UNIVERSITY OF HELSINKI
FACULTY OF SCIENCE

Course: Programming Projects in Molecular Modeling

Project 4

Authors: Mahdi Torabi, Ruoyu Tang, Iris Tamsalu

Instructor: Prof. Alex Bunker

Date: April 8, 2025

Contents

1 Introduction

This project builds on Project 3, where we developed a simple molecular dynamics (MD) program. In part 1, we analyze the time evolution of kinetic and potential energy to observe equilibration. From the kinetic energy, we compute an effective temperature and study its fluctuations once stabilized. A histogram shows these fluctuations, and we examine how fluctuations change with particle number.

In part 2, we implement Langevin and Berendsen thermostats, using the average temperature from part 1 as the target temperature. We study the effect of larger time steps and varying thermostat parameters and compute diffusion coefficients for comparison.

In part 3, we study 2D systems at various temperatures and densities, visualizing structural changes and comparing them with published results. Radial distribution functions (RDFs) are computed to quantify the observed behavior.

In part 4, we simulate the same two-dimensional Lennard-Jones system using Monte Carlo (MC) methods. We explore the effect of step size and acceptance criteria to find an optimal move distance. Then we compare the results and efficiency with MD simulations.

The code and visuals are provided in the supplementary material, available at the following GitHub repository: github.com/iristamsalu/KEM381_project4. All usage instructions are provided in the README file.

2 Theory

2.1 Lennard-Jones Potential

The Lennard-Jones (LJ) potential is a function used in physics and chemistry to describe the interaction between a pair of neutral atoms or molecules. It models van der Waals forces (attractive forces at long distances) and Pauli repulsion (repulsive forces at short distances) in molecular dynamics simulations. The LJ potential mathematical form is:

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

where ϵ is the potential well's depth, representing the interaction's strength. σ is the finite distance at which the interparticle potential is zero, which roughly corresponds to the diameter of the particles. The term $\left(\frac{\sigma}{r}\right)^{12}$ represents the short-range repulsion due to overlapping electron clouds, called the Pauli exclusion principle. The term $\left(\frac{\sigma}{r}\right)^{6}$ represents the long-range attraction due to van der Waals forces.

The force is calculated as the negative gradient of this potential.

2.2 Langevin Thermostat

The Langevin thermostat is a stochastic, dissipative thermostat that controls the temperature in MD simulations. The reason behind the name is that this method is based on the Langevin equation. The Langevin equation is Newton's second law with an extra term to represent the frictional force and another extra random force that represents the thermal fluctuations. The equation is as follows:

$$m\frac{d^2r}{dt^2} = -\nabla U(r) - \zeta\frac{dr}{dt} + \sqrt{2\zeta T}R(t) \tag{2}$$

where $U(r)$ is the LJ potential function, ζ is the friction coefficient, and $R(t)$ is a random Gaussian force representing thermal fluctuations. $R(t)$ must obey the fluctuation-dissipation theorem conditions:

$$\langle R(t)\rangle = 0, \quad \langle R(t)R(t')\rangle = 2\zeta T m\delta(t-t') \tag{3}$$

where $\delta(t-t')$ is the Dirac delta function.
The Langevin thermostat is implemented in an MD simulation by modifying the velocity calculation step as follows:

$$v(t+\Delta t) = v(t) + \left(-\frac{\zeta v(t)}{m} + \frac{F(t)}{m}\right)\Delta t + \frac{\sqrt{2\zeta T\Delta t}}{m}r_G \tag{4}$$

The friction coefficient is a parameter we choose for our simulation, which represents the level of damping.

For the velocity verlet algorithm:

$$r_i(t+\Delta t) = r_i(t) + v_i(t)\Delta t + \frac{1}{2}a_i(t)\Delta t^2 \tag{5}$$

$$a_i(t+\Delta t) = \frac{F_i(t+\Delta t)}{m_i} \tag{6}$$

$$v_i(t+\Delta t) = v_i(t)e^{-\zeta\Delta t} + \frac{1}{2}\left[a_i(t) + a_i(t+\Delta t)\right]\Delta t + \eta_i\sqrt{\frac{T(1-e^{-2\zeta\Delta t})}{m_i}} \tag{7}$$

The Langevin thermostat is efficient and useful for non-equilibrium simulations as well. However, it alters the dynamics of the simulation, and the choice of the ζ affects the results.

2.3 Berendsen Thermostat

Berendsen et al. introduced the Berendsen thermostat, coupling the system to an external heat bath. It is a widely used method for temperature control in MD, particularly for initial equilibration. It controls the temperature by rescaling the velocities of particles based on the difference between the system's current temperature T_m and the target temperature T_0. The temperature relaxation follows the equation:

$$\frac{dT}{dt} = \frac{T_0 - T_m}{\tau} \tag{8}$$

where T_m is the measured system temperature, T_0 is the desired reference temperature, and τ is the thermostat coupling time constant.

In practice, particle velocities v_i are rescaled at each timestep as:

$$v_i \to \alpha v_i \tag{9}$$

where the rescaling factor λ is given by:

$$\alpha = \sqrt{1 + \frac{\Delta t}{\tau_T}\left(\frac{T_0}{T_m} - 1\right)} \tag{10}$$

In the velocity verlet algorithm, the Berendsen thermostat is implemented as follows:

$$r_i(t+\Delta t) = r_i(t) + v_i(t)\Delta t + \frac{1}{2}a_i(t)\Delta t^2 \tag{11}$$

$$a_i(t+\Delta t) = \frac{F_i(t+\Delta t)}{m_i} \tag{12}$$

$$v_i(t+\Delta t) = v_i(t) + \frac{1}{2}[a_i(t) + a_i(t+\Delta t)]\Delta t \tag{13}$$

$$T = \frac{2K}{(3N-3)}, \quad K = \sum_i \frac{1}{2} m_i v_i^2 \tag{14}$$

$$v_i(t+\Delta t) \rightarrow \alpha v_i(t+\Delta t) \tag{15}$$

$$\alpha = \sqrt{1 + \frac{\Delta t}{\tau}\left(\frac{T_0}{T} - 1\right)} \tag{16}$$

Where in eq. (14), the denominator has $(3N-3)$. This is because we have to remove the degrees of freedom that are related to the motion of the center of mass (x, y, z). This is because the motion of the center of mass is simply 0. The system has no translational motion.

Berendsen thermostat is simple and computationally efficient. It's effective for rapid system equilibration and produces correct microscopic dynamics. However, a disadvantage is that it has the wrong fluctuations. Therefore, due to its inability to produce a true canonical ensemble, it is typically replaced with Nosé-Hoover thermostats for long production runs.

The coupling parameter τ determines the strength of the temperature control:

- **Small** τ (e.g., $0.1-1$): Strong coupling, fast equilibration, but suppresses fluctuations.
- **Large** τ (e.g., $5-10$): Weak coupling, preserves system dynamics better.

2.4 Radial Distribution Function

The radial distribution function (RDF), denoted as $g(r)$, describes how the particle density varies as a function of distance. Mathematically, the RDF is defined as:

$$g(r) = \frac{\rho(r)}{\rho_0} \tag{17}$$

where $\rho(r)$ is the number density of particles at a distance r from a reference particle. ρ_0 is the bulk density.

In an MD simulation, the RDF is calculated by choosing a particle and measuring distances to all other particles. Next, define small radial intervals dr and count the number of particles in each shell of thickness dr at a distance r. Finally, the count is normalized. In homogeneous systems:

$$g(r) = \frac{\text{Number of particles in shell } (r, r + dr)}{2\pi r^2 dr \cdot \rho_0} \tag{18}$$

RDF typically has distinct peaks in liquids corresponding to the nearest-neighbor shell structure. In solids, RDF shows well-defined peaks at characteristic distances. In gases, RDF is close to 1 for all distances, indicating little or no local ordering.

2.5 Monte Carlo Simulation

Monte Carlo (MC) simulation of N particles in a box is a computational method used to study the statistical mechanics and thermodynamic properties of a system of interacting or non-interacting particles. The technique relies on stochastic sampling and probability to explore the phase space of the system and estimate properties such as energy, pressure, and temperature.

The simulation steps are usually as follows:

- N particles are placed inside a 2D or 3D box of volume V.
- Periodic Boundary conditions are imposed.
- The LJ potential defines the interaction between particles.
- A particle is randomly selected, and a small displacement is proposed.
- The energy change ΔE due to the move is calculated.
- The move is accepted based on the following criterion:

  $$r > P = e^{-\frac{\Delta E}{k_B T}}$$

  where r is a random number from a uniform distribution. or $\Delta E < 0$
- If the move is accepted, the particle position is updated; otherwise, the move is rejected.

The average energy is calculated as follows:

$$\langle E \rangle = \frac{1}{M} \sum_{i=1}^{M} E_i$$

MC gives equilibrium properties efficiently but lacks dynamic information, such as particle trajectories or transport properties like diffusion or viscosity, which require methods like MD.

2.6 Autocorrelation

Autocorrelation measures how much a quantity, such as energy, correlates with itself at different time lags. In MC simulation, the higher the values, the less independent the samples are from each other, resulting in a worse exploration of the phase space.

When we have a sequence of energy values $E_1, E_2, \ldots, E_N$, the autocorrelation at lag t is defined as:

$$\rho(t) = \frac{\mathbb{E}[(E_i - \bar{E})(E_{i+t} - \bar{E})]}{\mathbb{E}[(E_i - \bar{E})^2]}$$

where $\mathbb{E}$ denotes the sample average, and $\bar{E}$ is the mean energy. A rapidly decaying $\rho(t)$ implies a better simulation, leading to more independent samples, which improves the efficiency of the simulation.

Integrated autocorrelation time (IACT), on the other hand, is a statistic that tells us how many correlated samples are equivalent to one independent sample. In other words, you can calculate the ratio of the independent steps to the total steps taken. The lower the value of IACT, the higher the number of independent steps.

IATC is defined as follows:

$$\tau_{\text{int}} = 1 + 2 \sum_{t=1}^{\infty} \rho(t)$$

In practice, the sum is truncated when $\rho(t)$ becomes negligible.

2.7 Diffusion Coefficient

The diffusion coefficient D characterizes how fast particles spread due to random motion. In an MD simulation, it is calculated from the Mean Squared Displacement (MSD) data using the Einstein relation. In d dimensions, the Einstein relation is:

$$D = \lim_{t \to \infty} \frac{1}{2dt} \left\langle |\mathbf{r}(t) - \mathbf{r}(0)|^2 \right\rangle$$

Where: $\mathbf{r}(t)$ is the position of a particle at time t and $\langle \cdot \rangle$ is the ensemble and time average. This implies that the slope of the MSD versus time is proportional to the diffusion coefficient.

The following steps are followed for the calculation of D from the trajectory:

1. Compute MSD:

$$\text{MSD}(\tau) = \left\langle |\mathbf{r}_i(t + \tau) - \mathbf{r}_i(t)|^2 \right\rangle$$

   Averaged over all particles i and multiple time origins t.

2. Fit Linear Region: Making sure that the system has equilibrated, the slope is fitted to the equilibrated region. The slope will be proportional to D:

$$\mathrm{MSD}(t) = 2dDt + C$$

In this project, we used the MSD method using particle positions, but it is also possible to calculate the diffusion coefficient from the velocity autocorrelation function (VACF). In this method, the diffusion coefficient is given by:

$$D = \frac{1}{d} \int_0^\infty \langle \mathbf{v}(t) \cdot \mathbf{v}(0) \rangle \, dt$$

where $\mathbf{v}(t)$ is the velocity of the particle at time t, and $\langle \cdot \rangle$ represents the time average.

