Project 5
Molecular dynamic simulation of Argon

Anisa Yaseen
UNIVERSITY OF OSLO

December 9, 2016

Abstract

The purpose of this project is to study a very popular simulation called Molecular dynamics(MD). Macroscopic properties are often determined at molecular level. Qualitative and quantitative information of macroscopic behavior of molecules can be obtained from simulation of a system at molecule's atomic level. MD is a simulation widely used in thermodynamic physics, chemistry and biology. Where the simulation is used to calculate the motion of atoms in a molecular collection by using Newtonian dynamics to determined the net force and the accelerations experienced by each atoms, which helps to determine several thermodynamic properties. We will in this project study the dynamics of argon atom and explore the phase space, which includes sampling the statistical properties such as; energy, temperature, and diffusion constant. The numerical ("practical") part of this project will be based on an object oriented code.

All source codes and programs used in this project are hosted on Github; see section (references)

Contents

1   Introduction

Molecular dynamics is a computer simulation widely used to simulate many-particle(body) systems ranging from solids, liquids, gasses and bio-molecules on Earth, to the motion of stars and galaxies in the Universe. This is a technique where the time evolution of a set of interacting atoms is followed by integrating the famous equations of motion. In molecular dynamics or MD(in short) one follows the laws of classical mechanics, and most notably the second law of Newton:

$$F_i = m_i a_i \tag{1}$$

for each atom i in a system constituted by N atoms, here m_i is the mass of atom, a_i is its acceleration, and F_i is the force acting upon it due to the interactions with neighboring atoms. We can say that it is a deterministic technique: with a given initial set of positions and velocities, the time evolution is in principle completely determined. In other words, atoms will move into the computer bumping into each other wandering around(in fluid), oscillating in waves in concert with neighboring atoms, and so on, in a sense pretty similar to how atoms in a real substance would behave. The computer calculates a trajectory in a $6 - N$ dimensional phase space(see later); $3N$ positions and $3N$ momenta. MD is statistical mechanics method, it is a way to obtain a set of configurations distributed according to some statistical distribution function or statistical ensemble(see later). According to statistical physics, physical quantities are represented by averages over configurations distributed according to certain statistical ensemble. A such set of configurations is obtained by MD. Therefor, a measurement of a physical quantity by simulation is simply obtained as an arithmetic average of the various instantaneous values assumed by that quantity during the simulation(MD) run. Statistical physics gives the links between the microscopic behavior and thermodynamics. In the limits of very long simulation times, one could expect the phase space to be fully sampled and in that limit this averaging process would yield the thermodynamics properties. In practice the runs are always of finite length and one should exert caution to estimate when the sampling may be good or not. In this way , the MD simulations can be used to measure thermodynamics properties in physics. MD is used to examine the dynamics of atomic-level phenomena that is unobserved directly, such as thin film growth and ion-subplantation. It is also used to examine the physical properties of nanotechnological devices that have not been or cannot yet be fabricate. Molecular dynamics is concerned with simulating the motion of molecules to gain a deeper understanding of several physical phenomena that derive from molecular interactions. These studies include not only the motion of many molecules as in a fluid, but also the motion of a single large molecule consisting of hundreds or thousands of atoms, such as in a protein. MD methods are routinely used to investigate the structure, dynamics and thermodynamics of biological molecules and their complexes. These include, for example; Protein stability, molecular recognition(proteins, DNA, membranes), ion transport in biological systems, and help to carry out studies such as drug design, structure determination(X-ray and NMR).

MD in chemistry is used to study the chemical reaction $A + A \rightarrow B + B$. The reaction rate follows the Arrhenius law both for Lennard-Jones (see later) and hard sphere interaction potentials between substrate particles. For denser systems the reaction rate is proportional to the value of the radial distribution function at the contact point of two hard spheres. In this way, the MD is a very useful tool in chemistry as well.

PART I: Theory

2   Molecular dynamic simulation of Argon

We consider N atoms of argon each with mass $m = 6.69 \times 10^{-26} kg$, argon is an inter gas; atoms behave almost like hard spheres which attract one another with weak van der Waals forces. With a given initial velocity according to Maxwell distribution:

$$P(v)dv = \left(\frac{m}{2\pi k_B T}\right)^{\frac{1}{2}} exp\left(\frac{-mv^2}{2k_B T}\right) dv \tag{2}$$

where m is the mass of atom, k_B is Boltzmann constant and T is the temperature. The force between two argon atoms is approximated by a Lennard-Jones potential energy function:

$$U(r_{ij}) = 4\epsilon \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \tag{3}$$

where $r_{ij} = |r_i - r_j|$ is the distance between the center of two atoms(from atom i to atom j), $\epsilon = k_B \cdot 119,8K = 1.65 \times 10^{-21}J$ is the strength(or depth) of the potential energy(well), and $\sigma = 3.405A = 3.4 \times 10^{-10}m$ is the distance at which the potential is zero. The first term $(1/r)^{12}$ in LHS represents a repulsive hard core interaction between the argon atoms and the seconds term $(1/r)^6$ represents an attractive dipole-dipole(van der Waals) interaction the non-polar atoms. This can be seen in figure 1. The potential has its minimum $U(2^{\frac{1}{6}}\sigma) = -\epsilon$ at $r = 2^{\frac{1}{6}}\sigma$. The Lennard Jones force is derived from the Lennard-Jones potential as:

