

Final report on Molecular Dynamics Simulations

Roger Bellido Peralta

The purpose of the code used is to compute a simulation of a dynamical molecular system that develops from a fixed SCC structure, on which it is imposed a bimodal velocity distribution. In this first part, a thermal bath is not considered so the system evolves from the given energy by the velocity distribution. Results show a compatibility between the initial and the final distribution of velocities, which makes sense as the main integrator used (Velocity Verlet) keeps the energy as a conserved quantity.

The next code has the objective to compute a simulation of argon atoms (fixing a certain ϵ , σ and m values) and extract results for different values of the initial density, ρ . All computations are made in reduced units, so the proper units transformations have to be applied at the end. The computed magnitudes are the kinetic, potential and total energies and the pressure of the system.

Simple Molecular Dynamics Simulation code

Let's start by describing the code used to compute this simulation.

First of all, we start considering a system formed by 125 particles with a density $\rho = 0.7m/\sigma^3$. The initial configuration of the system is a Simple Cubic Crystalline one with a bimodal velocity distribution so that its kinetic energy matches a temperature of $k_B T = 100\epsilon$. In this first part, two integrators have been used: Velocity Verlet and Euler, and its differences will be pointed out later on.

It has also been necessary to implement Periodic Boundary Conditions, so that the system could be manageable. For this matter, any particle that reached, so to say, the left boundary would reappear in the right one.

To control the temperature, the Andersen thermostat algorithm has been used, which consists in changing the velocity of a certain particle if a random number is under a stated probability. The new velocity is extracted from a normal distribution with a standard deviation that depends on the targeted temperature. This way, as the system develops, it tends to said temperature.

The potential that rules over interactions is the Lennard-Jones potential, that has two main parameters: ϵ and σ .

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

In general, these parameters have units of kJ/mol and Å, but using a convenient set of new variables: $r' = r/\sigma$, u'_{LJ}/ϵ :

$$u'_{LJ}(r) = 4 \left[\left(\frac{1}{r'} \right)^{12} - \left(\frac{1}{r'} \right)^6 \right]$$

we obtain the reduced version of the potential, where all magnitudes are adimensional.

However, as we are running a simulation, computing all interactions at all distances takes a huge amount of computational time. For that reason, a cut-off distance must be imposed. This parameter states a limitation to the distance at which interactions should be computed, thus reducing the amount of calculations needed. In general, the cut-off distance imposed is a third of the total length of the systems side: $D_{cutoff} = L/3$.

Due to this consideration, there is an issue to have in mind: if the Lennard-Jones expression is applied as usual, the potential can experience a jump at the cutoff distance. For that matter, it is necessary to apply a proper truncation so that the curve is well-behaved:

$$u_{LJ}(r) = \begin{cases} u_{LJ}(r) - u_{LJ}(r_{cutoff}) & \text{if } r \leq r_{cutoff} \\ 0 & \text{if } r > r_{cutoff} \end{cases}$$

The number of timesteps used to do the simulations has been fixed at $N_{steps} = 500.000$. With this value, it can be ensured to reach a stabilized regime of the system.

Simulation results

The first simulation involved initializing the system as a SCC with a bimodal velocity distribution and, imposing a Lennard-Jones potential, letting it evolve without considering any thermal bath whatsoever. The idea was to evaluate the energies and the total momentum as a function of time and discuss the results:

Figure 1 shows some differences between the time increments used in the simulations. As a bigger δt is used, an increment in fluctuations is observed. The main consequence can be seen in the total energy: in the left figure the fluctuations are broader while in the right figure are barely noticeable as the energy is plotted as a flat line.

This change can also be seen in the other magnitudes: both kinetic and potential energies also broaden their fluctuations, being wider for $\delta t = 10^{-3}$. In terms of the total momentum, the behaviour is somehow counterintuitive: for bigger time increments it also shows greater fluctuations, but not as wider as it could be expected compared to the $\delta t = 10^{-4}$ graphic.

