COMPUTATIONAL PHYSICS - EXERCISE SHEET 02
Simulating Lennard-Jones Fluids

Matteo Garbellini*
Department of Physics, University of Freiburg

(Dated: May 10, 2021)

The following is the report for the Exercise Sheet 02. The goal of this exercise is to implement a molecular dynamics code that computes the trajectories of a collection of N particles interacting through the Lennard-Jones (LJ) potential in an (NVE) microcanonical ensemble. Along this report, python scripts were also handed in. Additional code can be found at the following github repository https://github.com/matteogarbellini/Computational-Physics-Material-Science/tree/main/Exercise-Sheet-02. **After confrontations with other students regarding the energy results there might be some error in the code that I could not find.**

I. SYSTEM PARAMETERS

The following are the relationships between the main parameters describing the system volume and the initial cubic lattice distribution of the particles. Throughout the simulation natural/reduced units where used, namely $\sigma = 1$, thus the parameter p represents the lenght of the box edge in units of σ. Given a geometry with n particles *per dimension* (i.e. n^3 for a 3-dimensional system), the lattice parameter for the initial conditions is given by $a_{lat} = Ln^{-1}$. From these the relationships with ρ are given by:

$$\rho = \frac{N}{V} = \frac{N}{L^3} = \frac{N}{p^3} = \left(\frac{n}{L}\right)^3 = \left(\frac{1}{a_{lat}}\right)^3 = a_{lat}^{-3} \tag{1}$$

A. Initial distribution for a system of N=64

The following figure shows the initial distibution of $N = 64$ particles on a cubic lattice of given a_{lat}.

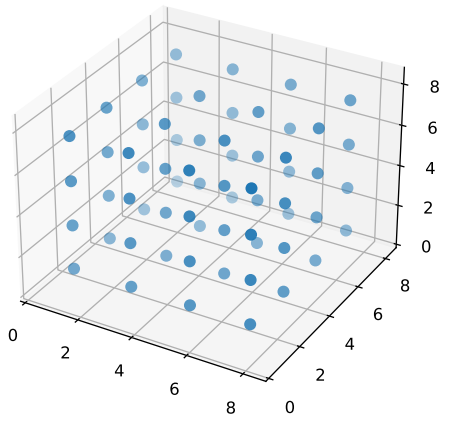

FIG. 1. Initial conditions of a system with $N = 64$ distributed on a cubic lattice.

* matteo.garbellini@studenti.unimi.it; https://github.com/matteogarbellini/Computational-Physics-Material-Science/

II. RESULTS FOR A SYSTEM OF N=64

The following are the results for a system of $N = 64$. The simulation consists of a *equilibration* run (2000 timesteps) in which the velocities of the system are rescaled every 10 integrations in order to achieve the desired target temperature. Follows then a *production* run (2000 timesteps) in which the system is free to evolve without any contraint (except the obvious boundary conditions). Both the equilibration and production runs are integrated using the Velocity-Verlet scheme with a timestep of $\Delta t = 10^{-4}$.

A. Equilibration

The following figure represents the total energy $E = U + K$ during the equilibration run.

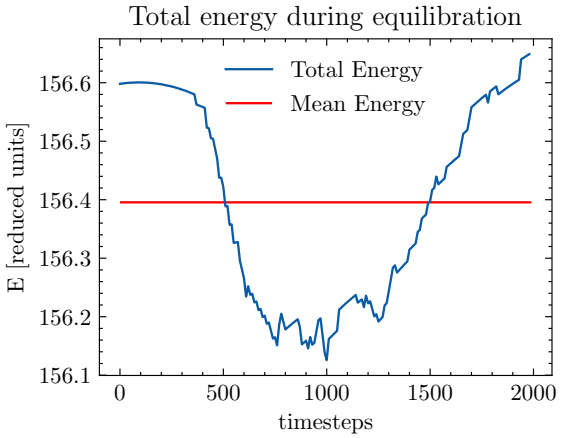

FIG. 2. The figure shows the total energy over time during the equilibration run

B. Production run

We now discuss the energy conservation during the production run. The following plot shows the total energy E of the system as a function of the number of timesteps. Although the energy varies throughout the evolution its value is fairly constant. Indeed the average energy is $< E >= 156.625$, with standard deviation of $\sigma = 0.116$ and variance $\sigma^2 = 0.0135$

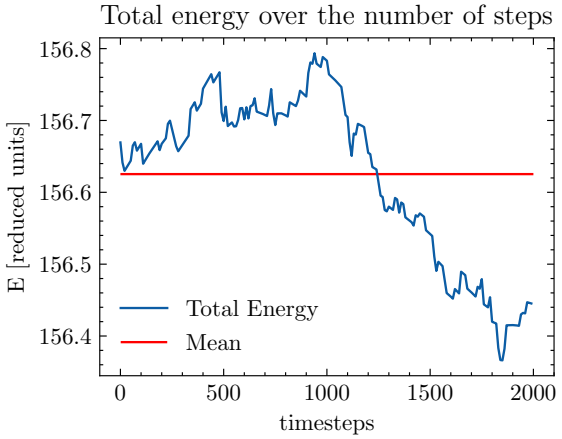

FIG. 3. The figure shows the total energy over tiem during the production run

We can also consider the evolution of the potential energy U and kinetic energy K, as seen in the following figures.

