Atomic modeling of argon

Project 5, FYS-3150

Ina K. B. Kullmann, candidate nr: 20

Abstract

Goal: Find the melting temperature of the solid.

The aim of this project is to numerically find the critical temperature for the two dimentional Ising model by using the metropolis algorithm. We will first test the implementation of the algorithm carefully, first by comparing with theoretical values calculated for a small system. Then we will see if the algorithm behaves as expected according to our physical intuition for a larger system.

When we have found a estimate for the critical temperature we will compare it to Lars Onsagers analytical result.

All source codes can be found at: `https://github.com/inakbk/molecular-dynamics-fys3150`.

Contents

1 Introduction

Molecular dynamics (MD) is a computer simulation method used to study atoms and molecule structure and movement. In a MD simulation the atoms or molecules are allowed to interact trough a force given by a potential for a given time. This makes it possible to study the systems development over time.

MD is a a type of N-body simulation since the simulation often consists of a large number of atoms or molecules. It is therefore possible to use MD to study stastistical properties of a large system consisting of N such atoms or molecules. For systems that obey the ergodic hypothesis the evolution of a single molecular dynamics simulation may be used to determine macroscopic thermodynamic properties of the system. This is because the time averages of an ergodic system correspond to microcanonical ensemble averages[1].

Often the main motivation to use Molecular dynamics is that it is not possible to determine properties of the system analytically because of the large number of particles. The main limitation for the numerical simulation is the computer recources available, but also cumulative errors in the numerical integration. The first is solved by applying periodic boundary conditions while the latter is solved by proper selection of algorithms and parameters. In this paper we will have a look at two numerical integration methods; the Euler-Cromer method and the Velocity Verlet integrator.

In this paper we will study the properties of a large system consisting of Argon atoms. And compare with experimental data(?). We will have a constant number of particles, a constant volume and a more or less constant engergy (depending on the integrator). We are more interested in the statistical properties of the system than in the individual motion of each of the particles. We want to sample microstates from the microcanonical ensemble (NVE). (?se over avsnittet over?)

The applications of MD is many ranging from chemical physics, materials science and the modelling of biomolecules. What areas of physics can it be used in? Chemistry and biology?(fra oppg) see 'Areas of application and limitations' at `https://en.wikipedia.org/wiki/Molecular_dynamics` and google

2 Theory

3 Numerical methods

3.1 Creating the lattice

We want to start out the simulation with the atoms in a crystal structure. We will choose the face-centered cubic (FCC) lattice which is a stable structure for the Lennard-Jones potential that will be used. The lattice is build up by unit cells, a group of attoms, so that a larger system can be created by repeating these cells in space. The FFC lattice unit cell is of size

[1]`https://en.wikipedia.org/wiki/Molecular_dynamics` 3.dec 11:25

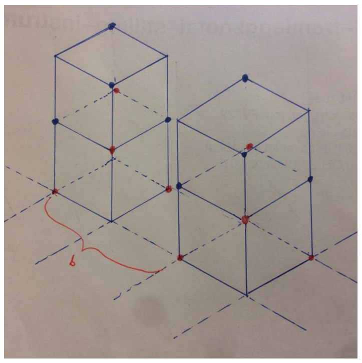

Figure 1: An illustration of the desired crystal structure when simulation an Argon crystal. The length of the sides of the unit cells used in this paper have half the length of the yellow box marked in the drawing.

bÅ and consists of four atoms with local coordinates

$$\mathbf{r}_1 = 0\hat{\mathbf{i}} + 0\hat{\mathbf{j}} + 0\hat{\mathbf{k}}, \tag{1}$$

$$\mathbf{r}_2 = \frac{b}{2}\hat{\mathbf{i}} + \frac{b}{2}\hat{\mathbf{j}} + 0\hat{\mathbf{k}}, \tag{2}$$

$$\mathbf{r}_3 = 0\hat{\mathbf{i}} + \frac{b}{2}\hat{\mathbf{j}} + \frac{b}{2}\hat{\mathbf{k}}, \tag{3}$$

$$\mathbf{r}_4 = \frac{b}{2}\hat{\mathbf{i}} + 0\hat{\mathbf{j}} + \frac{b}{2}\hat{\mathbf{k}}. \tag{4}$$

where $b = 5.26$ is the lattice constant. We want to construct a larger system with $N \times N \times N$ such unit cells next to each other. The orgin of unit cell (i, j, k) is

$$\mathbf{R}_{i,j,k} = i\hat{\mathbf{u}}_1 + j\hat{\mathbf{u}}_2 + k\hat{\mathbf{u}}_3, \tag{5}$$

where $i = 0, 1, ..., N_x - 1, j = 0, 1, ..., N_y - 1, k = 0, 1, ..., N_z - 1$. The unit vectors of the unit cells are scaled with the lattice constant b so that

$$\hat{\mathbf{u}}_1 = b\hat{\mathbf{i}}, \quad \hat{\mathbf{u}}_2 = b\hat{\mathbf{j}}, \quad \hat{\mathbf{u}}_3 = b\hat{\mathbf{k}}. \tag{6}$$

The coordinates of particle one to four is then given by:

$$x_1 = 0 + ib \qquad x_2 = \frac{1}{2} + ib \qquad x_3 = 0 + ib \qquad x_4 = \frac{1}{2} + ib$$
$$y_1 = 0 + jb \qquad y_2 = \frac{1}{2} + jb \qquad y_3 = \frac{1}{2} + jb \qquad y_4 = 0 + jb$$
$$z_1 = 0 + kb \qquad z_2 = 0 + kb \qquad z_3 = \frac{1}{2} + kb \qquad z_4 = \frac{1}{2} + kb$$

In figure 1 we see an illustration of two unit cells placed side by side. A cubic lattice is formed by placing $N \times N \times N$ such unit cells side by side.

