Delft University of Technology

Computational Physics

AP3082

Molecular Dynamics Simulation of Argon Atoms

Authors:

Kimmy Zhao (5709229)

Christina Sfetsou (5718384)

Rita Villar Parajó (5501148)

March 31, 2023

Abstract

The behavior of argon gas particles in a molecular dynamic system that interact under a Lennard - Jones potential has been examined for different values of temperature and densities. By applying the rescaling so the system to be in an equillibrium, the thermodynamic quantities of pressure and specific heat have been calculated.

Keywords: molecular dynamics, simulation, energy, rescailing, equilibrium

Introduction

In this report we will be revising the construction and utilization of a simulation of a molecular system. As a first approach to this field, we will be working with argon atoms: as a noble gas of not too large size, its (classical) self-interactions can be modeled via a quite simple model. In this case, we can disregard complex phenomena like charge transfer, dipole-dipole or even quadripole-quadripole interactions, and polarization.

The thermodynamic properties of the argon system will be extracted from the simulation on the basis of a series of assumptions and impositions to the system. In the appropriate limit $N \longrightarrow \infty$, the expected values of intensive properties (pressure and specific heat, in our case) can be retrieved from a microcanonical ensemble when we assume it to be ergodic[1] [3], which implies that the ensemble averages (the quantities that would interest us) equal the time averages.

These are only assumptions that we make to justify our simulation, but there are also some aspects to impose on the system. To warrant a microcanonical ensemble, we must have N (total number of particles), V (volume of the simulation) and E (total energy of the system) fixed. The first two requirements are rather trivial, but numerically integrating the equations motion of a complex system in such a way that total energy is conserved requires care.

When these conditions are fulfilled and assumptions are solid, the expectation values of thermodynamic quantities according to (1), where A describes the thermodynamic property to observe, and $\mathbf{Q}^N(t)$ and $\mathbf{P}^N(t)$ are the degrees of freedom of our system, namely coordinates and momenta. Our system is, then, fully characterized by the total energy, momentum, number of particles and volume.

$$\langle A \rangle_t = \frac{1}{T} \int_{t_0}^{t_0+T} A(\mathbf{Q}^N(t), \mathbf{P}^N(t))\, dt \tag{1}$$

Additionally, other considerations must be regarded, namely the thermalization of the system and the conditions at which equilibrium is reached, since they might be different from those imposed at the system's initialization.

In the following sections we will present how pressure and specific heat (as examples of intensive properties) can be analyzed from molecular dynamics simulations and to what extent the results of this first approach are valid approximations of the physical behavior of argon or not.

Physical Considerations

As mentioned previously, argon can be modeled sufficiently precisely by only considering the simple pairwise interactions described by the Lennard-Jones potential, r being the distance between the two particles:

$$U_{LJ}(r) = 4(r^{-12} - r^{-6}) \qquad (\epsilon) \tag{2}$$

Then, the only force that will impact the time-evolution of the system is that of (3), which shows the total energy exerted over one particle and in which r_{ij} refers to the distance between the chosen pair of particles.

$$\mathbf{F}(\mathbf{x}_i) = -\sum_{j \neq i} \frac{dU_{LJ}}{dr_{ij}} \frac{\mathbf{x}_i - \mathbf{x}_j}{r_{ij}} \tag{3}$$

This information is completely provided to the code, whose primary task is to numerically integrate the equation of motion (4). This is not a trivial task when having to maintain the total energy of the system constant, which we will discuss in following sections.

$$\frac{d^2\mathbf{x}_i}{dt^2} = \mathbf{F}(\mathbf{x}_i). \tag{4}$$

Since the goal of molecular dynamics simulations is to obtain information about physical systems, we will be using the information to study the changes in pressure and specific heat under different conditions. For this, from [2], and based on the virial theorem, the pressure is calculated from (5) and the specific heat from (6).

$$\begin{aligned} \frac{p}{\rho kT} = 1 - \frac{1}{6NkT} \left\langle \sum_i \sum_{j>i} \mathbf{r}_{ij} \frac{\partial U_{LJ\,j}}{\partial \mathbf{r}_{ij}} \right\rangle \\ - \frac{\rho}{6kT} \int_{r_v}^{\infty} \mathbf{r} \frac{\partial U_{LJ}}{\partial \mathbf{r}} g(\mathbf{r}) d\mathbf{r} \end{aligned} \tag{5}$$

where $g(\mathbf{r})$ refers to the pair correlation function and r_v refers to a cutoff distance for U_{LJ}. In our computations, this last term is not used, which will be subject of an upcoming discussion.