3 Method, Results and Discussion

3.1 Part 1

In this part, the code from the previous project was revisited, and the output data were analyzed further. First, the kinetic and potential energy of 3 different simulations with different numbers of particles were analyzed. The results for the two-dimensional case are in fig. 2. The equilibration happens at a faster pace as the particle number increases. In these figures, it is also visible that the energy fluctuations are higher for systems with a smaller number of particles.

Next, the temperature distribution was investigated by using the equation explained in the theory section and running 3 simulations with 100, 300, and 500 particles in 2D, respectively. The density of the system was kept constant at 0.35. The results are in fig. 1. The results clearly show that as the particle number increases, the distribution becomes narrower and the peak higher. Low particle number gives a wider distribution.

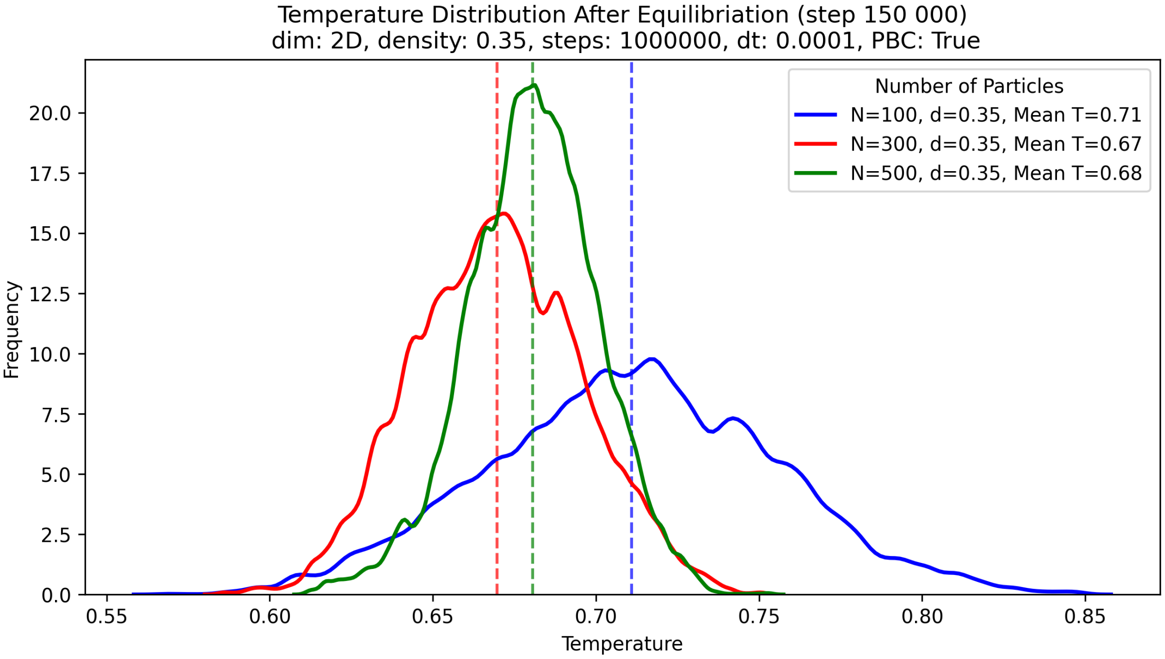
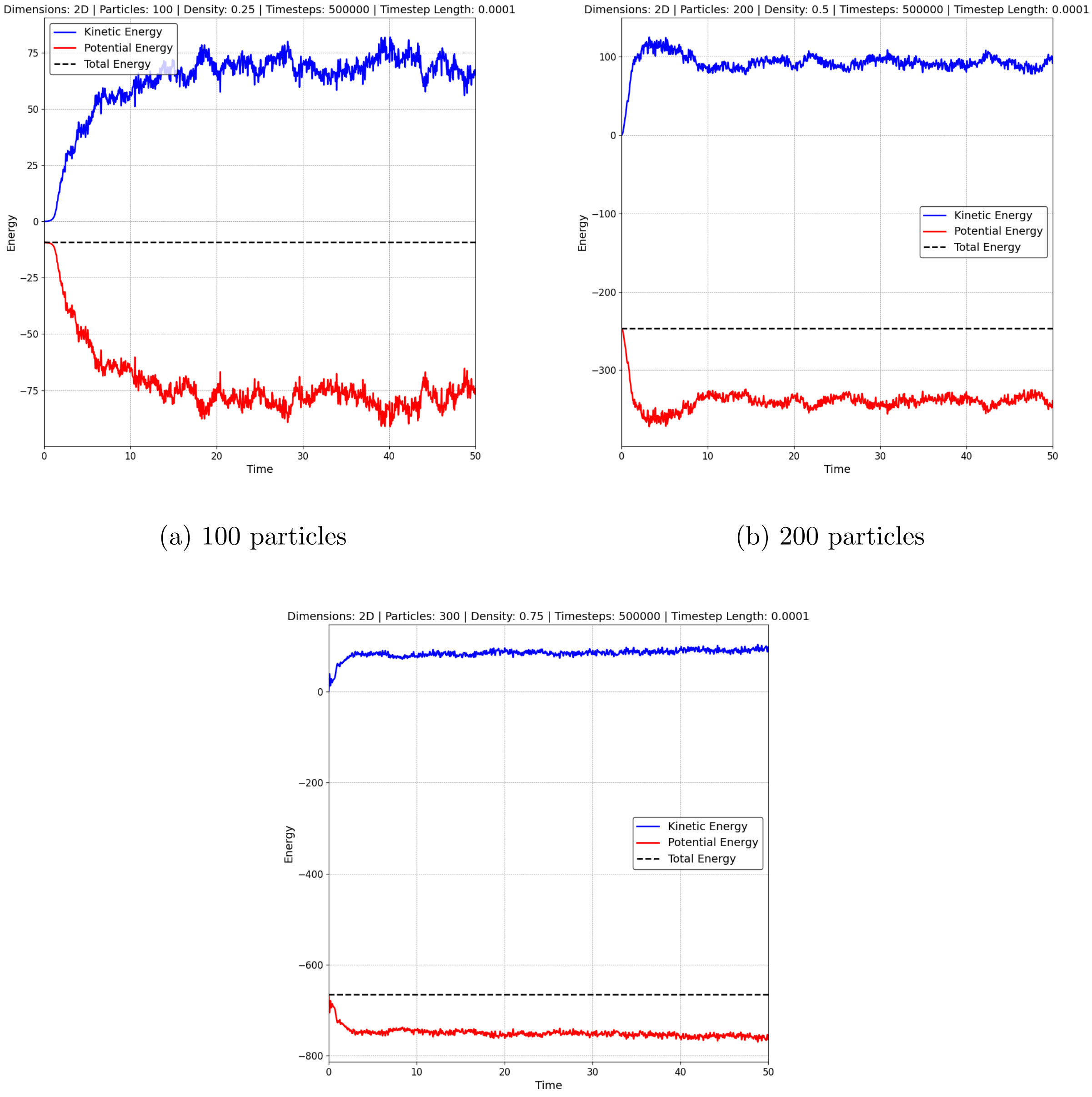


Figure 1: Temperature distribution of 3 different simulation runs.



(a) 100 particles

(b) 200 particles

(c) 300 particles

Figure 2: Three simulations of different numbers of particles at the same box size (densities were 0.25, 0.50, and 0.75 respectively) and 5×10^5 steps. The time step length was 0.0001.

3.2 Part 2

In this part, both the Langevin and Berendsen thermostats were implemented in the previous code by modifying the velocity Verlet algorithm, as explained in the theory part above. The thermostats are accessed from the `thermostats.py` module.

After making sure that our thermostats work, the stability of the simulation with different time step values was investigated. The results are in fig. 3. The simulation is stable until $\Delta t = 0.01$, which is two orders of magnitude larger than when we don't use a thermostat. When $\Delta t = 0.1$, the simulation fails.

Using $\Delta t = 0.01$, we performed simulations with thermostats set to the same temperatures observed after the energy equilibration in part 1, in fig. 1 as a benchmark for comparison. The results are in figs. 4 and 5. We can see that the temperatures are produced properly, as the mean temperatures are around 0.7 for both thermostats. That validates that our thermostats can keep the desired temperature.

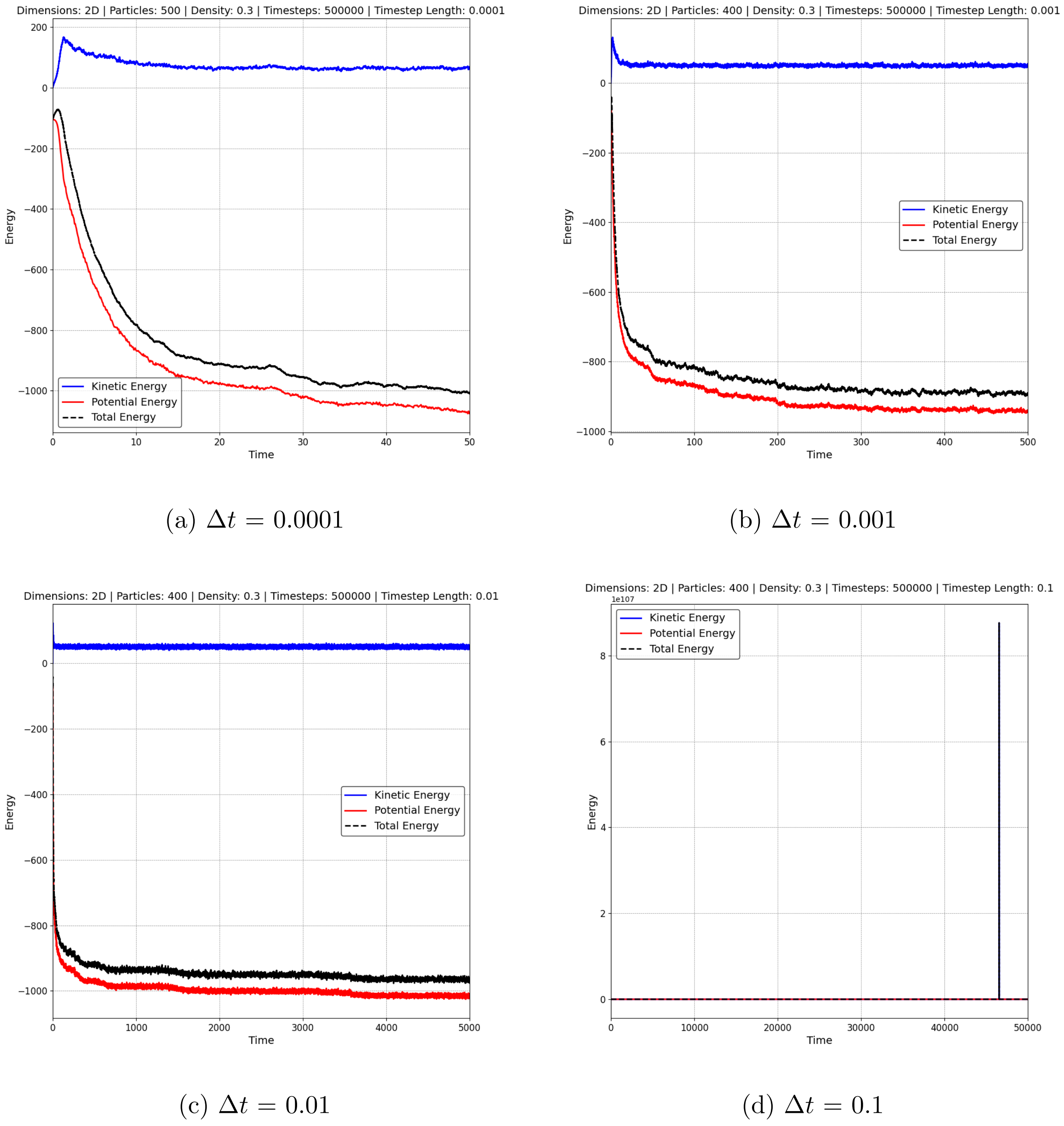
The Berendsen thermostat showed the wrong fluctuations indeed, and this is apparent from plotting the temperature vs. time in fig. 6. The fluctuation of the Langevin thermostat follows the proper dynamics, and the fluctuations have greater amplitudes compared to the Berendsen thermostat temperature fluctuations. In the case of Berenedsen, due to the rescaling of the velocities, the fluctuations are restricted, giving rise to incorrect dynamics. This difference can also be illustrated by plotting the velocity histograms of both as in fig. 7. The width of the Langevin thermostat histogram is more than the width of the Berendsen thermostat.

