

Simulating Argon Molecules using Molecular Dynamics

Computational Physics Project 1

Tim Koreman

Abstract

Molecular Dynamics is a method to simulate a collection of atoms by integrating the equations of motion for each atom. Using this simulation we calculate a set of thermodynamic observables of Argon. The pair distribution function (PDF), pressure and specific heat are calculated at multiple temperatures and pressures and are compared with simulations from literature. The form of the PDF has distinct forms for each state of matter which agrees with results from literature. The reduced pressure scales linearly with the temperature within a specific temperature range, which is also agrees with results from literature. The temperature dependence of the specific heat does not agree with results from literature.

1 Introduction

Molecular systems are often too complex to calculate analytically. Therefore it is useful to look at those systems numerically. The simulation of such systems can be done by simulating the trajectories of the particles, this process is called molecular dynamics (MD). Using such a simulation thermodynamic quantities of a system can be calculated. In this report we look at an MD simulation of Argon atoms to simulate the pairwise distribution function, the heat capacity and the pressure of the system.

2 Theoretical Background

To calculate the trajectories of those particles we need to integrate Newton's laws of motion. To do this we discretize the time in steps of size Δt . Then we calculate the positions and velocities at time $t + \Delta t$ by using the positions and velocities at time t and the forces on the particles at that time. The forces on the particles are governed by the inter particle potential. We want to simulate Argon atoms for which the Lennard-Jones potential is used given by

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

where r_{ij} is the distance between atom i and j and U the potential. The potential has 2 parameters σ and ϵ corresponding with the point where the potential is zero and the depth of the potential well respectively.

The straight forward way to do the integration is to use Euler's method given by

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \vec{v}(t)\Delta t \quad (2)$$

$$\vec{v}(t + \Delta t) = \vec{v} + \frac{1}{m}\vec{F}(\vec{x}(t))\Delta t \quad (3)$$

where $\vec{x}(t)$ and $\vec{v}(t)$ denote the position and velocity respectively of a particle at time t . $\vec{F}(\vec{x}(t))$ denotes the force on the particle at position \vec{x} at time t . The mass of the particles is denoted by m . This straight-forward way of integrating the laws of motion is very easy to compute but has problems with conserving energy. We want to look at a microcanonical ensemble which means energy has to be conserved within the system. In his paper published in 1967 Verlet [1] introduced an algorithm which we used in a slightly altered way as the Velocity-Verlet algorithm given by

$$\vec{x}(t + \Delta t) = \vec{x}(t) + \Delta t \vec{v}(t) + \frac{\Delta t^2}{2} \vec{F}(\vec{x}(t)) \quad (4)$$

$$\vec{v}(t + \Delta t) = \vec{v}(t) + \frac{\Delta t}{2} (\vec{F}(\vec{x}(t + \Delta t)) + \vec{F}(\vec{x}(t))) \quad (5)$$

where \vec{x} , \vec{v} and \vec{F} are used to denote the same as in equations 2 & 3. Using this algorithm we can simulate the trajectories of the particles from their initial positions and velocities. We want to simulate the system at a given temperature T . To this end we initialize the velocities randomly from a Maxwell-Boltzman distribution. After initialization the system equilibrates by exchanging potential and kinetic energy. This process is difficult to predict so we have to correct for this by rescaling the velocities to correspond to the Maxwell-Boltzmann distribution at the temperature we want to simulate. This factor is given by

$$\lambda = \sqrt{\frac{(N-1)3k_B T}{\sum_i m \vec{v}_i^2}} \quad (6)$$

where N corresponds with the number of particles in the system, k_B the Boltzman constant, T the temperature and $\sum_i m \vec{v}_i^2$ the sum over the kinetic energies of all the particles. This rescaling is repeated until the temperature converges.

From this simulation at a temperature T some observables can be calculated. The pair correlation function gives the probability to find a particle in a range from r to $r + \Delta r$ from a reference particle. The pair correlation function $g(r)$ is given by

$$g(r) = \frac{2V}{N(N-1)} \frac{\langle n(r) \rangle}{4\pi r^2 \Delta r} \quad (7)$$

where V is the volume of the simulation cell, N the number of particles within a cell and $\langle n(r) \rangle$ the ensemble average of the number of particles within a distance r and $r + \Delta r$.