Each of the sides of the simulation box has the length $L = bN$ so that the size of the whole system or the volume of the cubic lattice is given by $V = L^3 = (bN)^3$. The density of the system is then:

$$\rho = \frac{M_{tot}}{V} = \frac{4 * N^3 * m}{(bN)^3} = \frac{4m}{b^3}$$

where m is the mass of one atom and is equal for all the atoms. We can see that the density will be constant as long as the mass and the lattice constant is held constant.

should include particle density not mass density! (?)

3.2 The Maxwell-Boltzmann distrubution (giving velocities)

The atoms are usually given velocities according to the Maxwell-Boltzmann distribution so that

$$P(v_i)\mathrm{d}v_i = \left(\frac{m}{2\pi k_B T}\right)^{1/2} \exp\left(-\frac{m v_i^2}{2k_B T}\right)\mathrm{d}v_i, \tag{7}$$

where m is the mass of the atom, k_B is Boltzmann's constant and T is the temperature. We recognize this as a normal distribution with zero mean and standard deviation $\sigma = \sqrt{k_B T/m}$. This is a good choice because bla bla bla bla (?)

But this will result in a nonzero net momentum in the system which we will remove. The momentum in a given direction for one atom is $p_i = m v_i$ where the mass is m and v_i is the velocity in direction i. The total momentum in one direction, assuming that all particles have the same mass, is given by:

$$p_{tot} = m(v_{i,1} + v_{i,2} + v_{i,3} + \cdots + v_{i,n})$$

where n is the total number of atoms in the simulation.

We want to remove this momentum evenly from all the particles. The momentum we want to remove from each particle is then:

$$p_{rm} = \frac{p_{tot}}{n}$$

so that p_i' is the momentum for one particle after removing the total momentum:

$$p_i' = p_i - p_{rm}$$

If we write this out

$$\begin{aligned} m v_i' &= m v_i - \frac{m(v_{i,1} + v_{i,2} + v_{i,3} + \cdots + v_{i,n})}{n} \\ \Rightarrow v_i' &= v_i - \frac{v_{i,1} + v_{i,2} + v_{i,3} + \cdots + v_{i,n}}{n} \\ &= v_i - \overline{v_i} \end{aligned}$$

We see that to remove the total momentum we only have to substract the average velocity from each of the particles velocity.

3.3 Periodic boundaryconditions (PBCs)

To avoid problems with boundary effects we will apply periodic boundary conditions. This has a great analogy to 'old' video games such as Snake 2. If the snake head passes through one side of box, it re-appears on the opposite side with the same velocity. In the system of atoms this would mean that if an atom should leave the simulation box at one side it will enter on the oposite side with the same values for the physical parameters as it had before it left. This also implies that an attom at the edge of the box will interact with an atom at the opposite side of the box so that every atom have the same number of 'neighbours'.

MD simulations that use periodic boundary conditions have a large number of unit cells (defined above?), a group of atoms, so that a larger system can be created by repeating these cells in space.

The size of the simulation box must also be large enough to prevent unphysical behaviour. If the box is too small one unit cell (a few atoms) might interact with itself. In the Snake analogy this would mean that the "head" interacts with or bites its own "tail" trough the wall which is allowed in the game, but not very physical. Thus the box size have to be large enough relative to the size of a unit cell, length of the simulation and the desired accuracy[2].

3.4 Intergration methods

During the first implementation of the MD code we have used the Euler-Cromer method to find the next time step for the atoms. (bla bla legge til likninger?)

In MD the Velocity Verlet integrator is usually used. This is because other integration algorithms tend to drift the energy over time, which we want to avoid. The Velocity Verlet algorithm consists of three steps (in addition to computing the forces, see section 3.5)

$$\mathbf{v}(t+\Delta t/2) = \mathbf{v}(t) + \frac{\mathbf{F}(t)}{m}\frac{\Delta t}{2} \tag{8}$$

$$\mathbf{r}(t+\Delta t) = \mathbf{r}(t) + \mathbf{v}(t+\Delta t/2)\Delta t \tag{9}$$

$$\mathbf{v}(t+\Delta t) = \mathbf{v}(t+\Delta t/2) + \frac{\mathbf{F}(t+\Delta t)}{m}\frac{\Delta t}{2}. \tag{10}$$

sympletic integrator bla bla bla?

To measure the drift of the energy we will use the standard deviation of the total energy E:

$$\sigma_E = \sqrt{\langle E^2\rangle - \langle E\rangle^2}$$

that we will calculate using the Euler-Cromer method and the Velocity-Verlet integrator and compare the results.

Will also compare cumputing time (plot this!?)

say something more about energy drift?

