

---

---

# 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 dimensional 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

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Theory</b>	<b>4</b>
2.1	Creating the lattice . . . . .	4
2.2	Giving the particles initial velocities . . . . .	5
2.3	Periodic boundaryconditions (PBCs) . . . . .	6
2.4	Intergration methods . . . . .	7
2.5	The Lennard-Jones potential (interaction/forces) . . . . .	7
2.6	Calculating physical properties of the system . . . . .	10
<b>3</b>	<b>Program flow and folder structure</b>	<b>11</b>
<b>4</b>	<b>Numerical methods and approximations</b>	<b>11</b>
<b>5</b>	<b>Results</b>	<b>11</b>
5.1	Implementation and testing of the algorithm by using Ovito (code?) and choice of parameters . . . . .	11
5.2	Comparing the two integrators . . . . .	12
5.3	Equilibrium time? and conserved energy . . . . .	13
5.4	The mean square displacement and the diffusion constant . . . . .	17
<b>6</b>	<b>Conclusions and further perspectives/experiences</b>	<b>18</b>

# 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 through a force given by a potential for a limited time. This makes it possible to study the system evolving in time. The applications of MD is many ranging from chemical physics, materials science and the modelling of biomolecules<sup>1</sup>.

Molecular dynamics is often used to study statistical properties of a system consisting of a large number of atoms or molecules and thus MD is a type of N-body simulation. For systems that obey the ergodic hypothesis<sup>2</sup> 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 (NVE) ensemble averages<sup>3</sup>. In this paper we will study one such NVE ensemble, a system with constant number of particles (N), constant volume (V) and constant total energy (E).

Often the main motivation to use Molecular dynamics is that it is not possible to determine the properties of the system analytically, mainly because of the large number of particles. Therefore one has to apply numerical methods to be able to solve the problem. One of the main limitations for the numerical simulation is the computer resources available limiting the number of particles and the system size. The latter problem can be solved by applying periodic boundary conditions so that it is possible to simulate a system with infinite size. The first problem might be solved with a very efficient code and good approximations. Another problem with the numerical simulation is cumulative errors in the numerical integration when evolving the system in time, often drift in the total energy. This error can be avoided by proper selection of parameters and integration algorithms. In this paper we will have a look at two numerical integration methods; the Euler-Cromer method and the Velocity Verlet integrator. We will also use a set of units that we will refer to as 'MD units' to avoid calculations with small exponents.

In this project we will study the properties of a large system consisting of Argon atoms initially placed in a cubic lattice and try to determine the melting temperature of the solid. We are interested in the statistical properties of the system such as the mean square displacement and the diffusion constant. We will also have a quick look at the energy development in time. The individual motion of each particle is not of any physical interest, but we will look at the simulation with an visualization program to gain physical understanding of the system and to verify the implementation of the models and algorithms.

This report is organized in bla bla...sections (??) first the theory with the equations used to build up the algorithm

---

<sup>1</sup>[https://en.wikipedia.org/wiki/Molecular\\_dynamics#Areas\\_of\\_application\\_and\\_limitations](https://en.wikipedia.org/wiki/Molecular_dynamics#Areas_of_application_and_limitations), 11.dec 12:00

<sup>2</sup>[https://en.wikipedia.org/wiki/Ergodic\\_hypothesis](https://en.wikipedia.org/wiki/Ergodic_hypothesis) 11.dec 11:30

<sup>3</sup>[https://en.wikipedia.org/wiki/Molecular\\_dynamics](https://en.wikipedia.org/wiki/Molecular_dynamics) 3.dec 11:25

## 2 Theory

### 2.1 Creating the lattice

We want to start out the simulation with the atoms in a crystal structure and will choose the face-centered cubic (FCC) lattice. Later we will learn that the FCC lattice is a stable structure for the Lennard-Jones potential that we will use in the simulation. The lattice is build up by unit cells, a group of atoms, so that a larger system can be created by repeating these cells in space. The FCC lattice unit cell is of size  $b\text{\AA}$  and consists of four atoms with local coordinates

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

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

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

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

where  $b = 5.26\text{\AA}$  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 origin of unit cell  $(i, j, k)$  is

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

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

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

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

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

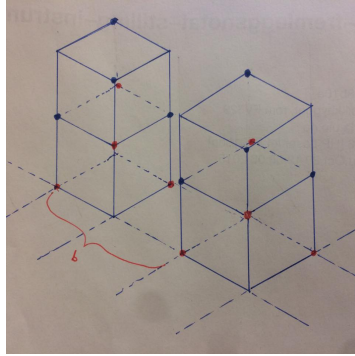
In figure 1 we see an illustration of two unit cells placed side by side. A cubic lattice is formed by placing  $N \times N \times N$  such unit cells side by side.

