

FYS4460 - First compulsory project

Mathilde Nygaard Kamperud

February 24, 2013

1 Abstract

The purpose of this project is to get an introduction to molecular dynamics modeling. We develop programs to generate systems of atoms, visualize the system, find the motion for the Lennard-Jones-potential, and eventually measure macroscopic properties of the system (temperature, pressure, internal energy). We start out with the brute force way of finding the forces; calculate the forces between all the atoms in the system. Then we move on to the less time consuming algorithm which includes neighbour lists. We use periodic boundary conditions and the minimum image convention.

Through this project we implement the molecular dynamics code to model the behavior of a system of Argon atoms. The code can be found by opening project1.pro in qtCreator.

2 Model development

We start the simulation with all the atoms on a face-centered cubic lattice, since this corresponds to the crystalline structure of Argon. See figure 1. To build this kind of lattice we need four base-vectors, and several base-position. Let the base position be one of the corners in the lattice-unit. When we have determined the base position \mathbf{r}_{base} we can build the unit with the four base-vectors.

$$\begin{aligned}\mathbf{r}_0 &= [0, 0, 0] \\ \mathbf{r}_{xy} &= \left[\frac{b}{2}, \frac{b}{2}, 0 \right] \\ \mathbf{r}_{yz} &= \left[0, \frac{b}{2}, \frac{b}{2} \right] \\ \mathbf{r}_{zx} &= \left[\frac{b}{2}, 0, \frac{b}{2} \right]\end{aligned}\tag{1}$$

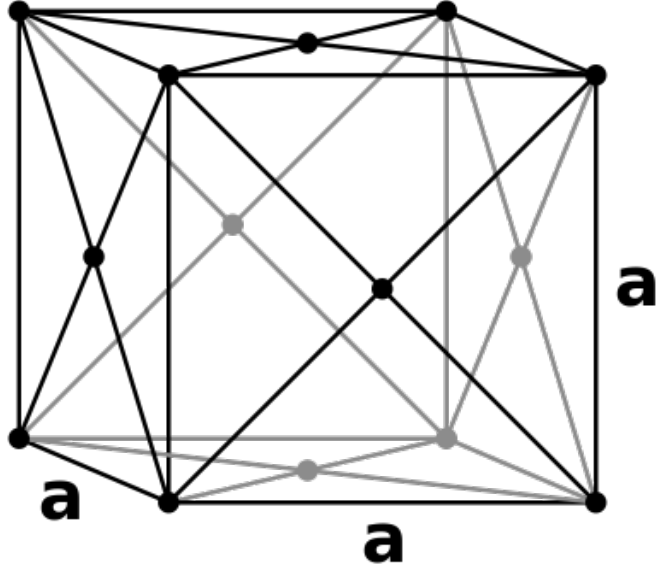


Figure 1: A face-centered cubic lattice unit.

where b is the distance between the base positions. For solid Argon $b = 5.260\text{\AA}$. To make this lattice you first make a grid consisting of the base positions (this is like a three dimensional cartesian coordinate-system, a system made of primitive cubic lattices) and for each of the base positions you generate four positions:

$$\begin{aligned}
 &\mathbf{r}_{\text{base}} + \mathbf{r}_0 \\
 &\mathbf{r}_{\text{base}} + \mathbf{r}_1 \\
 &\mathbf{r}_{\text{base}} + \mathbf{r}_2 \\
 &\mathbf{r}_{\text{base}} + \mathbf{r}_3.
 \end{aligned}
 \tag{2}$$

The natural choice of datastructure is an atom-class and a lattice class. The atoms has three aramdillo vectors; position, velocity, force (that works on the atom), and a string that states the type of element (Ar for argon). The Lattice-calls holds information about the lattice size, the size b , temperature and mass. It also holds a vector made of all the atoms in the system. (This is the brute force way of making the system).

To visualize the system we make state files; text files with .xyz format. The lines in the state file gives the type element, the position and the velocity of a atom at a given time. These state files can be loaded into VMD or OVITO.

2.1 Visualization

Visualization of the lattice made of $N_c \times N_c \times N_c$ face-centered cubic lattice units. This means we have a system of $N_c^3 * 4$ atoms. See figure 2.

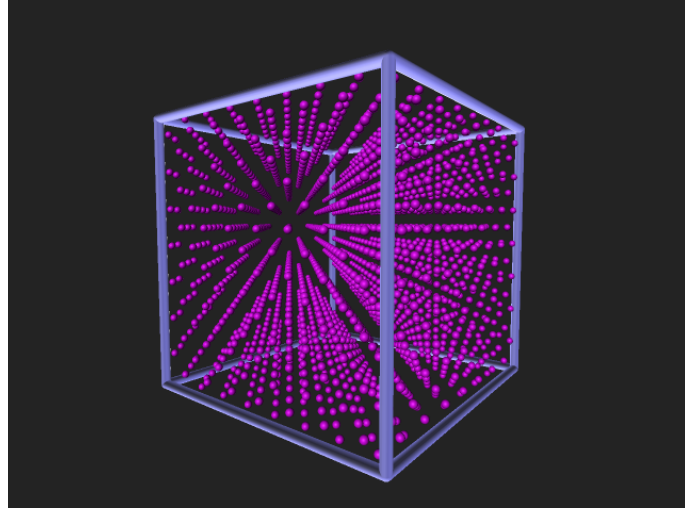


Figure 2: Visualization of solid Argon

To visualize the time development of the system, you make a .xyz file for each timestep and use VMD or OVITO to make an animation from these files.

