Molecular Dynamics Project

FYS-MEK1110

William Dugan

University of Oslo
March, 2022

Contents

1 **Introduction** **2**

a. Understanding the potential . . . 2

b. Forces and equation of motion . . . 3

c. Units . . . 4

2 **Two-atom simulations** **6**

a. Implementation . . . 6

b. Motion . . . 6

c. Energy . . . 7

d. Visualization . . . 9

3 **Large systems** **10**

a. Implementation . . . 10

b. Verification . . . 11

c. Initialisation . . . 12

d. Many atoms, open boundary . . . 13

e. Boundary conditions . . . 13

4 **Science** **15**

a. Temperature . . . 15

b. Velocity autocorrelation and diffusion coefficient . . . 15

c. Mean square displacement and diffusion coefficient . . . 17

d. Radial distribution function . . . 18

1 Introduction

The goal of this project is to model argon gas, where the atoms interact according to the famous Lennard-Jones potential,

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

where r is the distance between two atoms, $r = ||\vec{r_i} - \vec{r_j}||$. The parameters σ and ϵ determine which chemical compound is modelled.

The full code used to answer the tasks will not be shown in this document. See https://github.com/willidu/FYS-MEK1110/tree/main/md-prosjekt.

I am grateful for the guidance I have received from our advisor as well as for the advice my fellow students have paid me throughout this project. This project would not be a finished product without them.

a. Understanding the potential

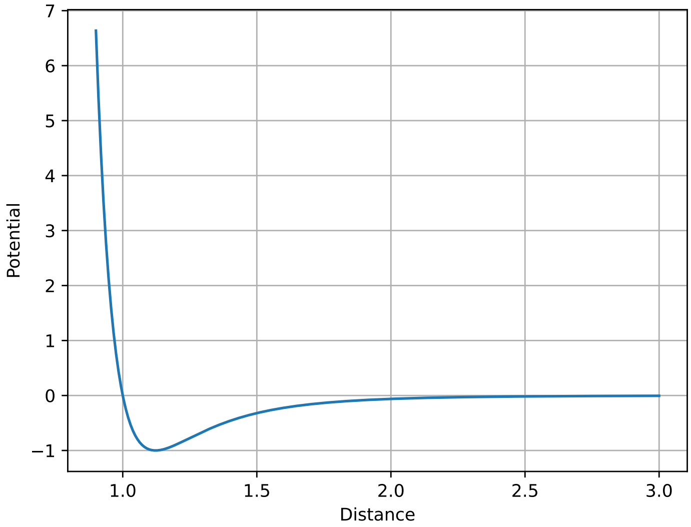

Figure 1: Plot of $U(r)$ for $r \in [0.9, 3], \epsilon = 1, \sigma = 1$.

The behaviour of $U(r)$ is vastly different for $r < \sigma$ and $r > \sigma$. For $r < \sigma$, the first term in equation 1 dominates and will make the potential go towards ∞ for small r. For $r = \sigma$ the terms are equal and the potential is 0. For $r > \sigma$, the second term in equation 1 dominates hence the negative potential. For $r \to \infty$ the potential will be 0 again. The two terms quickly become the same order of magnitude, effectively giving us zero potential a lot earlier than $r = \infty$.

Equilibrium is reached when the force between the two atoms is zero. The force is given by

$$\vec{F}(\vec{r}) = -\nabla U(\vec{r}) \tag{2}$$

Since we are only considering one dimension (for now), this will be

$$\begin{align} F(r) &= -\frac{dU}{dr} \tag{3} \\ &= -(-4 * 12\epsilon\sigma^{12}r^{-13} + 4 * 6\epsilon\sigma^{6}r^{-7}) \tag{4} \\ &= -\frac{24\epsilon\sigma^{12}(r^6 - 2\sigma^6)}{r^{13}} \tag{5} \end{align}$$

If we solve equation 5 equal to zero, we find the points of equilibrium. We have the following solutions:

$$r = \sqrt[6]{2\sigma^6} = \sigma\sqrt[6]{2} \qquad \wedge \qquad r = \infty$$

Since the potential is at a minimum at $r = \sigma\sqrt[6]{2}$, this is a stable equilibrium point. In contrast, $r = \infty$ is an unstable equilibrium since the potential is at a (local) maximum.

The motion of two atoms starting with a separation of 1.5σ can be described by the graph of the potential (Figure 1). At $r = 1.5\sigma$ the slope of U is positive, so the force between the atoms will be in negative r direction. This means that the atom will move towards the left until it reaches the potential $U(1.5\sigma)$ at the graph to the left of the equilibrium point. Since energy is conserved in this case, the atom will simply oscillate between these points.