$$\frac{\langle \delta K^2 \rangle}{\langle K \rangle^2} = \frac{2}{3N} \left(1 - \frac{3}{2c_V}\right) \tag{6}$$

[1] An ergodic system is that in which all states are equally probable to be reached, so over sufficiently long periods of time the ensemble average will equal the time average of the system.

Methods

Unless specified otherwise, all the expressions and data presented will be in their dimensionless form. For argon:

magnitude	dimensionless unit	equivalence
length	σ	$3.405 \cdot 10^{-10}$ m
energy	ϵ	$1.654 \cdot 10^{-21}$ J
temperature	ϵ/k_B	119.8 K
time	$\sqrt{m\sigma^2/\epsilon}$	$2.15 \cdot 10^{-12}$ s
mass	m	$6.6 \cdot 10^{-26}$ kg

Table 1: Equivalence of our system's dimensionless units to SI units.

We will set up two systems (108 particles and 256 particles) under the same conditions in a box of $L = 10$ (σ).

We will be working with a somewhat mall system embedded in a box with side $L = 10$ (σ) and a fixed amount of particles, but its study as an isolated system is not of great interest. Instead, with the right periodic boundary conditions, the system can be understood as a small part of a bulk material. Our box will then interact with its nearest neighbors, which are copies of itself. However, to ensure that the number of interacting particles is kept constant (and the microcanonical ensemble respected), a rule for what interactions are and are not allowed must be imposed: the minimal image convention. A particle will only interact with every other particle in the box or with whichever of its copies (in the neighboring boxes) is closest to the first particle.

Further detail on the implementation of all these conditions is discussed in the following section.

Code Structure

In this section we will discuss the more sophisticated aspects of our complex system, leaving out those rather trivial, as well as correctness checks for the main steps.

Periodic Boundary Conditons

The particle's trajectory can be ensured to stay confined to the box making use of the % function in Python. This ensures that all of the particle's coordinates remain within $[0, L]$ and re-enters from the opposite side if it goes beyond these limits.

```
"""
L: box side
pos_array: array with the particles' positions

"""
pos_array = pos_array % L
```

Minimal Image Convention

The more delicate task to take care of which particles interact is delegated to a function that calculates the relative positions of the closest image of each particle, relative to every other particle. For each coordinate:

```
"""
x2 : x coordinate of particle 2
x1 : x coordinate of particle 1

"""
 rel_x = ( x2 - x1 + L/2) % L - L/2
```

where the addition of $L/2$ ensures that $x1 - x2$ stays in an appropriate range for % to operate correctly, and then this is correcting by subtracting the same amount.

Numerical Integration of the Equation of Motion

There exist several algorithms to perform the pertinent numerical integration, of which we hereby present results for Euler (7) and Verlet (8) algorithms.

$$\begin{aligned} x_i(t_{n+1}) &= x_i(t_n) + v_i(t_n)h \\ v_i(t_{n+1}) &= v_i(t_n) + F(x_i(t_n))h, \end{aligned} \tag{7}$$

The equations represent the time evolution of the position and velocity of the particles, where h is the time step. The numerical integration algorithms are limited in their accuracy and require us to strike an optimal balance between speed and precision, while preserving the requirements for the simulation.

$$\begin{aligned} x_i(t_{n+1}) &= x_i(t_n) + hv_i(t_n) + \frac{h^2}{2}F(x_i(t_n)) \\ v_i(t_{n+1}) &= v_i(t_n) + \frac{h}{2}[F(x_i(t_{n+1})) + F(x_i(t_n))] \end{aligned} \tag{8}$$

Force

The periodic boundary conditions in molecular dynamics simulations create an effectively larger system than initially considered, where interactions between particles are not necessarily limited to those inside the initial system, but also include all particles in the surrounding copies.

As aforementioned, this is taken care of in a special function that returns an array with only the (relative) positions of the closest image of each particle with respect to every other particle. Therefore, one can just pass this as an argument to use according to (3).