Next, we investigated the effect of varying the thermostat parameter τ in the Berendsen thermostat. The results are in fig. 8. Thermostat parameter τ represents how tight the coupling is between the thermostat and the system, and the smaller the value, the tighter the coupling and the stronger the reduction of the fluctuations. This effect becomes even more apparent when comparing the velocity distributions shown in fig. 9, where the simulation with a smaller τ exhibits a much narrower distribution than that with a larger τ.

The Langevin thermostat ζ parameter was also varied to investigate its effect on the simulation. The results are in figs. 10 and 11. We can see that if the value of ζ is too low, then the temperature will fluctuate greatly, and the system is not coupled to the thermostat as it should. We took this to the extreme, and we compared two extreme values of the ζ to investigate the effects. The results are in figs. 12 and 13. This result shows that if the values are too large, then the temperature of the system will not be maintained at the intended temperature, and the velocity distribution is greatly shifted toward higher velocities. This effect was not observed in the Berendsen thermostat, as only the width changed. This shows that using a proper value of the thermostat parameter is crucial to reproducing the proper dynamics and correct temperature.

Finally, the diffusion coefficient was calculated as a function of the simulation time and computer time for Langevin and Berendsen thermostats and when no thermostats are used with

500 particles and 0.35 density. The results are in figs. 14 to 16. The calculated diffusion coefficient was always larger when we used computer time versus when simulation steps were used by one order of magnitude. Among the methods tested, the Berendsen thermostat showed the highest diffusion coefficient when measured using both simulation and real computer time, indicating greater sampling efficiency. The diffusion coefficient for Langevin was also significantly higher than for a system without a thermostat, although it was around two times lower than the diffusion coefficient with the Berendsen thermostat. The diffusion coefficient program is in `diffusion_coef.py`.



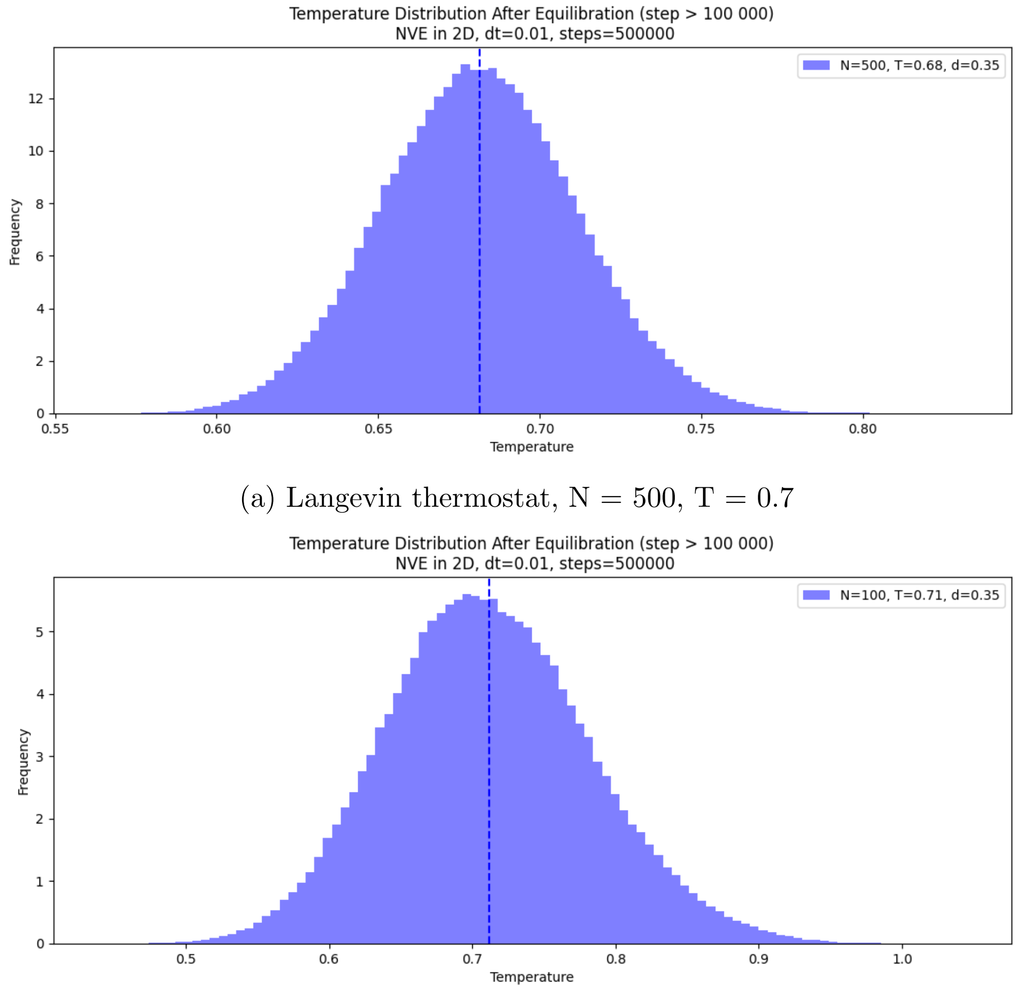
(a) $\Delta t = 0.0001$

(b) $\Delta t = 0.001$

(c) $\Delta t = 0.01$

(d) $\Delta t = 0.1$

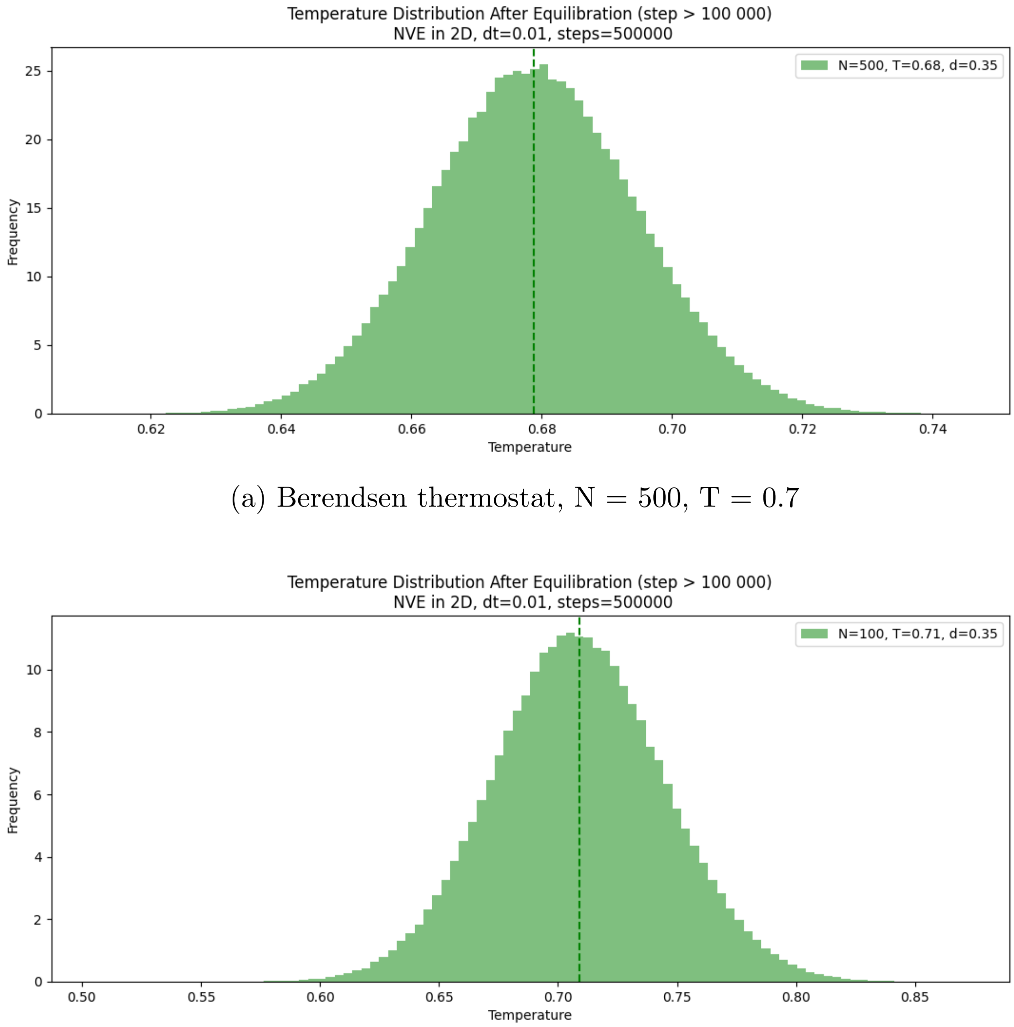
Figure 3: Four simulations with different Δt using the Langevin thermostat at temperature 0.5 show that simulation collapses at $\Delta t = 0.1$.



(a) Langevin thermostat, N = 500, T = 0.7

(b) Langevin thermostat, N = 100, T = 0.7

Figure 4: Two simulations were performed using the Langevin thermostat to verify whether the results from Part 1 are reproducible and to assess the thermostat's ability to maintain the target temperature. The desired temperature was set to 0.7. In the simulation with N=500, the mean temperature was 0.68, while for N=100, the mean temperature was 0.71.



(a) Berendsen thermostat, N = 500, T = 0.7

(b) Berendsen thermostat, N = 100, T = 0.7

Figure 5: Two simulations were performed using the Berendsen thermostat to verify whether the results from Part 1 are reproducible and to assess the thermostat's ability to maintain the target temperature. The desired temperature was set to 0.7. In the simulation with N=500, the mean temperature was 0.68, while for N=100, the mean temperature was 0.71.

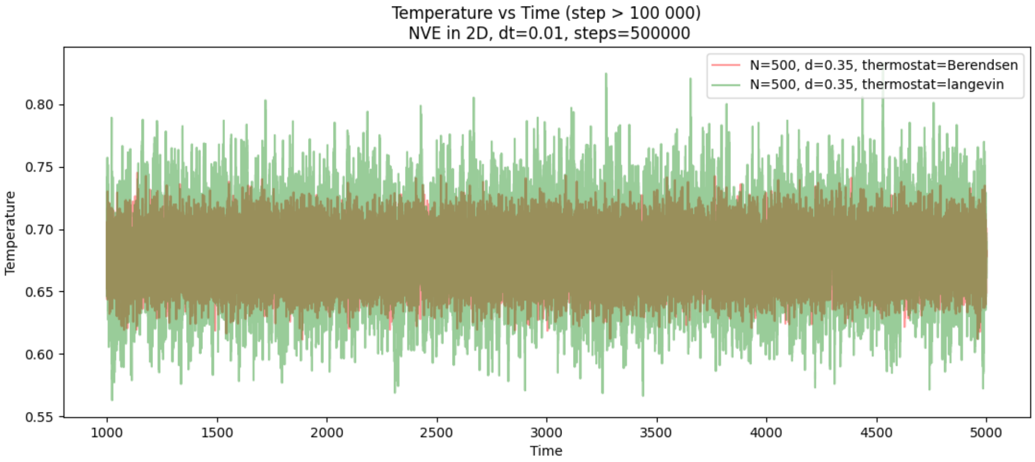


Figure 6: The temperature fluctuation of both Langevin and Berendsen thermostats. Langevin allows realistic fluctuations, while Berendsen overly damps them, leading to narrower distributions around the target temperature. The other greenish-beige color is for overlapping fluctuations.

