

Molecular Dynamics Simulation of Argon Gas

Stephen Mee¹, Mario Gely² and Hiran Daneshpour³

¹*Michigan State University (U.S.A.)*
^{2,3}*Delft University of Technology (Netherlands)*

Abstract:

We present a numerical integration of the equation of motion for a model of Argon gas. Only classical effects are considered with a Lennard-Jones interaction between particles. With this model we study free energy, pair correlation, temperature and pressure values for liquids, gas and solid states. Our model is confirmed by energy conservation and the ideal gas behavior at high temperature.

© 2015 Optical Society of America

1. Introduction

There exist essentially two methods for determining physical quantities such as temperature and pressure as statistical averages over a restricted set of states: the molecular dynamics and Monte Carlo methods. Molecular dynamics is a widely used method for studying classical many-particle systems. It consists essentially of integrating the equations of motion of the system numerically. It can therefore be viewed as a simulation of the system as it develops over a period of time. Its greatest advantage is that it not only provides a way to evaluate expectation values of static physical quantities; dynamical phenomena, such as transport of heat or relaxation of systems far from equilibrium can also be studied. The solutions of the equations of motion describe the time evolution of a real system although obviously the molecular dynamics approach is approximate for the following reasons. [1]

- Instead of a quantum mechanical treatment we restrict ourselves to a classical description for the sake of simplicity
- The system sizes in such simulations are much smaller than those of experimental systems. The finite-ness of the system size is felt through the presence of the boundary. The convention adopted in the vast majority of molecular simulations is to use periodic boundary conditions.
- The numerical integration algorithm is not infinitely accurate. This forces us to make some optimum choice between speed and accuracy: the larger the integration time step, the more inaccurate the results of the simulation...

In this report we shall implement a molecular dynamics simulation of an Argon gas. The complete octet in the outer atomic shell makes argon stable and resistant to bonding with other elements. Interaction with other particles can therefore be modeled accurately by the Lennard-Jones potential:

$$V = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6\right] \quad (1)$$

2. Theory

2.1. Initial conditions

The simulation is started in a face-centered cubic lattice with a distance σ between closest neighbors which is the lowest energy state corresponding to the Lennard-Jones potential. By starting in the lowest-energy bound state, we reduce the probability of the simulation creeping up and settling into a higher energy "false" bound state. This can be supported by running the simulation at different temperatures and comparing the final states of each run. For low temperatures, the Argon atoms are likely to settle back into the original cubic lattice. For higher temperatures, the atoms are likely to "overflow" the edges of the bound state in the potential and we can reach liquid or gas states. Each unit cell contains four particles, assuming we want to run our simulation in a square box containing a finite number of cells M^3 , the number of particles involved in the simulation is limited to the set of integers $4M^3$ with M an integer. Therefore we can run our simulation for 4, 16, 108, 256, 500, 864, .. particles.

The initial momenta for each particle are selected randomly from a Maxwell-Boltzmann distribution corresponding to the initial temperature of the system. Note that for such a finite number of particles, this method will necessarily yield a non-zero total velocity. This velocity is calculated and subtracted to all velocities at the beginning of the simulation.

2.2. Boundary conditions

The use of periodic boundary conditions is for the purpose of extending our "small" sample size to a infinite space. It is possible to think of the simulation as infinitely repeating cubic volumes of identical components. Without these boundary conditions, we would also have to worry about the inevitable possibility of atoms leaving the scope of the simulation and becoming extremely spread out. In practice, this has implications on the force and position calculation. To illustrate this, let us consider a one dimensional simulation in a one dimensional box of size L (i.e. the segment $[0, L]$). If the position of a particle exceeds L by a value dl , then it is repositioned at a distance dl of the origin as in the beloved computer game snake. For the force calculation let us assume for simplicity that we are calculating the forces on a particle situated at the origin. The contribution of a particle at a distance d superior to $L/2$ will then be taken as the interaction with a virtual particle situated at $-(L-d)$.