As shown in figure 1, the potential and force behave as expected. It also provides us with some insights into the nature and range of the interaction.

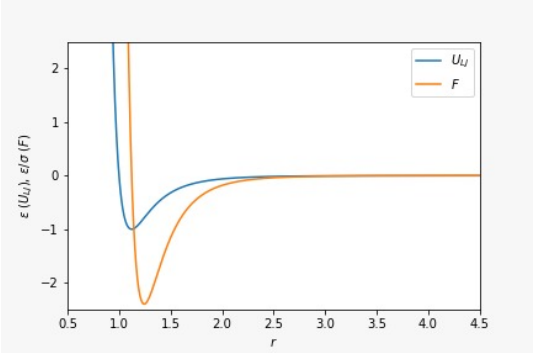


Figure 1: Force and potential between two particles, against their distance r_{ij}

Initial Conditions

Density and temperature are crucial parameters for accurate modeling of the system behavior. High densities result in more collisions and interactions between particles, increasing pressure and potential for phase transitions, while low densities may cause particles to behave more like free particles.

On the other hand, temperature affects the atoms' kinetic energy, hence also impacting the system's macroscopic properties.

Setting up the Particle System on an FCC Lattice

The particles are initialized in a faced-centered-cubic lattice, which is then copied in each direction (with a spacing equal to the lattice constant) as many times as necessary to achieve the desired N in our system. The lattice parameter in the cell depends on the desired density like $a = \sqrt[3]{4/\rho}$, and the result can be seen in figure 2.

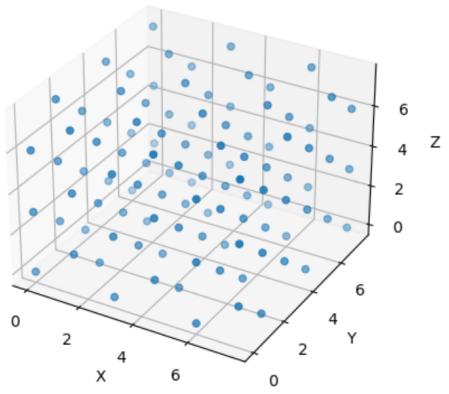
Maxwell-Boltzmann Distribution of Thermal Velocity

Since our system is not isolated at temperature $T = 0$K, a Maxwell-Boltzmann distribution is used to ensure that initial velocities conform to the thermal velocities at temperature T. This involves selecting the v_x, v_y, and v_z coordinates for each particle at random from a Gaussian distribution.

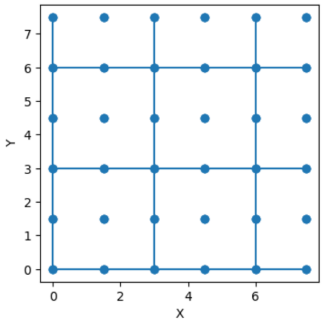
The velocities were normalized and centered at zero, but still corrected by subtracting the center of mass velocity (given to the finitude of our system, even when setting the mean to 0 there can be some net velocity in the working sample).

Table 2 outlines the Input parameters and calculated Values for fcc cell simulation.

To verify the correctness of our Gaussian velocity function, a histogram of the velocity distribution per dimension for 108 particles is shown in figure 3 and fitted to $e^{\frac{-v_{x,y,z}^2}{2T}}$.



(a) 3D Plot.



(b) Layout.

Figure 2: 3D plot of 108 particles in fcc configuration in a box with side length $L = 6$ and lattice constant $a = 3$, defined by the length L over the number of cells

density (ρ)	**number of fcc cells (n)**	**temperature (T)**
input	input	input
lattice constant (l)	**length (L)**	**number of particles (N)**
$\sqrt[3]{4/\rho}$	$n \cdot l$	$4 \cdot n^3$

Table 2: Input Parameters and Calculated Values for FCC Cell Simulation.