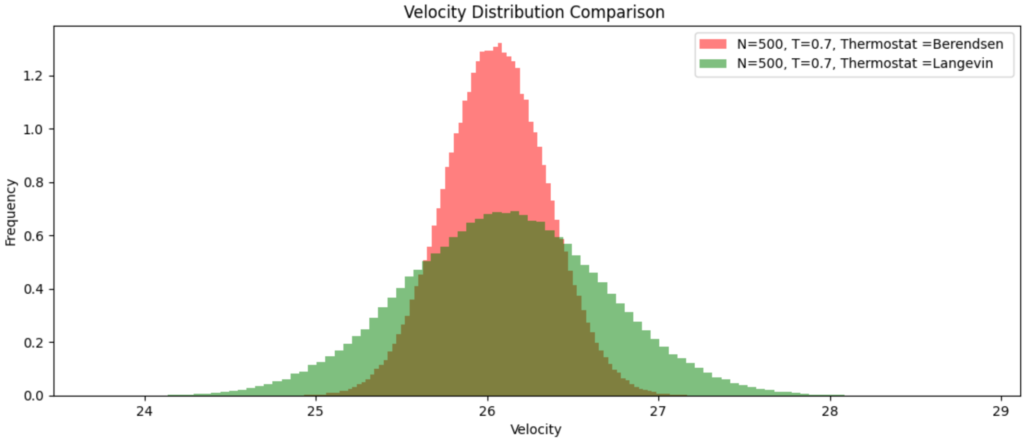


Figure 7: The comparison of the velocity distributions for both the Langevin and Berendsen thermostats shows that the distribution for Berendsen is significantly narrower, which is related to its stronger suppression of thermal fluctuations and its inability to correctly reproduce the canonical ensemble. The histograms for temperature and kinetic energy would show the same.

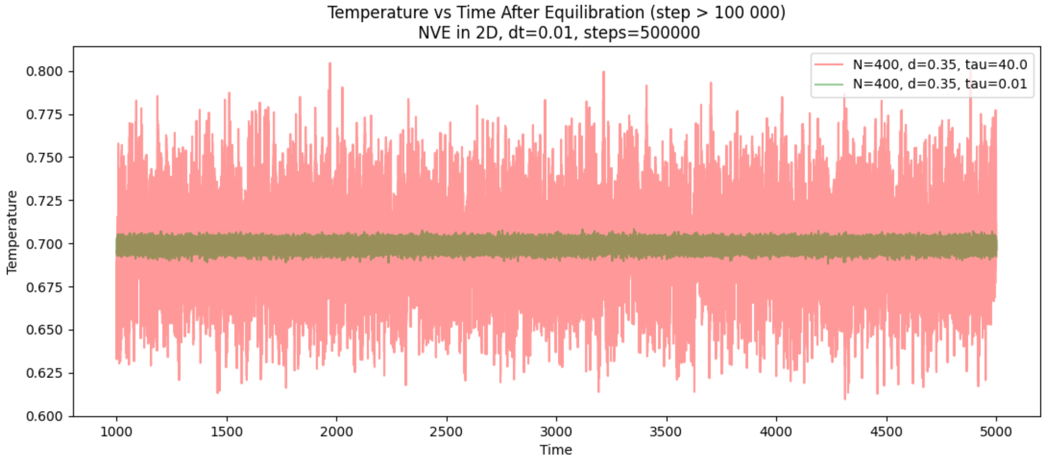


Figure 8: The effect of changing the thermostat parameter τ in Berendsen thermostats on the simulation. Smaller values of τ lead to reduced temperature fluctuations due to stronger coupling to the target temperature.

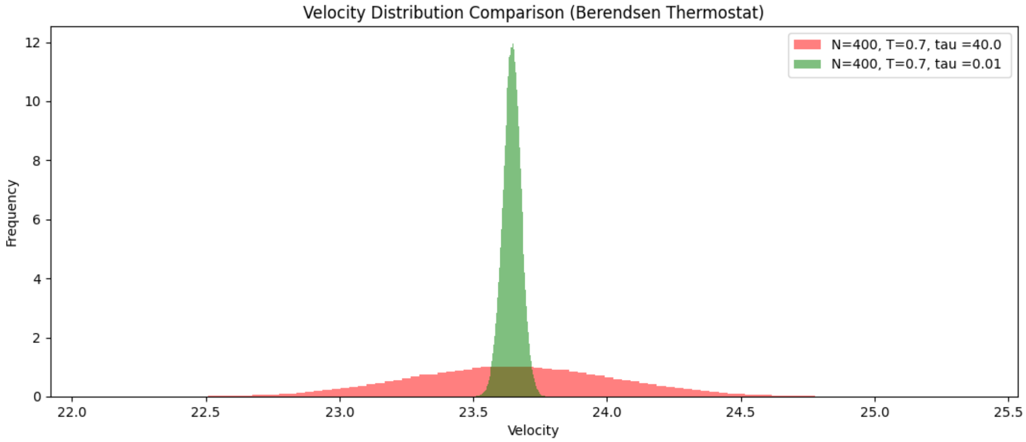


Figure 9: Velocity distribution of two simulations using the Berendsen thermostat with different τ values. The smaller the τ, the narrower the velocity distribution.

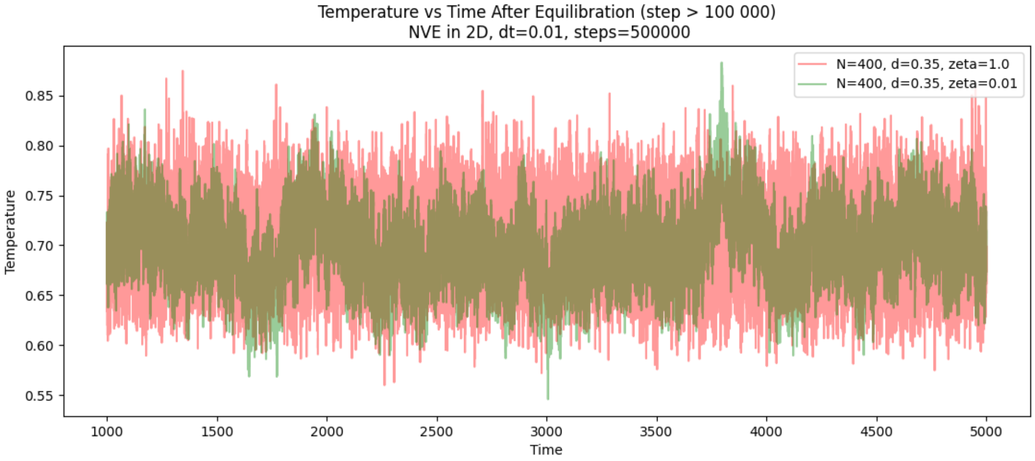


Figure 10: The effect of changing the thermostat parameter ζ in Langevin thermostats on the simulation. Smaller values of ζ lead to increased temperature fluctuations and less control of the simulation's temperature.

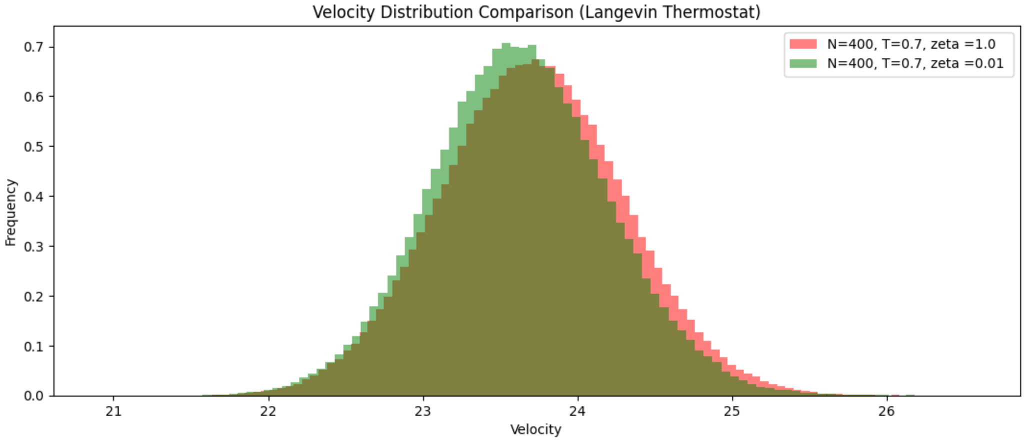


Figure 11: Velocity Distribution of two simulations using the Langevin thermostat with different ζ values. The larger the ζ, the more the histogram becomes shifted to a higher velocity with a slightly lower peak.

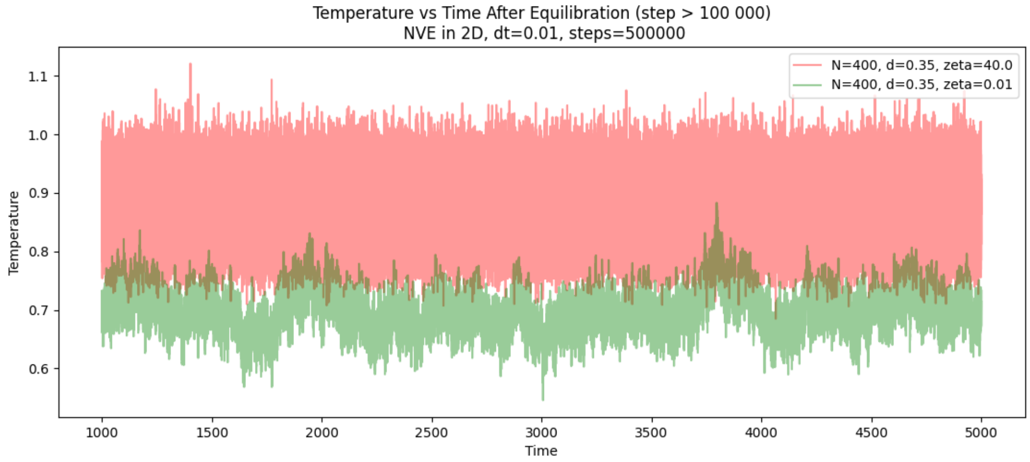


Figure 12: The effect of changing the thermostat parameter ζ in Langevin thermostats on the simulation. Too large values of ζ lead to higher temperatures than intended for the simulation.

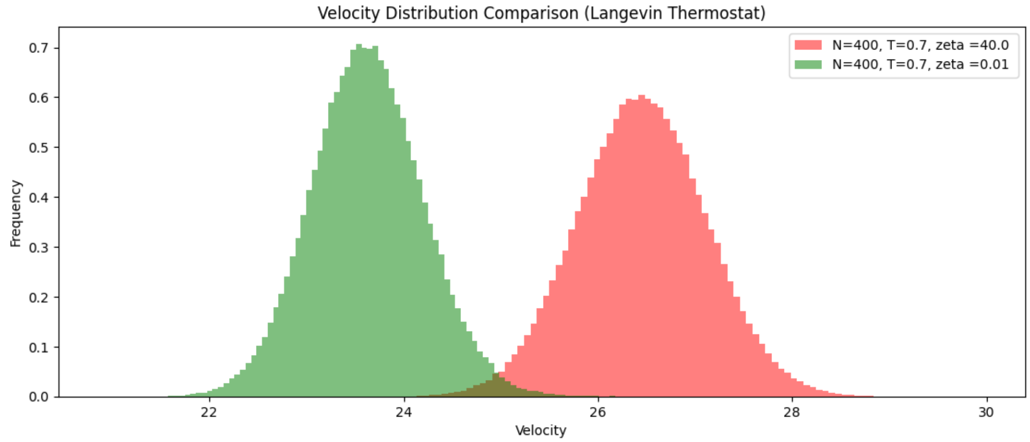


Figure 13: Velocity Distribution of two simulations using the Langevin thermostat with different ζ values. The larger the ζ, the more the histogram becomes shifted to a higher velocity with a lower peak.