(discussed in the lecture notes.?) see `https://en.wikipedia.org/wiki/Verlet_integration#Velocity_Verlet` on conserved quantities

bla bla bla

3.5 The Lennard-Jones potential (interaction/forces)

In this project we will use the Lennard-Jones potential which is a mathematically simple model that approximates the interaction between a pair of neutral atoms or molecules. Due to its computational simplicity, the Lennard-Jones potential is used extensively in computer simulations even though more accurate potentials exist[3].

[2] `https://en.wikipedia.org/wiki/Periodic_boundary_conditions#Practical_implementation:_continuity_and_the_minimum_image_convention`

[3] `http://en.wikipedia.org/wiki/Lennard-Jones_potential`

The potential calculates the energy between two atoms i and j as

$$U(r_{ij}) = 4\epsilon\left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right], \tag{11}$$

where $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$ is the distance from atom i to atom j, ϵ is the depth of the potential well (units energy) and σ is the distance at which the potential is zero.

The parameters in the potential can be fitted to reproduce experimental data. For argon, optimal values of the parameters are:

$$\frac{\epsilon}{k_B} = 119.8\text{ K}, \sigma = 3.405\text{ Å}. \tag{12}$$

With these parameters the L-J potential reproduces equilibrium thermodynamic properties that are in good agreement with experimental values for argon.

In our code, we use the so called MD units (see appendix **??**). (finne enheten og verdier til parameterne?)

The force is calculated by taking the negative gradient of the potential

$$\mathbf{F}(r_{ij}) = -\nabla U(r_{ij}), \tag{13}$$

giving the x-component (the other components are calculated the same way)

$$F_x(r_{ij}) = -\frac{\partial U}{\partial r_{ij}}\frac{\partial r_{ij}}{\partial x_{ij}}, \tag{14}$$

where x_{ij} is the x-component of $\mathbf{r}_{ij}$. Calculating the differensials separately:

$$\begin{aligned}
\frac{\partial U}{\partial r_{ij}} &= \frac{\partial}{\partial r_{ij}}4\epsilon\Big[\Big(\frac{\sigma}{r_{ij}}\Big)^{12} - \Big(\frac{\sigma}{r_{ij}}\Big)^{6}\Big] = 4\epsilon\Big[\sigma^{12}\frac{\partial}{\partial r_{ij}}r_{ij}^{-12} - \sigma^{6}\frac{\partial}{\partial r_{ij}}r_{ij}^{-6}\Big] \\
&= 4\epsilon\Big[-12\sigma^{12}r_{ij}^{-13} - (-6)\sigma^{6}r_{ij}^{-7}\Big] = \frac{24\epsilon}{\sigma}\Big[\Big(\frac{\sigma}{r_{ij}}\Big)^{7} - 2\Big(\frac{\sigma}{r_{ij}}\Big)^{13}\Big] \\
r_{ij} &= |\mathbf{r_i} - \mathbf{r_j}| = |x_i\hat{\mathbf{i}} + y_i\hat{\mathbf{j}} + z_i\hat{\mathbf{k}} - x_j\hat{\mathbf{i}} + y_j\hat{\mathbf{j}} + z_j\hat{\mathbf{k}}| = |(x_i - x_j)\hat{\mathbf{i}} + (y_i - y_j)\hat{\mathbf{j}} + (z_i - z_j)\hat{\mathbf{k}}| \\
&= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2} = \sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2} \\
\frac{\partial r_{ij}}{\partial x_{ij}} &= \frac{\partial}{\partial x_{ij}}\sqrt{x_{ij}^2 + y_{ij}^2 + z_{ij}^2} = \frac{1}{2r_{ij}}\cdot 2x_{ij} = \frac{x_{ij}}{r_{ij}}
\end{aligned}$$

gives

$$\begin{aligned}
F_x(r_{ij}) &= -\frac{24\epsilon}{\sigma}\Big[\Big(\frac{\sigma}{r_{ij}}\Big)^{7} - 2\Big(\frac{\sigma}{r_{ij}}\Big)^{13}\Big]\cdot\frac{x_{ij}}{r_{ij}} \\
&= \frac{24\epsilon}{\sigma^2}\Big[2\Big(\frac{\sigma}{r_{ij}}\Big)^{14} - \Big(\frac{\sigma}{r_{ij}}\Big)^{8}\Big]x_{ij}
\end{aligned}$$

The force in y and z direction can be calculated the same way giving:

$$\begin{aligned}
\mathbf{F} &= F_x\hat{\mathbf{i}} + F_y\hat{\mathbf{j}} + F_z\hat{\mathbf{k}} \\
&= \frac{24\epsilon}{\sigma^2}\Big[2\Big(\frac{\sigma}{r_{ij}}\Big)^{14} - \Big(\frac{\sigma}{r_{ij}}\Big)^{8}\Big]\Big(x_{ij}\hat{\mathbf{i}} + y_{ij}\hat{\mathbf{j}} + z_{ij}\hat{\mathbf{k}}\Big)
\end{aligned}$$

(another section???)

In section 3.4 we see that to calculate the next time step for the particles we need to calculate the forces. This is very time consuming because we have to sum over all pairs of particles to find the force for one particle. Therefore a lot of time can be saved if the code for the calculation of forces is as efficient as possible.