2.2 Motion

We use the Verlet algorithm to find the time development of the system. The velocity and position of each particle is calculated as follows.

$$\mathbf{v}_i(t + \Delta t/2) = \mathbf{v}_i + \frac{\mathbf{F}_i(t)}{2m} \Delta t \quad (3)$$

$$\mathbf{r}_i(t + \Delta t) = \mathbf{r}_i(t) + \mathbf{v}_i(t + \Delta t/2) \Delta t \quad (4)$$

$$\mathbf{F}_i(t + \Delta t) = -\nabla_i U_i(\mathbf{r}(t + \Delta t)) \quad (5)$$

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t + \Delta t/2) + \frac{\mathbf{F}_i(t + \Delta t)}{2m} \Delta t \quad (6)$$

where we use the Lennard-Jones potential for the interatomic interactions.

$$U(r) = 4\epsilon \left[\left(\frac{\sigma}{r} \right)^{12} - \left(\frac{\sigma}{r} \right)^6 \right] \quad (7)$$

where r is the distance between the interacting atoms. We need to calculate the force on each atom due to interactions with all the other atoms in the lattice. Let's find the force on one of the atoms from all the other atoms.

$$\mathbf{F}_i(t) = \sum_{j \neq i} -\nabla_i U_i(|\mathbf{r}_j(t) - \mathbf{r}_i(t)|) = \sum_{j \neq i} -24\epsilon \left(\frac{\sigma^6}{r^8} - \frac{2\sigma^{12}}{r^{14}} \right) \mathbf{r} \quad (8)$$

where \mathbf{r} is the vector between the two positions from r_i to r_j , and r is the distance between them. The Verlet algorithm: First you have to find the forces on all the atoms at the initial state. Using the force and the velocities, we calculate the new position of the atoms. Then we have to calculate the force once more, this time using the new positions (the positions at time step $t + \Delta t$ that we just calculated). This force is used to find the velocities at the next time step. And we're done! The Lennard-Jones potential is plotted in figure 3. To plot this I used $\epsilon = \sigma = 1$, but that is not important.

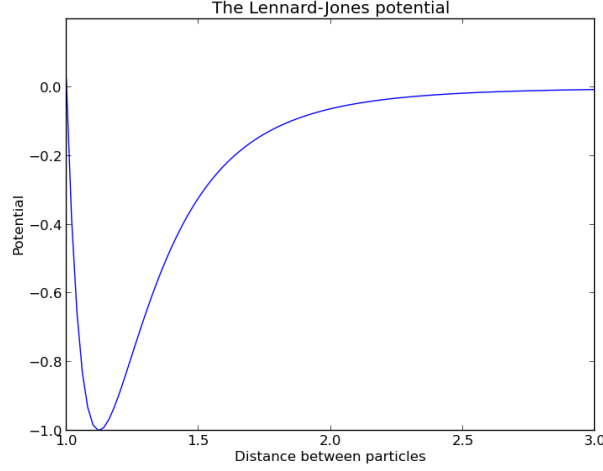


Figure 3: The Lennard-Jones potential

From the figure we see that the potential has an extremal point where it has a minimum. This minimum is at a distance we call equilibrium interatomic distance. When the particles are in a solid state, they will be oscillating with this distance between them. We can find this distance by setting the force between two particles to be zero, and solve for the distance between them.

$$-24\epsilon \left(\frac{\sigma^6}{r^8} - \frac{2\sigma^{12}}{r^{14}} \right) \mathbf{r} = 0 \quad (9)$$

$$\left(\frac{\sigma^6}{r^8} - \frac{2\sigma^{12}}{r^{14}} \right) = 0 \quad (10)$$

$$r^6 = 2\sigma^6 \quad (11)$$

$$r = 1.1225 * \sigma. \quad (12)$$

for Argon, the optimal value of $\sigma = 3.405\text{\AA}$. The equilibrium interatomic distance is then $r = 3.822\text{\AA}$.

2.2.1 Periodic boundary conditions

For our boundaries we choose periodic boundary conditions. This means that if a particle moves out of the cube on one side, it reappears inside the cube on the opposite side of where it disappeared, with the same velocity. We can think about it as a system of infinity identical cubes lying next to each other (they are copies). So when a particle moves out of the box, so does the copy-particle in the neighbors box. So our box gets a visit from the neighbors copy-particle. When we calculate the forces we need the distances between the particles. They are calculated using the minimum image convention.

Look at figure 4. This drawing illustrates the periodic boundary conditions. The lines in the drawing indicate the choices we have when finding the distance between the two particles. The minimum image convention states that we choose the smallest distance, in this case; the green line. This limits the interaction range to half the system size.

This is done by a set of complicated expressions in my program, but the bottom line is that I find the “real” distance $|r_i^x - r_j^x|$ in the x-direction, and the two other options: $|r_i^x - r_j^x \pm L|$, and use the smallest of these three lengths. Obviously I do the same evaluation for the y-, and z-direction as well.

