $R_{ik} > 0 \rightarrow R_{ik} = R_{ik} + L$
 - If $R_{ik} < 0 \rightarrow R_{ik} = R_{ik} - L$

This can be visually seen using Figure 2. When the two particles in the red unit cell are separated by a distance larger than half the unit cell size, then the second particle is translated by the unit cell dimension to compute the distance between the red particle and the green particle.

Using the PCBs for particles which leave the computational domain is much more straight forward. If the updated particle position is outside of the computational domain, then its position is merely shifted by the size of the computational domain $\vec{\mathbf{r}}_i \rightarrow \vec{\mathbf{r}}_i + L$.

1.5 Andersen Thermostat

An Andersen Thermostat was implemented to fix the temperature at a desired constant. This thermostat assumes that part of the system has an instantaneous interaction with fictional particles which exchange energy with the argon particles. Energy exchange between the thermostat and the Argon particles is implemented by replacing the particle momentum of some particles with a new momentum drawn from a Gaussian distribution. The Andersen Thermostat algorithm is:

- For velocity $\vec{\mathbf{v}}_i$, if (*Uniform random number* $\in [0, 1]$) $\leq 1 - \exp(-\nu \Delta t)$
 - $\vec{\mathbf{v}}_i \rightarrow \sqrt{\frac{k_B T}{2m}} \text{Gauss}(0, 1)$

Here, $\text{Gauss}(0, 1)$ is a Gaussian random number between 0 and 1.

2 Results

2.1 Temperature and Energy Measurements

For this report, four numerical measurements are taken in order to prove the validity of the code. These are: instantaneous temperature, kinetic energy, potential energy, and

total energy. The temperature at each time step T_n is computed using the equipartition formula:

$$3(N-1)k_B T = \frac{m}{2} \sum_{i=1}^N |\vec{v}_i|^2 \quad (7)$$

Total energy is computed as the sum of the kinetic and potential energy:

$$E = KE + PE = \frac{m}{2} \sum_{i=1}^N |\vec{v}_i|^2 + \sum_{i=1}^N \sum_{j=1}^N V(\vec{r}_i, \vec{r}_j) \quad (8)$$

where $V(\vec{r}_i, \vec{r}_j)$ is given by Eqn. (1).

2.2 $N = 256, T = 5$

The temperature for an $N = 256$ particle system with a desired temperature of $T = 5K$ is shown in Figure 3. A total of 20,000 time steps were used in this simulation with $\Delta t = 10^{-4}$, and the thermostat with $\nu = 1, 10$, and 100 was turned off after 10,000 time steps. This result proves the validity of the molecular dynamics code and at the

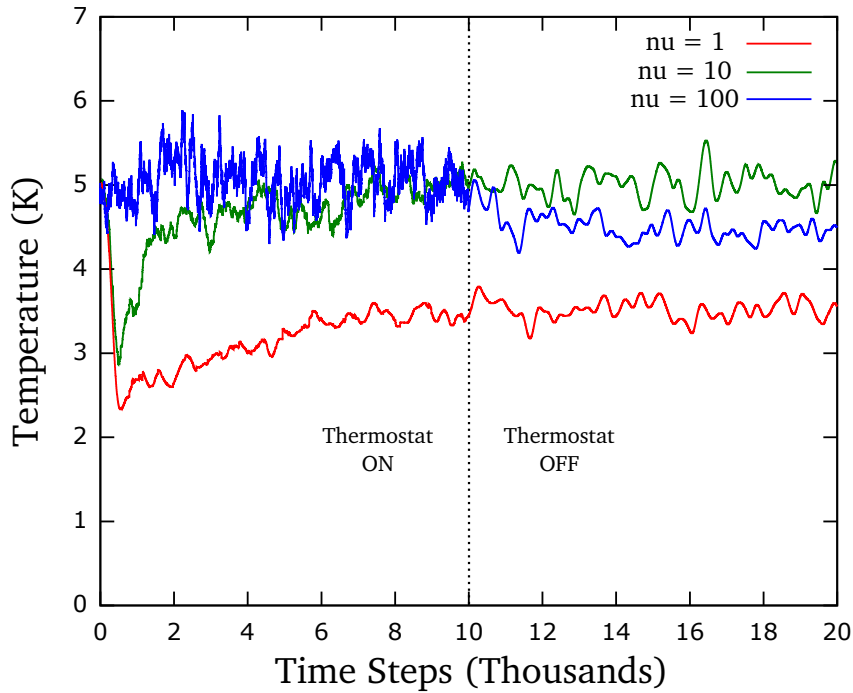


Figure 3: Molecular dynamics temperature result for $N = 256$ particles at $T = 5K$ after 20,000 time steps.

same time provides some interesting results. The first noticeable characteristic of the temperature is that it tries to drift away from the desired value for $\nu = 1$ and 10 , but the thermostat is able to slowly bring it back. The temperature with $\nu = 100$ suffers from much more wild oscillations than the others, however, and starts to drift away from the desired temperature after the thermostat is turned off.

Since the thermostat with $\nu = 10$ performs the best over this time range, this parameter will be used to calculate the energy of the system. Figure 4 shows the calculated kinetic, potential, and total energy with the same parameters and $\nu = 10$. After the thermostat is turned off, the total energy is constant. This matches intuition as after

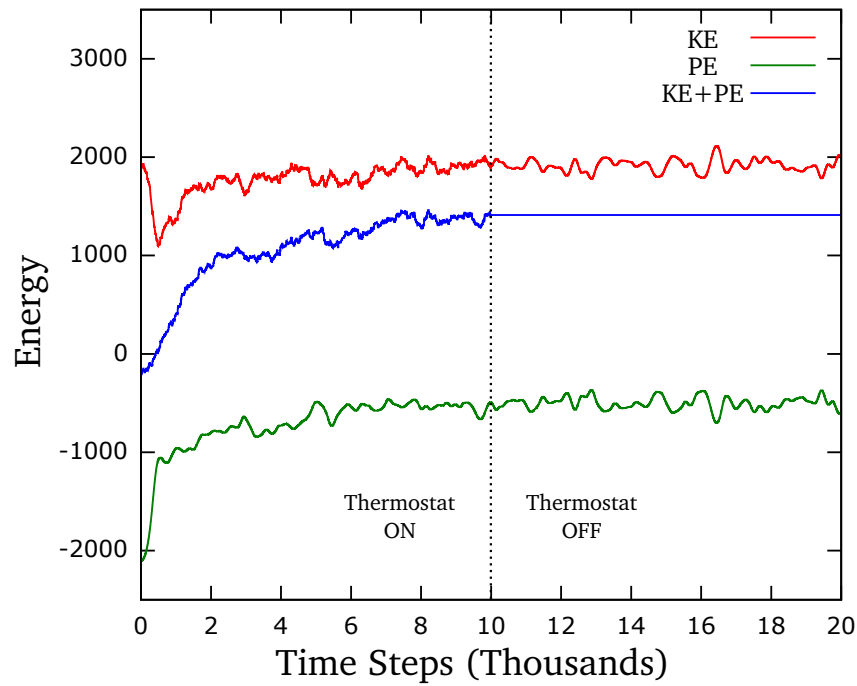


Figure 4: Molecular dynamics energy result for $N = 256$ particles at $T = 5K$ after 20,000 time steps. Here, KE stands for kinetic energy and PE stands for potential energy.

the thermostat is turned off, no energy is entering or leaving the system. This energy result also provides an excellent validation that the code is in fact valid and performing as expected.