If the atoms had an initial separation of 0.95σ, the force would initially be in the positive r direction. The atoms would then move away from each other forever since it cannot reach the potential $U(0.95\sigma)$ on the right side of the equilibrium point.

b. Forces and equation of motion

The (vector)force on atom i at position $\vec{r_i}$ from atom j at position $\vec{r_j}$ follows from equation 5 and is given by

$$\vec{F} = -\frac{24\epsilon\sigma^{12}(r^6 - 2\sigma^6)}{r^{13}} \frac{\vec{r}}{|\vec{r}|} \tag{6}$$

where

$$\frac{\vec{r}}{|\vec{r}|} = \frac{\vec{r_i} - \vec{r_j}}{||\vec{r_i} - \vec{r_j}||}$$

is the unit vector in the direction of the force.

To find the equation of motion for atom i we can use Newton's second law and equation 6.

$$\begin{aligned}
\sum \vec{F} &= m\vec{a} && (7)\\
\implies \vec{a_i} = \frac{d^2\vec{r_i}}{dt^2} &= \frac{\sum \vec{F_i}}{m} && (8)\\
&= \frac{1}{m}\sum_{j\neq i}\left[-\frac{24\epsilon\sigma^6(r^6-2\sigma^6)}{r^{13}}\right]\frac{\vec{r_i}-\vec{r_j}}{||\vec{r_i}-\vec{r_j}||} && (9)\\
&= \frac{24\epsilon}{m}\sum_{j\neq i}\left[-\frac{\sigma^6}{r^{11}}+\frac{2\sigma^{12}}{r^{13}}\right]\frac{\vec{r_i}-\vec{r_j}}{||\vec{r_i}-\vec{r_j}||} && (10)\\
&= \frac{24\epsilon}{m}\sum_{j\neq i}\left[2\left(\frac{\sigma}{r}\right)^{12}-\left(\frac{\sigma}{r}\right)^{6}\right]\frac{1}{r}\frac{\vec{r_i}-\vec{r_j}}{||\vec{r_i}-\vec{r_j}||} && (11)\\
&= \frac{24\epsilon}{m}\sum_{j\neq i}\left[2\left(\frac{\sigma}{||\vec{r_i}-\vec{r_j}||}\right)^{12}-\left(\frac{\sigma}{||\vec{r_i}-\vec{r_j}||}\right)^{6}\right]\frac{\vec{r_i}-\vec{r_j}}{||\vec{r_i}-\vec{r_j}||^2} && (12)
\end{aligned}$$

We sum for all $j \neq i$ since an atom i will interact with all other atoms except itself.

c. Units

To scale equation 7, we use $\vec{r_i}^* = \vec{r_i}/\sigma$ and $t^* = t/\tau$. We also have to assume $\sigma > 0$.

$$\begin{aligned}
\frac{d^2\vec{r_i}^*}{dt^{*2}} &= 24\sum_{j\neq i}\left[2\left(\frac{\sigma}{||\vec{r_i}^*\sigma-\vec{r_j}^*\sigma||}\right)^{12}-\left(\frac{\sigma}{||\vec{r_i}^*\sigma-\vec{r_j}^*\sigma||}\right)^{6}\right]\frac{\vec{r_i}^*\sigma-\vec{r_j}^*\sigma}{||\vec{r_i}^*\sigma-\vec{r_j}^*\sigma||^2} && (13)\\
&= 24\sum_{j\neq i}\left[2\left(\frac{\sigma}{|\sigma| * ||\vec{r_i}^*-\vec{r_j}^*||}\right)^{12}-\left(\frac{\sigma}{|\sigma| * ||\vec{r_i}^*-\vec{r_j}^*||}\right)^{6}\right]\frac{\vec{r_i}^*-\vec{r_j}^*}{||\vec{r_i}^*-\vec{r_j}^*||^2} && (14)\\
&= 24\sum_{j\neq i}\left[2||\vec{r_i}^*-\vec{r_j}^*||^{-12}-||\vec{r_i}^*-\vec{r_j}^*||^{-6}\right]\frac{\vec{r_i}^*-\vec{r_j}^*}{||\vec{r_i}^*-\vec{r_j}^*||^2} && (15)
\end{aligned}$$

To find the characteristic time scale τ, we perform a simple dimension analysis and compare the units of τ and the constants given.

$$\begin{aligned}
\sigma &= [m]\\
m &= [kg]\\
\epsilon &= [J] = \left[\frac{kg\cdot m^2}{s^2}\right]
\end{aligned}$$

Since we need τ to be dimensionless, it has to be the same unit as t, hence seconds. Therefore, τ is given by

$$\tau = \sqrt{\frac{kg\cdot m^2}{kg\cdot m^2}\cdot s^2} = [s]$$

which is achieved by expressing τ the following way:

$$\tau = \sqrt{\frac{m\sigma^2}{\epsilon}} \tag{16}$$

The values for argon are

$$\begin{aligned}\sigma &= 3.405 \cdot 10^{-10}\text{m},\\ m &= 39.95 \cdot 1.66 \cdot 10^{-27}\text{kg},\\ \epsilon &= 1.0318 \cdot 1.602 \cdot 10^{-21}\text{J}\end{aligned}$$

which gives $\tau \approx 2.26 \cdot 10^{-12}$s.