$$F(r_{ij}) = -\frac{dU(r_{ij})}{dr_{ij}} = \frac{24\epsilon}{\sigma}\left[2\left(\frac{\sigma}{r_{ij}}\right)^{13} - \left(\frac{\sigma}{r_{ij}}\right)^{7}\right] \tag{4}$$
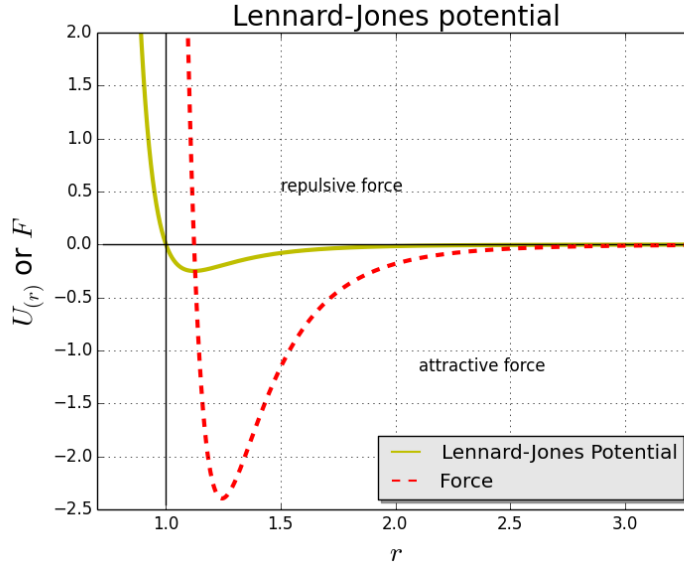

Figure 1: Plott of Lennard Jones potential and force.

We can further relate above equation with **Newton's equations of motion**, since we knows that the force vector between atoms with position r_i and r_j are given by:

$$F_{on\ i\ by\ j} = -F_{on\ j\ by\ i} = 24(r_i - r_j)\left[2\left(\frac{1}{r}\right)^{-14} - \left(\frac{1}{r}\right)^{-8}\right] \tag{5}$$

The net force on atom i due to all of the other atoms $(N-1)$ is given by:

$$F_i = \sum_{j=1, j\neq i}^{N} F_{onibyj} \tag{6}$$

3

The equation of motion for atom i follows as:

$$a_i(t) \equiv \frac{dv_i(t)}{dt} = \frac{d^2 r_i(t)}{dt^2} = \frac{F_i}{m} \tag{7}$$

where a_i and v_i are the acceleration and velocity of atom i. If initial conditions, that is the values of positions $r_i(t_0)$ and velocities $v_i(t_0)$ of the particle are specified at some initial time t_0, these equation 7 (the second order ordinary differential equations(ODE's)) have a unique solution as a function of time t. The equations can be integrated numerically by using Velocity Verlet algorithm one of many other algorithms developed by Verlet;

$$r_i(t + dt) = r_i(t) + v_i(d)dt + \frac{1}{2} a_i(t)dt^2$$

$$v_i(t + dt) = v_i(t) + \frac{1}{2}[a_i(t - dt) + a_i(t)]dt$$

It can be shown that the errors in this algorithm are of order $\mathcal{O}(dt^4)$, and that it is very stable in MD applications and in particular conserves energy very well.

2.1 Velocity distribution

The atoms are usually given velocities according to the Maxwell-Boltzmann distribution (equation 2) which result in a nonzero net momentum in the system. The Maxwell- Boltzmann distribution for momentum is given as:

$$P(p_x) = (2\pi m k_B T)^{\frac{1}{1}} \exp\left(-\frac{p_x^2}{2mk_B T}\right) \tag{8}$$

where p_x represents the momentum of the particle in x direction. For 3 dimensional system we have y and z components as well. We know that the kinetic energy of a particle can be written as:

$$\langle K \rangle = \frac{\langle p^2 \rangle}{2m} \tag{9}$$

where $\langle K \rangle$ mean the average of kinetic energy. The complete square momentum can be written as: $p^2 = p_x^2 + p_y^2 + p_z^2 = 3p_x^2$. We have considered that there are N particles in the simulation. Hence there are $3N$ components of the complete momentum. This can be written as:

$$\langle K \rangle = \frac{3N \langle p^2 \rangle}{2m} \tag{10}$$

$$\langle K \rangle = \frac{3N}{2m} \int_{-\infty}^{+\infty} P(p_x) p_x^2 dp_x \tag{11}$$