Also, it can be seen that the total energy is conserved (it is more clear in the right graphic) as well as the total momentum.

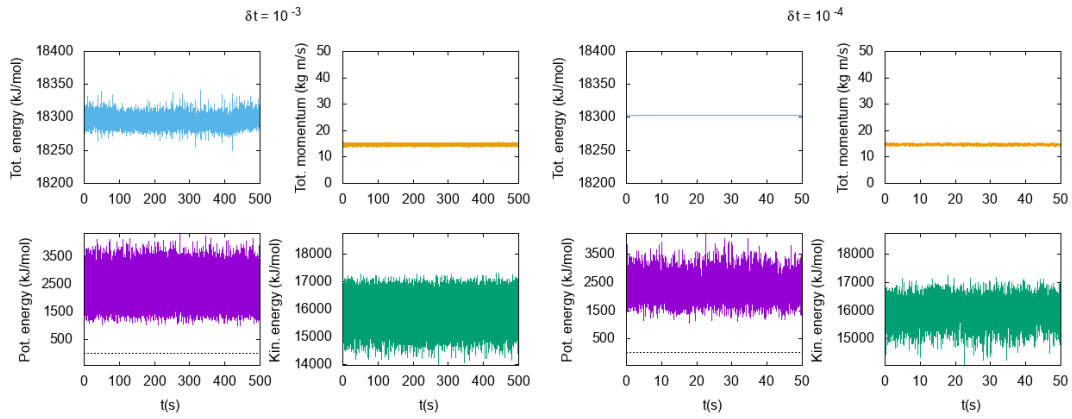


FIG. 1: Energies and momentum for a time increment of $\delta t = 10^{-3}$ (left) and $\delta t = 10^{-4}$ (right). The integrator used was the Velocity Verlet algorithm.

The results obtained with the Velocity Verlet algorithm can be compared with other algorithms that require less computational time but are not as precise. In this case, the comparison will be done with respect to the Euler algorithm. Before jumping into the conclusions, it is worth to do a brief review about the particularities of each algorithm.

The Velocity Verlet algorithm uses the current velocities to compute each step of the system's evolution. It is a convenient algorithm to apply when we want to also consider a thermal bath into our computation. A quick overview of its equations:

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t + \frac{f_i(t)}{2m_i}\Delta t^2$$

$$v_i(t + \Delta t) = v_i(t) + \frac{f_i(t) + f_i(t + \Delta t)}{2m_i}\Delta t$$

On the other hand, the Euler algorithm is based in the explicit updates of both distances and velocities. The main flaws of this algorithm are the energy drift (which means that the potential and kinetic energy are not completely coordinated) and the lack of symmetry under time reversal. Its equations are:

$$r_i(t + \Delta t) = r_i(t) + v_i(t)\Delta t + \frac{f_i(t)}{2m_i}\Delta t^2 + \mathcal{O}(\Delta t^3)$$

$$v_i(t + \Delta t) = v_i(t) + \frac{f_i(t)}{2m_i}\Delta t + \mathcal{O}(\Delta t^2)$$

Now, looking at figure 2 it can be seen that, even using a time increment of $\delta t = 10^{-4}$, the total energy fluctuations are even wider than the ones observed in the Velocity Verlet algorithm with $\delta t = 10^{-3}$. In terms

of potential and kinetic energy, on the other hand, fluctuations are not as wide as it could be expected. This is a consequence of the energy drift: as the potential and kinetic energy do not change in a coordinated way, the difference between them end up in the total energy, hence the wider fluctuations.