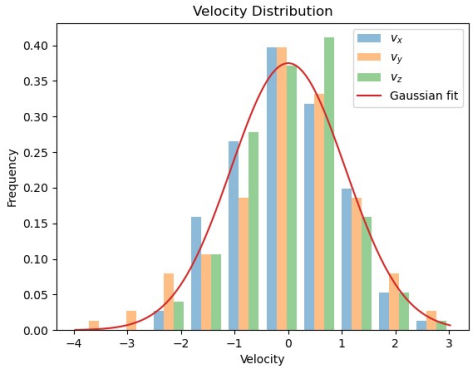


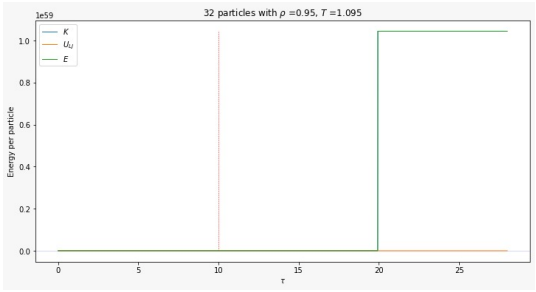
Figure 3: Velocity Distribution in Three Directions for 108 Particles.

Correctness Checks

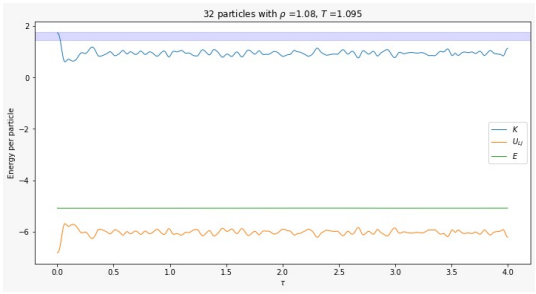
Conservation of Energy

The standard Euler method for integrating ordinary differential equations is not ideal for molecular dynamics simulations, as

it does not effectively conserve the total energy of the system, as can be seen in figure 4(a) (without rescaling). This method assumes that the forces acting on the atoms are constant over the time step, which is not necessarily true, and as a result, the computed trajectories can drift away from the true trajectory over time, leading to inaccuracies in the simulation. As a result, we chose to implement the Verlet algorithm (8).



(a) Euler Algorithm



(b) Verlet Algorithm

Figure 4: The energies of a system with 32 particles and 1000 steps for the two different algorithms without rescaling

Reaching Equilibrium and Rescaling

Upon initialization of the simulation, one can see (as in 4) that the kinetic tends to descend quite rapidly and stabilize at a lower value. This is expected behavior and points to a well-functioning system, since it corresponds to thermalization.

It is not allowed in our simulation, nonetheless, since the choice of initial velocity is given by the temperature at which the system is supposed to be studied. Therefore, rescaling (which can be understood as inputting energy into the system) must be applied. A condition for equilibrium at the desired temperature can be imposed and used to stop rescaling, but for simplicity in our simulation we decided to follow Thijssen's recommendations [1] and rescale every 20 timesteps (for $h = 0.004$ (τ) until $\tau = 10$ (τ). The rescaling factor is calculated by

$$\lambda = \sqrt{\frac{E_{kin}^2}{\sum_i v_i^2}}, \qquad E_{\text{kin}} = \frac{3}{2}(N-1)T$$

where E_{kin} is the theoretical kinetic energy. The time step at which rescaling occurs is recorded so only the data at equilibrium at the correct temperature are taken into account in future calculations.

A big part of our work has been dedicated to achieving the desired rescaling and equilibrium. Figure 5 shows the plot of evolution of the kinetic energy, potential and the total energy of the system over the time τ. The plot shows that the kinetic energy of the system oscillates around this value after the system reaches equilibrium.

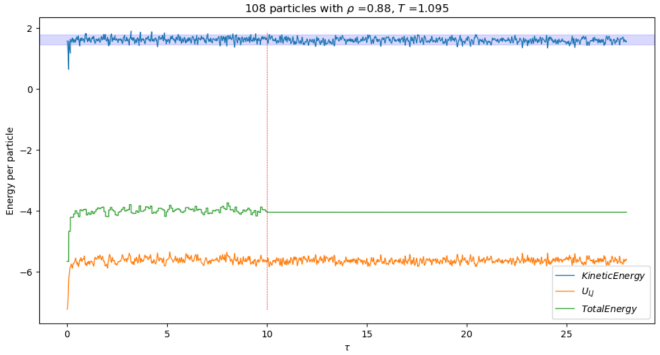


Figure 5: Energy per particle over the time with rescaling. The dashed line represents the last rescaling, and the blue band shows the target kinetic energy (for the desired T.