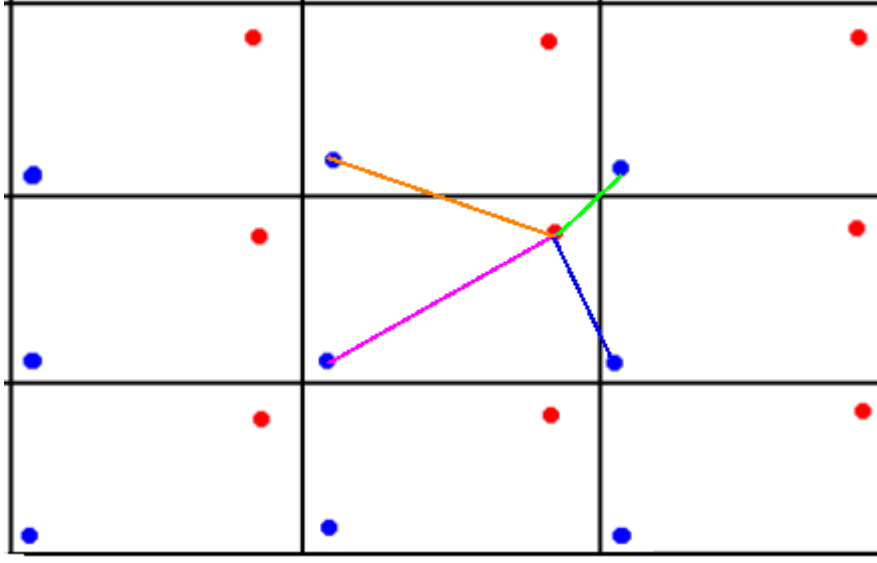


Figure 4: The system with periodic boundary conditions. Here with two particles. The lines indicate the choices we have when calculating the distance between the red and the blue particle.

2.2.2 MD-units

When doing computations on a computer it is not wise to use very large and very small numbers. Therefore we compute with dimensionless variables, using MD-units. For each variable A in our equations of motion, we insert $A = A' A_0$, where A' is the variable we end up computing with and A_0 is a conversion factor. The time step must also be treated this way. The equation of motion with MD-unit (after finding the equations with MD-units I switched back to variable-names without the apostrophe. $A' \rightarrow A$.)

$$\begin{aligned}
 \mathbf{v}_i(t + \Delta t/2) &= \mathbf{v}_i + \frac{\mathbf{F}_i(t)}{2} \Delta t \\
 \mathbf{r}_i(t + \Delta t) &= \mathbf{r}_i(t) + \mathbf{v}_i(t + \Delta t/2) \Delta t \\
 \mathbf{F}_i(t + \Delta t) &= -\nabla_i U_i(\mathbf{r}(t + \Delta t)) \\
 \mathbf{v}_i(t + \Delta t) &= \mathbf{v}_i(t + \Delta t/2) + \frac{\mathbf{F}_i(t + \Delta t)}{2} \Delta t \\
 U(r) &= 4 \left[\frac{1}{r^{12}} + \frac{1}{r^6} \right] \\
 \sigma_v &= \sqrt{T'}
 \end{aligned} \tag{13}$$

2.2.3 Initial velocity

For the initial velocity we draw random values from a normal distribution with average zero and standard deviation $\sqrt{k_B T/m}$ for each of the components. m is the mass of an atom, and k_B is the Boltzmann constant. Figure 5 shows the velocities before the first time step. The velocities are given in MD-units. For MD-units the standard deviation is $\sigma_v = \sqrt{T'} = \sqrt{T/T_0} = 0.914$. This fits quite good with the histogram in figure 5.

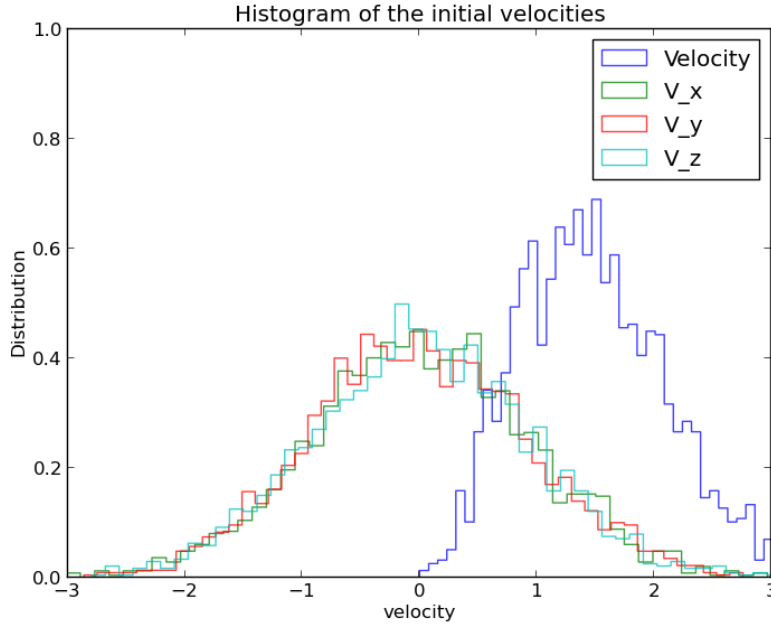


Figure 5: A histogram over the initial velocities of the system. Here in MD-units.