The above integral(the Gaussian integral) can be simple evaluated to:

$$\langle K \rangle = \frac{3N k_B T}{2} \tag{12}$$

We also know that the momentum and the kinetic energy are related through: $P = mv$ thus the Eqs. 8 and Eqs.9 can be rewritten as follow:

$$P(v_i)dv_i = \left(\frac{m}{2\pi k_B T}\right)^{\frac{1}{2}} \exp\left(-\frac{mv_i^2}{2k_B T}\right) dv_i \tag{13}$$

$$\langle K \rangle = E_k = \frac{1}{2}mv^2 \tag{14}$$

We recognize this as a normal distribution with zero mean and stander deviation $\sigma = \sqrt{\frac{k_B T}{m}}$. This is also recognized as the Boltzmann distribution:

$$P(E) \propto e^{-\beta E} \tag{15}$$

with $E = \frac{1}{2}mv^2$. Eqs. 12-15 tells us that the average kinetic energy is temperature related. The correct temperature can only be reproduced if all the atoms follow above relations for Kinetic energy. To ensure that one has the right averages for the Kinetic energy it is therefor necessary to use the Boltzmann distribution of velocities, because the Boltzmann distribution(and also Gaussian) gives the best probability function such that an atom will have a velocity at that given temperature. So, the average kinetic energy will give the required temperature.

2.2 Phase space

As I mentioned in section 1, one can calculate a trajectory in 6 dimensional phase space by MD computer simulation. The question here arrives what do we mean by **phase space**? The answer to this question is that; In physics, phase space is a concept which unifies classical mechanics and quantum mechanics. In classical mechanics, the phase space is the space of all possible states of a physical system. By "state" it does not mean the positions of all particle in the system which would occupy the physical space, but also their velocities or momenta which would occupy the momentum space. In order to determine the future behavior of a system one needs both the position and momentum of the system. We can think of phase space as a graph, but a point on this graph would represent the whole state of a system. Let us assume we have box with 4 gas particle inside, each point in the phase space for this system tells us where all 4 particle are located in the box.
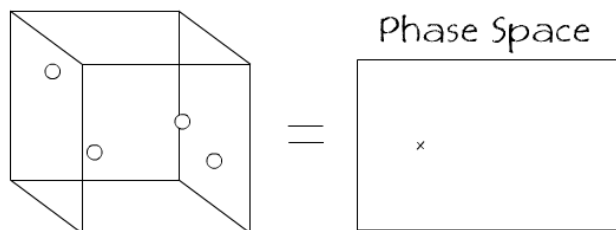

Figure 2: "Schematic" of phase space

If we now only consider the position of the 4 particles, so each point in phase space must contain an x, y, and z coordinate for each particle so our phase space is 3N dimensional, where N is the number of particles in the system. Hence in our case, the phase space is 12 dimensional.

2.3 Ensemble

An ensemble; is a collection of all possible systems which have different microscopic states but have an identical macroscopic or thermodynamic state. There are different ensembles with different characteristics.

1. Micro canonical ensemble(NVE); The state(thermodynamic state) characterized by a fixed number of atoms(N), a fixed volume(V), and a fixed energy(E), which corresponds to an isolated system.

2. Canonical Ensemble(NVT); The state is characterized by a fixed number of atoms, fixed volume and fixed temperature(T).

3. Isobaric-Isothermal Ensemble(NPT); This ensemble is characterized by a fixed number of atoms, a fixed pressure(P) and a fixed temperature.

4. Grand canonical Ensemble(mVT); This ensemble is characterized by a fixed chemical potential(m), a fixed volume and a fixed temperature.

We are dealing with NVE in this project.

3 Limitations

Molecular dynamics is indeed a very powerful technique but has limitations, let us examine the most important of them.

3.1 Use of classical forces

We know that systems at the atomic level obey quantum laws rather than classical laws, which means that Schrodinger's equation is the one to be followed, then how can we use Newton's law

to describe the motion of atoms? A simple test on validity of the classical approximation in MD is based the de Broglie thermal wavelength defined as:

$$\Lambda = \sqrt{\frac{2\pi\hbar^2}{mK_BT}} \tag{16}$$

The classical approximation yields if $\Lambda << a$, where a is the mean nearest neighbor separation. If we consider liquids as the triplet point(where all 3 state; solid, liquid and gaseous phase coexist), Λ/a is of order 0.1 for light elements such as argon, decreasing further for heavier elements. The classical approximation is not good for very light system such as H_2, He. Quantum effects become important in any system when T is sufficiently low. The drop in the specific heat of crystals below the Debye temperature ($\theta_D = \frac{hv_m}{k}$) is an example of measurable quantum effect in solids. Therefor the results of molecular dynamics should be interpreted with cautions in these regions.

3.2 Realism of forces

While studying MD we can ask ourselves how realistic is a molecular dynamics simulation? In molecular dynamics we study the interactions of atoms with each other, these interactions have origin from forces that act upon atoms and atoms move under the action of these instantaneous forces. With the motion of atoms, their relative positions and forces changes. The essence of the physics is therefor constituted by the forces. A simulation mimics the behavior of the real system, only to extent that interatomic forces are similar to those that real atoms would experience when arrange in same configuration. We know we see in later section, force are derived from the gradient of a potential function depending on the positions of the particles. Therefor the realism of MD simulation depends on the ability of the selected(constructed/chosen) potential to reproduce the behavior of the particles under the conditions at which the simulations is run.