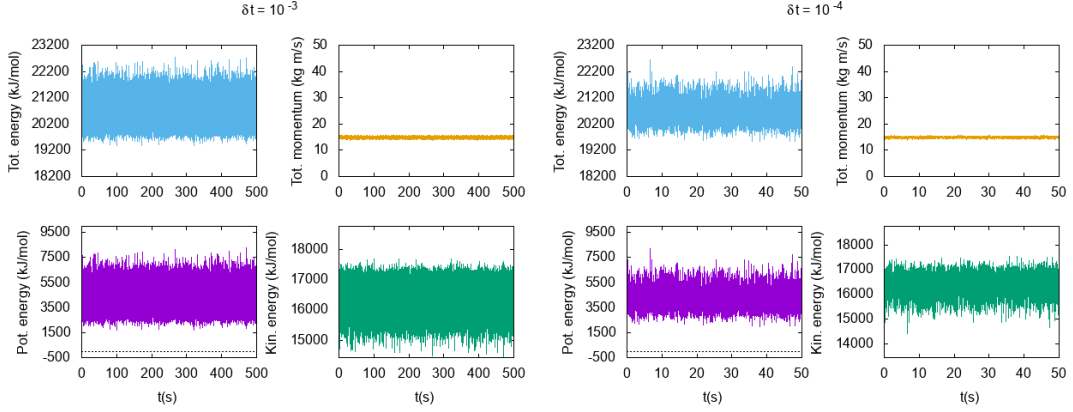


FIG. 2: Energies and momentum for a time increment of $\delta t = 10^{-3}$ (left) and $\delta t = 10^{-4}$ (right). The integrator used was the Euler algorithm.

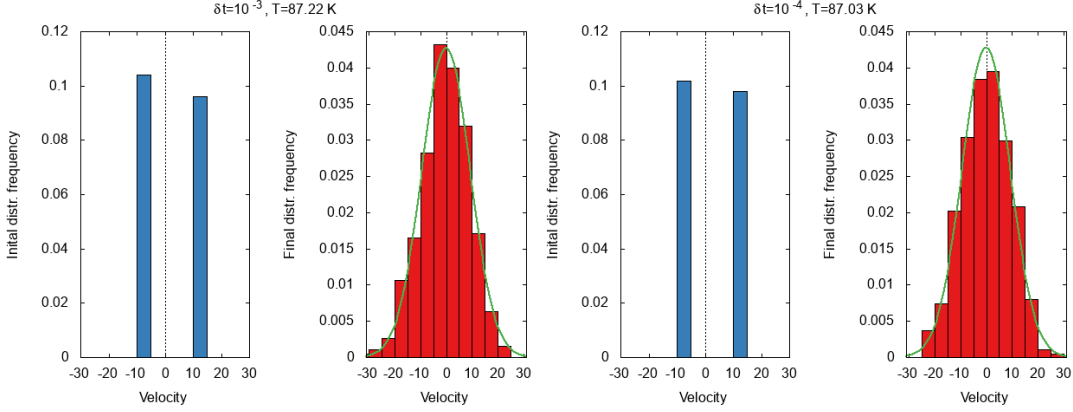


FIG. 3: Initial and final velocity distribution for a time increment of $\delta t = 10^{-3}$ (left) and $\delta t = 10^{-4}$ (right). The integrator used was the Velocity Verlet algorithm.

Finally, looking at the distribution of velocities, at first the system was forced to have a bimodal distribution compatible with a temperature of $k_B T = 100\epsilon$ with 0 mean. Then, after the system has been let develop, we expect to find it in a Maxwell-Boltzmann distribution centered at zero and with a temperature near the original one.

As we can see in figure 3, the system departed from the proposed bimodal distribution and ended in the targeted distribution, as can be proved by fitting the expression:

$$f(x) = \frac{1}{\sqrt{2\pi T}} e^{-\frac{x^2}{2T}}$$

Also, there is no apparent difference in computing the simulation with a $\delta t = 10^{-3}$ or $\delta t = 10^{-4}$.