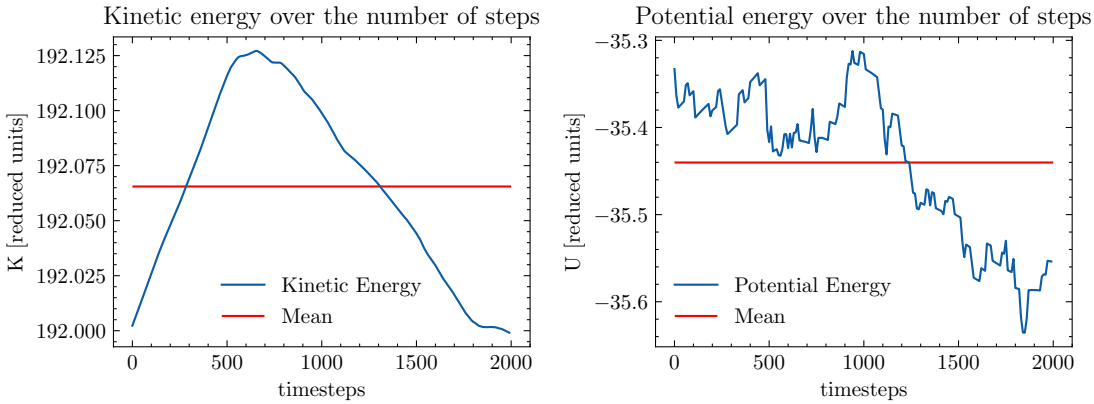

FIG. 4. The figures show the evolution of the potential energy (left) and kinetic energy (right) during the production run

C. Equipartition of energy

We then report the evolution of the quantities $\sum_i v_{i,\alpha}^2$ with $\alpha = x, y, z$ and $i \in \{0, N-1\}$. The followings are the mean values with standard deviation and variance along each axis. The values are quite different, suggesting that the energy is not equipartite among the directions of the system. This is indeed unexpected as, at least for the first iterations after the equilibration time the velocities should not have a distinct shift towards one direction. Chaotic behaviour in the production run could although lead to different velocities value for different axis.

$$< \sum_i v_{i,x}^2 >= 106.659 \qquad \sigma = 0.032 \qquad \sigma^2 = 0.001 \tag{2}$$

$$< \sum_i v_{i,y}^2 >= 131.223 \qquad \sigma = 0.043 \qquad \sigma^2 = 0.002 \tag{3}$$

$$< \sum_i v_{i,z}^2 >= 146.249 \qquad \sigma = 0.049 \qquad \sigma^2 = 0.002 \tag{4}$$

The following figures show the respective evolution over time.

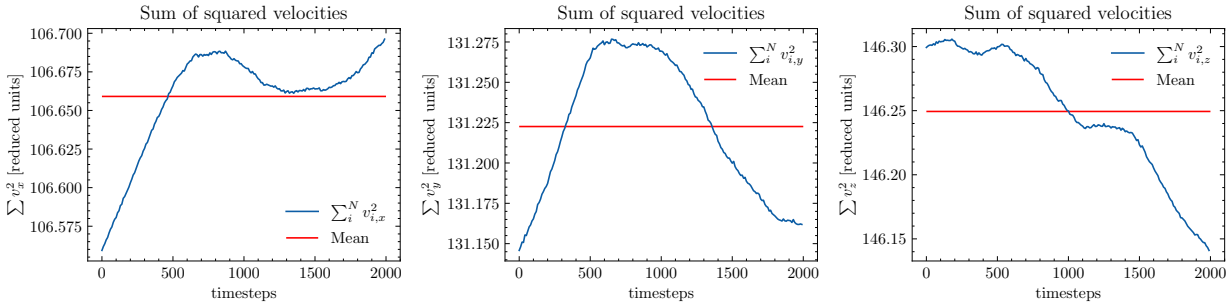

FIG. 5. The figures show the squared sum of the velocity along a particular axis, namely x (left), y (center), and z(right)

D. Stability of implementation using increased/decreased timesteps

The following are the results of a system simulated using an increased/decreased timestep by one order of magnitude. For better comparison the plots are put side by side with the regular timestep in the center $\Delta t = 10^-4$, increased on

the right $\Delta t = 10^-3$, and decreased on the left $\Delta t = 10^-5$. Note that the simulation was run with the same number of timesteps, therefore at equal timestep don't correspond to equal physical time of the system.
Given the results shown in the plots I was not able to discuss the stability of the system.