3.3 Duration(Time) and size limitations

There is a practical and a "theoretical" limit to simulation time in molecular dynamics simulations.

The practical limit; If we are looking at an atom and our goal is to accurately resolve something, we need to have a sufficient number of temporal integration points per time scale of interest. Since we are looking at atoms and molecules, everything of interest is related to the motion and vibration of the atoms in the lattice. Let's look at a very simple example of simulating a single Cesium-133 atom(the atom is used in atomic clocks), it oscillates at $1e9$ Hz. To resolve 1 period of motion for the atom, we might need for instance, 100 temporal points per period. We're looking at a time step of roughly 10^{-12}(pico)s. In order to increase duration time (physical times), whole lot of time steps are required to get anywhere more practical. In problems like this example, it's easy to have a longer duration time. But if it took us 6 months to get let's say 100 billion atom system to reach a point where we can get some statistical properties, another 6 months to get double the physical time may not be something affordable. Many of the simulations have a duration time roughly at 10th or 100th of 10^{-15}(femto)s in order to be stable and accurate, so just running a simulation longer is extremely expensive and undesirable.

The "theoretical" limit: The Explicit Euler scheme; is simple but it imposes a time step restriction, and most importantly it is unconditionally unstable. Therefor it can be a bad choice. It doesn't conserve energy either and it will blow up eventually. The other reason is that, it is first order, thus not really accurate. An alternative choice is a fully implicit method. These are unconditionally stable, allowing for arbitrarily large time steps, but in practice not a good choice. Because if we need to resolve the physical processes of interest, we must still select reasonable time scales. Before when we took 100 temporal steps per period, that is probably due to stability constraints, maybe we can only take 10 steps and still get good results if the method is stable. The down side is fully implicit methods typically require a global matrix inversion. These don't scale well in parallel as processor counts get large.

The semi-implicit methods, or symplectic methods are often used in MD. The most common method is the Verlet integration. As we have seen in previous projects we know that the Verlet (specifically), is a second-order accurate method, it allows much larger time steps compared to a fully explicit scheme, although smaller than a fully implicit scheme. The Verlet is energy conserving in theory, which is one of the most important aspect (NVE).The energy conserving in theory, but in practice no scheme is actually energy conserving. Finite precision in hardware means errors

accumulate over time. In other words if the duration time is at 0.01 femtoseconds and simulation ran 1 billion time steps, probably we accumulated some numerical error due just to finite precision. Which means that the result is drifted and is wrong, but we will experience an unstable simulation and will blow up. Verlet has a tendency to do the latter and it will be unstable for large numbers of time steps. This mean we can not just keep the simulation running longer, it is either inaccurate or unstable. As an positive side, we run into more numerical precision errors. If we try to add something like 1e-19 to 1e-7, could lead to underflow, which is fixable by using higher precision numbers (quad instead of double) but then it just doubled memory space and probably compute time, so it will take many times longer in CPU time to get to the same physical time.

As discussed; A simulation is considered "safe" from the point of view of its duration when the simulation time is much longer than the relaxation time of the quantities we are interested in. However, different properties (statistical properties)have different relaxation times. In particular, systems tend to become slow in the proximity of phase transitions, and it is not uncommon to find cases where the relaxation time of a physical property is orders of magnitude larger than times achievable by simulation.A limited system size can also constitute a problem. In this case one has to compare the size of the MD cell with the correlation lengths of the spatial correlation functions of interest. The correlation lengths may increase or even diverge in proximity of phase transitions, and the results are no longer reliable when they become comparable with the box length.

4   Periodic boundaries conditions

The system size is limited by available computer resources. A typical large MD simulation contains a few million atoms corresponding to a system much smaller than a cubic micron. In order to get rid to boundary effects, we apply periodic boundary conditions so that we simulate a system of infinite size. By applying periodic boundary conditions, particles are confined in a box, we can imagine that this box is have replicas to infinity in all the three Cartesian coordinates(x, y, z). If one of our particles is located at a position r in the box, we think that this particle really represents an infinite set of particles at $r + la + mb + nc$. All these "image" particles move together, and only one of them is represented in the computer simulation. The point is that now each particle i in the box should be considered as interacting not only with other particles j in the box, but only with their images in nearby boxes. Which means interactions can "penetrate through" box boundaries. We can thereby say that: we have virtually eliminated surface effects from our system there is no effect of box boundaries on the position(a translation of the box with respect to the particles does not change forces)

As an effect of applied periodic boundary conditions the number of pairs(interacting pairs) increases. This is not true in practice due to potentials like Lennard Jones, usually having a short interaction range. The minimum image criterion resolves this issue. Although the periodic boundary conditions allow us to describe an infinite system of particles that approach a real system, it is still not an exact representation of reality. In introduce the translational invariance not present in the real system.
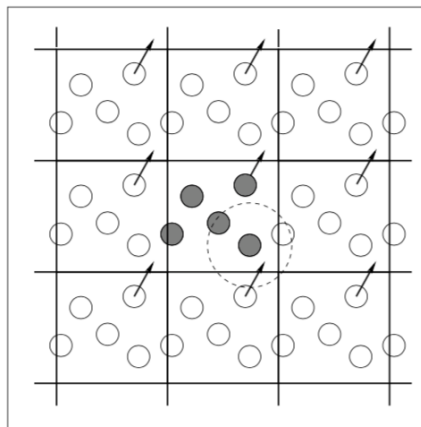

Figure 3: PBC, As a particle moves out of the simulation box, an "image" particle moves in to replace it.

