

Manual for Matlab Molecular Dynamics Simulation Script

Xiaoquan Sun

07/12/2015

Contents

1. Introduction	1
2. Parameters and output files	1
3. Algorithms	3
3.1. Periodic boundary conditions	3
3.2. Molecular Dynamics	4
3.2.1. Initial conditions	4
3.2.2. Compute forces	5
3.2.3. The Verlet integrator	6
3.3. Diffusion coefficient	7
4. Analysis	8
4.1. Energy	8
4.2. Temperature	8
4.3. Diffusion coefficient	9
5. Reference	10

1. Introduction

MD.m is a Matlab script to perform a classical molecular dynamics (MD) simulation for a simple system. The system contains 100 particles in a cubic box. Particles have only Lennard-Jones (LJ) Van der Waals interactions among each other. Initial velocities are assigned to particles according to a given temperature.

The script can be simply used by **matlab < MD.m** from command line interface. During the simulation, the script outputs kinetic energy, potential energy and total. Mean square displacement (MSD) is also generated to calculate diffusion coefficient.

Many inspiration and ideas are got from *Understanding Molecular Simulation: From Algorithms to Applications*¹.

2. Parameters and output files

In the script, there are some parameters that can be changed to modify the system.

1. **kb**: Boltzmann's constant. Default is 1 to simplify the simulation.
2. **natom**: Number of particles. Default is 100.
3. **boxsize**: The side-length of the cubic box. The unit is angstrom. Default is 14. This value is calculated from the density of water.
4. **mass**: Mass of one atom. Default is 1 to simplify the simulation.
5. **dt**: Time step. The unit is second. Default is 0.001.
6. **temp**: Temperature. The unit is K. Default is 300.
7. **step**: Total running step. Default is 100000.

8. **outstep**: Output frequency. Default is 50.
9. **epsilon**: A parameter for LJ potential, the depth of the potential well. Default is 1 to simplify the simulation.
10. **sigma**: A parameter for LJ potential, the distance at which the potential among particles is 0. Default is 2.5.
11. **rc**: Cutoff distance, the LJ potential is truncated to 0 at the cutoff distance. Default is $2.5 \times \text{sigma}$.

The script has four output files.

1. **md.out** contains kinetic, potential, total energies and temperature for each output. At the end of the file, it gives the total running time and the simulation speed per second.
2. **trajectory.xyz** is the trajectory file without Periodic Boundary Condition (PBC). The format is XYZ format. It can be visualized by VMD. This is for post-simulation analysis such as diffusion coefficient calculation.
3. **trajectory_imaged.xyz** is the trajectory file with PBC. All particles move in the cubic box. The format is XYZ format. It can be visualized by VMD. This is used to check if the interactions between particles are normal, and for better visualization.
4. **msd_r.out** is the MSD over the simulation time. The data is used to calculate the diffusion coefficient.

3. Algorithms

3.1. Periodic boundary conditions

Because the system size is small (only 100 particles), it has unwanted boundary with the surroundings. To avoid this situation, and simulate an infinite big system, the periodic boundary condition is used. All particles in copies of itself of a given cubic box will be translated into its surrounded environment. The extended system is comprised by infinite copies of the original box and has no boundaries any more (Figure 1).

During the simulation, only particles in the original box are used to calculate properties of the system. In the script, it can be done using **`position-boxsize*round(position/boxsize)`** to image particles outside the original box back for further calculations.

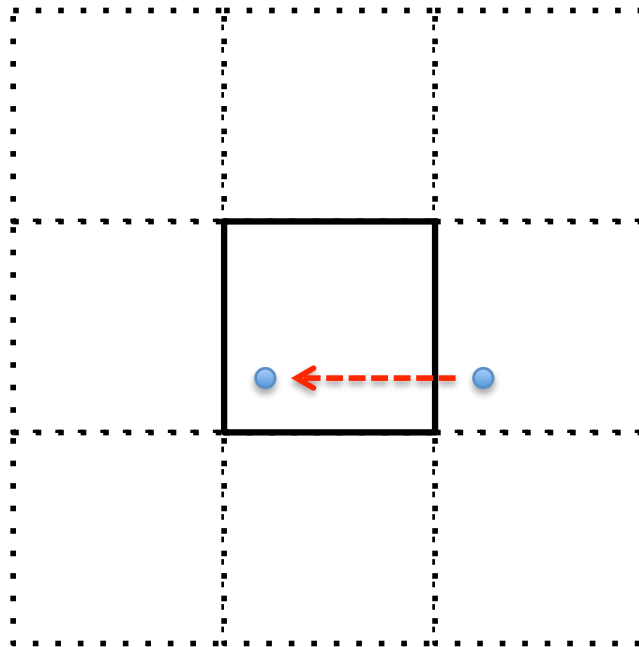


Figure 1. Illustration of PBC. The particle outside the original box (solid box) is translated back.