Analysis of properties of a Lennard-Jones liquid

For this simulation, the system was initialised again as a SCC configuration. This this, instead of letting it evolve freely only applying a certain potential and Periodic Boundary Conditions, an Andersen thermostat algorithm was implemented to keep the system at $T' = 100$. The code made it evolve during 10.000 timesteps, along which the system aquired velocities compatible with the thermal bath imposed. Once the system was initialised, the thermal bath was changed to one with $T' = 2$, and was let to develop along 500.000 timesteps and the kinetic, potential and total energies were computed, as well as the system's pressure. Also, all this computations were done for four different values of the density: $\rho = 0.2, 0.4, 0.6, 0.8m/\sigma^3$.

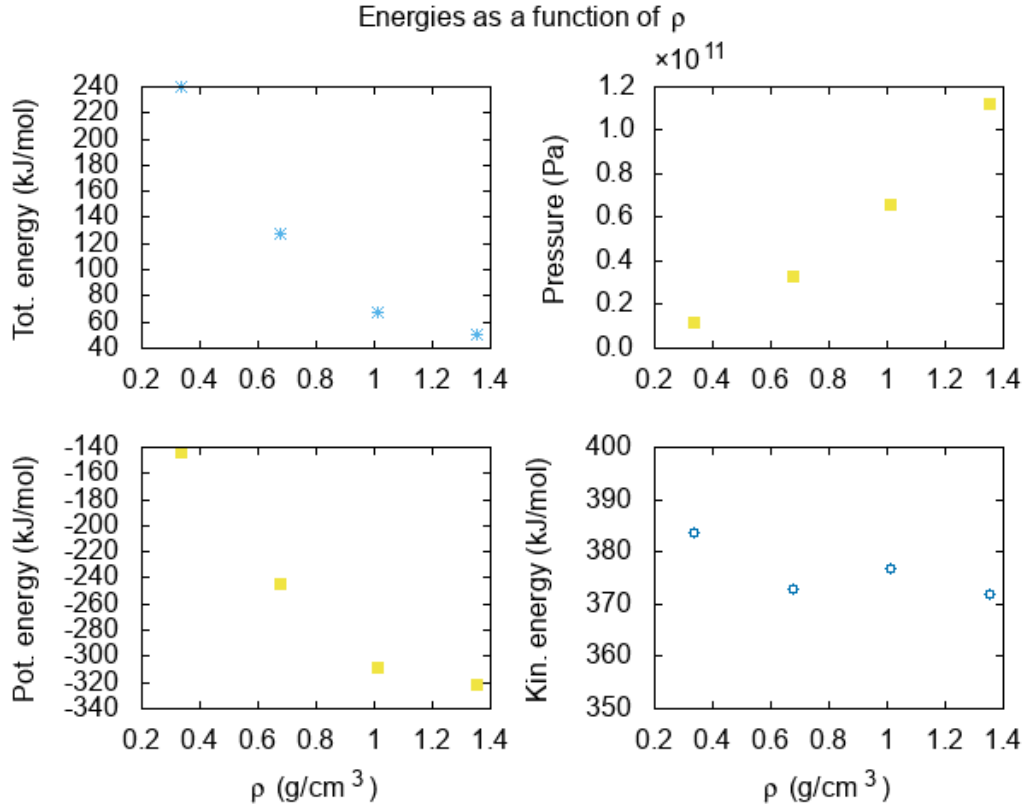


FIG. 4: Values for the expected value of energies and pressure once the system has equilibrated. Integration done with the Velocity Verlet algorithm.

As can be seen in figure 4, the potential energy is the one that rules over the behaviour of the total energy. The kinetic energy is fairly stable as it depends on the velocity and the velocity depends on the temperature, the values of which are bounded by the thermal bath independently of the density. Looking at the pressure, its values make sense: as the density rises, we have the same amount of particles in less space, which intuitively translates into an increment of the pressure.