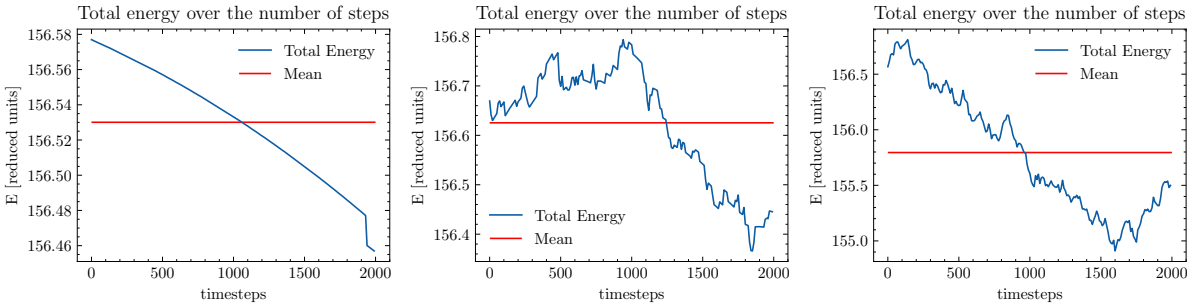

FIG. 6. The figures show the total energy over time using different timesteps: reduced (left), normal (center), and increased(right)

III. MD ENGINE EFFICIENCY

In order to estimate the efficiency of the implemented MD engine the same simulation was run using $n \in \{6, 8, 10\}$. The results were fitted with a function $f(N) = N^\alpha$, with the goal of estimating the parameter α. This allows us to obtain the scaling per single timestep of our implementation. As can be seen for the plot the parameter is given by $\alpha = 3.03 \pm 0.09$.
If a simulation was run with $n = 50$ it would take for a 4000 timestep run - of which 2000 equilibration and 2000 production - about ≈ 39 hours.

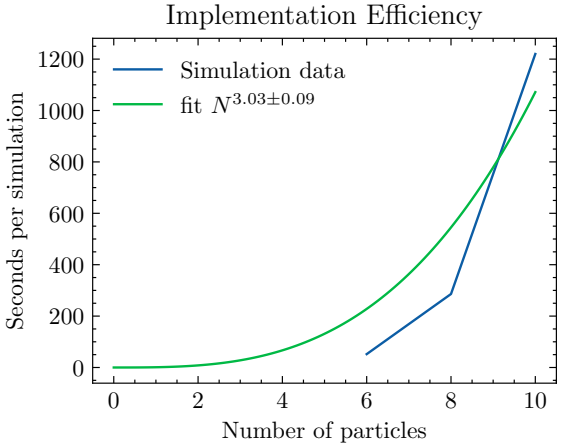

FIG. 7. The figure shows the time needed for a simulation (2000+2000 iterations).

IV. VELOCITY DISTRIBUTION FOR A SYSTEM OF N=1000

Using the results of the previous simulation with $n = 10$ we can plot the function $P(v^2)$, namely the distribution of the velocities of the system. This allows us to compare the results with the analytical for of the Maxwell-Boltzmann velocity distribution given by:

$$P(v_{i,\alpha}) = \frac{1}{\sigma\sqrt{2\pi}} exp - \{\frac{v_{i,\alpha}^2}{2\sigma^2}\} \tag{5}$$

where it is to be noted that the parameter $\sigma = \sqrt{\frac{KbT}{m_i}}$, and not the parameter given in the Lennard-Jones potential. The following is the distribution of the velocities squared, namely $P(v^2)$. I was not able to compare the results with the analytical form.

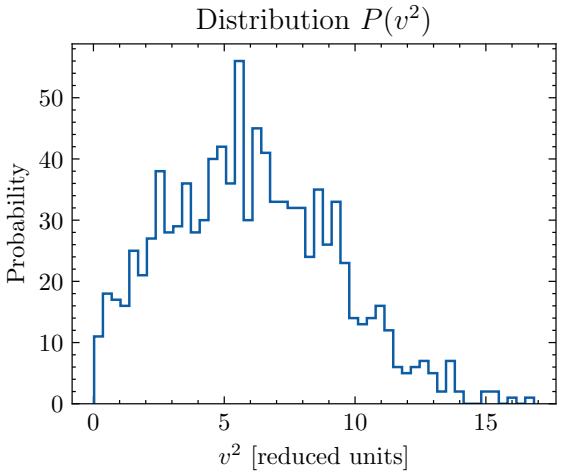

FIG. 8. The figure shows the velocity distribution $P(v^2)$.