Estimating errors

In order to approximate the error for the thermodynamic quantities (energy, pressure, heat capacity), we used the block bootstrap method. This method relies on the concept of data autocorrelation: since our data at any timestep is calculated from the data in the previous timestep, the values are of course correlated and cannot be treated with the statistics of normal, experimental data.

However, after a certain time (the correlation time), we can treat the data points after no longer correlated. To achieve this, the simulation data is divided into blocks of a certain amount of datapoints (b), which should be at least equal to the correlation time. All the values in each block are replaced by the block's average, and we can randomly pick a random sample (of the size of the original data) from them. In this case, we will have an observable obtained from the same amount of datapoints as we simulated for, only no longer autocorrelated.

For a dataset A divided into blocks a, we will have an error like:

$$\sigma_A(b) = \sqrt{\frac{1}{N_b - 1}\left(\langle a^2\rangle - \langle a\rangle^2\right)}$$

When plotting the error against the size of the data blocks b, we can pinpoint the correlation time as the b at which it plateaus. From figure 6 we concluded that $b = 100$ was a good estimation, without condensing our data into too few points. In our case, the block bootstrap was applied to the data passed

to the observables. For pressure, this was the su term in (5) and for the specific heat this was only the kinetic energy, both magnitudes per every timestep.

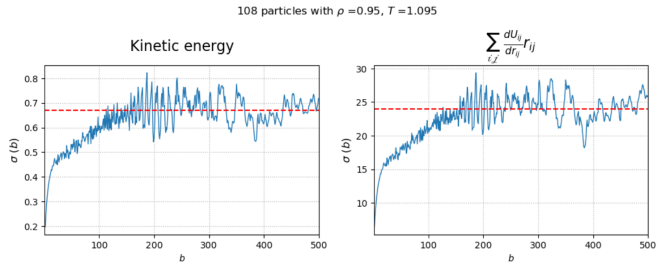


Figure 6: Left: Error of the kinetic energy over time after reaches equilibrium over the block size. Right: Error of the sum term of the pressure formula as obtained from the virial theorem over the block size.

Surprisingly, only a slight variation in the error was found when checking over different number of iterations. We chose to iterate the random picking 100 times since it was a minimum for both cases and it did not seem to segment the data too much.

Results

Starting the experiment

Our experiment run 36 minutes for 40 different data sets of (ρ, T) in a system with 108 particles and 4.32 hours for the same data in a system with 256 particles.

The main goal of the computation is the study of the thermodynamic quantities in the equilibrium: the pressure and the specific heat. In the following section we cite the results and present a comparison with the results in papers [2] and [3].

Pressure

The results we obtained for the pressure have been gathered in Table 3. Figure 7 shows the evolution of pressure for some isochores at densities $\rho = 0.88, 0.85, 0.75, 0.65$ and 0.45 and the comparison with the same simulation in [2].

Overall, our simulation results are consistent with widely accepted simulations results ([2]) and demonstrate good agreement with physical tendencies in molecular dynamic systems. Specifically, we observe that high density and high temperature result in more collisions and hence more pressure, which is consistent with known behavior. However, we did observe some negative pressure values in our simulation. While this is not physically meaningful in a molecular dynamic system, it is possible that these values were generated due to the method used to calculate force. We also observed that at high densities and low temperatures the deviation for the values in [2] is big. An explanation for that is we neglected the correction term in the pressure formula 5 for any condition. However, the correction term is relevant at high density and low temperature, when can be evaluated at around 1 ([2]). This can be the root of our more dissonant results.