2.3. Integrator

The equations of motion are integrated using the most widely used algorithm for molecular dynamics [1], the Verlet algorithm. This leads to an error per integration step of the order of dt^4 [1] where dt is the time step used in the implementation. Which is a considerable improvement on the naive algorithm.

Each time step, the force is calculated using the Lennard-Jones potential:

$$F = -\nabla V = 24\epsilon \left[2 \left(\frac{\sigma^{12}}{r^{13}} \right) - \left(\frac{\sigma^6}{r^7} \right) \right] \quad (2)$$

In order to compute the force for each each particle, we have use an two embedded loops that go through all the particles. The number of computational steps after a bit of optimization is proportional to $N^2/2$ where N is the number of particles considered in the simulation. This constitutes the most computationally expensive operation of the simulation. To save computational time, we only calculate the force if the distance between two particles is above a critical distance. Since distant particles do not contribute significantly to the value of the force. A critical distance of 2.5σ is chosen according to [1].

Although the temperature is fixed at the start through initial conditions, it does not maintain its original value throughout the simulation. In order to correct for this fact, we implement a thermostat which corrects every 10 time-steps the value of the velocities until the temperature is stabilized. This behavior is shown in fig. 1, the target temperature is 10 and we get a mean value of 9.986 with a variance of 0.009.

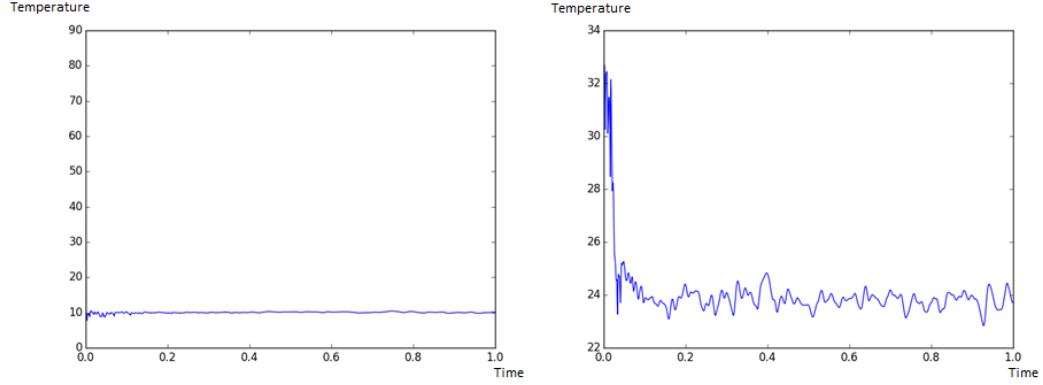


Fig. 1. On the left the thermostat is deactivated. On the right the thermostat is activated until $t=0.4$

3. Results

Whereas most of the model is implemented in python, the nested loop necessary for the force calculation (the bottleneck of this algorithm) is written in Fortran for speed. The Fortran code is linked to python with the Fortran to Python Interface called f2py. The constants corresponding to physical constants or properties of Argon are taken equal to one. These include the mass of each particle, σ and ϵ the two parameters in the Lennard-Jones potential and the Boltzmann constant. The number of particles is fixed to 864 at which the simulation takes a few seconds with a time step of 0.004.

A first result which proves the validity of our model is the conservation of the total energy. The potential energy is calculated as the sum of all the interaction potentials between two different particles, and the kinetic energy is taken to be the sum of the kinetic energy of each particle. The total energy is conserved with a standard deviation of 0.6 after the simulation converges as illustrated in fig 2 for $T=0.1$