2 Two-atom simulations

See `md-prosjekt/task2.py`.

a. Implementation

See `md-prosjekt/two_atom_sim.py`. Though the task asked for only one function, I chose to separate it into multiple functions. This made debugging easier.

b. Motion

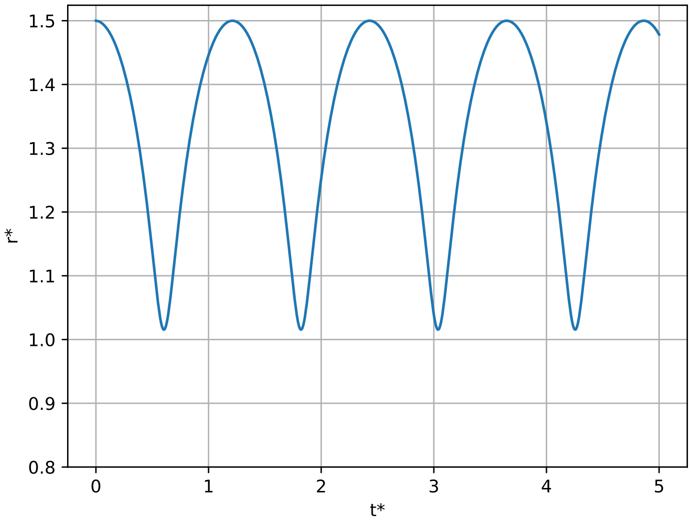

Figure 2: Plot of r^* for $t^* \in [0, 5]$. Initial separation 1.5σ.

As we found in *Forces and equation of motion*, the expected motion of two atoms with an initial separation of 1.5σ is periodic. From Figure 2 we see that the atoms oscillate periodically between two points. This tells us that our model works as expected.

We will now simulate with an initial separation of $.95\sigma$. See figure 3. Here we can see that the atoms move away from each other. They move with constant velocity as the acceleration goes to zero (equation 5). We also notice that the curve straightens out very quickly, which is explained by Figure 1 when the potential goes towards infinity for small r. This motion is also as expected.

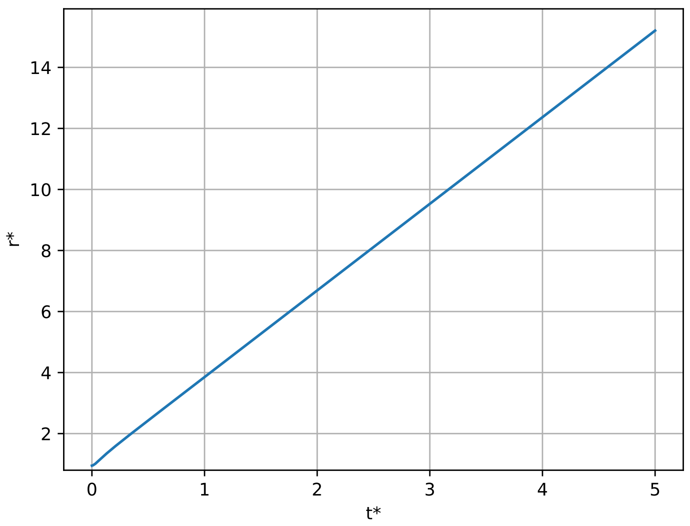

Figure 3: Plot of r^* for $t^* \in [0, 5]$. Initial separation $.95\sigma$.

c. Energy

We have to scale the potential to match the scaled kinetic energy. Remembering that r^* is already scaled by σ, $\vec{r_i}^* = \vec{r_i}/\sigma$, we get the following equation by dividing equation 1 with ϵ. This leaves

$$U^*(r^*) = 4\left(r^{*-12} - r^{*-6}\right) \tag{17}$$

where $r^* = ||\vec{r_i}^* - \vec{r_j}^*||$.

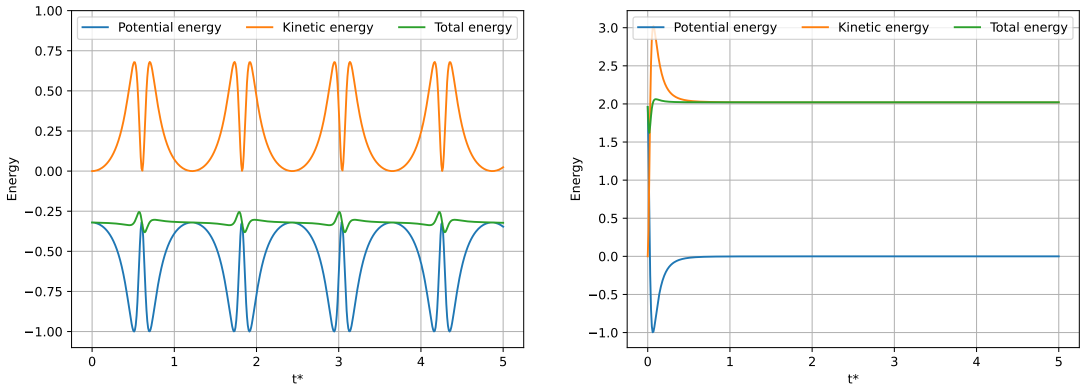

Figure 4: Energy plot with separation 1.5σ and $.95\sigma$ for $t \in [0, 5]$.