The pressure is given by

$$\frac{\beta P}{\rho} = 1 - \frac{\beta}{3N} \left\langle \frac{1}{2} \sum_{i,j} r_{ij} \frac{\partial U}{\partial r_{ij}} \right\rangle \quad (8)$$

where $\beta = (k_B T)^{-1}$, r_{ij} the distance between pairs of particles, N the number of particles and U the potential. The sum is over each of the pairs of particles.

The final observable looked at is the specific heat C_v given by

$$\frac{\langle \delta K^2 \rangle}{\langle K \rangle^2} = \frac{2}{3N} \left(1 - \frac{3N}{2C_V} \right) \quad (9)$$

where K is the kinetic energy and $\langle \delta K^2 \rangle = \langle K^2 \rangle - \langle K \rangle^2$. The number of particles is given by N .

3 Methods

3.1 Simulation Setup

Because the length- and timescales in the simulation are very small it is computationally more efficient to write the simulation in reduced units. We use length in units of σ , time in units of $\left(\frac{m\sigma^2}{\epsilon}\right)^{1/2}$ and energy in units of ϵ . From these definitions the other quantities in the simulation can be rewritten (see table 1). The plots in section 3 and 4 are in these dimensionless quantities where the tilde is dropped for notational clarity.

Natural Units:
$\tilde{r} = \frac{r}{\sigma}$
$\tilde{E} = \frac{E}{\epsilon}$
$\tilde{T} = \frac{T k_B}{\epsilon}$
$\tilde{t} = t \sqrt{\frac{\epsilon}{m \sigma^2}}$
$\tilde{P} = \frac{P \sigma^3}{\epsilon}$
$\tilde{c}_V = \frac{c_V}{k_B}$

Table 1: Definition of the natural units as used in the simulation. All plots and results given are in natural units where the tilde is dropped for notational clarity.

We want to study a system in the thermodynamic limit ($N \rightarrow \infty$). Simulating an infinite amount of particles is impossible so periodic boundary conditions are used to approach the thermodynamic limit. The system we want to study should go to a FCC lattice in the solid phase. Therefore the initial positions of the atoms were chosen at an FCC lattice with lattice constant a .

We calculate the observables by sampling our system n times and replacing the ensemble average with time averages. To calculate the errors for these observables bootstrapping is used. Bootstrapping re-samples with replacement from the n samples took from the system and the observables are calculated for each of those steps. This process repeated $N_{\text{bootstrap}}$ times and using the $N_{\text{bootstrap}}$ values found for the observables the error for an observables O calculated using

$$\sigma_O = \sqrt{\langle O^2 \rangle - \langle O \rangle^2}$$

where σ_O is the standard deviation for the observable.

The rescaling described in section 2 was applied every five steps until the system equilibrates at the desired temperature T . For the plots shown in section 3 and 4 this process was applied during the first 30 frames¹. After this rescaling the system was left to equilibrate for 20 frames and then we start sampling for the observables until the end of the simulation.

3.2 Validity Checks

To check whether the simulation evolves as expected we look at how the total energy evolves over time. The total energy plotted over time is shown in 1. The total energy should remain constant over time. As a secondary validity check we look at a system at $T = 1.0$ and $\rho = 0.8$ using a MD simulation with $M = 3$ and the others parameters as in table 2 we find that the reduced potential energy per atom equilibrates to $U = -5.305(5)$ where the error was calculated using the bootstrap method described in section 3.1. This value is close to the value of $U = -5.271(1)$ found in chapter 8 of Thijssen 2013 [5]. This discrepancy could be due to the relatively small system size used in this report due to time constraints.