The code of BPC is simple, consider a $1D$ "box" of length l, if a particle leaves the box by crossing boundary, we add or subtract l to the coordinate. One way of doing this to simply implement if-statement after the particle have been moves. Below we can see a code for $1D$, which can generalized to $3D$.

```
if (x > l_x)
        x = x - l_x;
else if (x < 0)
        x = x + l_x
```

4.1 The minimum image convention

By using a potential with a finite range(like Lennard Jones): when two particles are separated by a distance equal or larger than a cutoff distance R_c they do not interact with each other. Let us assume that we are using a box whose size is larger than $2R_c$ along each Cartesian direction.

With these conditions, it is clear that at most one among all the pairs formed by a particle i in the box and the set of all the periodic images of another particle j will interact. To demonstrate this further, let us suppose that particle i interacts with two images j_1 and j_2 of particle j. Two images must be separated by the translation vector bringing one box into another, and whose length is at least $2R_c$ (by hypothesis). Particle i should be within a distance R_c from each of them, in order to interact with both image particle j_1 and j_2, this is impossible since they are separated by more than $2R_c$. Limited by these conditions, we can safely use is the minimum image criterion: among all possible images of a particle j, select the closest, and throw away all the others. These operating conditions simplify the set up of a MD simulation, and are used commonly.

4.2 MSD

The mean square displacement of atoms in simulation is computed by its definition:

$$\delta r^2(t) = \left\langle |r(t) - r(0)|^2 \right\rangle \tag{17}$$

The MSD contains information such as atomic diffusivity. If the system is solid, MSD saturates to a finite value, and if the system is liquid, MSD grows linearly with time. In this case the systems behavior is characterized in terms of the slope, which is the diffusion coefficient D given by:

$$D = lim_{t \to \infty} \frac{1}{2dt} \left\langle |r(t) - r(0)|^2 \right\rangle \tag{18}$$

here d is the dimensionality of the system (3 in our case)

5 Energies

5.1 Potential energy

The potential energy is obtained by averaging its instantaneous value, which is obtained at the same time as the force computation is made. In interaction of two-body we have:

$$U(t) = \sum_i \sum_{j>i} \Phi\left(|r_i(t) - r_j(t)|\right) \tag{19}$$

In order to verify energy conservation, one must know all forces.

5.2 Kinetic energy

The instantaneous kinetic energy is given as:

$$K(t) = \frac{1}{2} \sum_i m_i [v_i(t)]^2 \tag{20}$$

and is easy to compute.

5.3 Total energy

The total energy is of course $E = K + U$ a conserved quantity(in Newtonian dynamics). In MD it is common to compute it at each time step in order to check that it is indeed conserved, (constant with time). We will see that during the run energy flows back and forth between kinetic energy and potential energy causing them to fluctuate while their sum remains fixed. There should be a small fluctuation in total energy as well and these fluctuations are caused by errors in the time integration.

6 The instantaneous temperature and The equipartition theorem

The equipartition theorem gives a connection between the average kinetic energy of particles and the temperature of the system. The theorem states that each degree of freedom contributes $1/2 k_B T$ to the internal energy of particle. We know that for a collection of atoms, there are only 3 degree of freedom corresponding to motion in the x, y, z directions. From the equipartition theorem the internal energy of the system is therefor:

$$U = \frac{3}{2} N k_B T \tag{21}$$

where N is the number of atoms and $3N$ is the number of degree of freedom. This energy(U) is also related to the kinetic energy as:

$$U = \langle T \rangle = \langle K \rangle = \left\langle \frac{1}{2} m v^2 \right\rangle \tag{22}$$

Assumed that the particles have identical masses. Equating the above Eq. 13, and solving for the temperature yields:

$$T = \frac{2}{3 N k_B} \left\langle \frac{1}{2} m v^2 \right\rangle = \frac{2}{3} \frac{K}{N k_B} \tag{23}$$

Hence the "instantaneous temperature" $T(t)$ is proportional to the instantaneous kinetic energy $K(t)$ by a relation analogous to the one above.