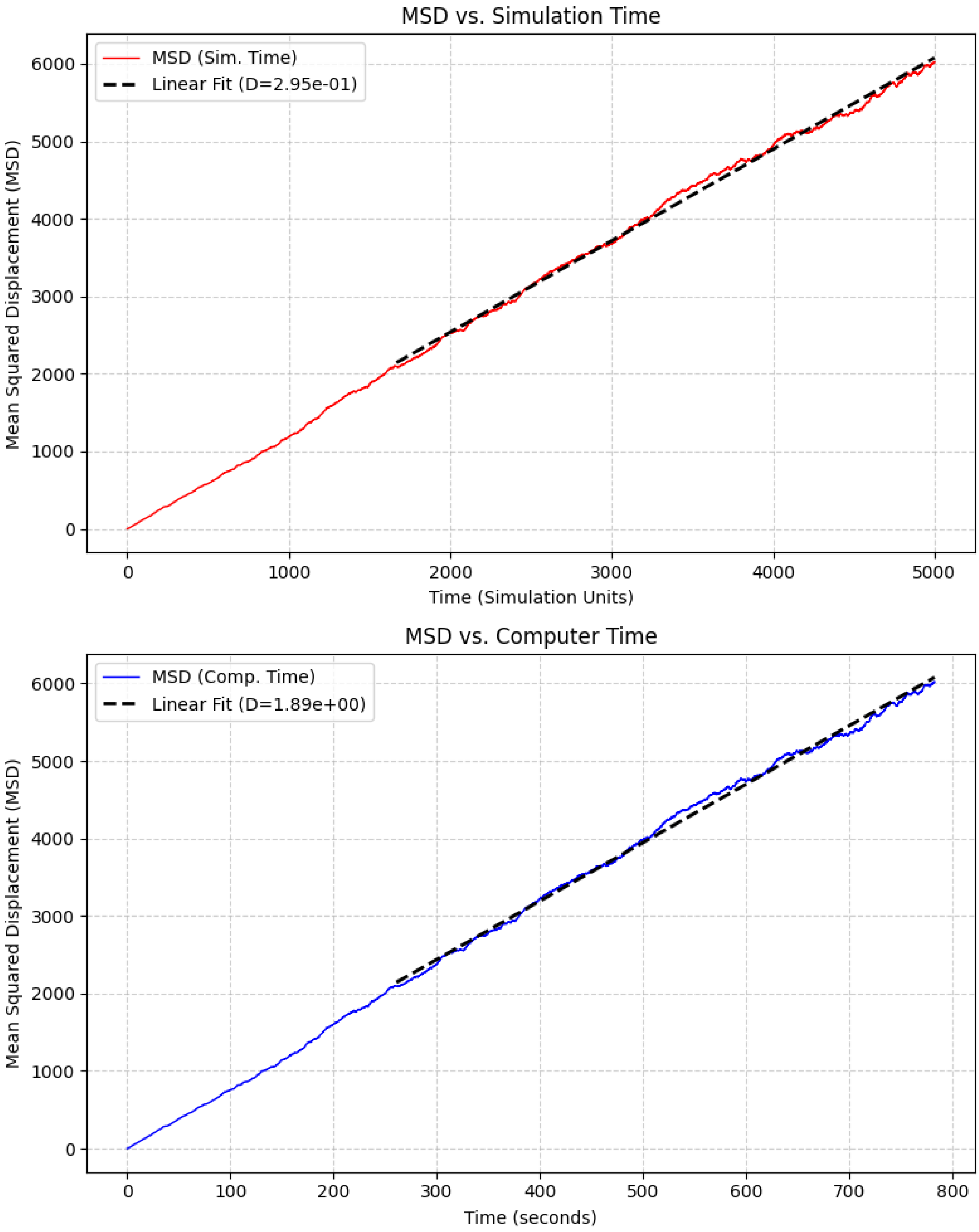


Figure 14: The diffusion coefficient as a function of the simulation time and the computer time in a simulation of $N = 500$ and using the Langevin thermostat. In simulation time units, $D = 2.95 \times 10^{-1}$, and in computer time units, $D = 1.89$.

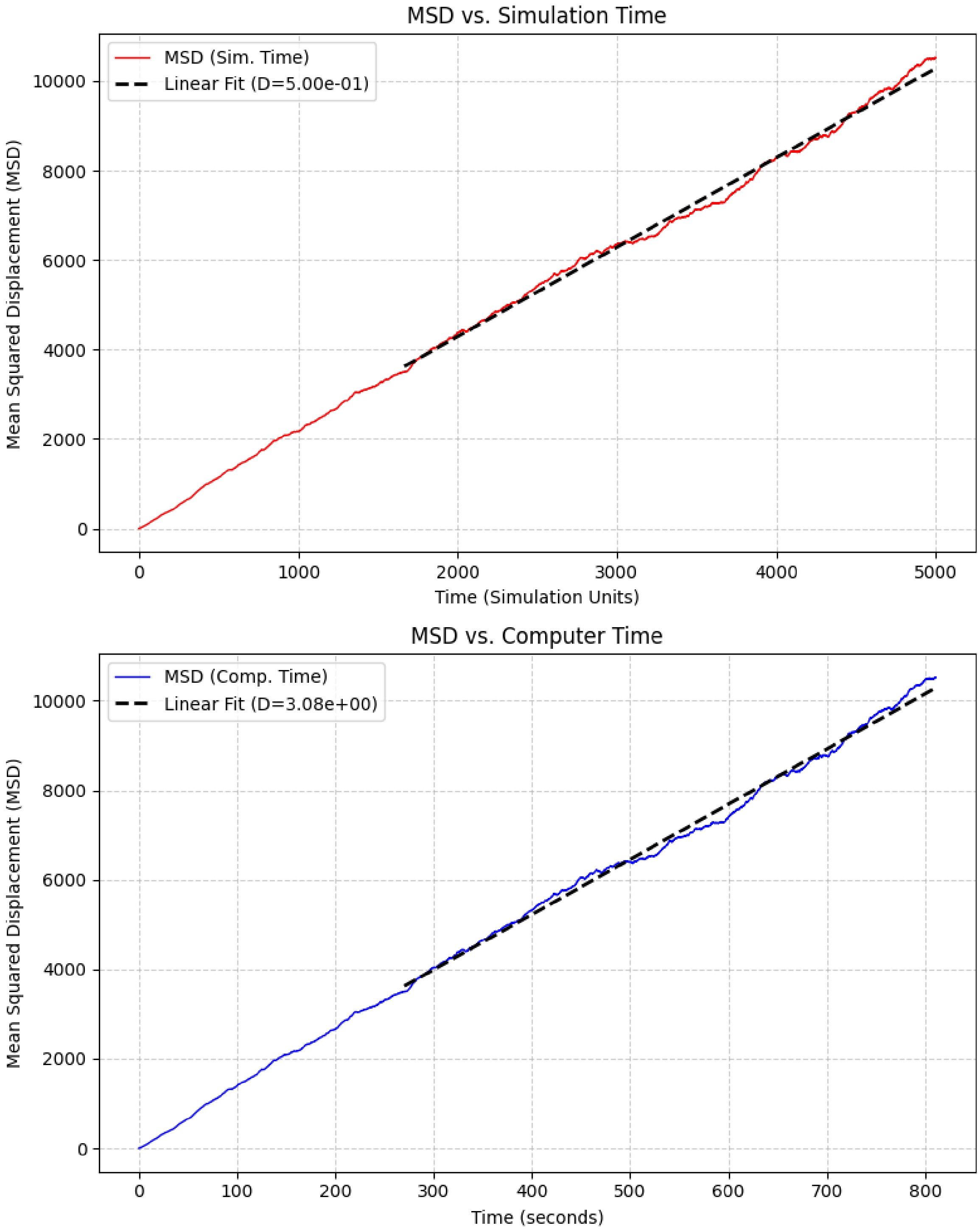


Figure 15: The diffusion coefficient as a function of the simulation time and the computer time in a simulation of $N = 500$ and using the Berendsen thermostat. In simulation time units, $D = 5.00 \times 10^{-1}$, and in computer time units, $D = 3.08$.

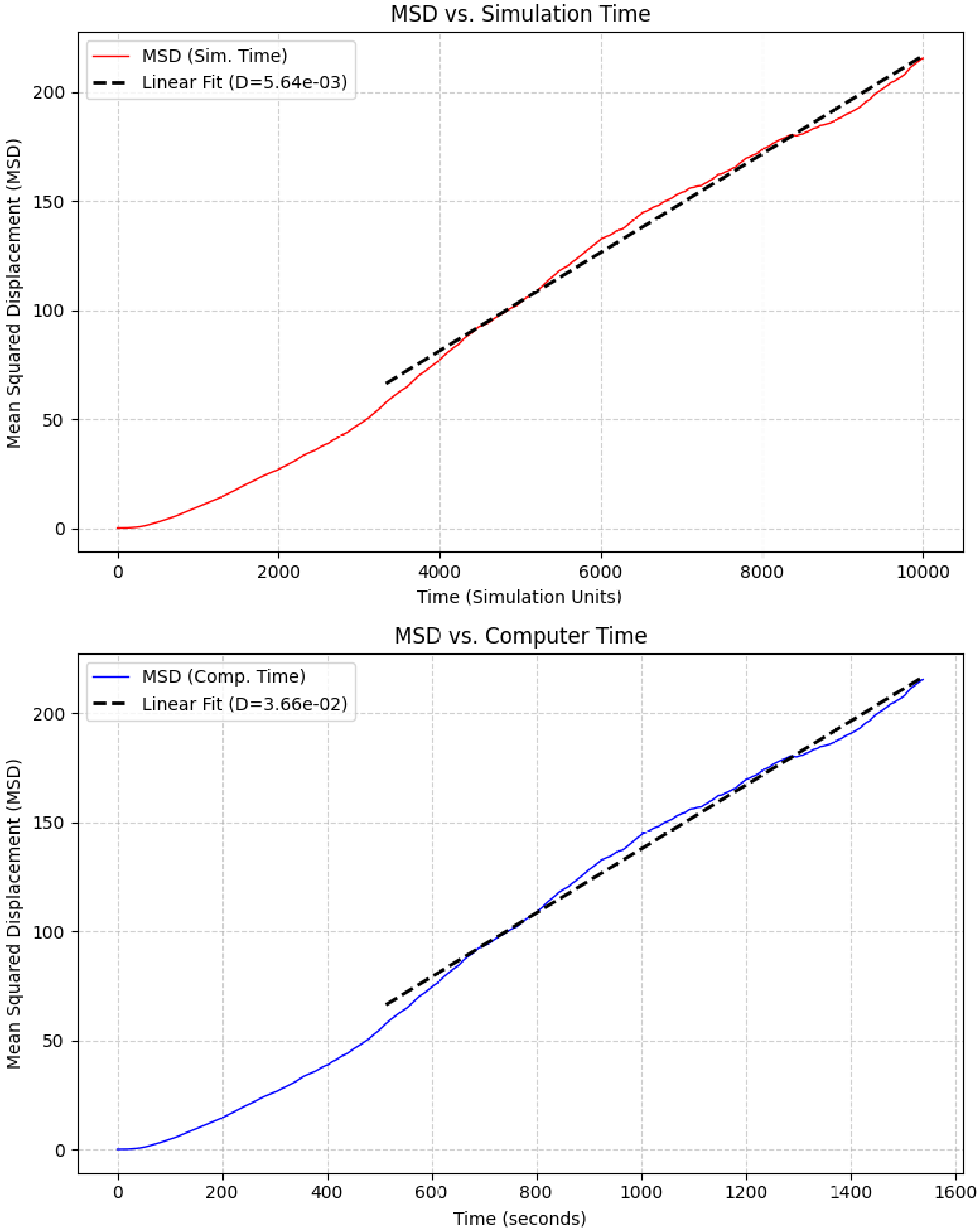


Figure 16: The diffusion coefficient as a function of the simulation time and the computer time in a simulation of $N = 500$ and no thermostat. In simulation time units, $D = 5.64 \times 10^{-3}$, and in computer time units, $D = 3.66 \times 10^{-2}$.