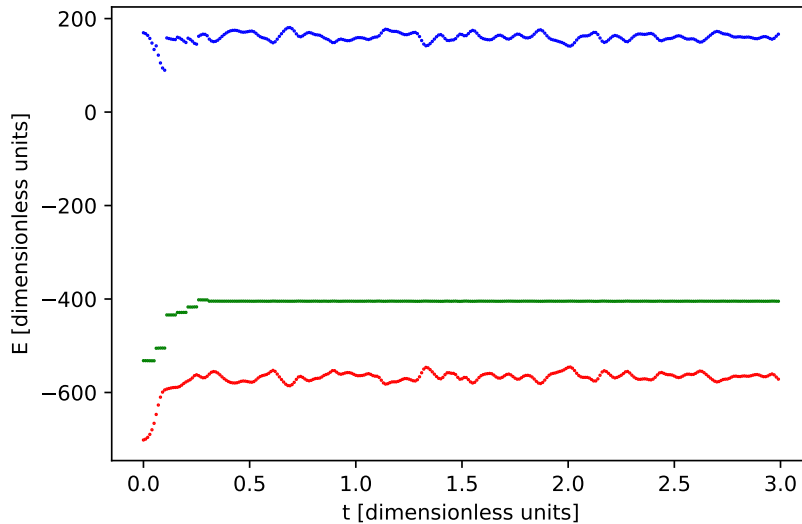


Figure 1: Energy during a MD run at $T = 1.0$ and $\rho = 0.8$ using the velocity-verlet algorithm with $M = 3$ and the other parameters from table 2. Red dots depict the potential energy, blue dots the kinetic energy and the green dots the total energy. The fluctuations in the first time steps are due to the rescaling to the temperature as described in section 2. After that rescaling the total energy remains constant up to three significant figures.

When we reverse the velocities of the particles halfway through the simulation the simulation should reverse and all the particles should reverse to their original position. To check this we look at the diffusion.

¹The number of times and the frequency the rescaling is applied with was chosen empirically.

The diffusion gives a measure of how much each particle on average has deviated from their starting position. We expect that if we reverse the simulation the particles return to their starting position and the diffusion goes to zero. In figure 2 we can see that the simulation used satisfies this validity check.

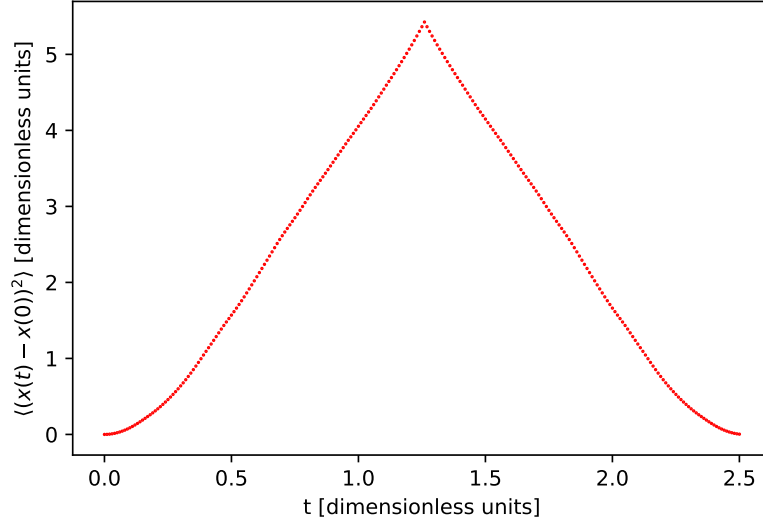


Figure 2: *Diffusion over time with $M = 3$ and the other parameters as in table 2 and $\rho = 0.3$ and $T = 3$. Halfway through the simulation the velocities of all the particles were reversed to study the reversibility of the simulation. After this reversal the diffusion goes to zero so the particles return to their initial positions. From this one can conclude this simulation is reversible.*

3.3 Implementation

The simulation was implemented using Python 3.6.3 and NumPy 1.14.2. The plots were made in Python using Matplotlib 2.1.0.

4 Results

For these results the MD simulation as described in section 2 and 3 was implemented for 256 atoms with the parameters given in table 2.

We want to look at the system in three phases of matter. The first result to look at to this end is the pair distribution function as described in section 2. In plot 3 one can see the pair distribution function for the three phases. These plots agree with results from literature [4].