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  $V = L^3 = (bN)^3$  which is constant as long as  $b$  and  $N$  is constant. The number of unit cells in each direction  $N$  is also a measure of the number of particles in the system because it is 4 particles in each unit cells. The total number of particles in the simulation box is therefore  $4N^3$ . The mass density of the system is then:

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

where  $m$  is the mass of one atom and is equal for all the atoms. The particle density is then:

$$\rho = \frac{4N^3}{(bN)^3} = \frac{4}{b^3}$$



**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.

We can see that both the mass and particle density will be constant as long as the mass and the lattice constant is held constant in the simulation. We therefore see that we have constructed a NVE ensemble since the number of particles and the volume is constant in the simulation. The system will not interact with any surroundings implying a conserved total energy.

## 2.2 Giving the particles initial velocities

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

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

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  $\sigma = \sqrt{k_B T/m}$ .

The Maxwell-Boltzmann distribution describes particle speeds of ideal gasses where the particles move freely inside a stationary container. An ideal gas is an simplification of real gasses where the particles is assumed to not interact with one another except for very brief collisions in which they exchange energy and momentum with each other or with their thermal environment. The system of particles is also assumed to have reached thermodynamic equilibrium<sup>4</sup>.

In real gases and in the Argon gas that we will study there are various effects such as van der Waals interactions. These effects imply that the gas does not fulfill the assumptions of an ideal gas. However the ideal gas is often a good approximation implying that the Maxwell speed distribution is also an good approximation. The distribution also depends on the temperature of the system and the mass of the particle which is beneficial in our case. We will therefore use the Maxwell-Boltzmann distribution to give initial velocities to the particles in the simulation.

Using the Maxwell-Boltzmann distribution to give initial velocities to the particles will result in a nonzero net momentum in the system. We will therefore remove this total momentum right after the initial velocities have been given.

---

<sup>4</sup>[https://en.wikipedia.org/wiki/Maxwell-Boltzmann\\_distribution](https://en.wikipedia.org/wiki/Maxwell-Boltzmann_distribution)

The momentum in a given direction  $i$  for one atom is  $p_i = mv_i$  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:

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

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

$$\begin{aligned} mv'_i &= mv_i - \frac{m(v_{i,1} + v_{i,2} + v_{i,3} + \dots + v_{i,n})}{n} \\ \Rightarrow v'_i &= v_i - \frac{v_{i,1} + v_{i,2} + v_{i,3} + \dots + v_{i,n}}{n} \\ &= v_i - \bar{v}_i \end{aligned}$$

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

## 2.3 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 opposite side with the same values for the physical parameters as it had before it left. This also implies that an atom 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<sup>5</sup>.

---

<sup>5</sup>[https://en.wikipedia.org/wiki/Periodic\\_boundary\\_conditions#Practical\\_implementation:\\_continuity\\_and\\_the\\_minimum\\_image\\_convention](https://en.wikipedia.org/wiki/Periodic_boundary_conditions#Practical_implementation:_continuity_and_the_minimum_image_convention)

## 2.4 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 2.5)

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

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

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

symplectic 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](https://en.wikipedia.org/wiki/Verlet_integration#Velocity_Verlet) on conserved quantities  
bla bla bla

## 2.5 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<sup>6</sup>.

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

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

where  $r_{ij} = |\mathbf{r}_i - \mathbf{r}_j|$  is the distance from atom  $i$  to atom  $j$ ,  $\epsilon$  is the depth of the potential well (units energy) and  $\sigma$  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:

$$\frac{\epsilon}{k_B} = 119.8 \text{ K}, \sigma = 3.405 \text{ \AA}. \quad (12)$$

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

---

<sup>6</sup>[http://en.wikipedia.org/wiki/Lennard-Jones\\_potential](http://en.wikipedia.org/wiki/Lennard-Jones_potential)

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

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

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

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

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

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

gives

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

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

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

(another section???)