2.3 Implementation - Verlet solver

I ran my program for $8 \times 8 \times 8$ -lattice, with initial-temperature at 100 K (without thermostat), for 1000 timesteps. The result can be viewed in the gif called movie_Verlet.gif.

In that video we see that the particles stay in the crystal structure, and vibrates. So we are now in solid state for argon.

2.4 Neighbour lists

The Lennard-Jones interactions is short ranged, and can be neglected for distances over $r_{cut} = 3\epsilon_0$. There is not much point in calculating the forces outside this range. We therefore divide the space into quadratic cells of size r_{cut} , and let the atoms only interact with the atoms in the neighboring cells (and their own cell).

2.5 Implementation - Neighbour lists

I ran my program for $8 \times 8 \times 8$ -lattice, with initial-temperature at 100 K (without thermostat), for 1000 timesteps. The result can be viewed in the gif with the name movie_Cells.gif.

I wanted to compare the time usage for the two implementations; Verlet-solver and Cell-solver. So I ran my program for different number of time steps, and different sizes:

For $8 \times 8 \times 8$ -lattice:

# time steps	Verlet Solver-tid(s)	CellSolver-tid(s)
100	24	13
500	120	65
1000	241	130

For $12 \times 12 \times 12$ -lattice:

# time steps	Verlet Solver-tid(s)	CellSolver-tid(s)
100	274	44
500	1388 (23 min)	220 (3.6 min)
1000	2766 (46 min)	441 (7.35 min)

For $16 \times 16 \times 16$ -lattice:

# time steps	Verlet Solver-tid(s)	CellSolver-tid(s)
100	1825 (30 min)	106
500	8742 (2 hours og 15 min)	546 (9 min)
1000	17223 (4 hours og 27 min)	1059 (17.7 min)

I startet the program before I went home one night, and did not mean to run it for $16 \times 16 \times 16$ -because I had an idea of how much time it would take. But I made a mistake, and the program ran for this size as well. For the $N = 8$ lattice, the difference between the solvers are not that big. The Cell-solver uses about half as much time as the Verlet-solver, which is reasonable because we use four cells in each direction. In that case each particle interact with about half of the other particles in the system, as opposed to all of the particles (Verlet). For the $16 \times 16 \times 16$ -lattice the difference in time usage is very large!

2.6 The central limit theorem

According to the central limit theorem, the velocity distribution of the particles will eventually evolve into a Maxwell-Boltzmann distribution independent of the initial conditions. We test this by starting a simulation with velocities that are uniforml distributed random numbers in a interval. I made histograms of the x-componentents of the velocities at every time step, and put them together to make a animation. This can be found in movie_velocity.gif. Here we see that the x components are uniformly distributed at first, but turn into the typical shape of a normal distribution. This happens for the y- and z- components as well. When you have a normal distribution for the three space components, the magnitude of the velocity will become a Maxwell-Boltzmann-distribution.

3 Thermostats

In order to simulate the canonical ensemble, interactions with an external heat bath must be taken into account. In this project we use two thermostats; the Berendsen thermostat and the Andersen thermostat.

3.1 The Berendsen thermostat

The Berendsen thermostat rescales the velocities by multiplying them with a factor γ to regulate the temperature:

$$\gamma = \sqrt{1 + \frac{\Delta t}{\tau} \left(\frac{T_{bath}}{T} - 1 \right)} \quad (14)$$

Here τ is the relaxation time, tuning the coupling to the heat bath. In my program I gave τ ten times the value of Δt ($\tau = 10\Delta t$).

3.2 The Andersen thermostat

The Andersen thermostat works like this: For all atoms we generate a random number (uniformly distributed) in the interval $[0,1]$. If this number is smaller than $\frac{\Delta t}{\tau}$ (same τ as for Berendsen), the

atom is assigned a new velocity. For my program that means that about 10% of the atoms get a new velocity at every time step the thermostat is on. This new velocity is again a random number, but this time collected from a normal distribution with standard deviation $\sigma_v = \sqrt{k_B T_{bath}/m}$.
key-words: Berendsen, Andersen

4 Macroscopic observables

4.1 Energy

To find the total internal energy of the system, we have to calculate the kinetic and potential energy, based on the particles velocities and the distances between them. The potential energy is just the Lennard-Jones potential we used to find the forces, and the kinetic energy is the sum of all the kinetic energies due to the particles $\frac{1}{2}m(v_x^2 + v_y^2 + v_z^2)$. The total energy is just the sum of the kinetic and potential energy.

$$E_{tot} = \sum_{particles} \frac{1}{2}m(v_x^2 + v_y^2 + v_z^2) + \sum_{i < j} 4\epsilon \left[\left(\frac{\sigma}{r_{ij}} \right)^{12} - \left(\frac{\sigma}{r_{ij}} \right)^6 \right] \quad (15)$$

In my program, the kinetic, potential and total energy is measured and stored in a file every time step. After running my system with two different $dt = 0.01, 0.001$, I made energy vs. time-plots and calculated the fluctuations in energy. The plots can be seen in figure 6 and 7.