Parameter:	Value:
Δt	4×10^{-3}
M	4
N	$4M^3$
a	$(4/\rho)^{1/3}$
L	Ma
$N_{\text{bootstrap}}$	10

Table 2: *Parameters used in the MD runs used to generate the plots as given in section*

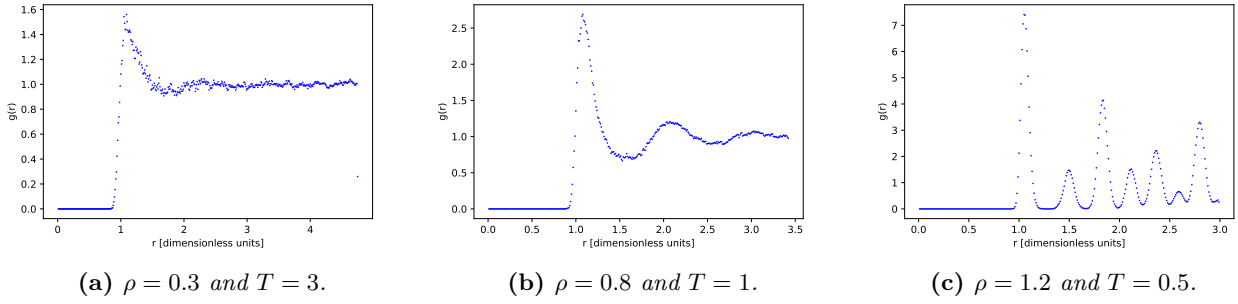


Figure 3: Pair distribution function for a MD run using the velocity-verlet algorithm with the parameters given in table 2 and a bin size of $\Delta r = 0.01$. Sub plots for gas, liquid and solid phase of matter respectively with ρ and T as given in sub caption. The structure for each of the phase agrees with results from literature [4].

We want to look at the temperature dependence of the pressure. The pressure was calculated using the definition as given in section 2. In plot 4 the reduced pressure is plotted against the reduced temperature for $\rho = 0.8$. In this temperature range we expect the pressure to scale linearly with the temperature [2].

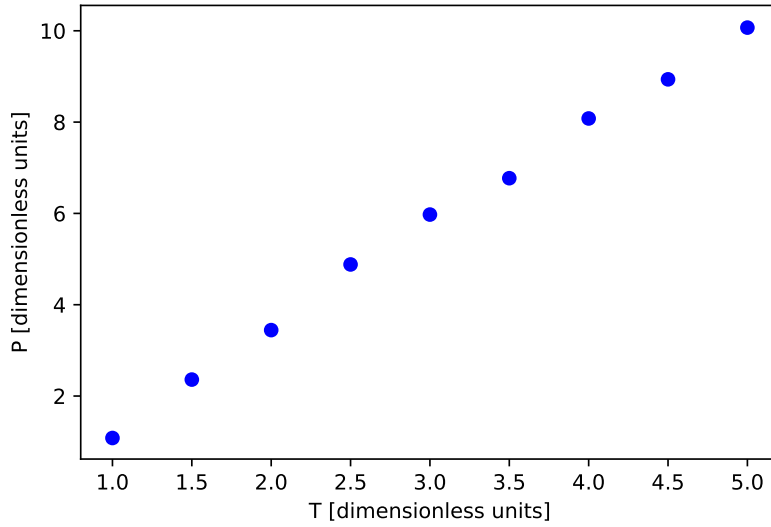


Figure 4: Reduced pressure against the reduced temperature for a MD run with $M = 3$ and the other parameters as given in 2 and $\rho = 0.8$. The error were calculated using the bootstrap method and plotted but are not visible because they fall within the dots. In this temperature range the pressure scales linearly with the temperature which agrees with results from Nichele et al. 2017 [2]

In the gas phase we want to compare the specific heat of the simulation with the heat capacity of an ideal gas. For an ideal gas we know that the reduced heat capacity per atom is given by $c_V = \frac{3}{2}$. For our simulation in the gas phase ($T = 3.0$ and $\rho = 0.3$) we find $c_V = 1.561(4)$ which is higher for the ideal gas what is to be expected since we look at a system with an interatomic potential which is not the case for the ideal gas. For the solid phase ($T = 0.5$ and $\rho = 1.2$) we compare the reduced specific heat per atom with the Dulong-Petit law which states that for some models of solids the reduced specific heat is given by $c_V = 3$. For our simulation we find $c_V = 3.7(4)$

When we look at the temperature dependence of the specific heat (figure 5) the achieved results don't agree with the expected results. For this temperature range an asymptotic decrease is expected [2] but this is not the achieved result.