A better numerical expression for the force is

$$\begin{aligned}\mathbf{F}(r_{ij}) &= \frac{24\epsilon}{\sigma^2}\left(\frac{\sigma}{r_{ij}}\right)^2\left(\frac{\sigma}{r_{ij}}\right)^6\left[2\left(\frac{\sigma}{r_{ij}}\right)^6 - 1\right]\left(x_{ij}\hat{\mathbf{i}} + y_{ij}\hat{\mathbf{j}} + z_{ij}\hat{\mathbf{k}}\right)\\ &= 24\epsilon\sigma^6 r_{ij}^{-2} r_{ij}^{-6}\left(2\sigma^6 r_{ij}^{-6} - 1\right)\end{aligned}$$

and for the potential energy

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

so that the powers of r_{ij} is calculated as few times as possible, r_{ij}^{-6} can be calculated just once for each loop (define?).

Another smart way to make a more efficient code when calculating the forces is to realize that in the limit $r_{ij} \to \infty$ both the force and the potential energy is zero. In figure 2 the potential energy (red) and the force (blue) plotted against the distance between two atoms r_{ij}. We can see that the energy and the force reaches zero for large values of r_{ij}. The point for which the potential energy and the force is nearly zero, r_{cut}, is marked with a dot in the figure. A good approximation is therefore to stop calculating forces and energies for distances larger than r_{cut} and set all forces and energies for $r_{ij} > r_{cut}$ to zero.

The implementation of r_{cut} will save the algorithm a lot of time and still be a good approximation since the force and the potential energy is esensially zero anyway. The only problem is that the potential energy function will be discontinuously at the point r_{cut} when the energy is set to zero after r_{cut}. A solution to this is to shift the potential function slightly so that the energy is zero at r_{cut} and still set to zero for $r_{ij} > r_{cut}$. The shifted potential function is marked with green in figure 2. This will only shift the values for the potential and total energy, but it will not affect the physics of the problem. The statistical properties of the system will not be affected by the shift in energy.

The minimum image convention (due to calculation of forces) When an atom leave the simulation box it must re-enter the box on the opposite side. An atom at the edge of the simulation box must also interact with an atom at the other side of the box 'through' the wall, or interact with the image of the simulation box. But this atom is also contained inside the box so the atoms interact twice, once through the wall and once from across the box. To avoid counting this interaction twice we will apply the minimum image convention. An atom will interact with the other atoms only once and then choose the image (or the simulation box) that gives the minimum distance[4]:

```
if(periodic_x) then
  dx = x(j) - x(i)
  if (dx >   x_size * 0.5) dx = dx - x_size
  if (dx <= -x_size * 0.5) dx = dx + x_size
endif
```

[4]https://en.wikipedia.org/wiki/Periodic_boundary_conditions#Practical_implementation:_continuity_and_the_minimum_image_convention

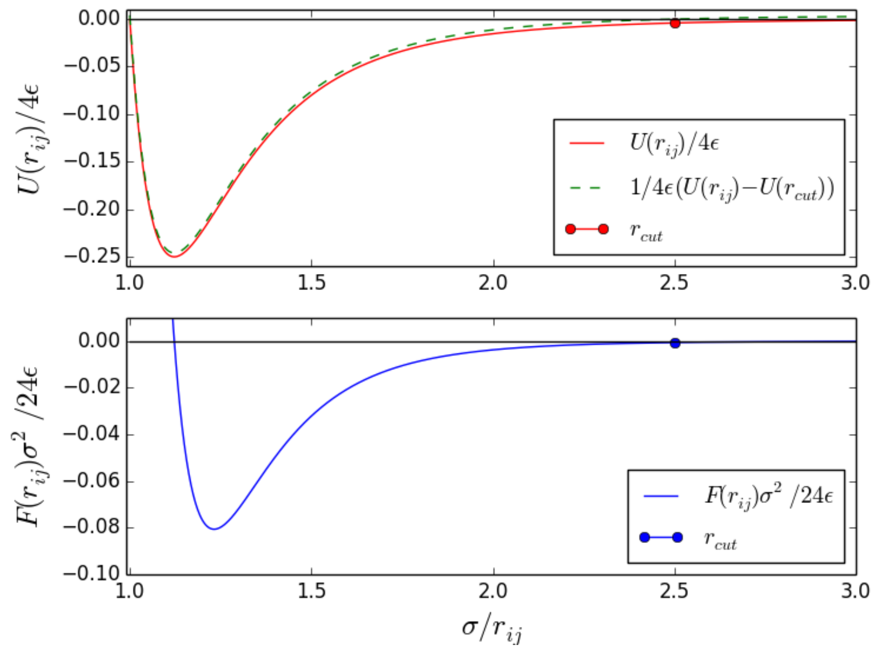

Figure 2: Top: The Lennard-Jones potential (red), or the potential energy $U(r_{ij})$ between two atoms i and j plotted against the distance between atom i and j, r_{ij}. The point $(r_{cut}, U(r_{cut}))$ is marked with a red dot. The shifted potential function is also plotted in green. **Bottom:** The force (blue) between two atoms i and j plotted against the distance between atom i and j, r_{ij}. The point $(r_{cut}, F(r_{cut}))$ is marked with a blue dot.

(include drawing??)