In section 2.4 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.

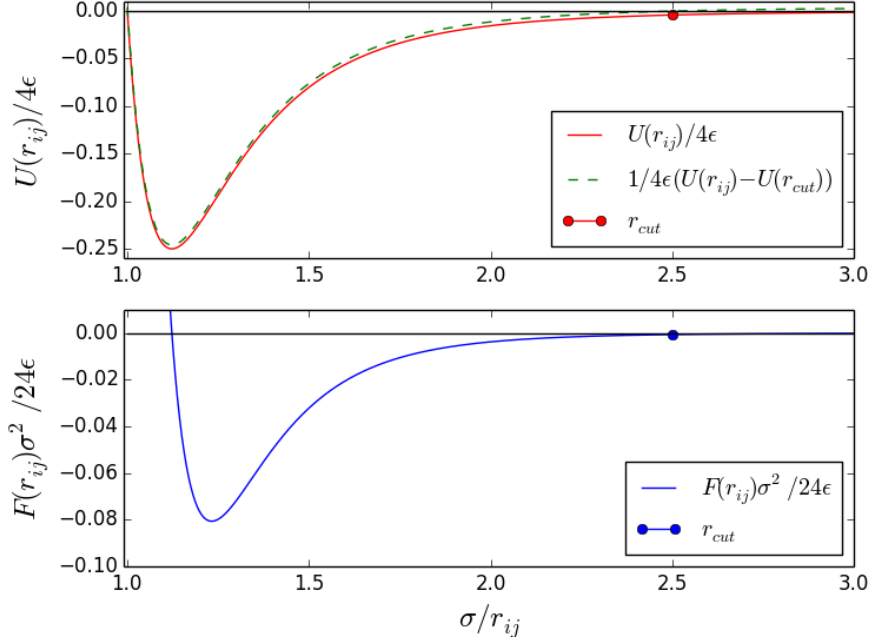
A better numerical expression for the force is

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

and for the potential energy

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





**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.

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  $r_{ij} \rightarrow \infty$  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 essentially 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<sup>7</sup>:

```

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

```

(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) 'fold back' 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 is interested in mean displacements, diffusion lengths, etc., the second option is preferable.

## 2.6 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)

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

where  $U(r_{ij})$  are the Lennard-Jones potential given in section 2.5. The kinetic energy is defined as

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

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  $\sigma_E$  ??)

We will also calculate an estimate of the temperature through the equipartition theorem<sup>8</sup>

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

We can use this to define an *instantaneous* temperature

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

---

<sup>7</sup>[https://en.wikipedia.org/wiki/Periodic\\_boundary\\_conditions#Practical\\_implementation:\\_continuity\\_and\\_the\\_minimum\\_image\\_convention](https://en.wikipedia.org/wiki/Periodic_boundary_conditions#Practical_implementation:_continuity_and_the_minimum_image_convention)

<sup>8</sup>See for example D. Schroeder's *An Introduction to Thermal Physics* for details. better reference?

**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

$$\langle r^2(t) \rangle = 6Dt, \quad (19)$$

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

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

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

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 reveal the melting temperature because the solid melts when the diffusion constant is non zero.

write something about ovito introduction(?)

### 3 Program flow and folder structure

### 4 Numerical methods and approximations

### 5 Results

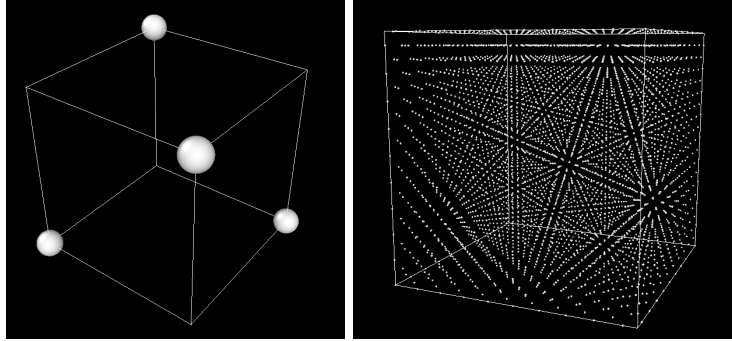
#### 5.1 Implementation and testing of the algorithm by using Ovito (code?) and choice of parameters

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 2.1 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.