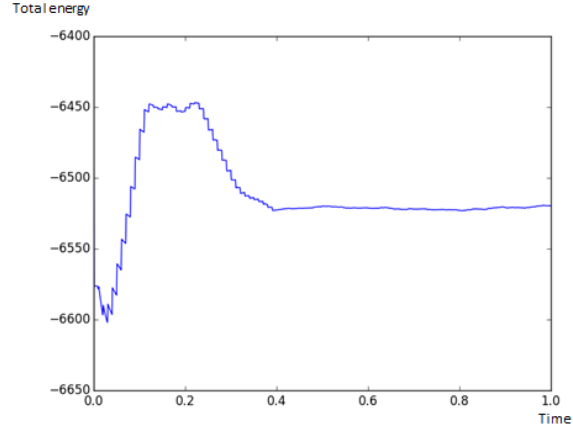


Fig. 2. Time evolution of the total energy

The computation of the correlation function leads to knowledge about the state of the gas of Argon atoms. Indeed, depending on the initial temperature we get different results. In fig. 3 we see, from left to right the correlation function plotted for temperatures $T = 0.1, 1, 10$ corresponding to a solid, a liquid and a gas respectively. Also plotted are a 2 dimensional representation of the final positions of the particles.

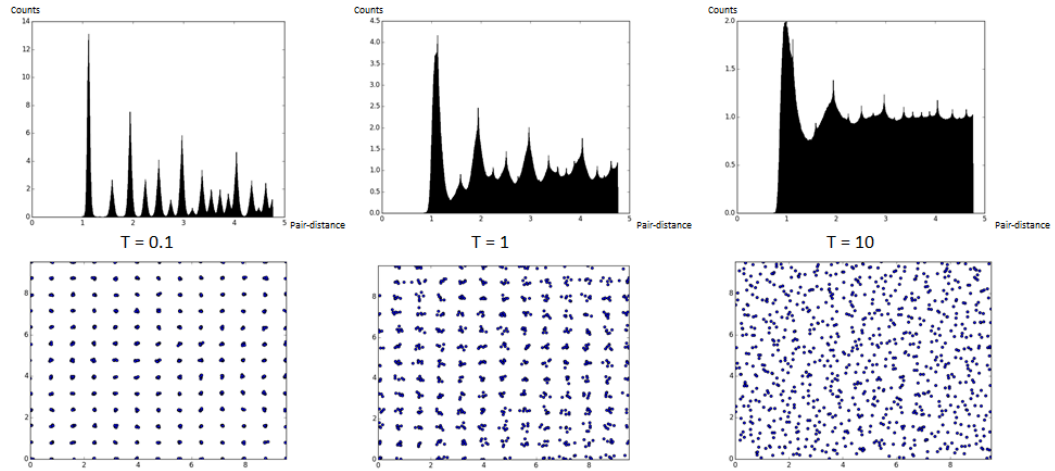


Fig. 3. From left to right, the final positions and correlation functions for simulations with $T = 0.1, 1$ and 10

The pressure is calculated with the Virial equation corrected for the cut-off in the potential calculation [1]

$$\frac{p}{nk_B T} = 1 - \frac{1}{3Nk_B T} \left[\sum_i \sum_{j>i} r_{ij} \frac{\partial U(R)}{\partial r_{ij}} \right]_{cut-off} - \frac{2\pi N}{3k_B T V} \int_{r_{cut-off}}^{\infty} r^3 \frac{\partial U(r)}{\partial r} g(r) dr \quad (3)$$

This allows the calculation of pressures for different temperatures as presented in the following. Note the convergence to an ideal gas state as the temperature increases (P/T converges to

1).

Temperature	Pressure	Variance
0.1	1.17	4.22e-07
1	2.12	1.13-04
10	11.15	8.32-03
50	51.2	0.23

4. Conclusion

We have achieved a realistic simulation of an Argon gas as shown by the energy conservation as well as the high temperature, ideal-gas limit. This allows us to explore properties such as the pressure in solid liquid phase, phase transitions or the validity range of an ideal gas approximation.

The simulation speed could be further improved by keeping track of the particles which are close to each other with a method such as the linked-cell method. This would allow simulations with more particles. We could also investigate other properties of the Argon gas such as the heat capacity.

5. References

[1] - "Computational Physics" J.M. Thijssen