ρ	T	(1) P	(2) P	(3) P
0.88	1.095	3.48	3.555 ± 0.0036	3.7013 ± 0.0025
0.88	0.94	2.72	2.5686 ± 0.006	2.901 ± 0.0022
0.88	0.591	-0.18	-2.6875 ± 0.0051	-3.5561 ± 0.0027
0.85	2.889	4.36	4.4747 ± 0.0023	4.3548 ± 0.0016
0.85	2.202	4.2	4.5283 ± 0.0026	4.2525 ± 0.0018
0.85	1.214	3.06	3.3733 ± 0.0037	3.2728 ± 0.0022
0.85	1.128	2.78	3.1445 ± 0.0042	2.9289 ± 0.0026
0.85	0.88	1.64	1.9035 ± 0.004	1.6957 ± 0.002
0.85	0.782	0.98	1.7002 ± 0.0044	1.1465 ± 0.0028
0.85	0.786	0.99	1.6017 ± 0.004	1.3234 ± 0.0031
0.85	0.76	0.78	1.3276 ± 0.0044	0.8318 ± 0.003
0.85	0.719	0.36	0.7719 ± 0.0049	0.7462 ± 0.0031
0.85	0.658	-0.2	-2.0676 ± 0.0051	-2.6156 ± 0.0034
0.85	0.591	-1.2	-3.4111 ± 0.0044	-3.956 ± 0.0037
0.75	2.849	3.1	3.3257 ± 0.003	3.0371 ± 0.0013
0.75	1.304	1.61	2.2141 ± 0.0042	1.7485 ± 0.0021
0.75	1.069	0.9	1.1534 ± 0.0033	0.9761 ± 0.002
0.75	1.071	0.89	1.087 ± 0.0034	0.8557 ± 0.0027
0.75	0.881	-0.12	0.2374 ± 0.0037	-0.0661 ± 0.0027
0.75	0.827	-0.54	-0.2495 ± 0.0046	-0.389 ± 0.0033
0.65	2.557	2.14	2.2308 ± 0.0028	2.2585 ± 0.0015
0.65	1.585	1.25	1.5914 ± 0.0033	1.3286 ± 0.002
0.65	1.036	-0.11	0.1747 ± 0.0048	-0.0451 ± 0.003
0.65	0.9	-0.74	-0.4983 ± 0.0054	-0.7312 ± 0.0031
0.55	2.645	1.63	1.663 ± 0.002	1.6191 ± 0.0017
0.543	3.26	1.86	1.8353 ± 0.0023	1.8212 ± 0.0017
0.543	1.404	0.57	0.5443 ± 0.0027	0.542 ± 0.0021
0.543	1.326	0.42	0.481 ± 0.0039	0.4597 ± 0.0022
0.5	1.36	3.4	0.4773 ± 0.0043	0.4137 ± 0.002
0.45	4.625	1.68	1.6873 ± 0.0016	1.6808 ± 0.0012
0.45	2.935	1.38	1.4047 ± 0.0028	1.3919 ± 0.0014
0.45	1.744	0.74	0.8596 ± 0.0027	0.8162 ± 0.0019
0.45	1.764	0.76	0.892 ± 0.0027	0.8006 ± 0.0017
0.45	1.71	0.74	0.9237 ± 0.0034	0.7613 ± 0.002
0.45	1.552	0.75	0.6975 ± 0.0037	0.5767 ± 0.002
0.4	1.462	0.41	0.5469 ± 0.0025	0.5127 ± 0.0019
0.4	1.424	0.38	0.4421 ± 0.0027	0.3904 ± 0.0019
0.4	1.35	0.29	0.3999 ± 0.0027	0.3171 ± 0.0022
0.35	1.62	0.58	0.6284 ± 0.0029	0.6185 ± 0.0018
0.35	1.418	0.4	0.4456 ± 0.003	0.4012 ± 0.0018

Table 3: List of thermodynamic results for 108 and 256 particles: ρ is the density; T is the temperature; (1) P is the pressure in the Verlet's paper; (2) P the pressure of our simulation for 108 particles and its error; (3) P the pressure of our simulation for 256 particles and its error.

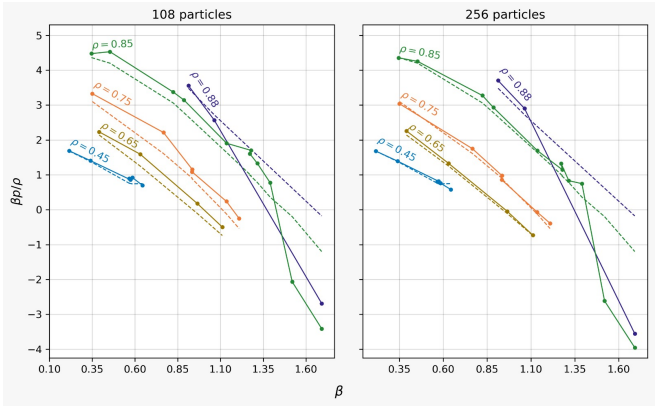


Figure 7: Evolution of the pressure of isochores under variation of T. Dashed lines correspond to results for these same simulations in [2], largely in agreement with experimental data. Solid lines and dots correspond to our simulated pressure.