**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



**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.

as the particles 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 opposite 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  $T_i = 300K$ . 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??

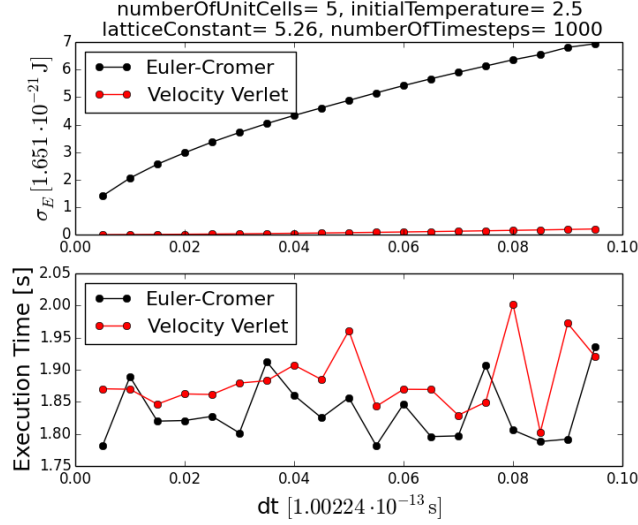
## 5.2 Comparing the two integrators

In the top plot in figure 4 we can see the standard deviation of the total energy  $\sigma_E$  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  $\sigma_E$  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  $b = 5.26\text{\AA}$  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,  $t_{exe} \approx 800s \approx 13min$  so that the deviation in the execution



**Figure 4: Top:** The standard deviation of the total energy  $\sigma_E$  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  $b = 5.26\text{\AA}$  and 1000 timesteps in the simulation. **Bottom:** The execution time 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  $b = 5.26\text{\AA}$  and 1000 timesteps in the simulation.

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  $\sigma_E$  is as small as possible, we should therefore use the Velocity Verlet integrator. The further calculations are done with the Velocity Verlet integrator.

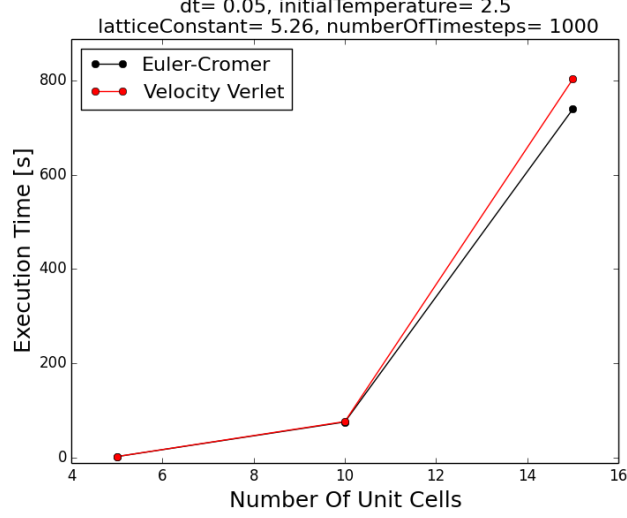
number of unit cells is also a measure of how many particles..include?