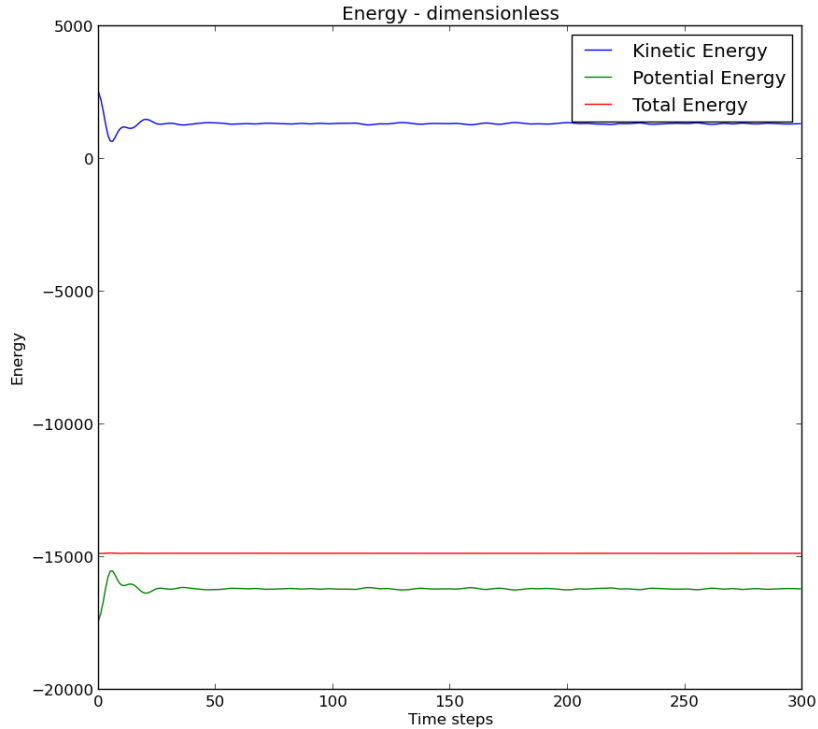


Figure 6: Energies as a function of time. Here with $dt = 0.01$.

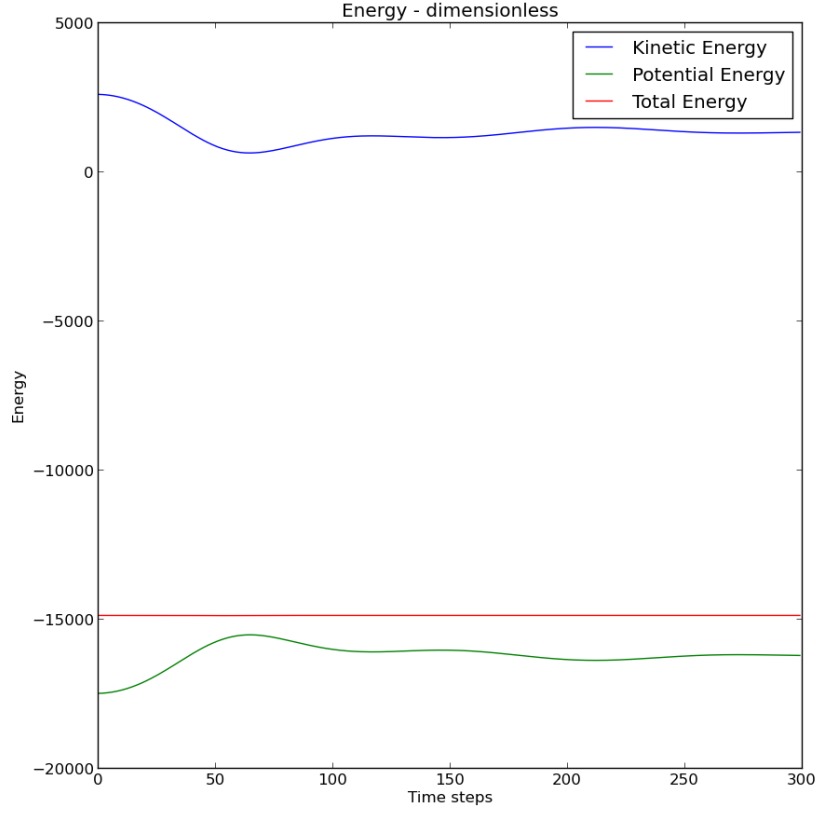


Figure 7: Energy as a function of time. Here with $dt = 0.001$.

These two plots are made with the same amount of time steps, meaning that one with the largest time step evolves further in time than the one with the small time step. That is why the kinetic and potential energy in plot in figure 7 looks less stable than in figure 6. In figure 7, the system uses almost all the time steps to equilibriate. Perhaps it would be better to run the program for the same amount of time instead of the same amount of time steps.

It is a bit hard to tell which of the two plots who has the least fluctuation in energy, so we better look at the numbers. In the table below you see the results.

dt	Average energy	Fluctuations
0.01	-14848.868	1.58
0.001	-14857.48	0.087

Based on these two measurements it looks like the energy fluctuations decrease as we decrease the time step.

4.2 Temperature

According to the equipartition theorem, the average total kinetic energy is

$$\langle E_k \rangle = \frac{3}{2} N k_B T \quad (16)$$

We can use this to measure the temperature per time. Let's see what happens when we start out with temperature equal to 100 K (this gives the standard deviation of the initial velocities). In figure 8 we see that even though we start out with $T = 100\text{K}$, the system stabilizes around 50 K, but with quite large fluctuations.