In theory, the total energy should be conserved since we do not have any non-conservative forces. If we compare the total energy for the two simulations, we obtain results consistent with the findings in *Forces and equation of motion.* If energy had been lost, the particles would slow down and the time period for each oscillation would increase. This is not the case. Hence, energy is conserved. We see from Figure 4 that momentum is conserved if the initial separation is 1.5σ, but it is not conserved when it is 0.95σ.

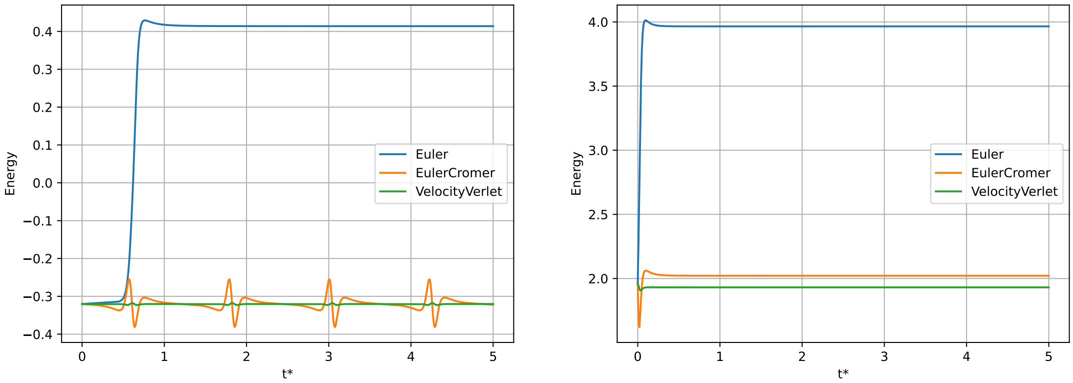

Figure 5: Total energy with initial separation 1.5σ for $t \in [0, 5]$ and $.95\sigma$ and $t \in [0, 1]$ respectively.

From Figure 5 it is clear that energy is not conserved with the Euler method. Let us consider the situation with an initial separation of 1.5σ. It is clear that even though there are some fluctuations with both the Euler-Cromer and Velocity Verlet methods, the changes are periodic and does not increase with time. This is not the case for the Euler method. we can observe the same occurrence then the initial separation is $.95\sigma$. One last observation is that the total energy is one order of magnitude larger when the initial separation is $.95\sigma$ when compared to when the separation is 1.5σ.

I was unable to obtain a stable motion with Euler's method, even with unreasonably small time steps. With using $\Delta t = 0.01$ I got stable motion and conservation of energy with Euler-Cromer and Velocity Verlet. It is well known that Euler-Cromer is far better than Euler when working with periodic movement. The differences between these two methods are minor, and Euler-Cromer is not much harder to work with. Velocity Verlet is a more intuitive integration method as it is based on Taylor approximations of the equations for motion. We will use this method further on in the project.

d. Visualization

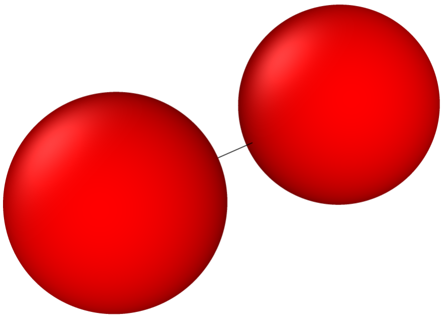

Figure 6: Representation of .xyz file in Ovito, showing one frame only.

3 Large systems

See `md-prosjekt/task3.py`.

a. Implementation

See `md-prosjekt/n_atom_sim.py`. Note that the code does not use Newton's third law to reduce the number of force calculations. To reduce the simulation time we set a cutoff in the potential r_c^*; we see from Figure 1 that the potential is effectively zero at $r = 3$, which is the value we choose for our cutoff.

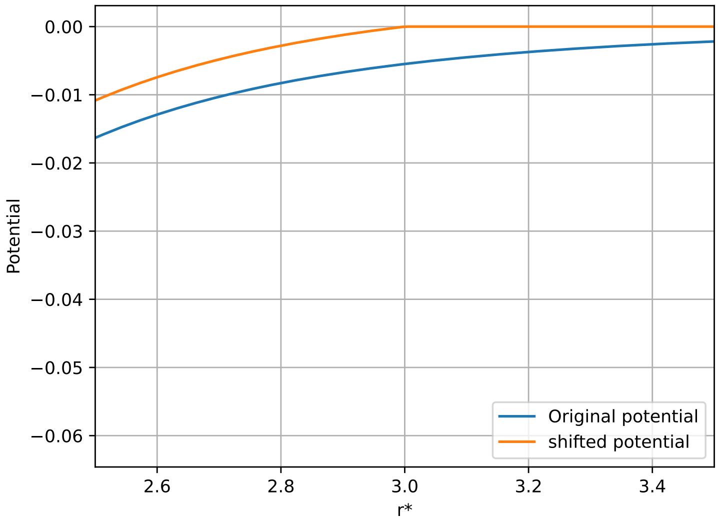

Figure 7: Plot of original and shiftet potential for $r^* \in [2.5, 3, 5]$ with $r_c^* = 3$.