### 5.3 Equilibrium time? and conserved energy

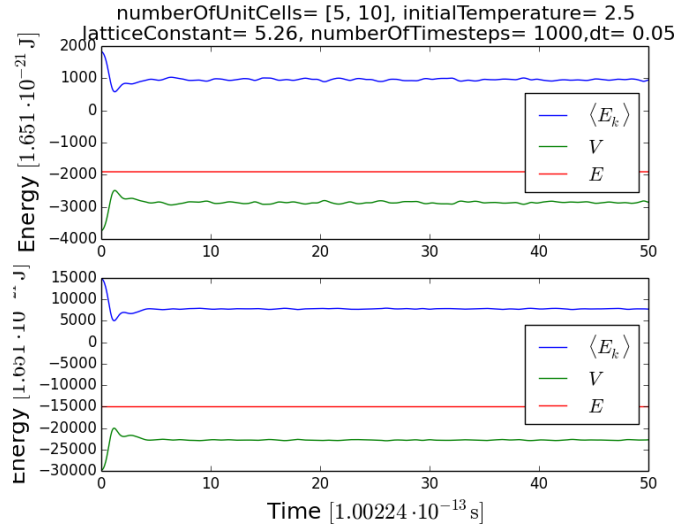
(conserved, ref to rcut?) In figure 6 we see the energy plotted versus the time with initial temperature  $T_i = 2.5 \approx 300$  K for two different number of unit cells in each direction. We see that the total energy is conserved, or constant. This corresponds well with the low value of  $\sigma_E$  found in section 5.2 (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 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 stopped, 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.

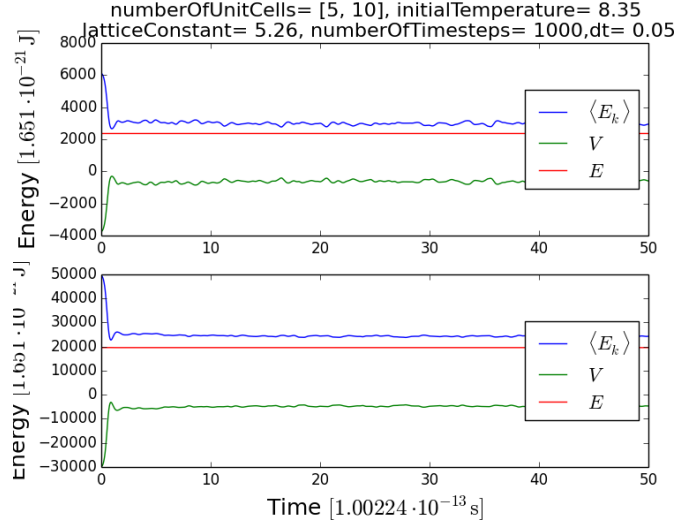


**Figure 5:** The execution time plotted versus the number of unit cells in each direction for the Euler-Cromer and Velocity Verlet integrator with initial temperature  $T_i = 2.5 \approx 300$  K, lattice constant  $b = 5.26 \text{ \AA}$ ,  $dt = 0.05 \approx 5 \cdot 10^{-15} \text{ s}$  and 1000 timesteps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



**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  $b = 5.26 \text{ \AA}$ ,  $dt = 0.05 \approx 5 \cdot 10^{-15} \text{ s}$  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 temperature  $T_i = 8.35 \approx 1000$  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 fluctuations after reaching equilibrium is larger for 5 unit cells (top) than for 10 unit cells (bottom) in figure 7. The fluctuations are even larger for  $T = 8.35$  than for  $T = 2.5$ .

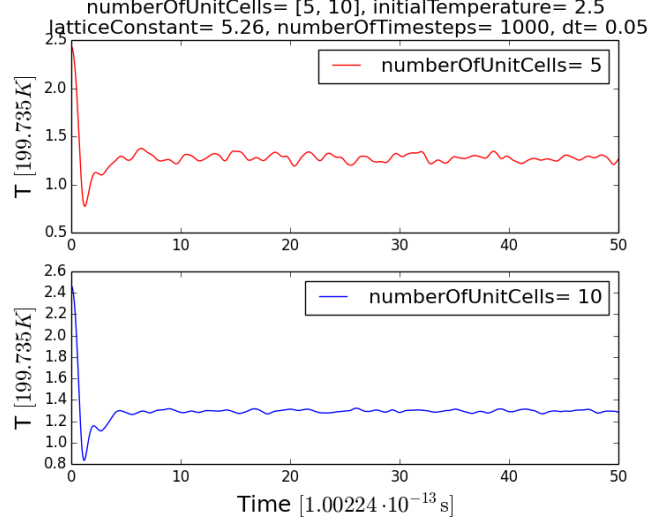


**Figure 7:** The kinetic (blue), potential (green) and total (red) energy plotted versus the time with initial temperature  $T_i = 8.35 \approx 1000$  K, lattice constant  $b = 5.26 \text{ \AA}$ ,  $dt = 0.05 \approx 5 \cdot 10^{-15} \text{ s}$  and 1000 timesteps in the simulation. **Top:** 5 unit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

