

# Molecular dynamic simulation of fluids

Dajka Bence

November 25, 2022

## 1 Motivation

Studying the physical properties or dynamics of large molecular systems, such as proteins, viruses or nanotechnological devices is a crucial task in the field of chemistry, biophysics and material science. The calculations required to determine the properties of such complex systems are often too complex for our computers. Molecular dynamics (MD) seek to overcome this limitation by using numerical methods to simulate the motion of atoms or molecules. In this project I will simulate a simple fluid (argon) in 3D using MD.

## 2 Introduction and Background

### 2.1 Introduction

In MD the trajectories of atoms and molecules are determined by numerically solving Newton's equations of motion for a system of interacting particles, where forces between the particles and their potential energies are often calculated using interatomic potentials or molecular mechanics force fields. This solution is conceptually simple, but when we apply to a very large number of particles it becomes very time consuming and mathematically ill-conditioned, meaning it generates cumulative errors in numerical integrations. To avoid this we have to make some approximations and limit the number of particles from  $10^{23} - 10^{25}$  to  $\sim 10^3 - 10^8$  to describe macroscopic systems.

### 2.2 Lennard-Jones Potential

For our model we will representation of the interactions between noble-gas atoms, such as between two Argon atoms. For the interaction between two noble-gas atoms we have two main contributions:

- An attractive force due to a dipole-dipole interaction, the *van Der Waals* force. It is proportional to  $(1/r^6)$ , where  $r$  is the distance between atoms.
- A repulsive force which is a quantum mechanical effect proportional to  $(1/r^n)$ , commonly chosen to be  $n = 12$ .

The two component together gives us the Lennard-Jones potential:

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

where  $\epsilon$  is the strength of the interaction and,  $\sigma$  determines the length scale. From this we can calculate the forces between the particles:

$$\mathbf{F} = -\frac{du}{dr} \frac{\mathbf{r}}{r} = \frac{48\epsilon}{r^2} \left[ \left( \frac{\sigma}{r} \right)^{12} - \frac{1}{2} \left( \frac{\sigma}{r} \right)^6 \right]$$

Because the Lennard-Jones potential falls off rapidly for large  $r$ , far-off molecules do not contribute significantly to the motion of a molecule, so we pick a cutoff  $r_{Cut}$  distance beyond which we ignore the effect of the potential, so atom interacts with only the nearest image of another atom.

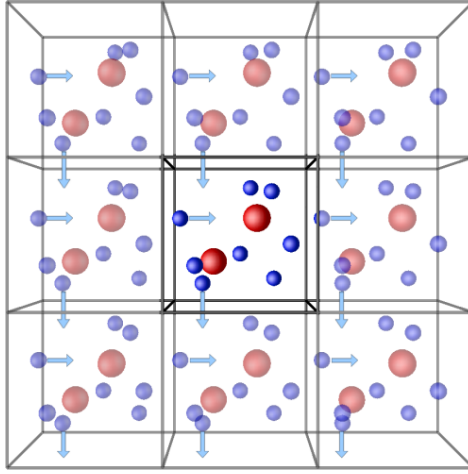


Figure 1: Schematic representation of the idea of periodic boundary conditions.

### 2.3 Periodic boundary conditions

Periodic boundary conditions allow for an approximation of an infinitely sized system by simulating a simple unit cell. We strive to minimize the shortcomings of both the small numbers of particles and of artificial boundaries. After each time-integration step we examine the position of each particle and check if it has left the simulation region. If it has, then we bring an image of the particle back through the opposite boundary 1.

### 2.4 Velocity-Verlet Integration

For the integration of the Newton equations, I will be using the Velocity-Verlet algorithm. The velocity is calculated simultaneously with the position, at both time  $t$  and at intermediate times  $t + \Delta t/2$ , where  $\Delta t$  is the time-step, whereas the forces are only calculated at the full time-steps, ensuring a good trade-off between precision in the integration algorithm and the number of times forces are calculated. The method is:

- Calculate  $\mathbf{v}(t + \frac{1}{2} \Delta t) = \mathbf{v}(t) + \frac{1}{2} \mathbf{a}(t) \Delta t$ .
- Calculate  $\mathbf{x}(t + \Delta t) = \mathbf{x}(t) + \mathbf{v}(t + \frac{1}{2} \Delta t) \Delta t$ .
- Derive  $\mathbf{a}(t + \Delta t)$  from the interaction potential using  $\mathbf{x}(t + \Delta t)$ .
- Calculate  $\mathbf{v}(t + \Delta t) = \mathbf{v}(t + \frac{1}{2} \Delta t) + \frac{1}{2} \mathbf{a}(t + \Delta t) \Delta t$

## 3 Discussion

### 3.1 Model

MD is a powerful simulation tool and it can be used in various real life problems, such as understand the characteristics of proteins [LLPDS11], studying the conductivity of solid thin films [LLLT00], or even simulating 2019-nCoV protease model [CDSG20]. There are a number of MD simulation software, that can do complex tasks and can be used for visualizing molecular systems, but building a simple version from scratch can be useful to understand the basics of Md.