We observe that the potential is shifted with $U^*(3)$ for $r^* < 3$. This does not affect force calculations as it is defined by the derivative of the potential, thus the addition of a constant does not have any effect.

b. Verification

The solver for N atoms also works for two atoms. It yields the same results as *Two-atom simulations*. We will now simulate the motion of four atoms. The first simulation is run with initial positions [1, 0, 0], [0, 1, 0], [-1, 0, 0] and [0, -1, 0]. For the second simulation, we alter the first atom to have a starting position of [1, 0.1, 0].

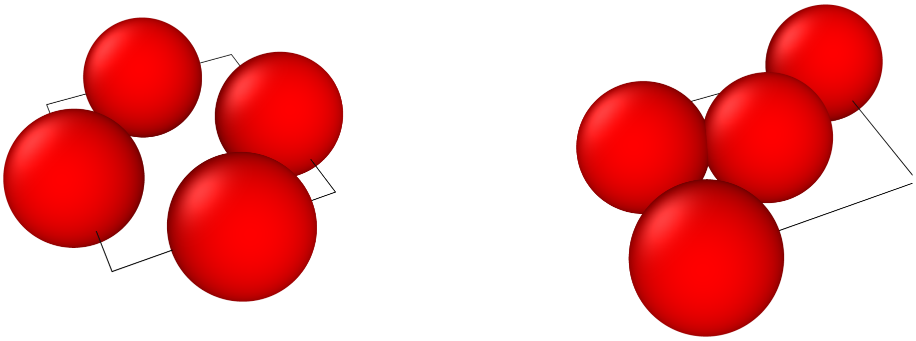

Figure 8: Ovito visualization for the last simulation, showing frame 1/500 and 415/500.

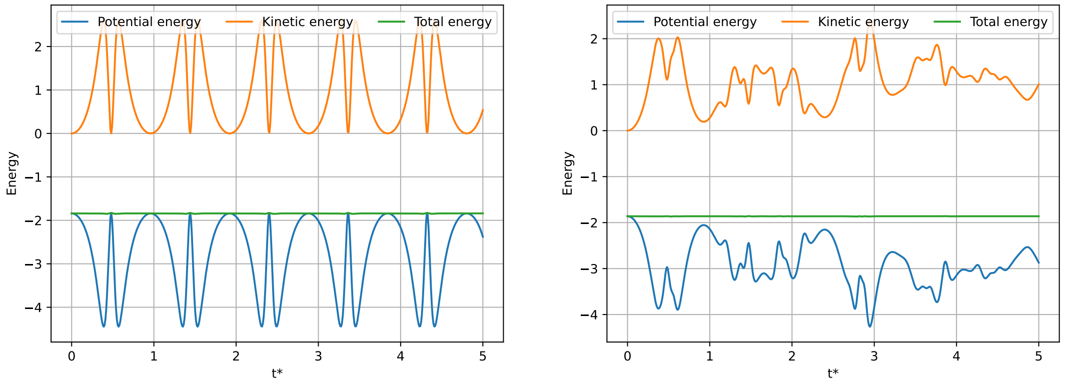

Figure 9: Energy plot for simulations with four atoms starting from rest.

We observe that we achieve a neat periodic movement for the atoms if the system is symmetric, which is reasonable as the forces on the atoms have different directions but the same magnitudes. We also observe that the energy is conserved in both simulations.

c. Initialisation

See `md-prosjekt/box.py`. Running the script with $n = 3$ and $d = 20/3$ we get $4 * 3^3 = 108$ atoms. See `md-prosjekt/xyz_files/3_c.xyz`.