PART II: The machinery

7 Algorithms or The structure of program

I have mentioned in previous section that in molecular dynamic simulation it is common to use the verlet time integration algorithm. I have also mentioned the reason behind using this method in MD. We have studies the verlet method in project 3 and the same algorithm can be reused in this project. The code used in this project is a prepared code in an object oriented framework. The implementation consists of a collection of nested functions called by former one. We begin with using the *System* class in order to implement the periodic boundary conditions. Secondly we set up the lattice by positioning each particle into face-centered cubic(FCC) crystal structure of the argon by implementing the function *createFCCLattice*(...), and give them random velocities based on the Boltzmann distribution as a function of initial temperature (T_i), We also remove the total initial momentum. Beginning with the step function of the system class that, before advancing one step forward in time, class the function *integrator* of the integrator class (velocity verlet). The integrator calls the function *calculateForces* from the potential class. We integrate Lennard-Jones potential analytically as a function of the distance r_{ij}. The integrator derives the new positions and velocities in order to compute the based forces calculated previously. Since we have applied periodic boundary conditions in order to compute forces and positions of the particles, the particles can not escape the lattice. For loop repeats the step function a number of times and at each time steps thermodynamic properties of the system are calculated and stored into a file "*statistics.txt*". All the positions of the particles are also stored into a file and can be visualized with animation tool like OVITO. The flow chart is shown in figure below (figure 4).
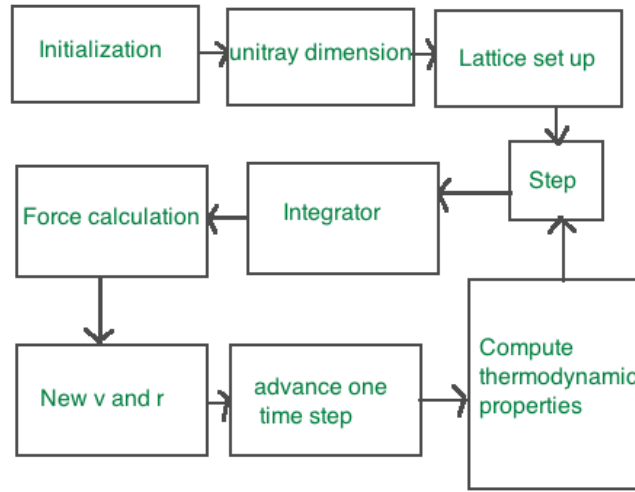
Figure 4: Flow chart; nested functions of the program
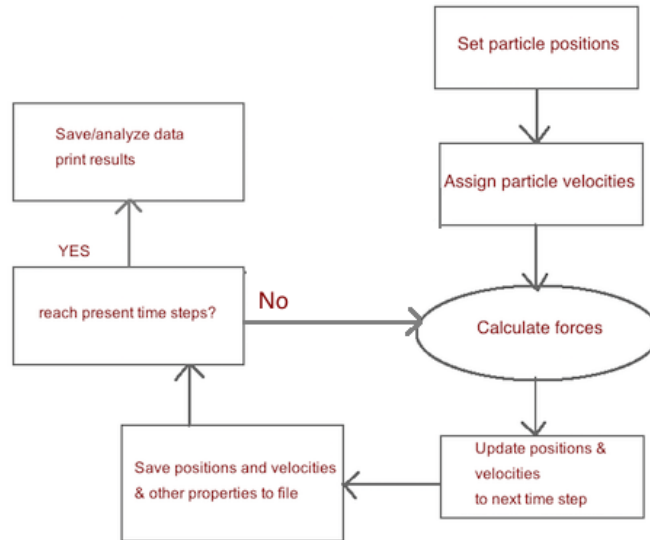
7.1 Flow chart of MD

Figure 5: Flow chart of the code

The program is constructed as follow:

1. Read parameters that specify the conditions of run: initial temperature, No of particles and time step

2. Initialize the system, initial positions and velocities

3. Compute force on all particles

4. Integrate Newton's equations of motion.

5. After completion of the central loop, averages of measured properties are computed and printed and stop.

PART III: Analysis

8 Implementation and results

Initially we had a system of 1000 atoms arranged in $10 \times 10 \times 10$ cube. Nearly half the atoms in this sample size are on the outer faces, and these will effect our the measured properties. Even for 10^6 atoms, the surface atoms amount to 6 of the total. By surrounding the cube with replicas of itself take cares of this problem, given that the potential range is not too long, we can there adopt the minimum image criterion as discussed above. This is also shown in figure 3. We set the particles in a cubic-closed structure as shown in figure(), because this is the real crystal structure of argon(s).
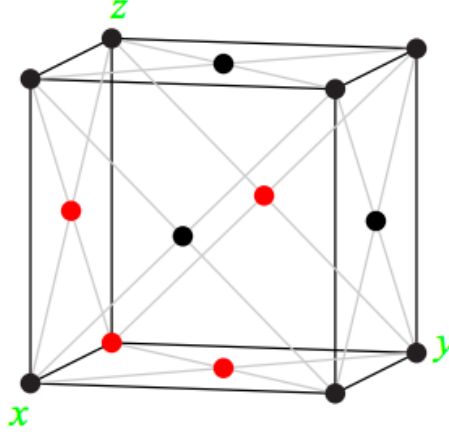
Figure 6: Face Centered Cubic(FCC) structure