3.3 Part 3

In Part 3, we ran MD simulations using the Langevin thermostat across various temperature and density combinations. We analyzed the resulting systems both visually, using OVITO, and quantitatively, using the radial distribution function (RDF). The RDF program is in `RDF.py`.

For the suitable density and temperature combinations, we explored the phase diagram by Li et al. [1]. In the article they used systems with $N = 32^2 - 512^2$, we sticked to the $N = 32^2$. We were able to detect three different phases with our MD model: fluid gas-like, fluid liquid-like, hexatic, and solid phase. Our results were qualitatively comparable to those of Li et al.

The results for a fluid-liquid system are shown in fig. 17. In the OVITO snapshot, particles form loose clusters but still move freely due to lower density. The RDF shows a prominent first peak that decays fast, with subsequent peaks that do not vanish to zero. The simulation video is in the GitHub /videos folder: `liquid.mp4`.

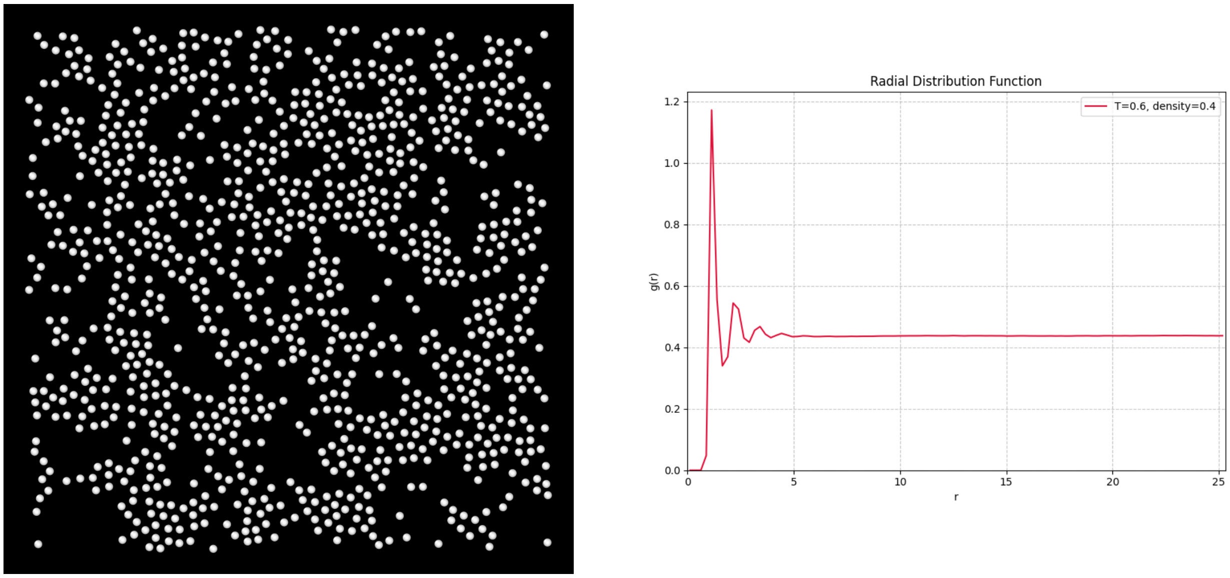


Figure 17: Liquid phase at T=0.6 and density=0.4. RDF is averaged over 50,000 frames after equilibration.

In fig. 18, the configuration is dense and solid-like, where particles remain near their lattice positions with minimal thermal motion. The RDF shows sharp, regularly spaced peaks. While the first peaks decay to zero, the later peaks maintain a higher baseline. The simulation video can be found in the GitHub /videos folder: `solid.mp4`.

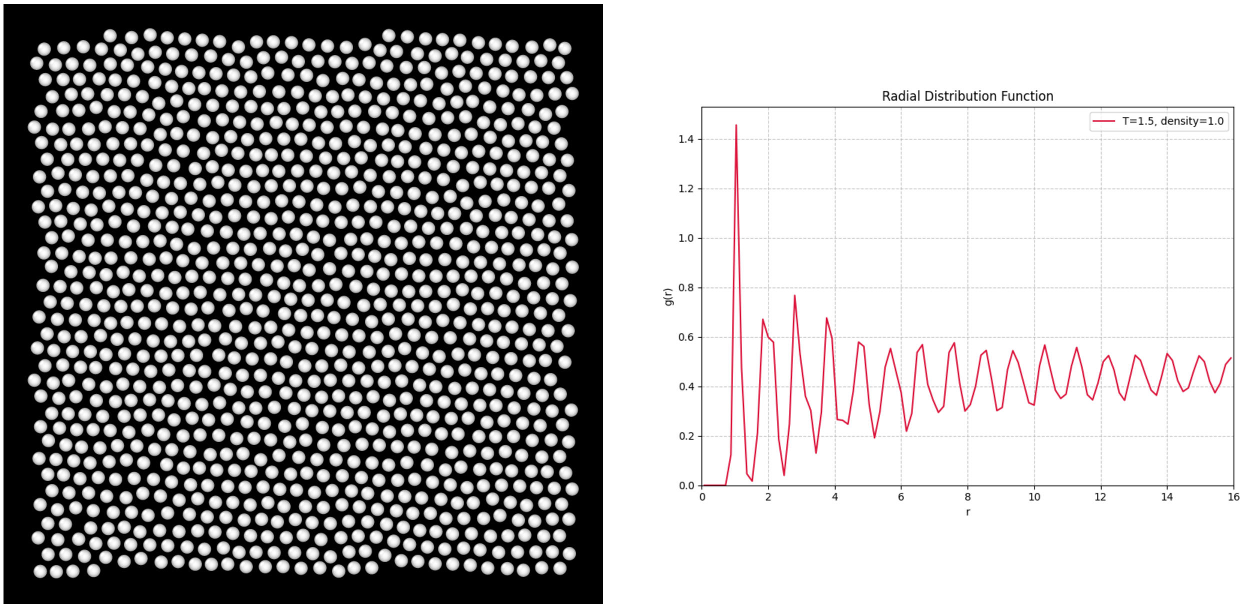


Figure 18: Solid phase at T=1.5 and density=1.0. RDF is averaged over 10,000 frames after equilibration.

As shown in fig. 19, particles in the gas-like phase are sparsely distributed and highly mobile. Occasionally, they form brief contacts but quickly separate. The RDF shows a single dominant peak followed by no significant structure. The simulation video is in the GitHub /videos folder: `gas.mp4`.

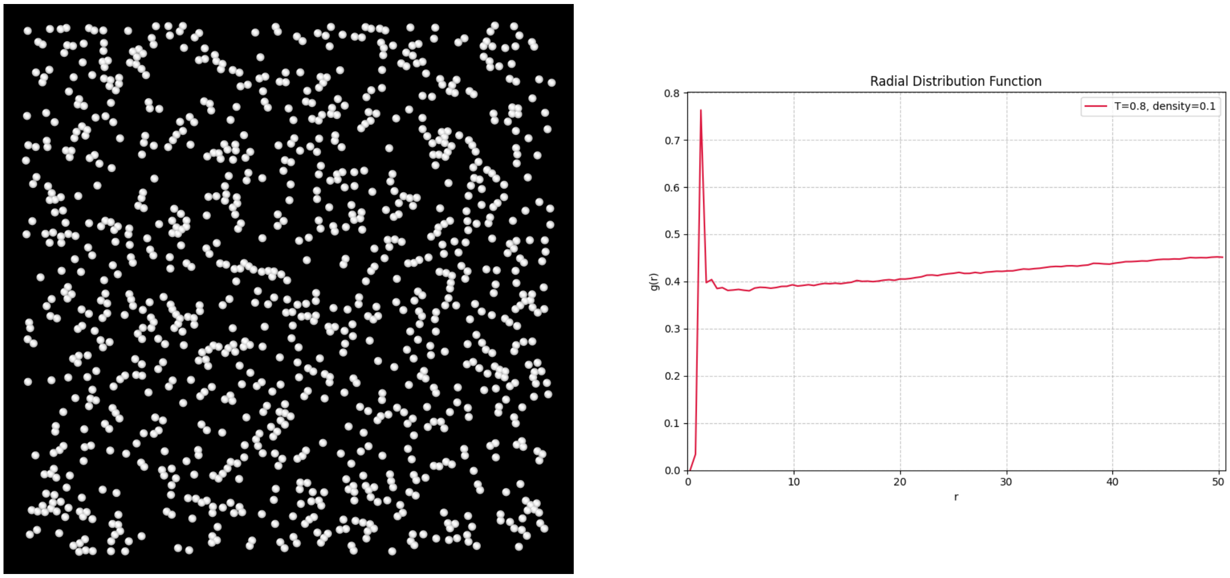


Figure 19: Gas phase at T=0.8 and density=0.1. RDF is averaged over 10,000 frames.

In fig. 20, the hexatic phase shows a structured local order, with small voids interrupting the lattice. The RDF contains well-defined peaks but with a higher baseline than the solid and lower than the liquid phase, indicating intermediate order. However, visual observations may not be sufficient to identify the hexatic phase, and an analysis of the bond order parameter would be useful to confirm the presence of six-fold local ordering. The simulation video is in the GitHub /videos folder: hexatic.mp4.

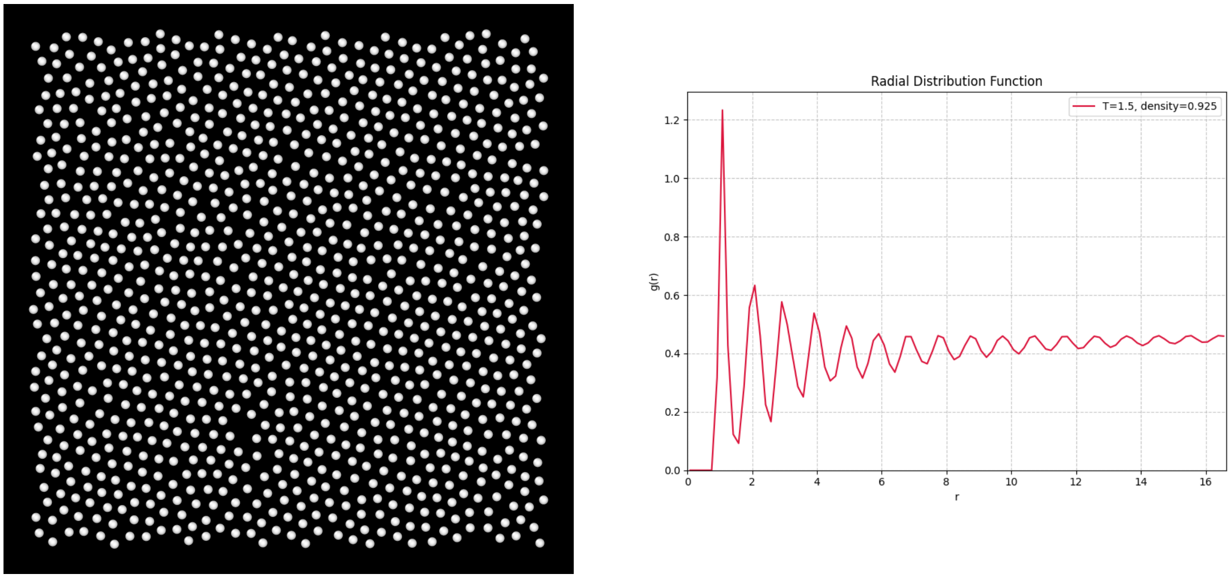


Figure 20: Possibly hexatic phase at T=1.5 and density=0.925. RDF is averaged over 10,000 frames.

