

Assignment 2: Extended System

Marta Casado Carrasquer

March 24, 2025

1 Introduction

Molecular dynamics (MD) is a technique used to study the physical movements of atoms and molecules over time. By solving Newton's equations of motion, MD enables the prediction of particle trajectories based on interactions governed by forces and potential energies. This method provides insights into the dynamic behaviour, structural properties, and thermodynamics of systems at the atomic level.

Section 2 in this report gives an understanding into how a MD simulation, for liquid Argon, is implemented. Section 3 discusses such implementation by giving some insight into the system's properties and behaviour.

2 Theory and Methods

2.1 Periodic Boundary Conditions and Minimum Image Convention

In order to account for the finite size of the simulation box, Periodic Boundary Conditions (PBC) can be implemented to approximate an infinite system. This ensures that atoms leaving one side of the box re-enter from the opposite side, preserving continuity. The minimum image convention is applied to account for interactions only with the nearest periodic image of each particle.

The minimum image convention is implemented our code with the function `minimum_image`, which ensures that the displacement vector \mathbf{r} between two particles is computed considering the closest periodic image:

$$\mathbf{r}_{\text{closest}} = \mathbf{r} - L \cdot \text{rint} \left(\frac{\mathbf{r}}{L} \right), \quad (1)$$

where L is the size of the box.

PBC are explicitly applied during position updates using the function `apply_PBC`, which keeps particle positions within the box by wrapping coordinates:

$$\mathbf{x} \leftarrow \mathbf{x} - L \cdot \lfloor \mathbf{x}/L \rfloor. \quad (2)$$

2.2 Initial Conditions: Lattice Initialization

The simulation initializes particles in a simple cubic lattice to ensure a uniform distribution. For a given cube of size L , the number of atoms per dimension is n , leading to a total of $N = n^3$ atoms. The lattice spacing d is determined to match the target density:

$$d = \frac{L}{n}. \quad (3)$$

The function `generate_lattice` assigns initial positions (x, y, z) by iterating over all integer indices i, j, k :

$$x = id, \quad y = jd, \quad z = kd. \quad (4)$$

2.3 Velocity Initialization and Rescaling

Initial velocities are drawn from a normal distribution using `random_velocities`. To ensure consistency with the desired temperature T , velocities are rescaled to match the equipartition theorem:

$$K = \frac{1}{2} \sum_{i=1}^N m v_i^2, \quad T = \frac{2K}{3k_B N}. \quad (5)$$

To rescale velocities, a scaling factor a is applied:

$$a = \sqrt{\frac{T_{\text{target}}}{T_{\text{current}}}}, \quad (6)$$

where T_{target} is the desired temperature. The function `rescale_velocities` applies this scaling factor to ensure the system starts with the correct kinetic energy. Furthermore, to remove center-of-mass drift, the center-of-mass velocity is subtracted:

$$\mathbf{v} \leftarrow \mathbf{v} - \frac{1}{N} \sum_{i=1}^N \mathbf{v}_i. \quad (7)$$

This ensures that the total momentum of the system is zero so we are in the inertial frame of reference of the system.

2.4 Molecular Dynamics Integration

The equations of motion are integrated using the Velocity Verlet algorithm in the function MD. The algorithm proceeds as follows:

1. Update positions:

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t)\Delta t + \frac{\Delta t^2}{2}\mathbf{F}_i(t). \quad (8)$$

2. Apply periodic boundary conditions using `apply_PBC`.
3. Compute potential energy and new forces based on the Lennard-Jones potential.
4. Update velocities:

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{2} [\mathbf{F}_i(t) + \mathbf{F}_i(t + \Delta t)]. \quad (9)$$

5. Rescale velocities periodically during equilibration.

This ensures energy conservation and stability in the simulation.

3 Results and Discussion

Here, we present the output data for our MD simulation which contains plots of positions, velocities and energies used to analyze our system's behaviour and implementation. Figures that study the change in our configuration over time are also added.