The project is based on this description, so the analysis of the simulation is modelled after the requirements described there. The main goal is to a Python MD code to simulate liquid Argon. In order to simulate liquid argon we are using the Lennard-Jones parameters introduced in the 2.2 section. The simulations are to be performed at constant volume  $V$ , particle number  $N$ , and total energy  $E$ , for a given density  $\rho$  and initial temperature  $T_0$ .

The first step is to initialise the simulation to a system of the prescribed  $N$ ,  $\rho$ ,  $\epsilon$ ,  $\sigma$ , also generate a cubic simulation cell with length  $L$ . The next step is to get the position of the atoms from a file I got from the site where I found the [description](#). The main parts of the simulation are:

- Use periodic boundary conditions in the cubic simulation cell.
- Calculate forces between particles, using the minimum-image convention, with cutoff distance  $r_{Cut} = L/2$ .
- Integrate the equations of motion using the Velocity-Verlet integration.
- Calculate and store for each time step:  $U(t)$  potential energy,  $T_{kin}(t)$  kinetic energy,  $E(t)$  total energy,  $T(t)$  temperature and  $P$  total linear momentum.
- Write out the coordinates every 100  $ps$  to a file.

### 3.2 Results

The input parameters for the simulation were:  $N = 64$ ,  $m_{Ar} = 39.948 u$ ,  $\rho = 1.374 gcm^{-3}$ ,  $T_0 = 94.4 K$ ,  $\epsilon = 0.99774 kJmol^{-1}$ ,  $\sigma = 0.34 nm$ ,  $L = 10 nm$ . There were  $N_{steps} = 10000$  steps and the stepsize were  $dt = 0.0032$ . The initial velocities were chosen randomly in the  $[-1, 1]$  interval, the initial accelerations were  $a_i = 0$ .

Figure 2 shows the energies of the system in the first 600 steps. We can see, that the energies fluctuate around a certain value, also the shape of the total energy curve looks like the potential energy curve, but the kinetic energy is significantly larger than the potential, so the total energy value is closer to the kinetic energy value.

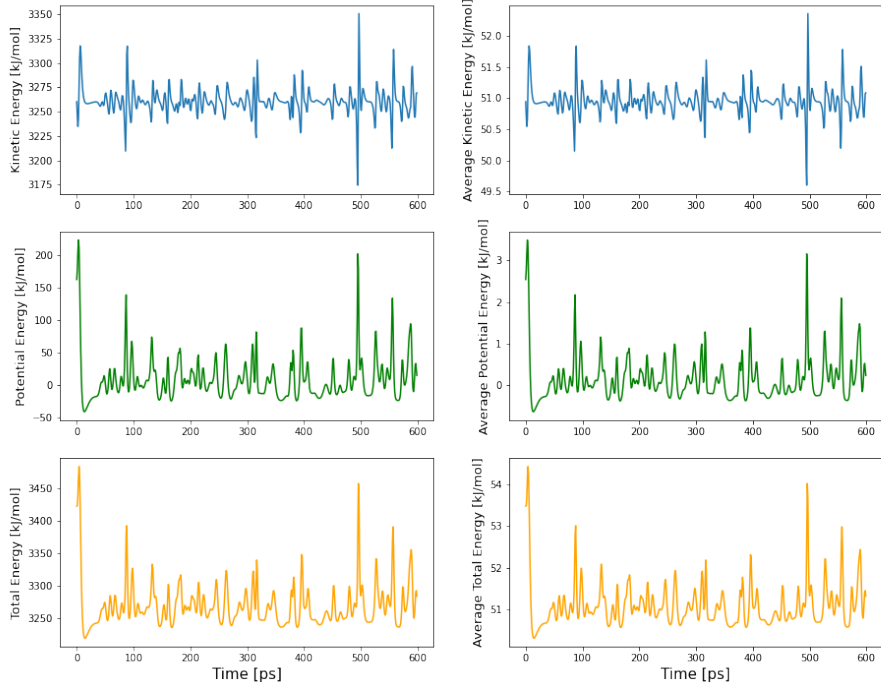


Figure 2: The energy properties of the simulation.

Figure 3 shows the change of the temperature and the total linear momentum, as well as the frequency of the temperature and the average potential energy. The temperature oscillates at a certain value, the temperature histogram resembles a normal distribution, while the total linear momentum increases.