This leads to two possible strategic choices: (A) 'fold back' particles into the simulation box when they leave it, or (B) let them go on into the other side of the box?

....ahh, this is relevant for computing forces/potentials! An atom which has passed through one face of the simulation box should re-enter through the opposite face-or its image should do it. Evidently, a strategic decision must be made: Do we (A) âĂIJfold backâĂİ particles into the simulation box when they leave it, or do we (B) let them go on (but compute interactions with the nearest images)? The decision has no effect on the course of the simulation, but if the user in interested in mean displacements, diffusion lenghts, etc., the second option is preferable.

3.6 Calculating physical properties of the system

The total potential energy V in the system is computed by summing over all pairs of atoms (counting each pair only once)

$$V = \sum_{i>j} U(r_{ij}). \tag{15}$$

where $U(r_{ij})$ are the Lennard-Jones potential given in section 3.5. The kinetic energy is defined as

$$E_k = \sum_{i=1}^{N_{\text{atoms}}} \frac{1}{2} m_i v_i^2, \tag{16}$$

where m_i and v_i is the mass and the speed of atom i. The total energy of the system is then $E = V + E_k$. (which we will study if is conserved or not, see σ_E ??)

We will also calculate an estimate of the temperature through the equipartition theorem[5]

$$\langle E_k \rangle = \frac{3}{2} N_{\text{atoms}} k_B T. \tag{17}$$

We can use this to define an *instantaneous* temperature

$$T = \frac{2}{3} \frac{E_k}{N_{\text{atoms}} k_B}. \tag{18}$$

The diffusion constant of the system can be used to measure the melting temperature. We use the Einstein relation that relates the so called mean square displacement (MSD) to the diffusion constant

$$\langle r^2(t) \rangle = 6Dt, \tag{19}$$

where D is the diffusion constant, t is time. The mean square displacement for atom i is calculated as

$$r_i^2(t) = |\mathbf{r}_i(t) - \mathbf{r}_i(0)|^2, \tag{20}$$

where $\mathbf{r}_i(t)$ is the position of atom i at time t.

[5] See for example D. Schroeder's *An Introduction to Thermal Physics* for details. better reference?

The diffusion constant is then given by

$$D = \frac{\langle r^2(t) \rangle}{6t} =$$

(for a given temperature (T_i ?).)

Atoms in a solid will not diffuse (move?) much, so a solid will have a diffusion constant close to zero. A plot of the diffusion constant as a function of temperature will therefore reveale the melting temperature because the solid melts when the diffusion constant is non zero.

write something about ovito introduction(?)

---

4 Results

4.1 Implementation and testing of the algorithm by using Ovito (code?) and choice of parameters

bla bla bla

Looking in `Ovito`

Initial positions To the left in figure 3 we see a screen shot of the initial frame of the simulation box with one unit cell from `Ovito` (ref?). We see that this is exactly the unit cell introduced in section 3.1 and figure 1. To the right in figure 3 we see a screen shot of the initial frame of the simulation box with 10 unit cells from `Ovito`, creating a cubic lattice. After looking at the initial frame of the simulation box in `Ovito` it seems safe to assume that the program produces the desired initial positions for the Argon atoms.

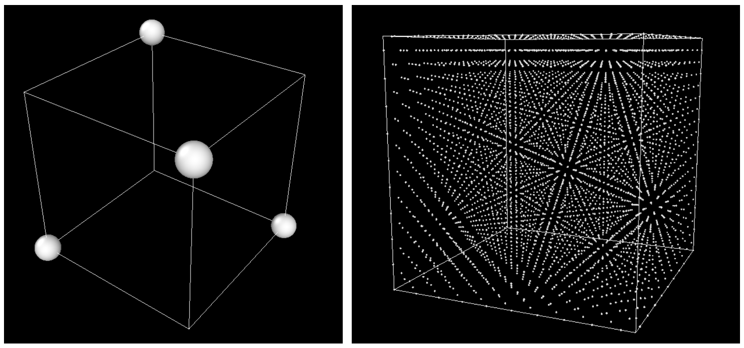

Figure 3: Left: A screen shot of the initial frame of the simulation box with one unit cell from `Ovito`. **Right:** A screen shot of the initial frame of the simulation box with 10 unit cells from `Ovito`, creating a cubic lattice.

Before forces At first the particles was unaffected by forces in the simulation. Before the Lennard-Jones potential and the calculation of the forces was implemented the particles was only given an initial speed by the maxwell boltzmann distrubution and should thus move in a straight line. In `Ovito` the particles moved in a straight line, and the simulation box expanded as the particels reached further out. After the periodic boundary conditions was implemented in the code the simulation box maintained a constant size when looked at in `Ovito`. One could also see that the particles still moved in a straight line, but they would jump to the oposite side of the box if they where to move out of the box.

After the implementation of the forces one could easy see (in `Ovito`) that the particles was contained inside the initial crystal structure, for instance at the temperature $T_i = 300$K. Just by looking at the simulation one could determine that the atoms formed a solid. At a high temperature, for instance $T_i = 1000K$ one could see that the atoms moved freely and that the crystal or solid most likely had melted. Screen shots igjen??