3.1 Simulation Parameters

The system is initialized with $N = 125$ atoms in a cubic lattice with side-length $L \approx 18.1 \text{ \AA}$, computed from the target density: 1.4 g/cm^3 . The target temperature is set to $T = 95K$. The integration is performed using a time-step of $\Delta t = 0.001 \text{ ps}$ over 10 ps of simulation time i.e. 10,000 simulation steps.

3.2 Bouncing Test

In order to verify the correct implementation of PBC and coordinate wrapping, we redo the bouncing test from Assignment 1 before implementing our simple cubic Initial Conditions.

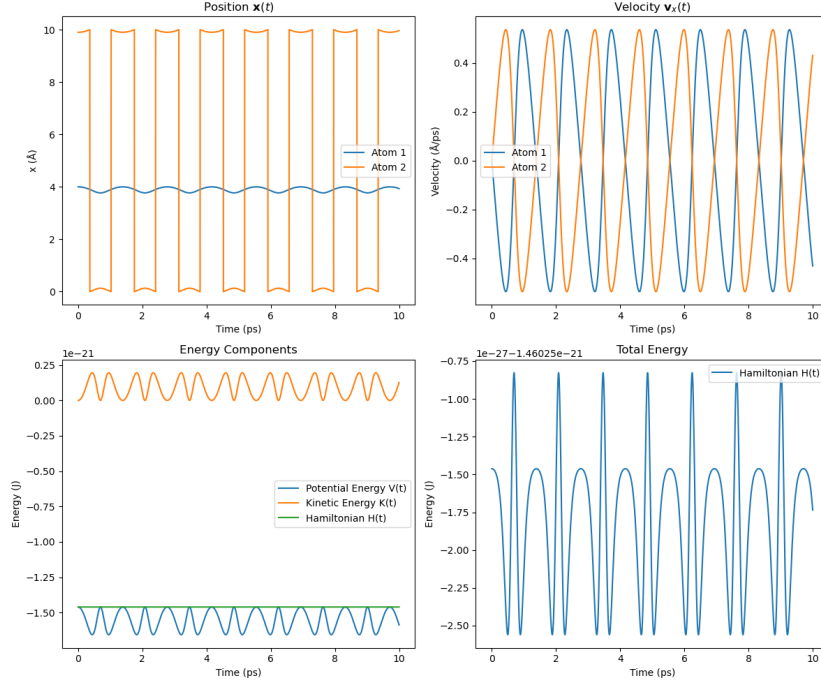


Figure 1: Plots for position, velocity and different energy components for 2 atoms separated by a distance dependent on L .

In Assignment 1, two atoms were separated in the x -direction by 4 Å. Now, the separation between the atoms depends on length L given by:

$$L = \frac{10}{3.405} \quad (10)$$

in reduced units, corresponding to 10 Å. The two atoms are initialized with the following properties:

- Atom 1 starts at $x_0 = 0.4L$ (as in Assignment 1).
- Atom 2 starts at $x_0 + 0.59L$, placing it near the opposite end of the box.
- Initial velocities are set to zero, ensuring the system begins at rest.

The position plot $\mathbf{x}(t)$ of Atom 2 from Figure 1 exhibits great discontinuities, indicating that it crosses the boundary and wraps around. Atom 1, on the other hand, oscillates within a small range. Because of how PBC are applied, we ensure that all positions remain within $[0, L)$.

Furthermore, plots for the energies and velocity show their corresponding antisymmetric oscillatory behaviour (just as in Assignment 1) and the Hamiltonian is constant.

3.3 Change in Configuration

Figure 2 studies the change in configuration of our system over time.