3.2. Molecular Dynamics

Classical MD simulation is based on Newton's equation of motion. The forces are the derivatives of the potential function. In this case, the potential function is LJ potential². The global flow scheme for MD simulation is in Figure 2. At the beginning, a set of initial coordinates and velocities are given. LJ potentials as well as forces can be calculated according the initial positions. Then the simulation runs in a small time step to predict future positions, velocities, and forces using the integration method. Velocity Verlet algorithm³ is used in this script. When the simulation time is up, the loop is ended. The script will output the energies and trajectory of this system.

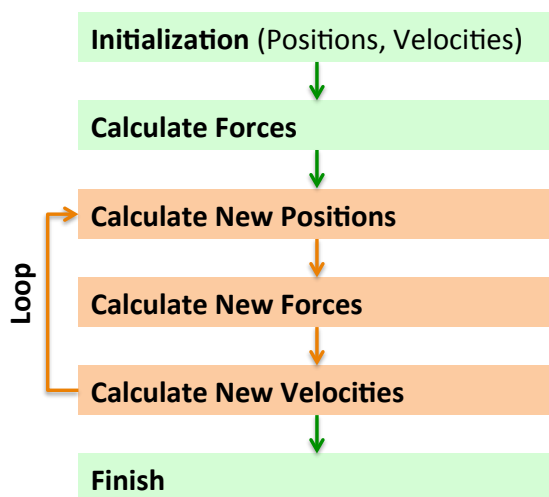


Figure 2. The global flow of MD simulation in this script.

3.2.1. Initial conditions

All particles are located in a simple cubic lattice. Since 100 particles cannot distribute in the cubic uniformly. A bigger box will be build that can contain N^3 particles evenly. In this case,

N is 5. Then 100 of them will be chosen randomly. The potential energy and forces are also calculated.

Initial velocities are given randomly. They are directly related to the system temperature.

According to the equipartition theorem, the relationship in this system is

$$\frac{\sum_{i=1}^{100} \frac{1}{2} m v_i^2}{100} = \frac{3}{2} k_b T$$

In this script, the initial random number is between 0 and 1. They are scaled to make sure the total momentum to 0 to avoid the system move as a whole. Then all velocities multiply a factor to make the system to the given temperature. Kinetic energy is calculated in this step as well.

3.2.2. Compute forces

The forces are the negative derivatives of the potential energy. Only LJ Van der Waals potential is considered in this case. LJ potential is a simple mathematical model of Van der Waals interactions among particles.

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

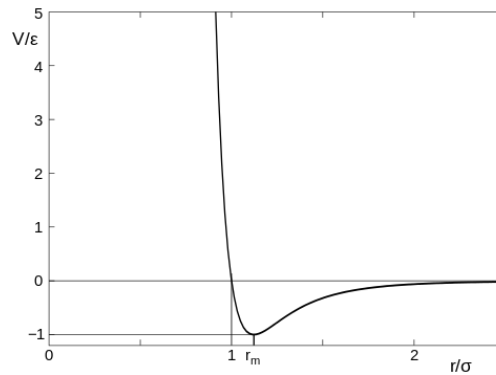


Figure 3. LJ potential changes with the distance between two particles.

The derivative of LJ potential at x direction is below.

$$F_x = -\frac{\partial V_{LJ}(r)}{\partial x} = -\left(\frac{x}{r}\right)\left(\frac{\partial V_{LJ}(r)}{\partial r}\right) = -\left(\frac{x}{r}\right)\left(\frac{24\epsilon\sigma^6(r^6 - 2\sigma^6)}{r^{13}}\right) = \frac{24x\epsilon\sigma^6(2\sigma^6 - r^6)}{r^{14}}$$

Two particles have strong repulsive force when the distance between them is less than σ . When the distance is σ , there is no interaction between them and the LJ potential is 0. When the distance is larger than σ , the two particles will attract each other. With the increase of the distance, the attractive interaction will increase and then decrease and become 0 at the infinite distance. The potential of the strongest attractive interaction at the turning point is ϵ . ϵ is also called the depth of potential well. These two parameters can be calculated using Quantum mechanics simulation numerically. In this case, ϵ is set to 1. And σ is adjusted according to the cutoff distance to make sure it less than half of the box.