We made a FCC lattice of unit cell size b[A], the lattice constant. Each cell contains particle in position given:

$$r_1 = 0\hat{i} + 0\hat{j} + 0\hat{k}$$
$$r_2 = \frac{b}{2}\hat{i} + \frac{b}{2}hatj + 0\hat{k}$$
$$r_3 = 0i\hat{i} + \frac{b}{2}\hat{j} + \frac{b}{2}\hat{k}$$
$$r_4 = \frac{b}{2}\hat{i} + 0\hat{j} + \frac{b}{2}\hat{k}$$

where $\hat{i}, \hat{j}, \hat{k}$ are orthogonal vectors. We create a $N_x \times N_y \times N_z$ such unit cell next to each other, so that the origin of unit cell(i, j, k) is:

$$R_{i,j,k} = ib\hat{i} + jb\hat{j} + kb\hat{k} \tag{24}$$

where i, j, k are integers. The lattice constant $b = 5.26A$, which corresponds to real unit cell size in argon crystal. The density was calculated as:

$$\rho = \frac{\sum m_i}{V} = \frac{Nm}{Nb^3} = \frac{m}{b^3}$$
$$\Rightarrow \frac{6.63 \times 10^{-26kg}}{1.45 \times 10^{-28}m^3}$$
$$\rho = 4.57 \times 10^5 \frac{kg}{m^3}$$

where $\sum m_i$ is the sum over all particles mass(argon atoms), V is volume. After having initialized the particles positions and velocities we make the total momentum zero. The forces were

computed according to equations described in theory section. The number of couple particles is $\frac{N^2}{2}$ and the number of particles N scales as 4 times the cube of the number of unit cells per edge of the lattice, the number of calculations increased rapidly as the lattice dimension zcreased. In order to study the systems evolution and its dependency over initial conditions we computed several thermodynamic properties each time steps. The energies(kinetic, potential and total energy) were computed. We observed that the total energy was constant over time, hence no exchange of particles or energy. Therefor we verified the reliability of our algorithm by this simple test implementation. We also computed the temperature of the system as function of the average kinetic energy. We also analyzed data by using OVITO, and observed that as the temperature grew, the system evolved to a more chaotic state. We noticed that above a certain temperature(hence its melting point) our system lost its crystal structure. In order to estimate the melting temperature we computed the diffusion constant D of the system. D is defined as:

$$\left\langle r^2(t) \right\rangle = 6Dt \Rightarrow D = \frac{\left\langle r^2(t) \right\rangle}{6t} \tag{25}$$

We know that $D = 0$ when the system is solid, so we expected it to be about *zero* when argon has crystal structure, and around its melting point the diffusion constant grows steeply. We have ran our simulation for different values of initial temperature and visualized the system evolution, and took "snapshots" of the system. The lattice we used consists of $5 \times 5 \times 5$ unit cells, which corresponds to 500 atoms. The results can be seen in figures("snapshots") below.
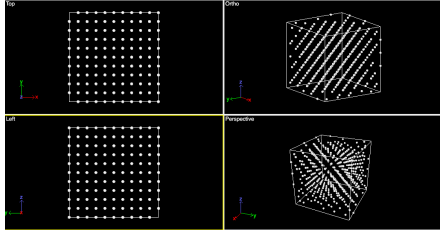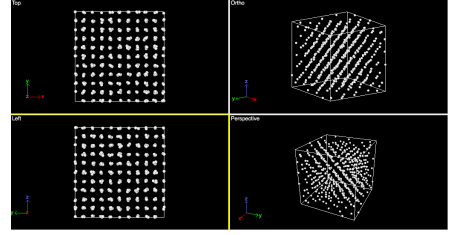

Figure 7: $T = 100K$
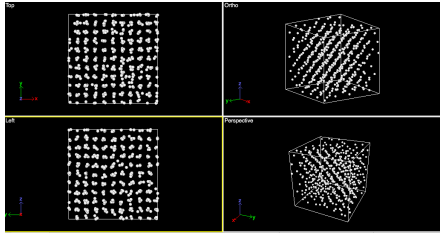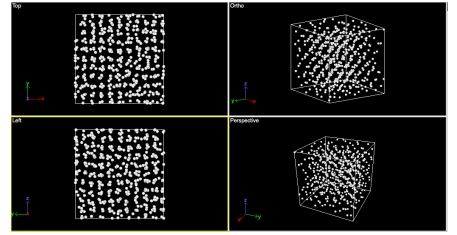

Figure 8: $T = 300K$

Figure 9: $T = 550K$

Figure 10: $T = 780K$

In figure 7 to figure 10 we see the state of the system after few time steps is plotted for different final temperatures. We observe as expected that as the temperature grow the system becomes more and more chaotic and at this point, the atoms are no longer bonded to their positions in the crystal structure and the system make phase transition, i.e. becomes fluid. We can see that in figure 10 . Although this phase transition indicates the melting point, it is not easy to detect the precise value of melting point graphically, therefor we needed to compute a new thermodynamic quantity namely, the diffusion constant, which is defined in Eqs. 25. We will see that result soon but before this let's include the energy plots.