The hexatic phase is a characteristic feature of 2D systems. During melting, dislocation pairs (defects in the crystal lattice) first separate and move freely, breaking positional order while maintaining orientational order—this intermediate state is known as the hexatic phase. As the temperature increases further, disclination pairs unbind, leading to the loss of orientational order and a transition to the liquid phase. [2]

All in all, simulations are only as accurate as the models and parameters they rely on. Phase diagrams generated through MD or MC simulations across different temperature–pressure conditions can reflect real physical behavior, but one must consider their limitations. It is also essential to ensure proper convergence and allow the system sufficient time to reach equilibrium for meaningful results.

3.4 Part 4

In this part, we performed Monte Carlo (MC) simulations (link: `monte_carlo.py`) and compared the effects of various step sizes. The energy autocorrelation function (ACF) and the integrated autocorrelation time (IACT) were used as the basis for comparison. We performed three simulations, each having 400 particles and 0.3 density and temperature 0.9, with 3 for different maximum displacements. The results are in figs. 21 to 23. The results show that when the time step is small, the autocorrelation function remains high, the steps are not independent of each other, and the integrated correlation time is extremely high, meaning the number of independent samples in the entire simulation is very low. The number of independent samples can be given as follows:

$$\text{no. of independent samples} = \frac{\text{no. of steps}}{2 \cdot \tau} \tag{19}$$

Where τ is the integrated correlation time. In addition, the acceptance ratio of the proposed moves assisted us in evaluating the MC simulation. For a step of 0.01, we got an acceptance ratio of 0.997, proving that the steps are way too small, and for displacement equivalent to 1.2, we got 0.299 as an acceptance ratio, meaning that most proposed moves were rejected. A good acceptance ratio is usually between 0.4 to 0.6, and for a step size of 0.3, we got 0.591.

Next, using the Monte Carlo algorithm, we performed a series of simulations to reproduce the results of the paper [1]. We took four points from figure 7 of the paper as the following, corresponding to different regions of the plot:

- $\rho = 0.9$ and $T = 0.4$: Solid Region
- $\rho = 0.94$ and $T = 1.52$: Hexatic Region
- $\rho = 0.45$ and $T = 0.55$: Fluid region (more like liquid)
- $\rho = 0.1$ and $T = 0.8$: Fluid Region (more like gas)

We could successfully reproduce these results, as shown in figs. 24 to 27.
The results of the simulation have reproduced the data published in the paper. The hexatic phase was the most difficult to produce, and the results in fig. 25 are the best-achieved outcome.

We did not observe significant differences in the outcome when comparing the results of MD and MC. RDFs from MC simulations look neater and stable.

The MC methods provided a much more efficient way to compute the properties we sought out of the simulation, such as the pair correlation function, as all the MC simulations ranged from 150 to 222 seconds for 10^6 steps and 500 particles. However, the fastest simulation we performed with 500 particles was 782 seconds with Langevin thermostat and 811 seconds with Berendsen and 1539 seconds with no thermostat. This shows that understanding the data required to answer a question or solve a problem can be cleverly and efficiently acquired if we know what we are looking for. Sampling the equilibrium configuration and inquiring for static properties are done more efficiently by the Monte Carlo method.

To have a valid comparison, we ran a series of simulations, fixing other variables but the particle number to properly compare the different methods. Both MC and MD with or without thermostat were tested for particle numbers $N = \{100, 200, 300, 400, 500\}$ and the results are in fig. 28. These results prove the superiority of the MC algorithm when it comes to computational time.

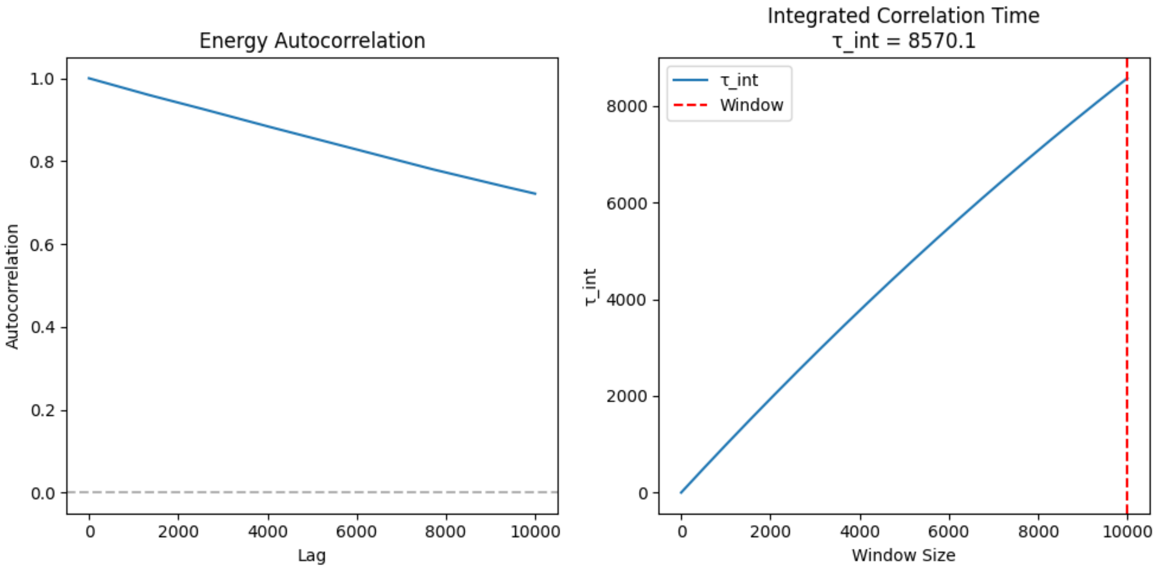


Figure 21: Monte Carlo simulation with a step size 0.01.

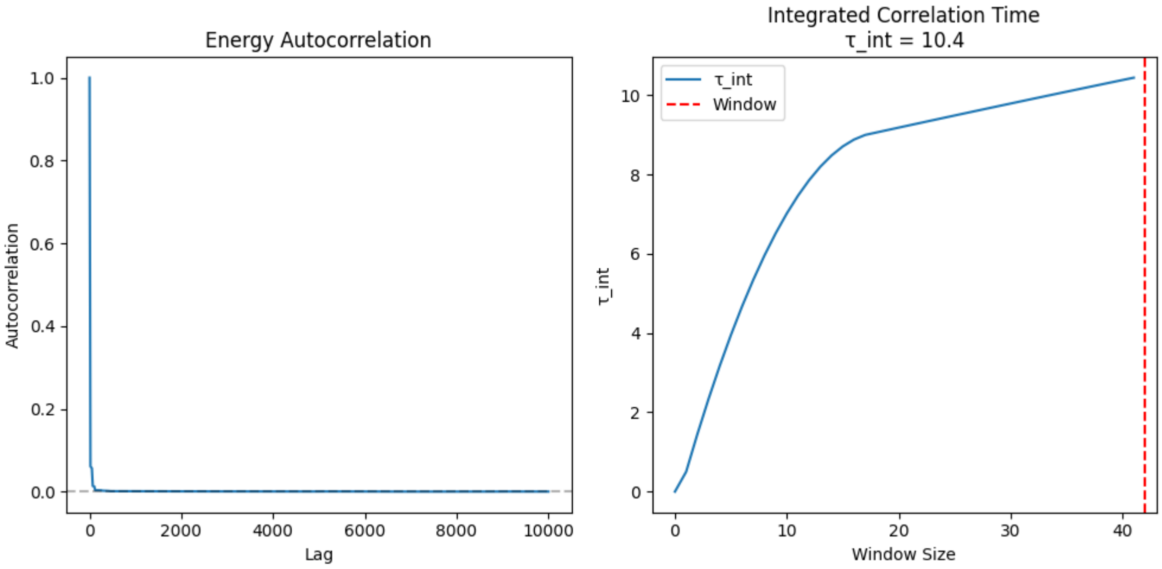


Figure 22: Monte Carlo simulation with a step size 0.3.

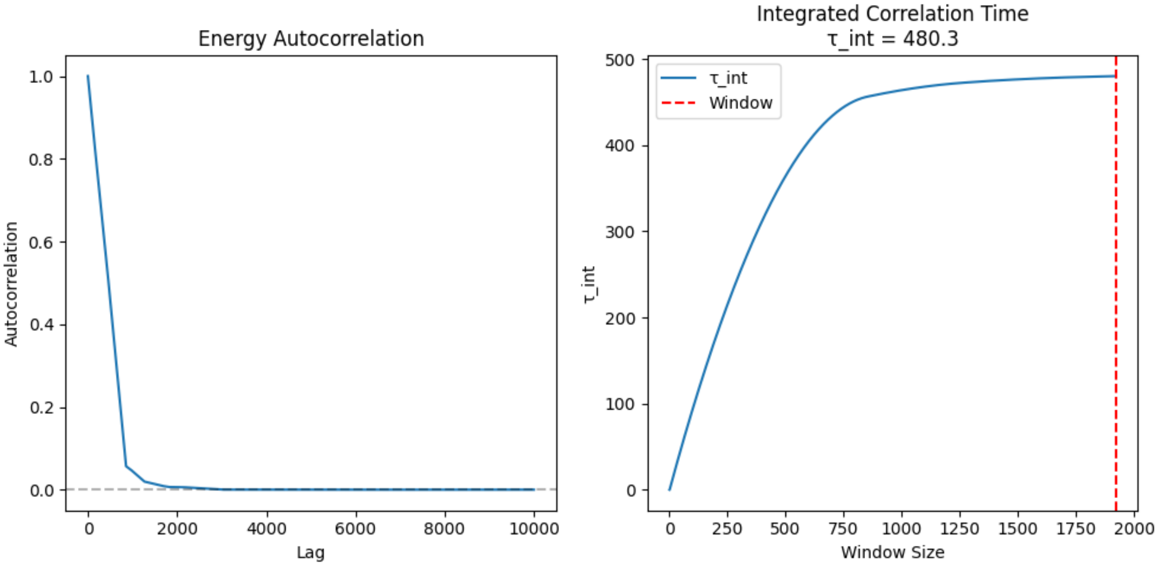
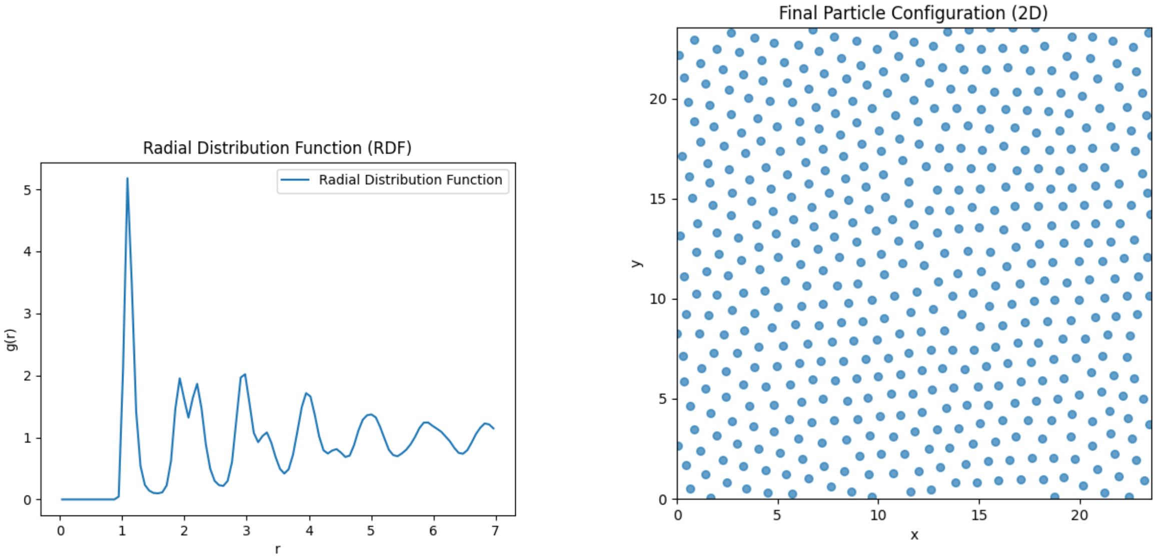