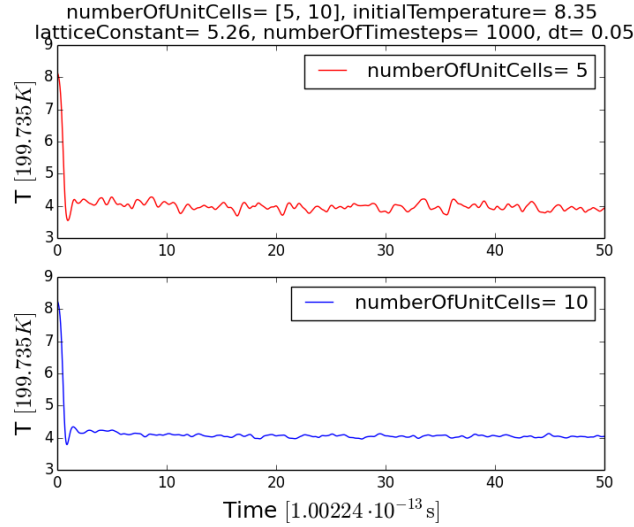
In figure 8 we see the instant temperature  $T$  plotted versus the time with initial temperature  $T_i = 2.5 \approx 300$  K. Right after initializing the temperature drops quickly below the equilibrium value. This behaviour can also be observed in the kinetic energy in figure 6 and 7. Since the instant temperature is proportional to the kinetic energy (see section 2.6) 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 quickly 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.

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 appears 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.



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

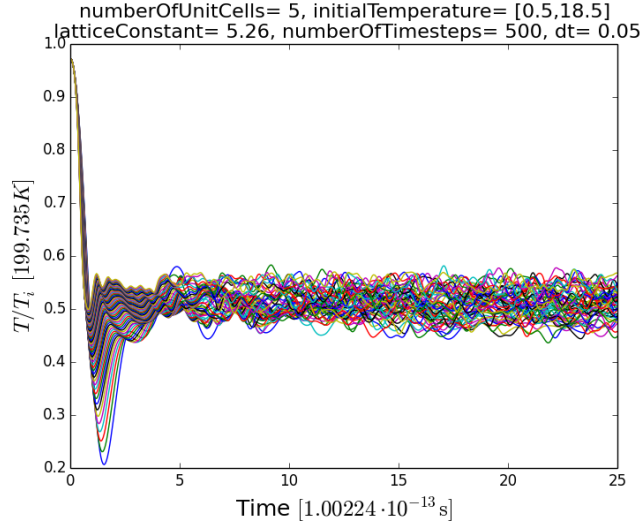


**The temperature ratio** In figure 10 we see the ratio between the instant temperature  $T$  and the initial temperature  $T_i$  plotted versus the time for the initial temperatures  $T_i \in [0.5, 18.5]$ . We see that the equilibrium state is reached for all of the temperatures after the time  $t =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

$$\begin{aligned} T/T_i &\approx \frac{1}{2} \\ \Rightarrow T &\approx \frac{T_i}{2} \end{aligned}$$

so from the initial value the temperature will approximately drop to half after the equilibrium state is reached.

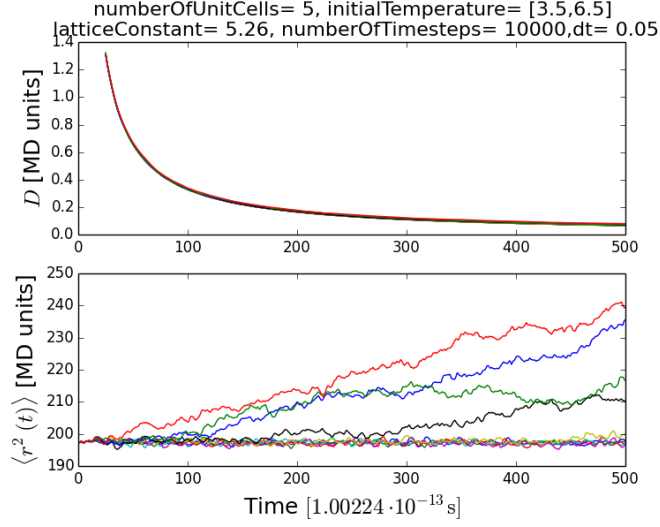


**Figure 10:** The ratio between the instant temperature  $T$  and the initial temperature  $T_i$  plotted versus the time (first 500 timesteps) for initial temperatures  $T_i \in [0.5, 18.5]$ , lattice constant  $b = 5.26\text{\AA}$  and  $dt = 0.05 \approx 5 \cdot 10^{-15}$ .