4.2 Comparing the two integrators

In the top plot in figure 4 we can see the standard deviation of the total energy σ_E plotted versus the timestep dt for the Euler-Cromer and Velocity Verlet integrator . The units on the axis are MN unist explained in section (?do it? explain in caption?) We see that the drift in the energy is significantly larger for the Euler-Cromer method than for the Velocity Verlet integrator and that it increases with increasing dt for the Euler-Cromer method. For the Velocity Verlet integrator the choice of dt does not matter much, the energy drift, or the value for σ_E is small for all values of dt. In the further calculations we have therefore chosen dt=0.05.

In the bottom plot in figure 4 we can see the execution time plotted versus the timestep for the Euler-Cromer and Velocity Verlet integrator. We see that there is not much difference in the execution time between the two integrators and that the dependence on the value of dt seems rather random.

Both plots in figure 4 where obtained with 5 unit cells in each direction, initial temperature $T_i = 2.5 \approx 300$ K, lattice constant $b = 5.26$Å and 1000 timesteps in the simulation. Similar plots are obtained when running with 10 unit cells in each direction.

In figure 5 we see the execution time plotted versus the number of unit cells in each direction for the Euler-Cromer and Velocity Verlet integrator. We see that for small number of unit cells the execution time is about the same for both integrators, they are equally fast. For 15 unit cells in each direction the two integrators deviate slightly in the execution time, but the execution time is quite large, $t_{exe} \approx 800$s ≈ 13min so that the deviation in the execution time between the integrators are small compared to the execution time. Thus the choice of integrator does not affect the execution time much and we should choose the integrator that gives the best physical results. The best physical results are when the drift in the energy or σ_E is as small as possible, we should therefore use the Velocity Verlet integrator. The further calculations are done with the Velocity Verlet integrator.

number of unit cells is also a measure of how many particles..include?

4.3 Equilibium time? and conserved energy

(conserved, ref to rcut?) In figure 6 we see the energy plotted versus the time with initial temperature $T_i = 2.5 \approx 300$ K for two different number of unit cells in each direction. We see that the total energy is conserved, or constant. This coresponds well with the low value of σ_E found in section 4.2 (figure 4). We can see that the system is initialized with high kinetic energy and low potential energy and that the two interchanges energy giving that the total energy is conserved. The system stabilizes after a short period of time, it reaches an

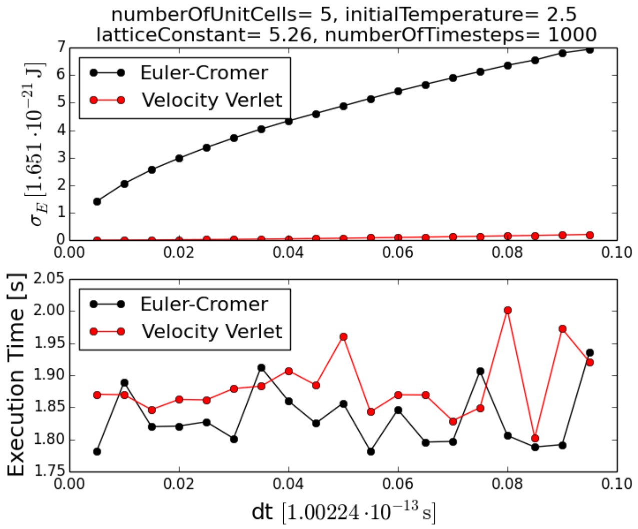

Figure 4: Top: The standard deviation of the total energy σ_E plotted versus the timestep dt for the Euler-Cromer and Velocity Verlet integrator with 5 unit cells in each direction, initial temperature $T_i = 2.5 \approx 300$ K, lattice constant $b = 5.26$Å and 1000 timesteps in the simulation. **Bottom:** The execution time plotted versus the the timestep dt for the Euler-Cromer and Velocity Verlet integrator with 5 unit cells in each direction, initial temperature $T_i = 2.5 \approx 300$ K, lattice constant $b = 5.26$Å and 1000 timesteps in the simulation.

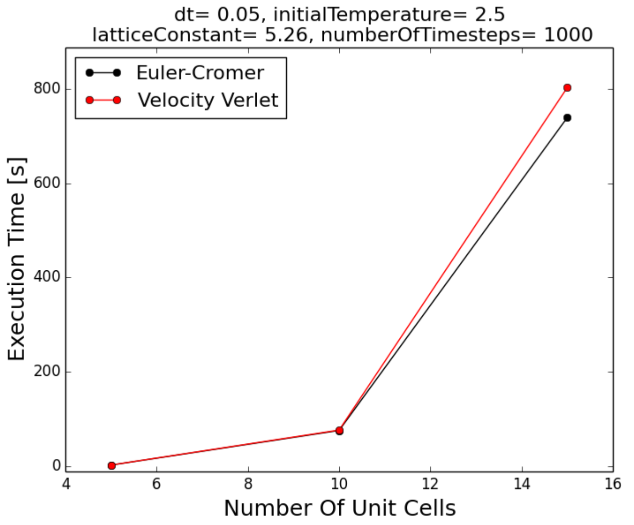

Figure 5: The execution time plotted versus the number of unit cells in each direction for the Euler-Cromer and Velocity Verlet integrator with initial temperature $T_i = 2.5 \approx 300$ K, lattice constant $b = 5.26$Å, $dt = 0.05 \approx 5 \cdot 10^{-15}$s and 1000 timesteps in the simulation.

equilibrium state when the kinetic and potential energy fluctuates around a constant energy. The equilibrium state is reached at least for $t = 10$, or after 20 timesteps.