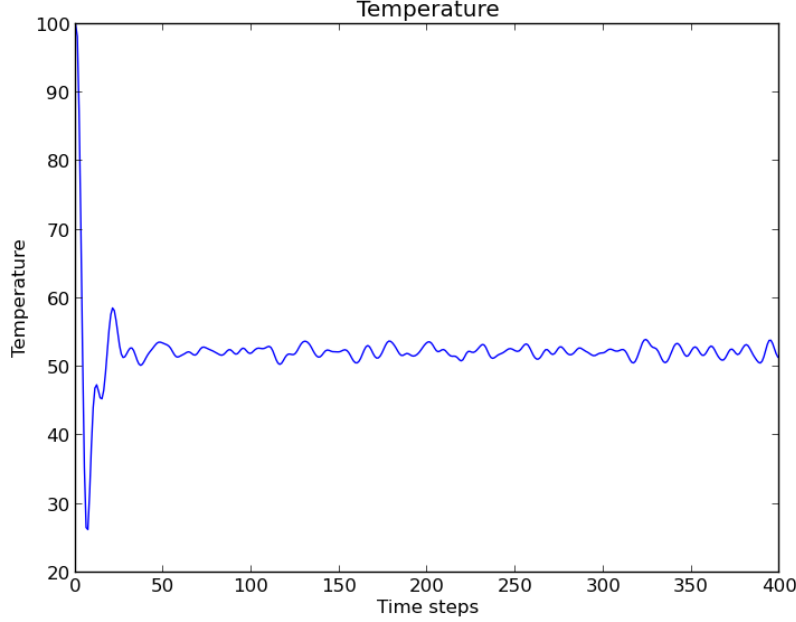


Figure 8: Temperature as a function of time. Initial temperature: 100 K

4.3 Pressure

The method we use for measuring the pressure is derived from the virial equation. The average pressure in a volume V with N particles:

$$P = \frac{Nk_B T}{V} + \frac{1}{3V} \sum_{i < j} \mathbf{F}_{ij} \cdot \mathbf{r}_{ij}$$

This pressure is valid for the micro-canonical ensemble only. To make the $P(T)$ -plot, I equilibrated my system many times with different temperatures, and found the mean pressure every time and plotted the result. In figure 9 we see the pressure-temperature-plot. It looks quite sensible that the pressure increases as the temperature increases. Around 250 K it looks like we have a phase-transition. But there is only one of them, and it is at a very high temperature, so I am not sure what that means.

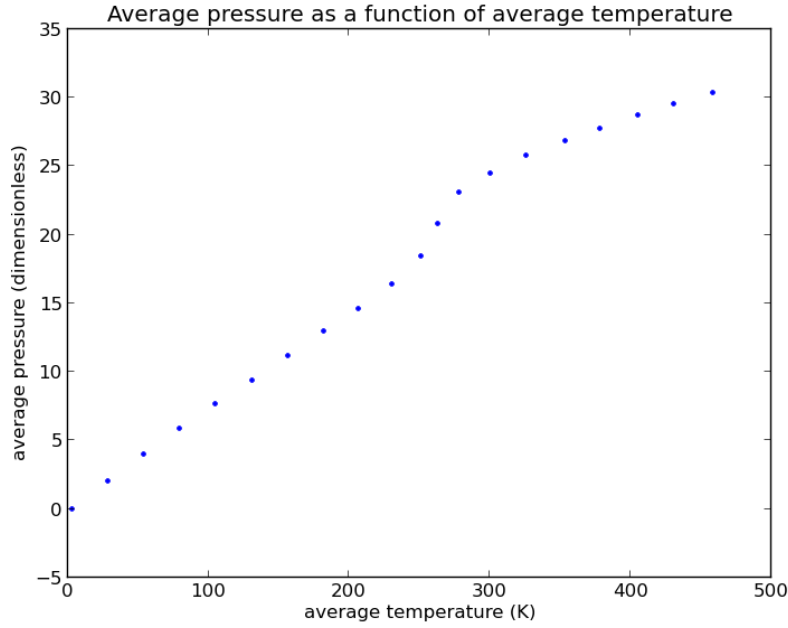


Figure 9: The pressure as a function of temperature.

4.4 The diffusion constant

We want to characterize transport in a fluid by measuring the self-diffusion of an atom. That is, we find the mean square displacement of all the atoms

$$\langle r^2(t) \rangle = \frac{1}{N} \sum_{i=1}^N (\mathbf{r}(t) - \mathbf{r}_{initial})^2 \quad (17)$$

From this we can find the diffusion constant by

$$\langle r^2(t) \rangle = 6Dt \quad (18)$$

when $t \rightarrow \infty$.

It is important to remember that we use periodic boundary conditions in our algorithm. When measuring the diffusion constant we have to follow the atoms as if they did not jump from one boundary to another. I have solved this by letting the atoms know their “real” position, the position they would have if they did not jump back into the box when they cross the boundaries. And I use this “real”-position when finding the square displacement. I have found the diffusion constant for these temperatures $T = 500, 700$. For $T = 500$ the average displacement is plotted in figure 10. I have found a linear approximation to the curve by using linear regression, as you can see in the plot. Based on the slope of the linear approximation, I found the diffusion constant to be $D_{500} = 0.002568$ in dimensionless units.