#### 5.4 The mean square displacement and the diffusion constant

In figure 11 we see the diffusion constant (top) and the mean square displacement (bottom) for the particles plotted versus the time for initial temperatures  $T_i \in [3.5, 6.5]$ . We can see that the diffusion constant diverges against a constant value for large values for the time. The highest temperatures have the largest values for all temperatures, but it is difficult to tell the different initial temperatures apart by looking at this plot. After 10000 timesteps the diffusion constant have still not stabilized entirely, but we will take the diffusion constant after 10000 timesteps to be the diffusion constant after equilibrium.

It is easier to tell the initial temperatures apart in the bottom plot in figure 11. We can see that the mean square displacement of the particles is zero for the low temperatures and that it for higher temperatures starts to grow for larger values of the time. When the system is in a solid state it is expected that the diffusion constant is be zero, or that  $\langle r^2(t) \rangle$  is zero. For



**Figure 11: Top:** The diffusion constant  $D$  plotted versus the time with 10000 timesteps for initial temperatures  $T_i \in [0.5, 18.5]$ . **Bottom:** The mean square displacement  $\langle r^2(t) \rangle$  of the particles plotted versus the time with 10000 timesteps for initial temperatures  $T_i \in [0.5, 18.5]$ . The units are the MD units listed in section (ref???!).

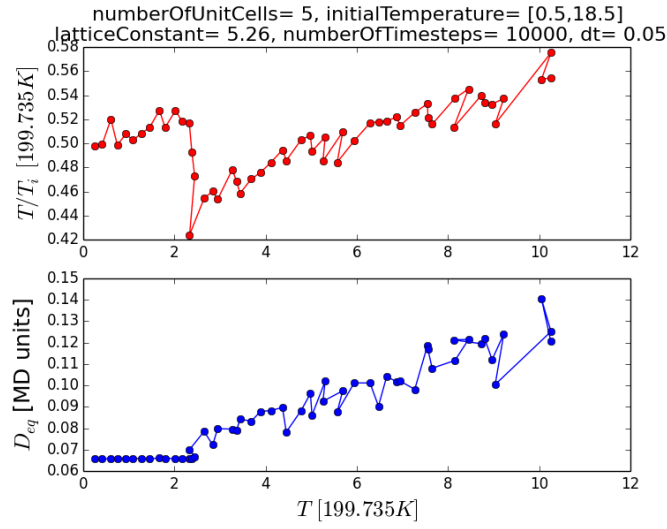
higher temperatures the diffusion constant would be non-zero giving the slope seen for higher temperatures in the  $\langle r^2(t) \rangle$  plot. It is hard to tell exactly for which instant temperature the system melts, but it must be in the range with initial temperatures  $T_i \in [3.5, 6.5]$ .

In figure 12 we see the temperature ratio  $T/T-i$  (top) and the diffusion constant (bottom) after reaching equilibrium  $D_{eq}$  plotted versus the instant temperature  $T$  with 10000 timesteps. Clearly something is happening at  $T = 2.3$ ....but also a lot of noise probably from that unit cells is 5

2.3 in correct units!  
or observations in ovito here?  
units and experimental data?

## 6 Conclusions and further perspectives/experiences

2.3 in correct units! With pressure bla bla bla



**Figure 12: Top:** The temperature ratio  $T/T - i$  plotted versus the instant temperature  $T$  with 10000 timesteps. **Bottom:** The diffusion constant after reaching equilibrium  $D_{eq}$  plotted versus the instant temperature  $T$  with 10000 timesteps. The units are the MD units listed in section (ref???!).