The kinetic energy drops below the equilibrium value before it stabilizes. This is because the particles are all initialized with different velocities and move until they are stoped, at about the same time, by the forces of the surrounding particles before they find an equilibrium position and oscillates around this position.

The top plot in figure 6 is obtained with 5 unit cells in each direction and the bottom plot is obtained with 10 unit cells in each direction. We see that the plot with only 5 unit cells fluctuates more after equilibrium than the plot with 10 unit cells.

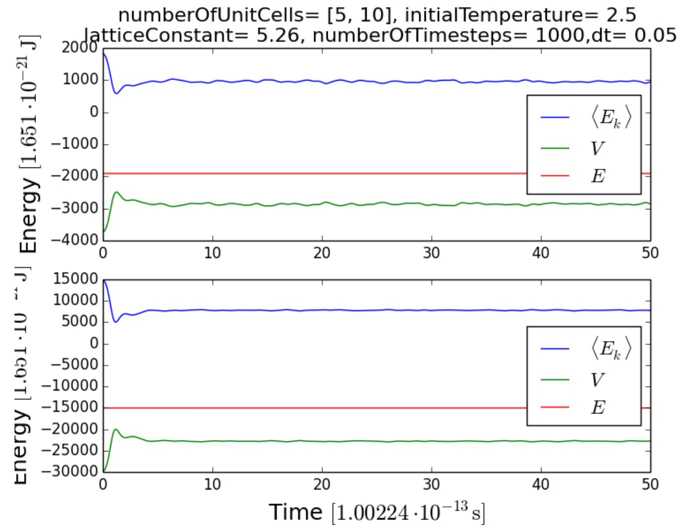

Figure 6: The kinetic (blue), potential (green) and total (red) energy plotted versus the time with initial temperature $T_i = 2.5 \approx 300$ K, lattice constant $b = 5.26$Å, $dt = 0.05 \approx 5 \cdot 10^{-15}$s and 1000 timesteps in the simulation. **Top:** 5 unit cells in each direction. **Bottom:** 10 unit cells in each direction.

In figure 7 we also see the energy plotted versus the time, but with the initial tempereraturе $T_i = 8.35 \approx 1000$ K. The initial temperature is a lot larger than in figure 6 so the system have a higher initial kinetic and potential energy giving a higher total energy. The total energy is still conserved but a larger amount of the energy now comes from the kinetic energy than in figure 6. The time the system uses to reach equilibrium is about the same as for $T_i = 2.5$. We notice that the fluctiations after reaching equilibrium is larger for 5 unit cells (top) than for 10 unit cells (bottom) in figure 7. The flutuations are even larger for $T = 8.35$ than for $T = 2.5$.

In figure 8 we see the instant temperature T plotted versus the time with initial temperature $T_i = 2.5 \approx 300$ K. Right after initializing the temperature drops quickly below the equilibrium value. This behaviour can also be observed in the kinetic energy in figure 6 and 7. Since the instant temperature is proportional to the kinetic energy (see section 3.6) it is expected that the two curves have the same behaviour.

As in the plots with the energy the value for the instant temperature stabilizes quicly and oscillates around a constant value. It is therefore safe to assume that the system has reached equilibrium after at least 100 timesteps. In the following calculations we have used 10000 timesteps which is long after the system have reached equilibrium.

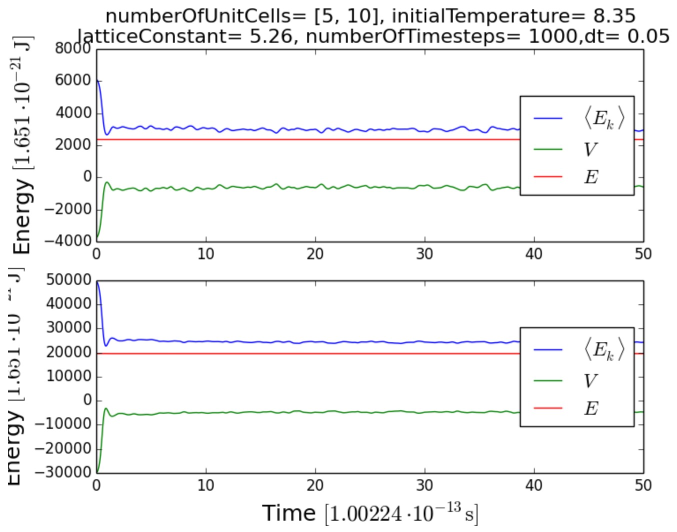

Figure 7: The kinetic (blue), potential (green) and total (red) energy plotted versus the time with initial temperature $T_i = 8.35 \approx 1000$ K, lattice constant $b = 5.26$Å, $dt = 0.05 \approx 5 \cdot 10^{-15}$s and 1000 timesteps in the simulation. **Top:** 5 unit cells in each direction. **Bottom:** 10 unit cells in each direction.