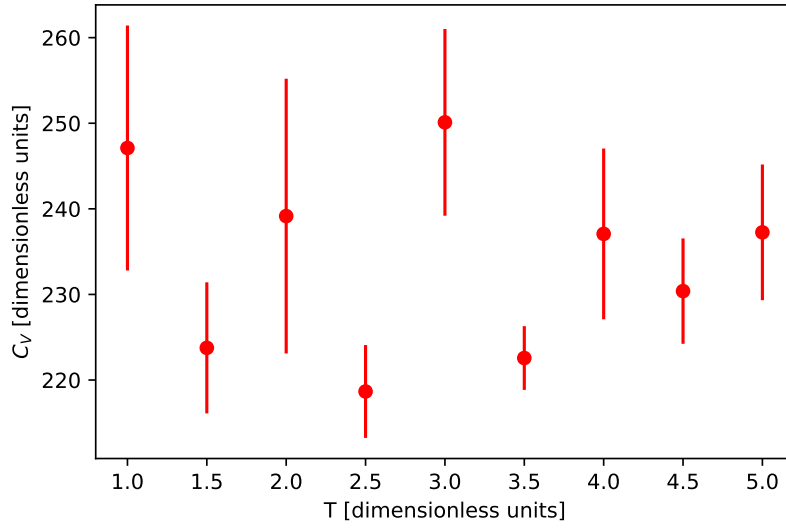


Figure 5: Specific heat against the reduced temperature for a MD run with $M = 3$ and the other parameters as given in 2 and $\rho = 0.8$. The errors were calculated using the bootstrap method. The temperature dependence of the specific heat does not agree with the results from literature[2].

5 Discussion & Conclusion

We looked at a Molecular Dynamics simulation for Argon atoms. To verify that the simulation functions as expected we verified that the energy remains constant over time and that the system is reversible, looked at the pair distribution function for 3 phases of matter and verified that these agree with literature. The temperature dependence of the pressure agrees with the result from literature [2]. The specific heat in the gas phase is just above the specific heat of an ideal gas which agrees with physical intuition and results from literature [2]. The temperature dependence of the specific heat does not agree with expected results. The validity checks, the pair distribution function and the temperature dependence of the pressure agrees with expectation. Therefore it is probable that something went wrong with the calculation of the specific heat.

In order to further improve the simulation the next step would be to simulate more particles than done for this report. In this report the forces were calculated for each pair of particles at each step. Since we look at short range interactions this is inefficient. To speed up this process we could implement an algorithm that keeps track which particles are close together (called the neighbour list) and only calculate the force pairwise for those particles close together. The trade off is that this neighbour list needs to be updated. Using the fact that the particles move quite slow this updating of the neighbour list would not have to be done every frame (see for example Chialvo & Debenedetti 1990 [3]).

References

- [1] L Verlet (1967), *Computer "Experiments" on Classical Fluids. I. Thermodynamical Properties of Lennard-Jones Molecules*, Physical Review Vol. 159 N. 1.
- [2] J Nichele et al. (2017), *Accurate calculation of near-critical heat capacities CP and CV of argon using molecular dynamics*, Journal of Molecular Liquids 237 65–70.
- [3] A Chialvo & P. Debenedetti (1990), *On the use of the Verlet neighbor list in molecular dynamics*, Computer Physics Communications Vol. 60, Is 2.
- [4] S Franchetti (1975), *Radial Distribution Functions in Solid and Liquid Argon*, Il Nuovo Cimento B Vol. 26 N. 2.
- [5] J.M. Thijssen (2013), *Computational Physics*, Cambridge University Press, Cambridge.