Figure 11 shows the energies as function of time steps, we see that the kinetic energy drops to roughly half its initial value, while the potential increase by the same amount. The growth of potential happens because the system loses its initial ordered crystalline structure, for which the energy had a minimum. We see from figure 12 that the temperature curve predicts same behavior, this is because the temperature is proportional to the kinetic energy. After this the system is thermalized and temperature and energies(kinetic and potential) fluctuate about a certain value. The total energy remains constant for every time steps as expected. This also confirms the reliability of algorithm. Figure 13 shows the ratio between final and initial temperature is "almost"
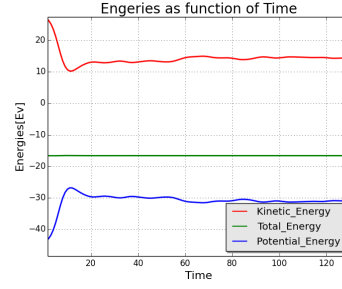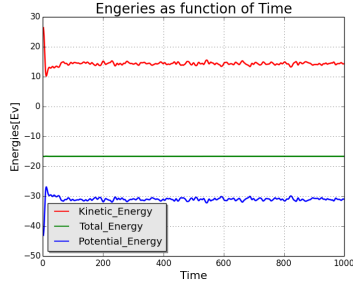
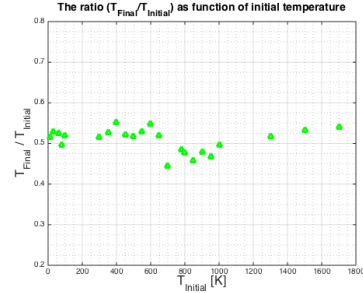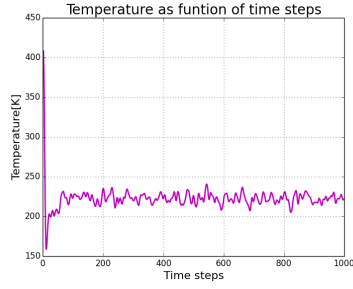Figure 11: Plot of E_k, E_p and E_{tot} as function of $time_{steps}$.

Figure 12: T as function of time steps

Figure 13: The ratio $T/T_{initial}$ as function of $T_{initial}$.

constant. The data is computed for a fixed volume, particularly for which the particles are confined in their crystalline structure.
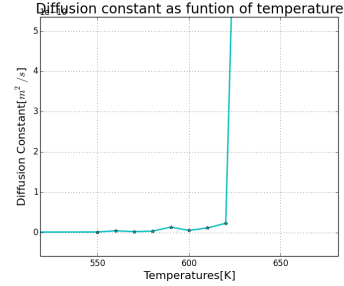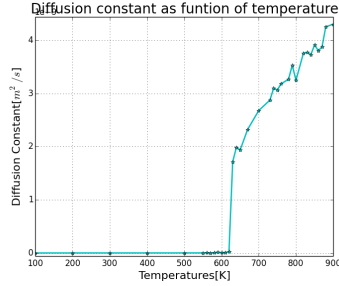

Figure 14: Diffusion coefficient (D) as function of temperature(T)

If we now look at figure 14, D remains zero as expected and when the crystal melts is grow sharply and then scales as the temperature according to Einstein's relation ($D = \mu k_B T$, where μ is the mobility). For lower temperatures D is negligible as expected, then it grows rapidly between $600K$ and $630K$ and keeps growing. Thus The melting temperature is about ($600K < T_{melting} > 630$), much higher than the melting temperature ($84K$) at standard conditions.

9   Conclusion and experiences

Implementation worked nicely, the most time-consuming part of simulation was to compute forces and diffusion coefficient in last part of the project. The observation of data through plots and visualizing in OVITO have been very interested. The data observation and visualization gave a deeper understanding of the essence of molecular dynamic. When the number of particles were increased from 1000 particles to 5000 we notice a stable value of the final temperature. This is due to the fact that the variance of average energies decreases with the increasing number of particles (As mentioned in theory section the temperature is proportional to energies).

13

10   References

[1] Github address : `https://github.com/AnisaYn/Project_5`

[2] M. Hjort-Jensen. Computational physics, lecture notes fall 2015. Department of Physics, University of Oslo, 2015

[3] Daniel V. Schroeder: "An Introduction to Thermal Physics".

[4] http://www.thp.uni-due.de/Paper/entel/entel-heraeus-springer-2004.pdf

[5] http://www.physics.buffalo.edu/phy411-506-2008/chapter9/index.html

[6] http://phycomp.technion.ac.il/ talimu/md2.html

[7] $https://www2.msm.ctw.utwente.nl/sluding/THESIS/BSc_Bosch.pdf$

[8] https://udel.edu/ arthij/MD.pdf

[9] $http://www.ch.embnet.org/MD_tutorial/pages/MD.Part1.html$

[10] http://comp-phys.net/2013/03/19/optimizing-your-c-code-for-molecular-dynamics/