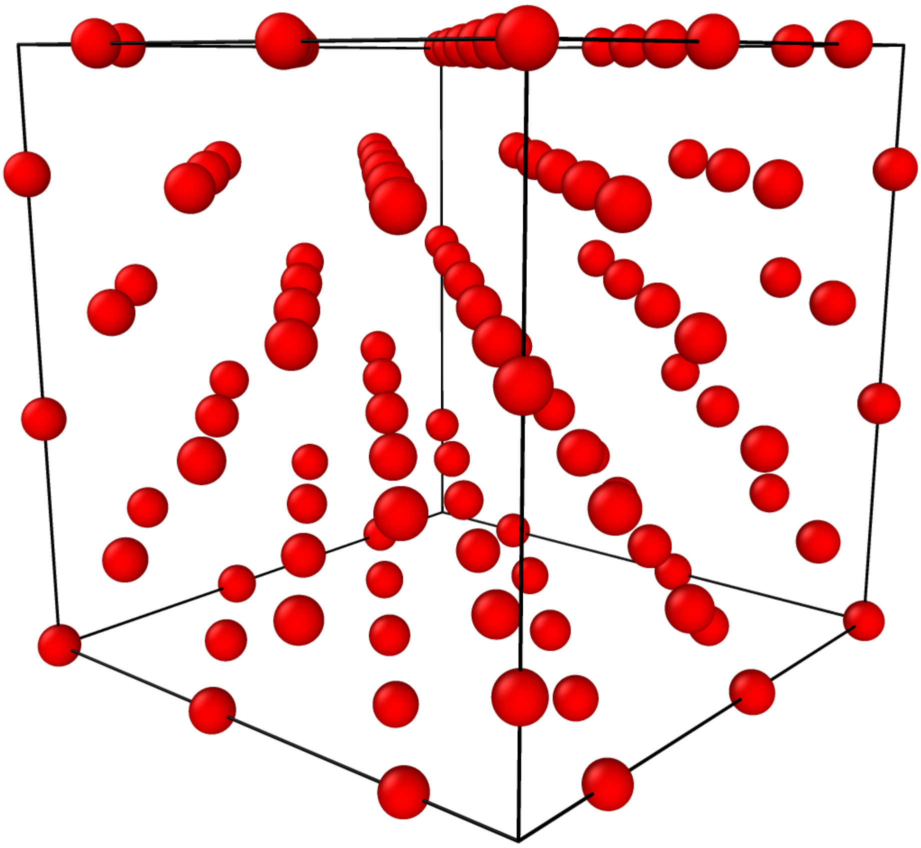

Figure 10: Ovito visualisation of the face-centered lattic structure with 108 atoms.

To show that the unit cell size corresponding to the density $\rho = 1.374$ g/cm^3 is $d = 1.7\sigma$, we use the general formula for density, $\rho =$total mass/total volume.

$$\begin{aligned} \rho &= \frac{n \cdot m}{L^3} \\ &= \frac{4 \cdot 39.95 \cdot 1.66 \cdot 10^{-27} \cdot 10^3 \mathrm{g}}{(1.7 \cdot 3.405 \cdot 10^{-10} \cdot 10^2 \mathrm{cm})^3} \\ &\approx 1.3676 \left[\frac{\mathrm{g}}{\mathrm{cm}^3}\right] \end{aligned}$$

where we have used that $n = 4$, $m = 39.95$ u, $L = 1.7\sigma$.

d. Many atoms, open boundary

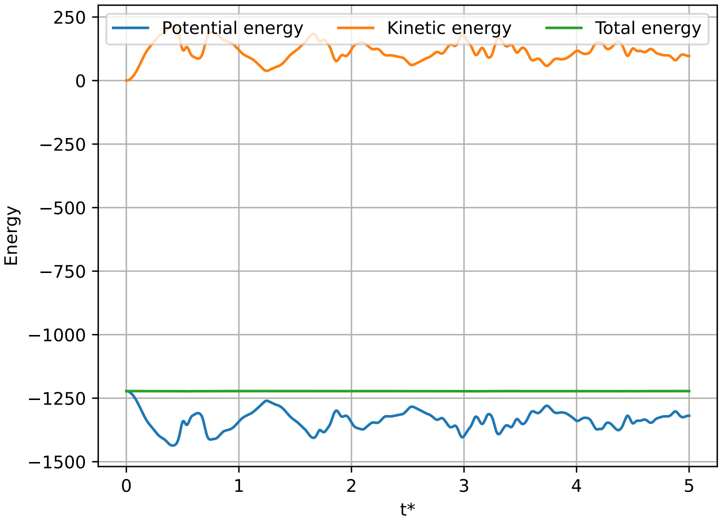

Figure 11: Energy plot for simulation with 256 atoms starting from rest.

The differences between the simulation with 256 atoms and the non-symmetrical simulation with four atoms are minor. We observe the same type of motion but simply with more atoms. Energy is also conserved in this case. Also note that the changes in potential and kinetic energy is initially larger, but decrease quickly as the system reaches equilibrium.

e. Boundary conditions

I chose to implement periodic boundary conditions. It is needed to calculate the shortest distance between two atoms and the shortest path might pass through the walls of the bounding box. I added the following to the function that calculates the distance between atoms:

```
if self.bound:
    r -= np.around(r/self.L)*self.L
```

To move the atoms through the walls I added the following line to my integration loop:

```
if self.bound:
    x_ = np.floor(x[i+1]/self.L)*self.L
    x[i+1] -= x_
    self.wallcount[i+1] = self.wallcount[i] + x_
```

The wallcount variable will be used later in the project to calculate mean square displacement and is an array with zeros as default.

To test the boundary conditions, we simulate a single atom starting at [1, 0, 0] with velocity [1, 0, 0]. The length of the bounding box is set to 2.