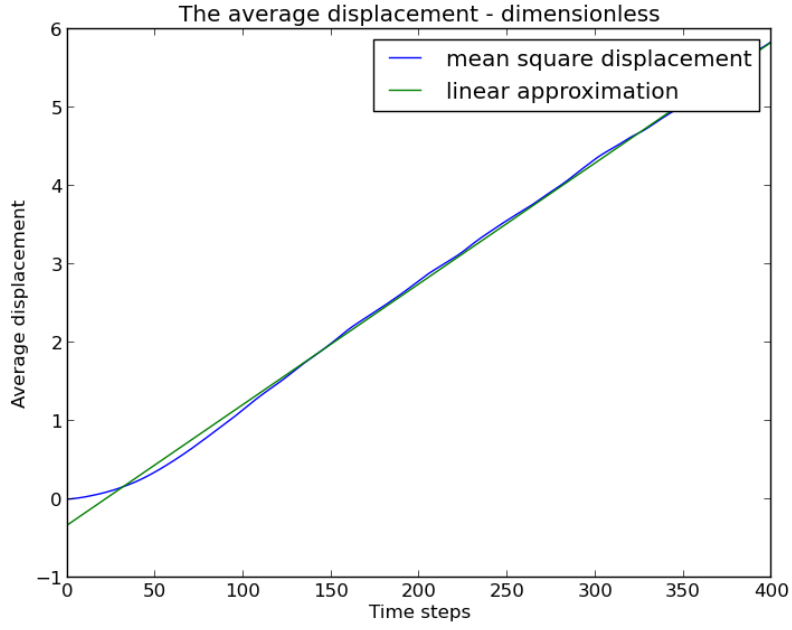


Figure 10: The average displacement for $T=500$ kelvin

Did the same thing for $T = 700$ K. See figure 11.

Here the diffusion constant was $D_{700} = 0.00507$ in MD-units. It appears the diffusion constant increases with temperature, which is reasonable. Larger temperature means larger velocities in the system, and that enables the particles to move further.

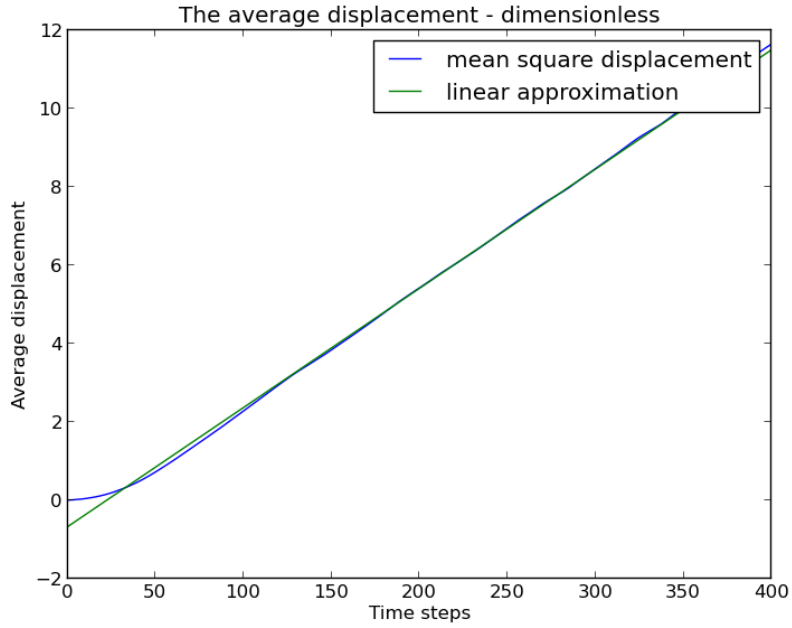


Figure 11: The average displacement for temperature $T = 700$.

5 Microscopic structure - radial distribution functions

The radial distribution function $g(r)$ is the radial probability for finding another atom a distance r from an arbitrary atom. To find this distribution we measure the distances between all the atoms, and make a histogram of the distances. The largest distance we can find is $L/2$ because we have periodic boundary conditions, so our distance interval is $[0, L/2]$. We divide the distance interval into bins, and count how many atom-pairs-distances belong in each bin. I first plot the $g(r)$ for the perfect crystal we have before time-evolving the system, to check if I counted right. The result is in figure 12. Here we see that there are some positions that are much more likely than others, which is what we would expect from a perfect crystal. This plot is not timeaveraged.