The fluctuations in the temperature is as for the energy larger for the plot with 5 unit cells in each direction than with 10 unit cells. The fluctuations are even larger in figure 9 where the instant temperature T plotted versus the time with initial temperature $T_i = 8.35 \approx 1000$ K. It apeares as if the system is more unstable and 'noisy' for small temperatures and few unit cells. Even though the trend is that the data obtained with a larger number of unit cells is more stable than with few unit cells, we have chosen to use 5 unit cells in each direction in the following calculations. This is to save time, the execution time for 10 unit cells are quite long compared with the execution time for 5 unit cells.

The temperature ratio In figure 10 we see the ratio between the instant temperature T and the initial temperature T_i plotted versus the time for the initial temperatures $T_i \in [0.5, 18.5]$. We see that the eqiulibrium state is reached for all of the temperatures after the time $t = 10$, as also seen in the energy plots.

The ratio T/T_i is not equal for all of the initial temperatures, but it does center around the value 0.5 so that:

$$T/T_i \approx \frac{1}{2}$$
$$\Rightarrow T \approx \frac{T_i}{2}$$

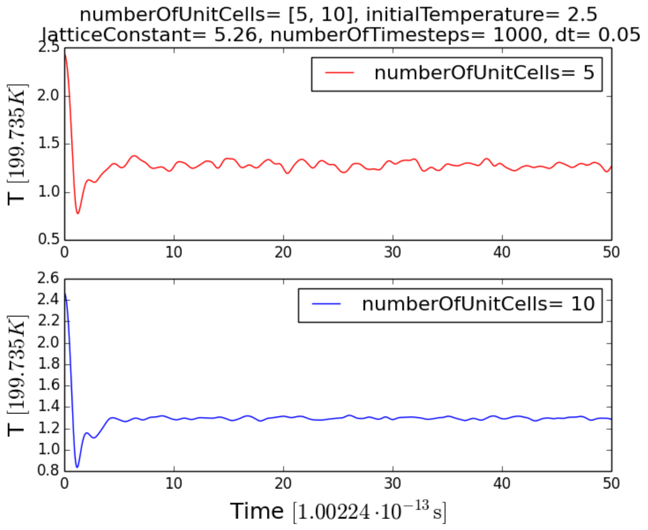

Figure 8: The instant temperature T plotted versus the time with initial temperature $T_i = 2.5 \approx 300$ K, lattice constant $b = 5.26$Å, $dt = 0.05 \approx 5 \cdot 10^{-15}$s and 1000 timesteps in the simulation. **Top:** 5 unit cells in each direction. **Bottom:** 10 unit cells in each direction.

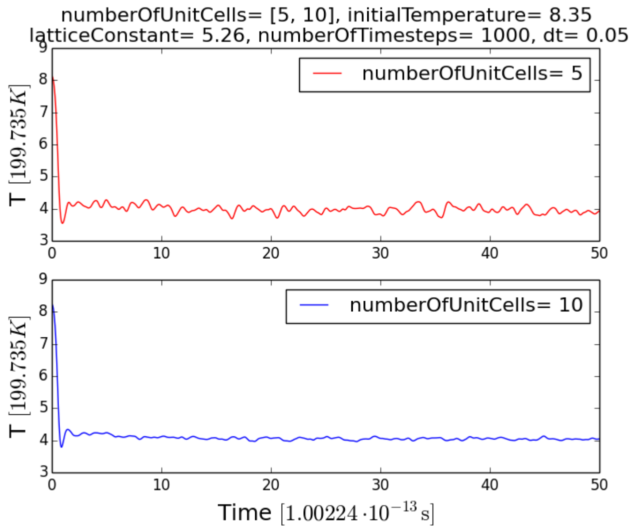

Figure 9: The instant temperature T plotted versus the time with initial temperature $T_i = 8.35 \approx 1000$ K, lattice constant $b = 5.26$Å, $dt = 0.05 \approx 5 \cdot 10^{-15}$s and 1000 timesteps in the simulation. **Top:** 5 unit cells in each direction. **Bottom:** 10 unit cells in each direction.

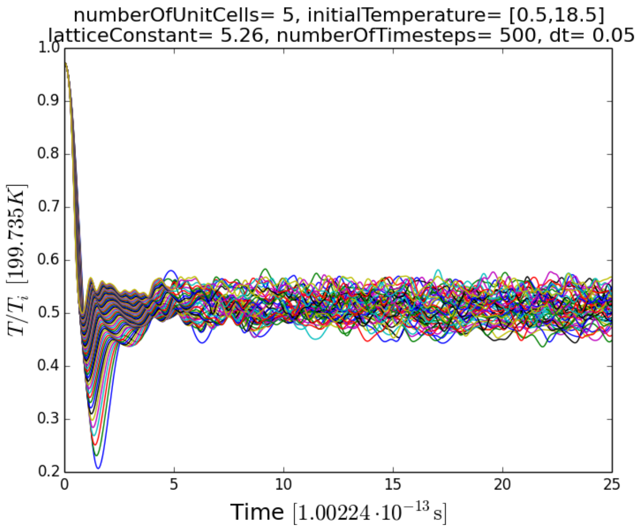

Figure 10: The ratio between the instant temperature T and the initial temperature T_i plotted versus the time (first 500 timesteps) for initial temperatures $T_i \in [0.5, 18.5]$, lattice constant $b = 5.26$Å and $dt = 0.05 \approx 5 \cdot 10^{-15}$.