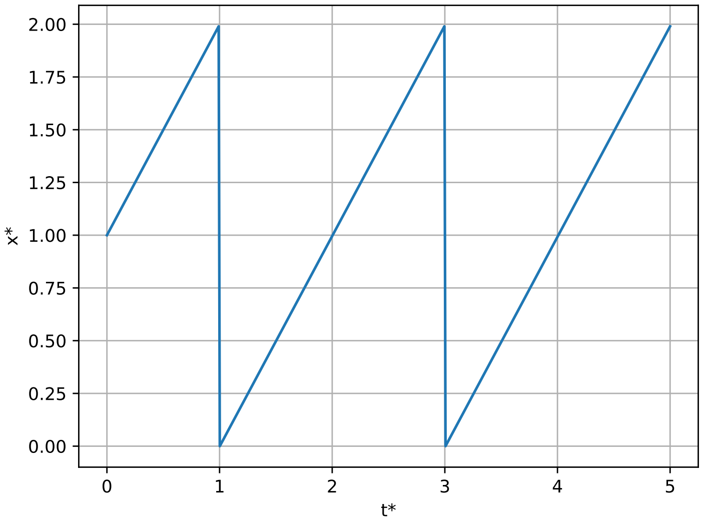

Figure 12: Plot of x-coordinate of atom with periodic boundaries.

The motion is as expected, and our implementation of periodic boundaries works.

4 Science

See `md-prosjekt/task4.py`. All the simulations are run with $\Delta t = 0.01, d = 1.7$ and periodic boundary conditions. The following graphs are from a simulation with 4000 atoms for 10 t^*. It took almost three hours to simulate.

a. Temperature

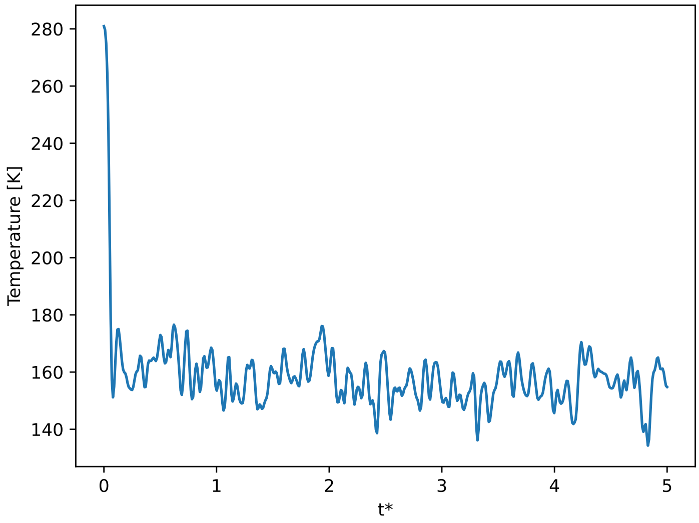

Figure 13: Plot of temperature for 108 atoms with initial temperature of 300K.

To find an initial temperature that yields equilibrium temperature around 95K, I ran 10 simulations with 108 atoms for 5 t^* and averaged the temperature for the last 4.5 t^*. Through trial and error I found that an initial temperature of 180K yields an equilibrium temperature around 95K. The rest of the simulations are run with this initial temperature.

b. Velocity autocorrelation and diffusion coefficient

I am too happy with my one-liner to not include it here:

```python
def vac(self) -> np.ndarray:
    return np.sum(np.einsum('ijk,jk->ij', self.v, self.v0)/np.einsum('ij,ij->i',
        self.v0, self.v0), axis=1)/self.n
```

The velocity autocorrelation plotted in figure 14 is similar to that of A. Rahman and looks to be stable around 0 as it should. We observe minor changes in calculating autocorrelation from equilibrium, but the curve does indeed flatten out a bit. It might be better to run multiple simulations and averaging the results.

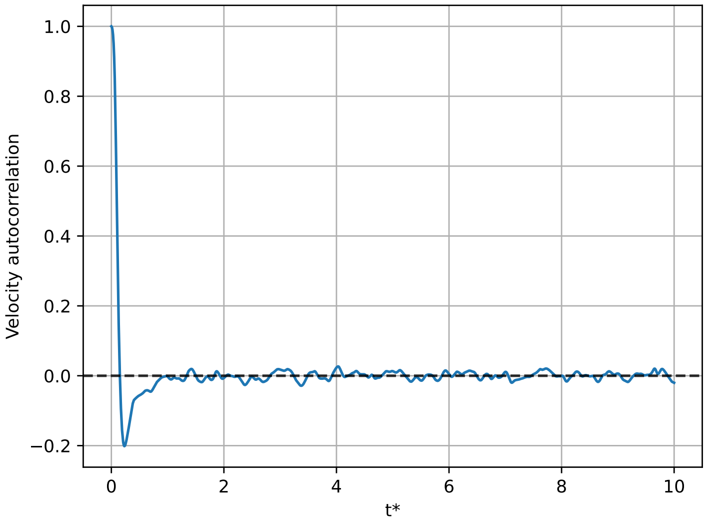

Figure 14: Velocity autocorrelation for 4000 atoms.

To estimate the diffusion coefficient we integrate the velocity autocorrelation from 0 to 3. We see that it converges after around 1.5 t^*. The value differs for each simulation, but is approximately 10^{-2}. Note that this is in scaled units, and cannot be directly compared to the value found by A. Rahman.