Specific heat

Table 4 gathers the results we obtain for the specific heat. Figure The results we obtained have been gathered in Table 4. Figure 8 shows the evolution of specific heat under variation of T.

ρ	T	(1) c_v	(2) c_v	(3) c_v
0.85	2.889	0.73	1.5859 ± 0.0017	1.626 ± 0.0031
0.85	2.202	0.79	1.6041 ± 0.0022	1.6306 ± 0.003
0.85	1.214	0.95	1.7106 ± 0.0051	1.6778 ± 0.0043
0.85	1.128	0.99	1.7379 ± 0.0044	1.705 ± 0.0051
0.85	0.88	1.11	1.6852 ± 0.0041	1.6686 ± 0.0041
0.75	2.849	0.56	1.5890 ± 0.0018	1.5773 ± 0.0019
0.75	0.827	0.88	1.6928 ± 0.0044	1.7418 ± 0.0049
0.45	4.625	0.2	1.5353 ± 0.0009	1.5324 ± 0.0007
0.45	2.935	0.26	1.5571 ± 0.0011	1.5377 ± 0.0007
0.45	1.71	0.28	1.6026 ± 0.0022	1.6154 ± 0.0025
0.4	1.462	0.54	1.6936 ± 0.0034	1.6877 ± 0.0038

Table 4: List of thermodynamic results for 108 and 256 particles: ρ is the density; T is the temperature; (1)c_v is the specific heat in the Verlet's paper; (2)c_v the specific heat of our simulation for 108 particles and its error; (3)c_v the specific heat of our simulation for 256 particles and its error.

The overall behavior of our values is opposite to what it is supposed to, though we do not observe a lot of variation. What we were expecting was specific heat to decrease at high densities, where particles interact more strongly, as well as to decrease at low temperatures as particles have less thermal energy. However, our findings were somewhat inconsistent with this prediction. One possible explanation for this inconsistency is the relatively low number of particles used in our experiment (256), which may not have been sufficient for a molecular system. The number of particles can greatly affect specific heat by influencing the degree of particle interaction.

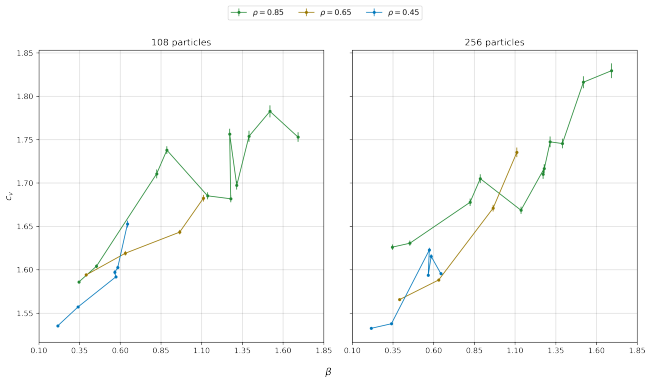


Figure 8: Evolution of the specific heat of isochores under variation of T. Solid lines and dots correspond to our results.

Room for improvement

For a better preformance of our code structure, we have to focus on reducing computation time. An analysis of the time consumed by different parts of the code would help us to identify where optimization strategies could be implemented. The improvement of computation times will contribute to a deeper study of accuracy over various parameters, such as number of steps and number of particles, that might affect the results.

References

[1] Thijssen, J.M. (1999). *Computational Physics*. Cambridge University Press.

[2] Verlet, L. (1967). *Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules*. Belfer Graduate School of Science, Yeshiva University, New York, New York.

[3] Lebowitz J.L., Percus J.K., Verlet L. (1966). *Ensemble Dependence of Fluctuations with Application to Machine Computations*. Belfer Graduate School of Science, Yeshiva University, New York, New York.

[4] Swimmer, M.T., Akhmerov, A. R. *Notes for Computational Physics, project 1*.