The mentioned behaviour of the pressure can be better seen in figure 5. The pressure begins at really high values as the initial configuration is at a much more high temperature than the one the system has once it reaches stability. That happens approximately at $t \approx 15$ ps, from where the values of the pressure only fluctuate around a certain value for each density. A zoomed graphic is also provided, where the stability values can be guessed.

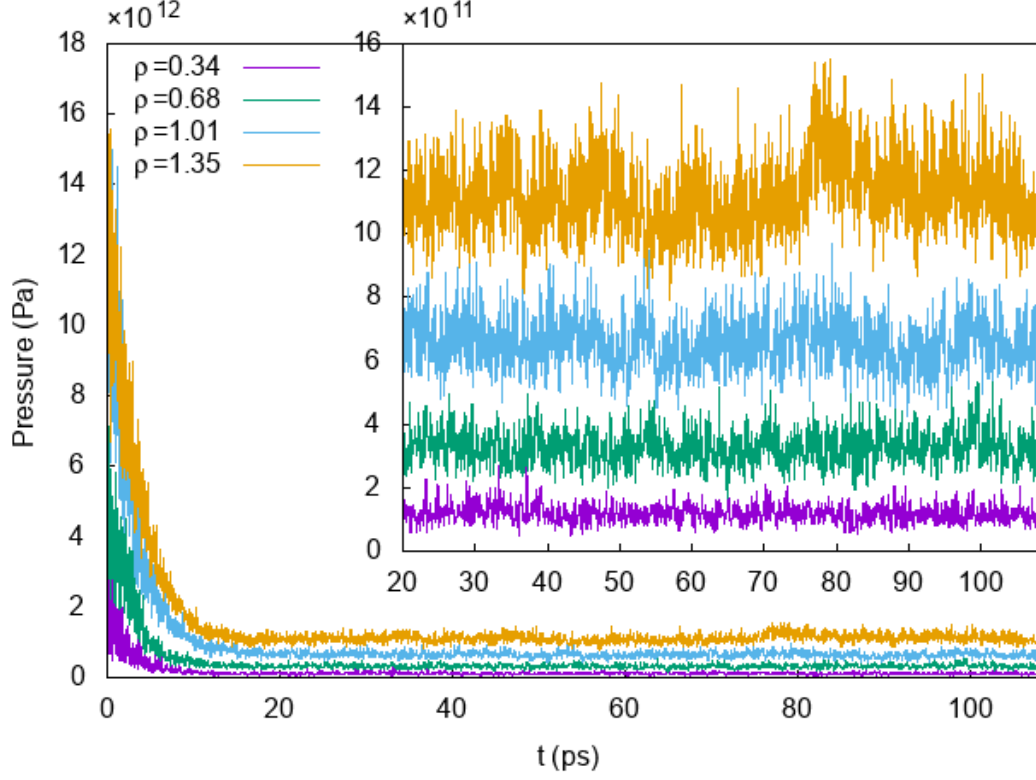


FIG. 5: Evolution of the pressure with t (in picoseconds) for each density. Integration done with the Velocity verlet algorithm.

The transformation applied to change from reduced units to the proper ones are the following:

$$\begin{aligned}
 \text{Energy} : E &= \epsilon E' \\
 \text{Temperature} : T &= \frac{\epsilon 10^3}{N_{Av} k_B} T' \\
 \text{Pressure} : p &= \frac{\epsilon 10^3}{N_{Av} (\sigma 10^{-10})^3} p' \\
 \text{Density} : \rho &= \frac{m}{N_{Av} (\sigma 10^{-8})^3} \rho' \\
 \text{Time} : t &= \sqrt{\frac{m 10^{-3}}{\epsilon 10^3}} \sigma 10^{-10} t'
 \end{aligned}$$

The codes used to compute all data and to generate the figures will be provided via GitHub, in the MoMo repository.

-
- [1] GitHub repository: https://github.com/sowor14/Molecular_Dynamics.
[2] Carles Calero, *Class slides on Molecular Dynamics*, UB, 2021.