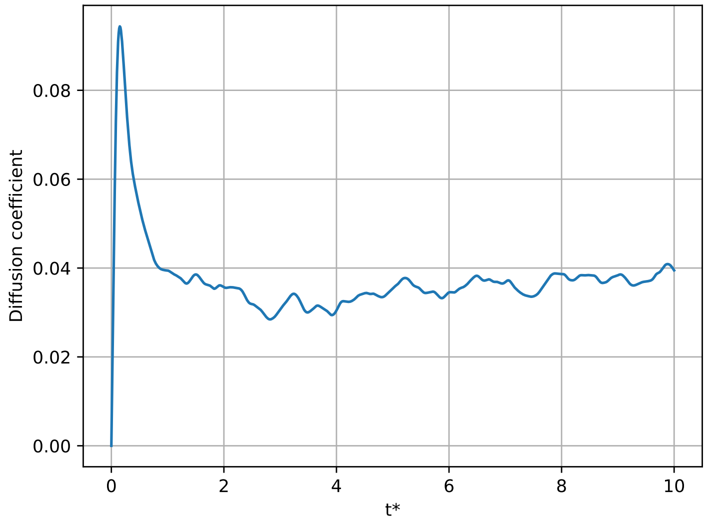

Figure 15: Diffusion coefficient for 4000 atoms.

c. Mean square displacement and diffusion coefficient

We can also find the diffusion coefficient by looking at the slope mean square displacement over time. This is where we have to use our wallcount, which is the variable that counts how many times each atom moves through any wall in any direction.

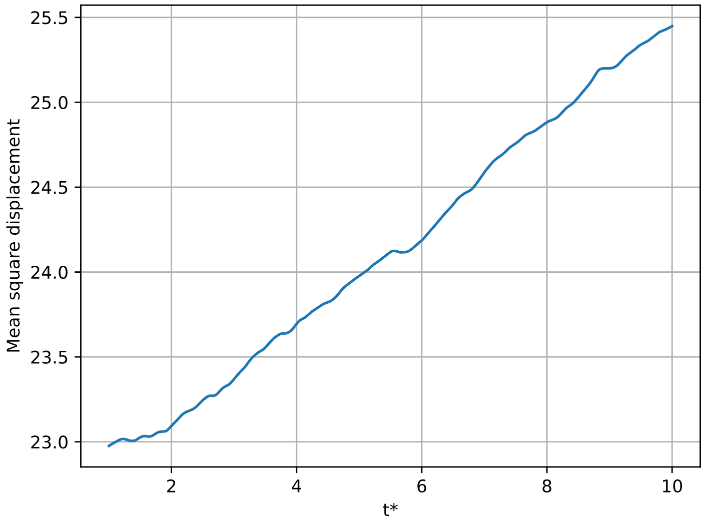

Figure 16: Mean square displacement for 4000 atoms.

The slope of the linear part is approximately $0.3{r^*}^2/t^*$. This is $1.54 \cdot 10^{-4}$ $\mathrm{cm^2 s^{-1}}$, which is only a single order of magnitude larger the value found by A. Rahman.

d. Radial distribution function

The radial distribution function is not vectorized, and therefore takes quite a while. I added a progress bar instead. It used a bit less than 2 hours for 4000 atoms for 10 t^*. The results are shown below.

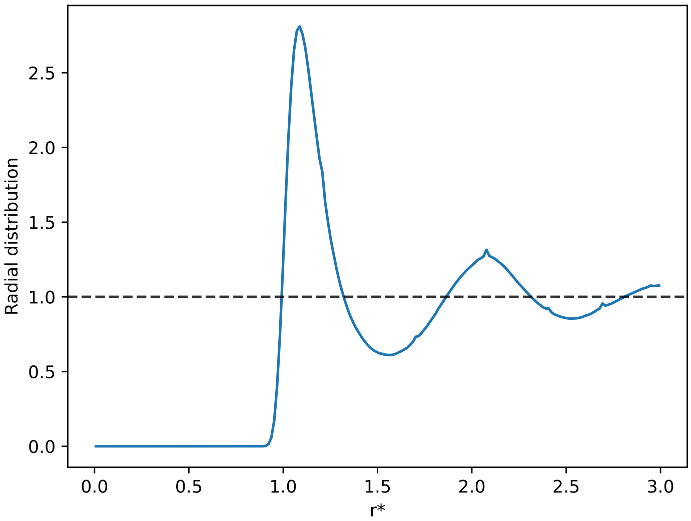

Figure 17: Radial distribution for 4000 atoms.

A. Rahman used around 45 seconds per iteration when simulation 864 atoms. Note that he used a potential cutoff $r_c^* = 2.25\sigma$, while my program uses $r_c^* = 3\sigma$, meaning that I have to calculate more per iteration. Yet, my program averages 12 iterations per second with $\Delta t = 0.01$ for 864 atoms. This is approximately 500 times faster as A. Rahman.