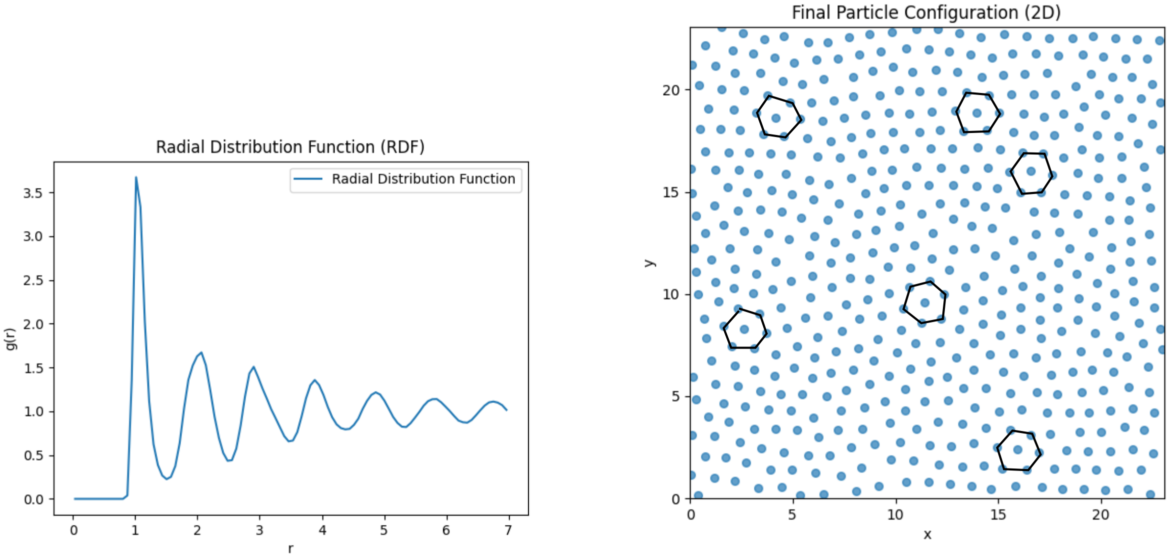
Figure 23: Monte Carlo simulation with a step size 1.2.



(a) Radial Distribution Function

(b) Final configuration of the system

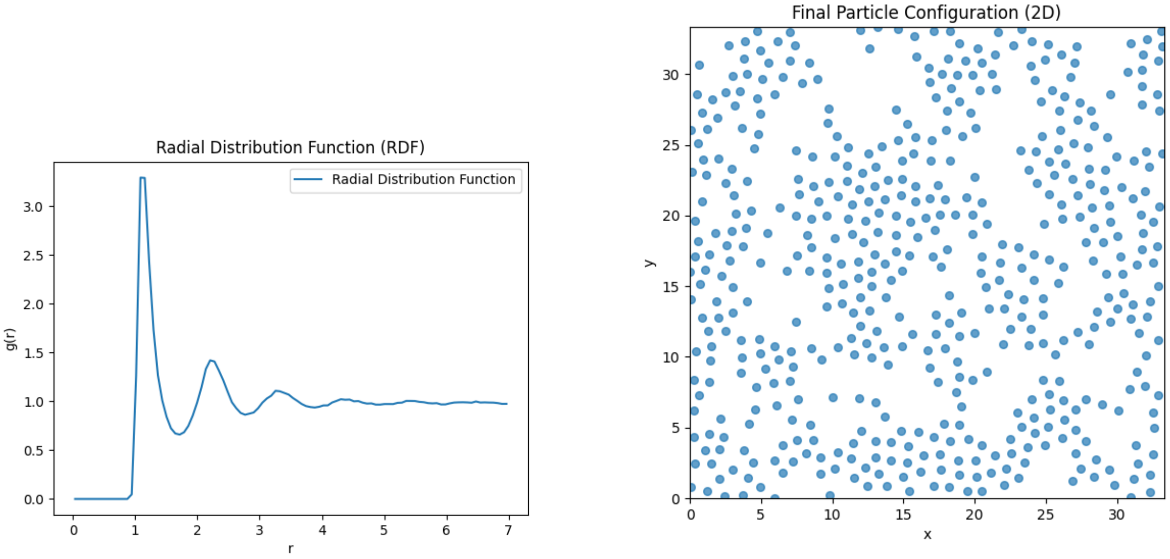
Figure 24: The results for MC simulation at $\rho = 0.9$ and $T = 0.4$. This corresponds to the solid region.



(a) Radial Distribution Function

(b) Final configuration of the system

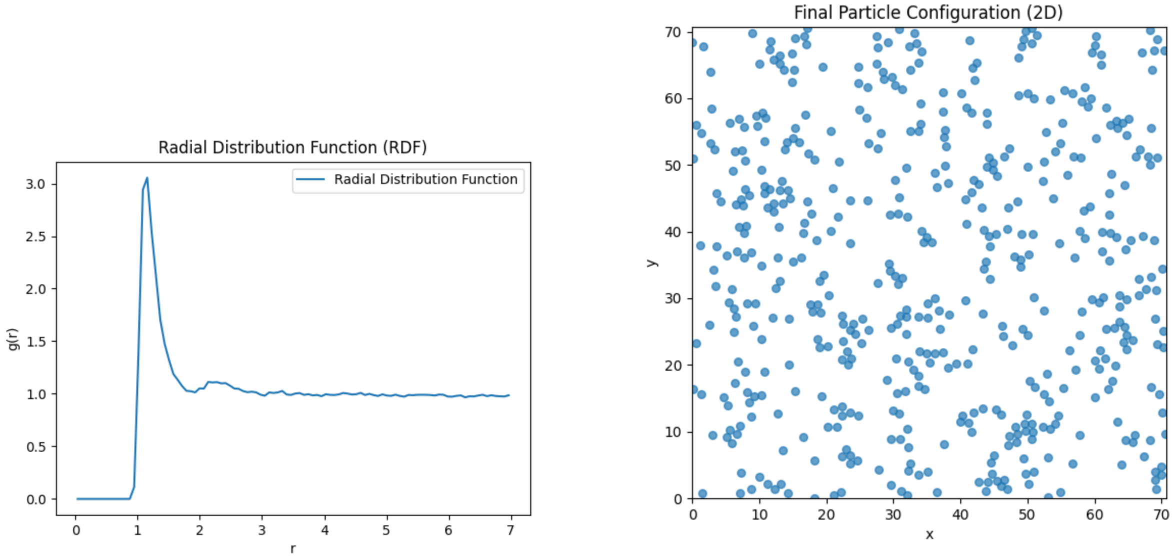
Figure 25: The results for MC simulation at $\rho = 0.94$ and $T = 1.52$. This corresponds to the hexatic region. Some hexagons were highlighted for better visualization, but more can be found.



(a) Radial Distribution Function

(b) Final configuration of the system

Figure 26: The results for MC simulation at $\rho = 0.45$ and $T = 0.55$. This corresponds to the fluid region with a liquid-like character.



(a) Radial Distribution Function

(b) Final configuration of the system

Figure 27: The results for MC simulation at $\rho = 0.1$ and $T = 0.8$. This corresponds to the fluid region with a gas-like character.

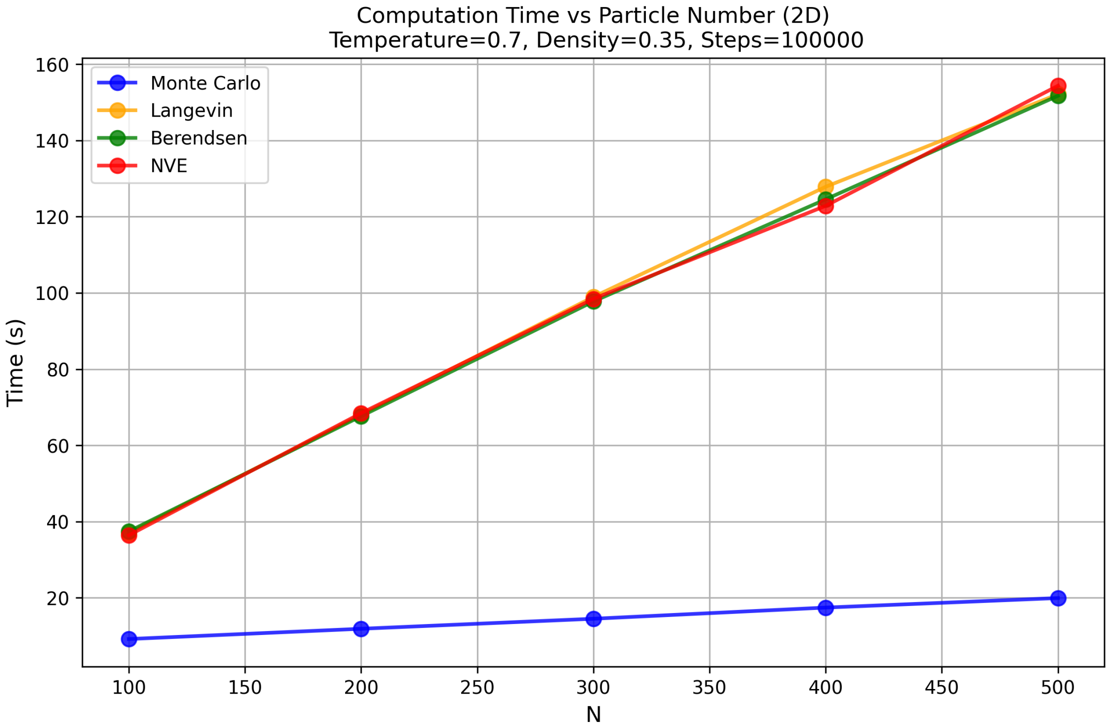


Figure 28: Comparison between the simulation time of MC and MD with and without thermostats. All the MD had a step size of 0.0001, and the MC had a maximum displacement of 0.3.

References

[1] Y.-W. Li and M. P. Ciamarra, “Phase behavior of lennard-jones particles in two dimensions,” *Phys. Rev. E*, vol. 102, p. 062 101, 6 Dec. 2020. DOI: 10.1103/PhysRevE.102.062101. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.102.062101.

[2] W. F. Brinkman, D. S. Fisher, and D. E. Moncton, “Melting of two-dimensional solids,” *Science*, vol. 217, no. 4561, pp. 693–700, 1982. DOI: 10.1126/science.217.4561.693. eprint: https://www.science.org/doi/pdf/10.1126/science.217.4561.693. [Online]. Available: https://www.science.org/doi/abs/10.1126/science.217.4561.693.

[3] A. Bunker, “Molecular modelling,” *Lecture at University of Helsinki (KEM 342)*, 2025.

[4] A. Bunker, “Programming projects in molecular modelling,” *Lecture at University of Helsinki (KEM 381)*, 2025.