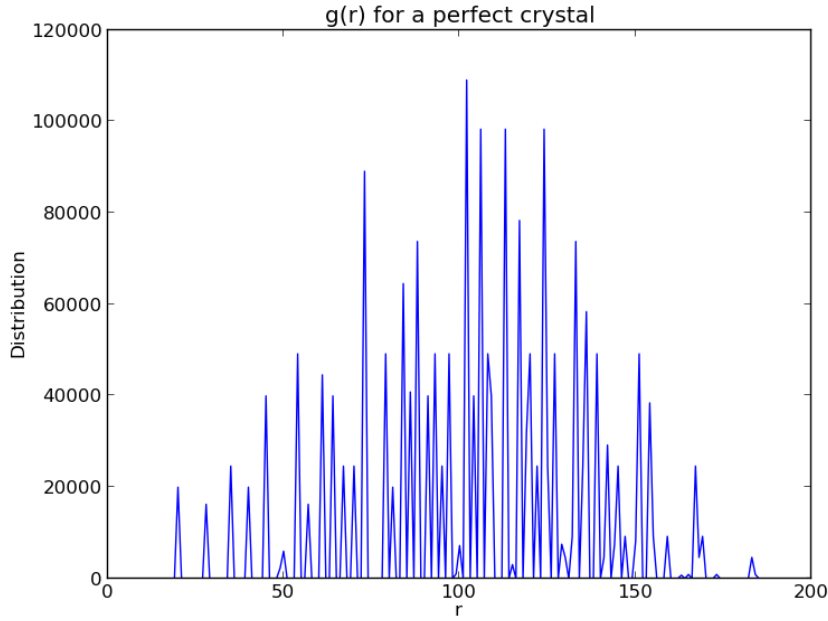


Figure 12: The $g(r)$ for the perfect crystal.

Now we look at the same system after letting it evolve some time, and timeaveraging the result. I start counting the distances after a few time steps, so that the system reaches the desired temperature. In figure 13 you can see the distribution for $T = 100$ K. This is all the measurements over all time steps, so we should divide by the number of time steps, but it doesn't matter. The shape is what interest us. We see that it looks a bit like figure 12; it has the same shape, but the spikes are not as sharp, and they blend into each other so there is no distance between them.

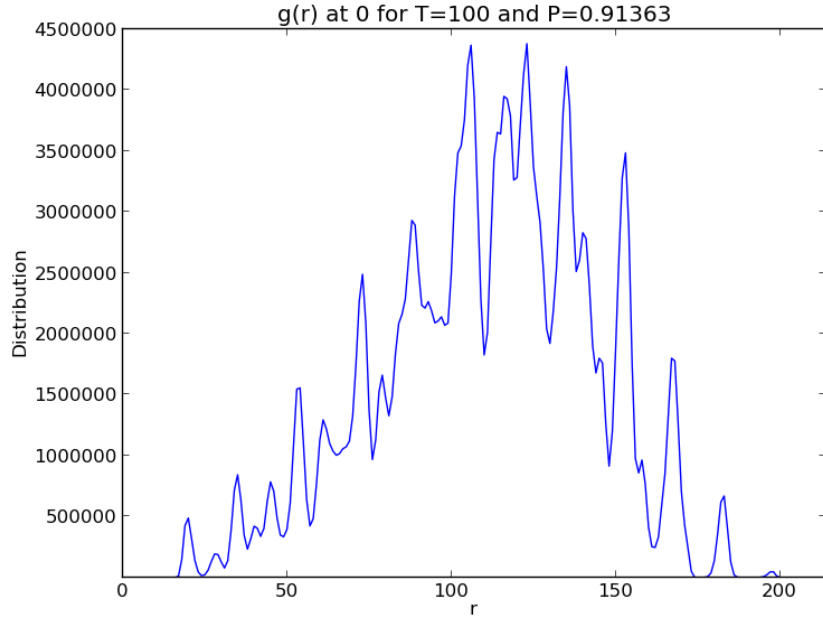


Figure 13: The $g(r)$ for temperature 100 kelvin.

Now I wanted to find $g(r)$ for a temperature larger than the temperature where we got the phase-transition, namely $T \approx 250$ K. The result is in 14. The general shape is still there, but now the spikes are totally gone.

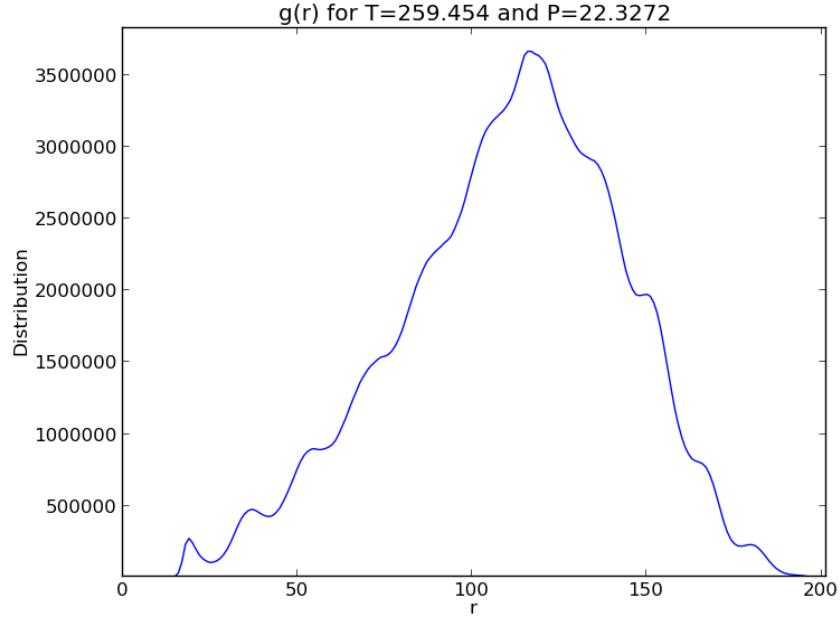


Figure 14: The $g(r)$ at $T=259$ K. After the phase transition