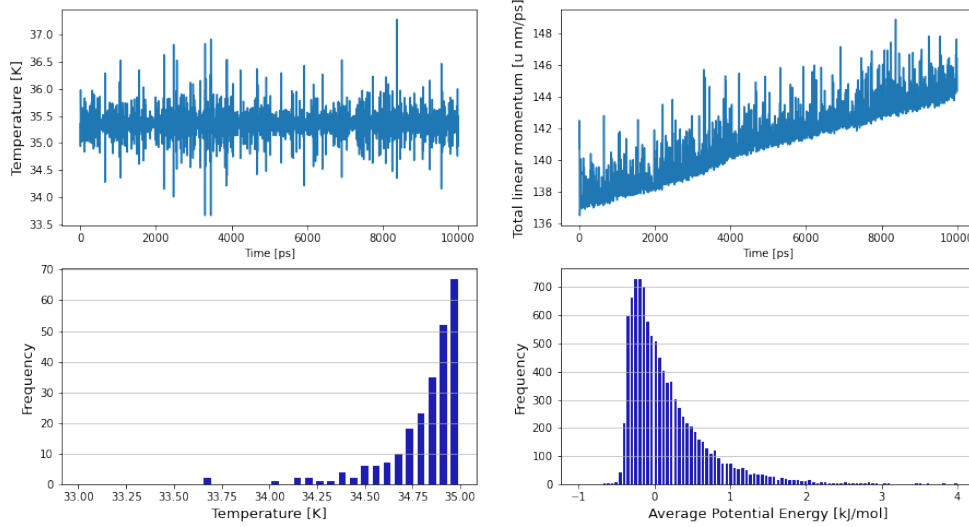


Figure 3: Some properties of the system during the simulation.

In table 1, there are the expected values of selected system properties alongside the energy drift which we get by calculating the following equation:

$$\Delta E = \frac{1}{N_{steps}} \sum_{i=1}^{N_{steps}} \left| \frac{E(0) - E(i\Delta t)}{E(0)} \right| = \left\langle \left| 1 - \frac{E(t)}{E(0)} \right| \right\rangle$$

Property	Expected value
T	$35.36 \pm 0.16$
U	$0.166 \pm 0.54$
$T_{kin}$	$53.2 \pm 0.54$
Energy drift	7.13

Table 1: Expected values of system properties and the energy drift.

## 4 Summary

We saw, that even simple MD simulations can be used to look at molecular systems and their macroscopic properties. The simulation could be more advanced by introducing surfaces which the material can interact, or by swapping the simple argon to a more complex system like a protein. To conclude I took the file with the coordinates of the atoms and made a 3D visualization of a state of the system with the Visual Molecular Dynamics (VMD) [HDS96] software 4.

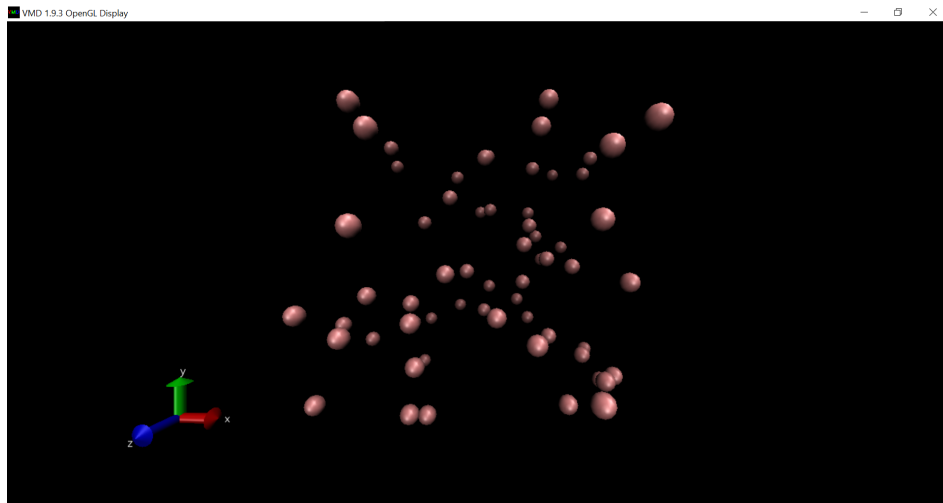


Figure 4: A state of the simulated system in 3D.

## References

- [CDSG20] Marco Cespugli, Vedat Durmaz, Georg Steinkellner, and Christian C. Gruber. 6 molecular dynamics simulations of coronavirus 2019-ncov protease model in complex with different conformations of lopinavir., Jan 2020.
- [HDS96] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, 14:33–38, 1996.
- [LLLT00] J. R. Lukes, D. Y. Li, X.-G. Liang, and C.-L. Tien. Molecular Dynamics Study of Solid Thin-Film Thermal Conductivity . *Journal of Heat Transfer*, 122(3):536–543, 03 2000.
- [LLPDS11] Kresten Lindorff-Larsen, Stefano Piana, Ron O. Dror, and David E. Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.