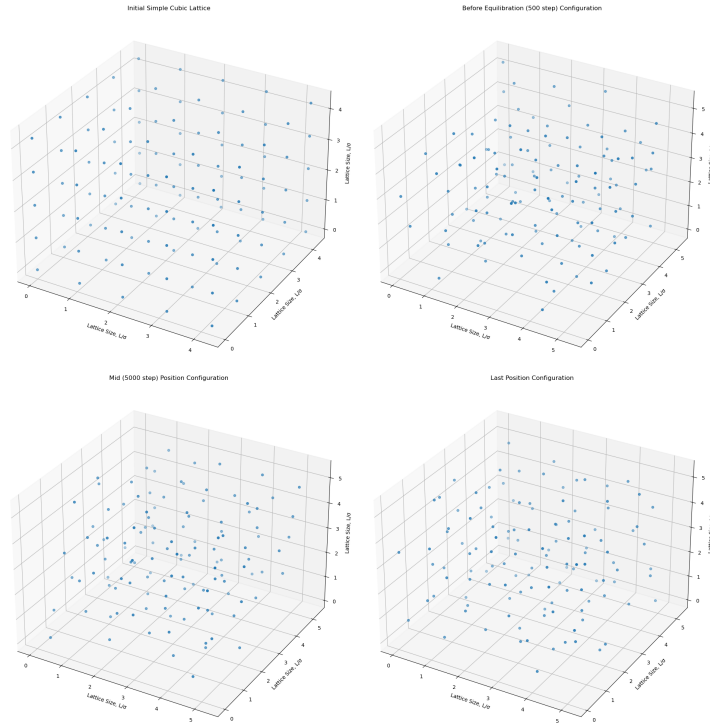


Figure 2: Different snapshots of MD simulation over time.

At first, the atoms are arranged in a simple cubic lattice, as shown in Figure 2. This choice of Initial Conditions ensures that no initial overlapping occurs, as could be the case if choosing to arrange the atoms randomly.

After 500 time steps, the atom's positions begin to deviate from their original lattice points due to thermal motion. In this time-step, the system is equilibrating to the target temperature.

At 5000 steps, the movement of the atoms becomes more pronounced with them varying significantly from their initial positions (Initial Simple Cubic Lattice). The system has reached equilibrium.

By the final stage of the simulation, atoms are distributed in a uniform and random manner. The system has had enough time to equilibrate and its structure resembles that of a liquid.

3.4 Evolution of Energy Over Time

What may have seemed odd from the previous subsection is how were we able to tell equilibration of our system. We resolve this question by looking at Figure 3.

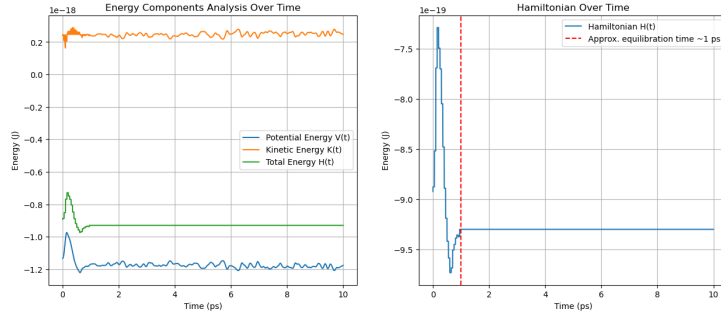


Figure 3: Time evolution of kinetic, potential, and total energy.

The left plot in Figure 3 presents the time evolution of potential, kinetic and total energy of the system. Initially, there are significant fluctuations in both the potential and kinetic energy which indicates that the system is rapidly adjusting from its initial configuration. As time progresses, the kinetic and potential energies stabilize, reaching a steady-state equilibrium after approximately 1 ps, i.e. 1000 steps.

The right plot in Figure 3 focuses on the time evolution of the Hamiltonian. Initially we give our system some Initial configuration Conditions and apply velocity rescaling (thermostat) in order to give it its target temperature. As a result, $\mathcal{H}(t)$ exhibits a transient behaviour with noticeable fluctuations until it adjusts to the desired conditions stated by the system. That is way, after approximately 1 ps, the total energy stabilizes, denoting that the system has reached an equilibrium state. Once in that state, the Hamiltonian remains constant over time and energy is conserved. This is crucial to determine we are, indeed, working in the NVE ensemble.