FYS4460 -
First compulsory project

Mathilde Nygaard Kamperud

February 22, 2013

1 Abstract

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.

$$\begin{aligned} \mathbf{r_0} &= [0,0,0] \\ \mathbf{r_{xy}} &= \left[\frac{b}{2},\frac{b}{2},0\right] \\ \mathbf{r_{yz}} &= \left[0,\frac{b}{2},\frac{b}{2}\right] \\ \mathbf{r_{zx}} &= \left[\frac{b}{2},0,\frac{b}{2}\right] \end{aligned} \tag{1}$$

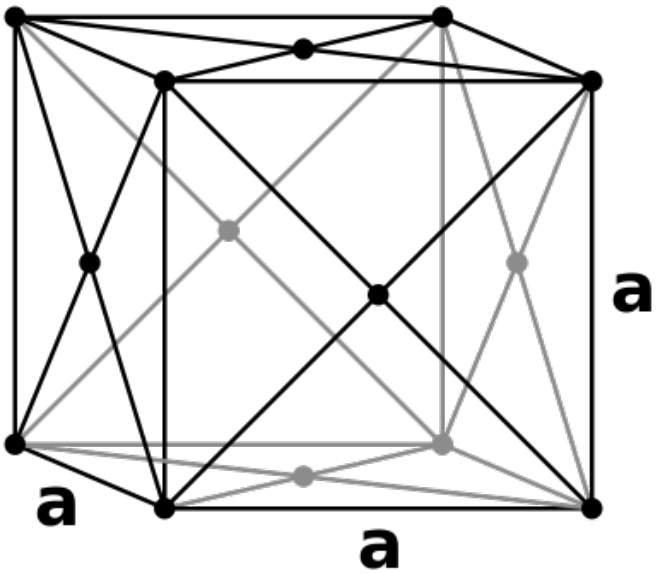

Figure 1: A face-centered cubic lattice unit.

where b is the distance between the base positions. For solid Argon $b = 5.260$ÃĚ 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:

$$\begin{aligned}\mathbf{r_{base}} + \mathbf{r_0}\\ \mathbf{r_{base}} + \mathbf{r_1}\\ \mathbf{r_{base}} + \mathbf{r_2}\\ \mathbf{r_{base}} + \mathbf{r_3}.\end{aligned} \tag{2}$$

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.

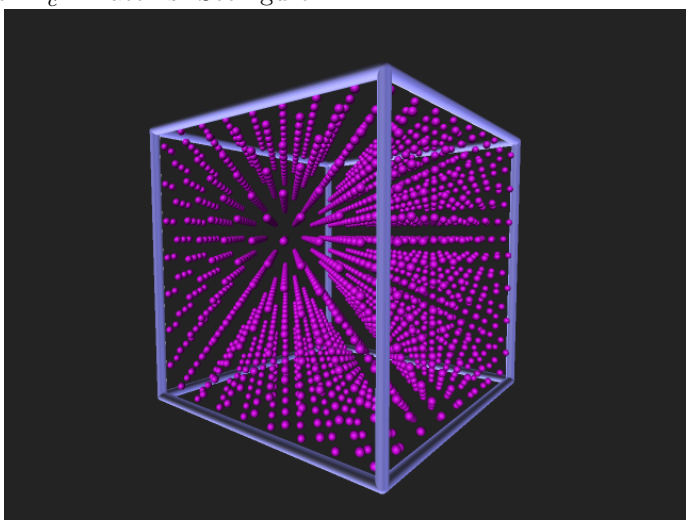

Figure 2: Visualization of solid Argon

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.

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

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

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

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

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

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

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.

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

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 plottet in figure 3. To plot this I used $\epsilon = \sigma = 1$, but that is not important.

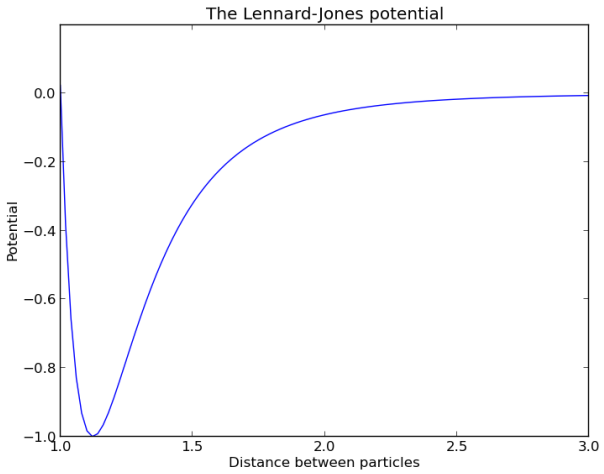

Figure 3: The Lennard-Jones potential

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.