V. EFFECTS OF CUTOFF RADII IN A SYSTEM OF N=216

Thus far the simulations were using a cutoff radius of $r_{cutoff} = 2.5\sigma$ - a standard in MD simulations. We can now investigate the effects of increasing the cutoff radius of our system, without modifying the system parameters. In particular we use $n = 6$ and the usual 2000+2000 timesteps (equilibration and production). The following plot show the difference in energy and velocity (for convenience only the x axis are shown) for $r_{cutoff} \in \{2.4\sigma, 3.25\sigma, 4\sigma\}$. It is to be noted that the results for $r_{cutoff} = 4\sigma$ are probably incorrect, since they don't show any form of fluctuations. It could be an error implementation, although the only parameter changed was the radius.

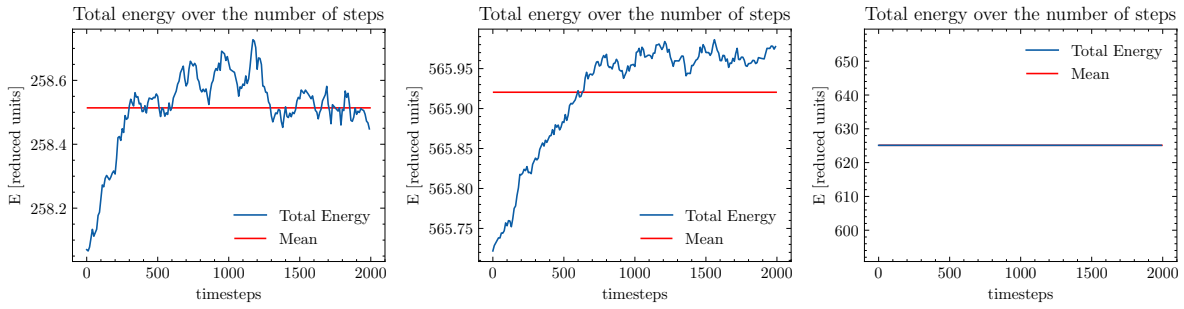

FIG. 9. The figures show the total energy evolution for different cutoff radii: 2.5σ, 3.25σ, and 4σ

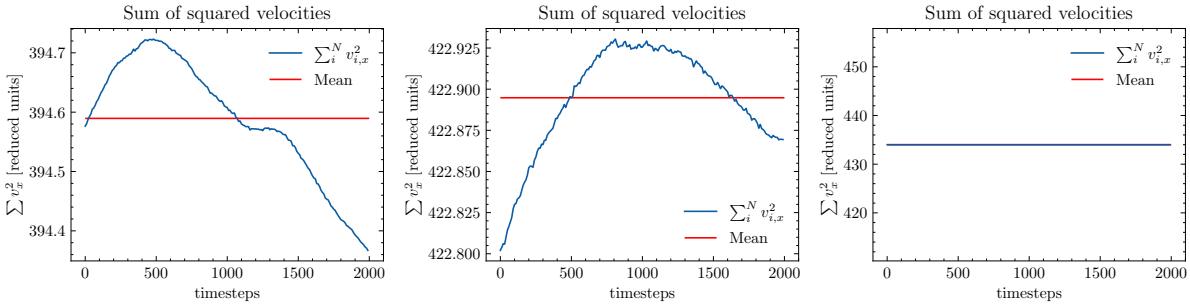

FIG. 10. The figures show the evolution of the sum of squared velocity (x axis) for different cutoff radii: 2.5σ, 3.25σ, and 4σ

VI. REAL UNITS VS REDUCED/NATURAL UNITS

In the simulations reduced units were used, namely $\epsilon = 1$, $Kb = 1$, $m = 1$ and $\sigma = 1$. In order to do so, the following conversions were used:

	Conversion
Length, x^*	$x^* = \frac{x}{\sigma}$
Energy, E^*	$E^* = \frac{E}{\epsilon}$
Mass, m^*	$m^* = \frac{m}{m} = 1$
Time, t^*	$t^* = t\sigma\sqrt{\frac{m}{\epsilon}}$
Force, f^*	$f^* = f\frac{\epsilon}{\sigma}$
Temperature, T^*	$T^* = T\frac{k_B}{\epsilon}$
Velocity, v^*	$v^* = v\frac{t^*}{\sigma} = v\sqrt{\frac{m}{\epsilon}}$