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

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

$$r^6 = 2\sigma^6 \tag{11}$$

$$r = 1.1225 * \sigma. \tag{12}$$

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

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 disappeard, 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 neightbors copy-particle. When we calculate the forces we need the distances between the particles. They are calculated using the minimum image convention.

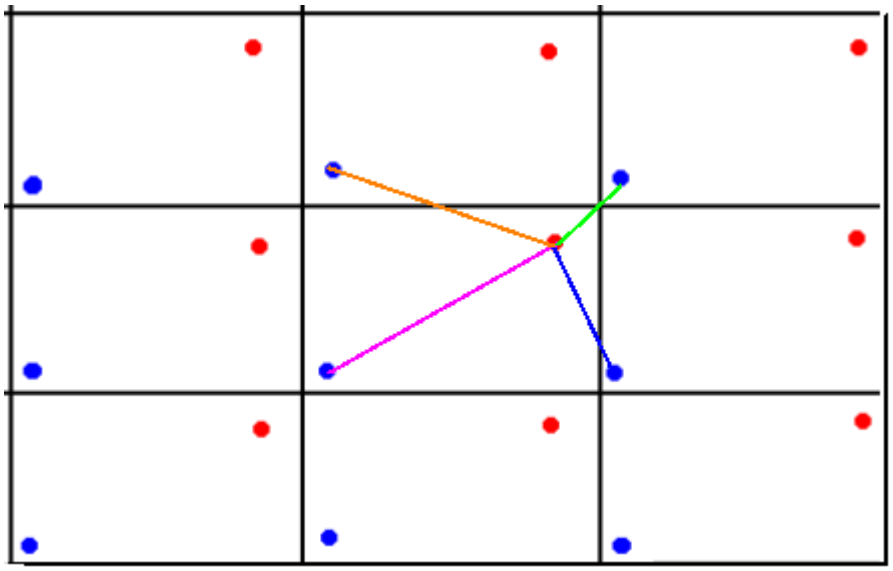

Figure 4: The system with periodic boundary conditions. Here with two particles. The lines indicate the choices we have when calculating the distance between the red and the blue particle. Look at figure 4. This drawing illustrates the periodic boundary conditions. The lines in the drawing indicate the 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.

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 equtions 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 aposthrophe. $A' \to A$.)

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

2.2.3 Initial velocity

For the intial 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 standar deviation is $\sigma_v = \sqrt{T'} = \sqrt{T/T0} = 0.914$. This fits quite good with the histogram in figure 5.

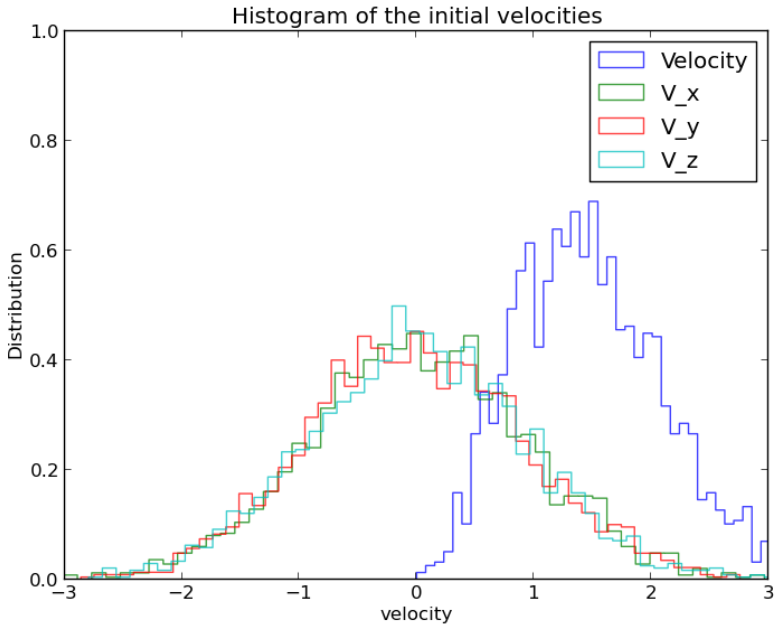

Figure 5: A histogram over the intital velocities of the system. Here in MD-units.

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. 500 timesteps -> 121 s

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

# time steps	Verlet Solver-tid(s)	CellSolver-tid(s)
100	24	25
500	120	123
1000	241	245

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

# time steps	Verlet Solver-tid(s)	CellSolver-tid(s)
100	274	182
500	895 (15 min)	
1000		

2.4 Neighbour lists

The Lennard- Jones interactions is short ranged, and can be neglected for distances over $r_{cut} =$ 3Ł0. 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_cell.gif.

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-compontents 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:

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

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

key-words: velocity distributions, tidsutvikling av energi, fluktuasjoner, temperature, pressure, results.

5 The diffusion constant

key-word: find it! radial distribution functions.