To speed up the simulation, LJ potential is truncated at the cutoff distance, which is usually set as 2.5σ . The potential at the cutoff distance is thought to be 0, and all potential energies in the simulation are scaled by the potential at the cutoff distance as below.

$$V_{LJ(truncated)}(r) = \begin{cases} V_{LJ}(r) - V_{LJ}(r_{cutoff}), & r \leq r_{cutoff} \\ 0, & r > r_{cutoff} \end{cases}$$

3.2.3. The Verlet integrator

The Velocity Verlet algorithm is implemented in this script to integrate Newton's equation of motion. This is one of Verlet integration methods. It is derived by Taylor expansion of the position and velocity at time t. And the update position and velocity are below.

$$r(t + \Delta t) = r(t) + v(t)\Delta t + \frac{1}{2}a(t)\Delta t^2$$

$$v(t + \Delta t) = v(t) + \frac{1}{2}a(t)\Delta t + \frac{1}{2}a(t + \Delta t)\Delta t^2$$

In the loop of the script, the update position $r(t + \Delta t)$ is calculated first. And the update LJ potential and forces can be calculated using the new positions. Update velocities as well as the kinetic energies are then calculated at the end of a loop.

3.3. Diffusion coefficient

The diffusion coefficient can be calculated by MSD of all particles in the system. The relation is first derived by Albert Einstein in 1905. The script outputs MSD at the end of the simulation. It also has the trajectory file without PBC (trajectory.xyz) that can be used for the post-simulation analysis. MSD of the system with 100 particles can be calculated by the following equation.

$$MSD = \frac{1}{100} \sum_{i=1}^{100} [r_i(t) - r_i(0)]^2$$

The diffusion coefficient can be calculated by the following equation.

$$D = \lim_{t \rightarrow \infty} \frac{MSD}{6t}$$

4. Analysis

4.1. Energy

The total energy should be approximately stable during the simulation. Figure 4 shows how the total energy changes with time. Since some parameters are dimensionless, the unit of total energy is not specified in the figure. The change of the total energy is big (red line), but the average remains relatively constant (black line). This might be caused by improper parameters of LJ potential or the big value of **dt**. In a real MD simulation, **dt** is usually set to around 10^{-15} .

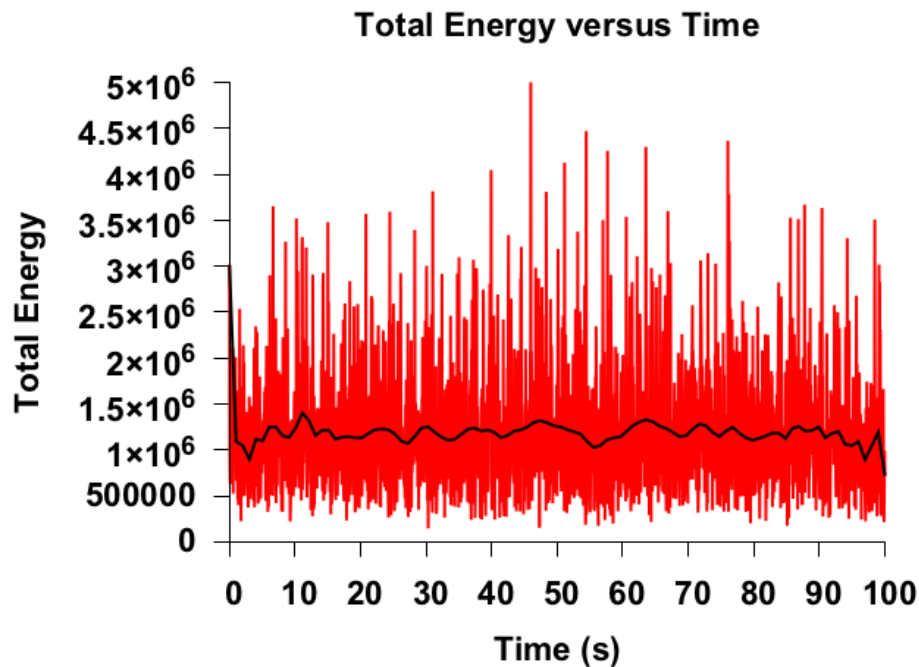


Figure 4. Total energy changes with the simulation time.

4.2. Temperature

The system has constant volume and constant number of particles. Because it also does not have energy exchange with environment, this is a NVE system. There is no thermostat in the script. The change of temperature is shown in Figure 5. In this short simulation, the temperature is approximately stable. Although it shows a trend to increase at the end of the simulation, it might decrease again for a longer simulation.

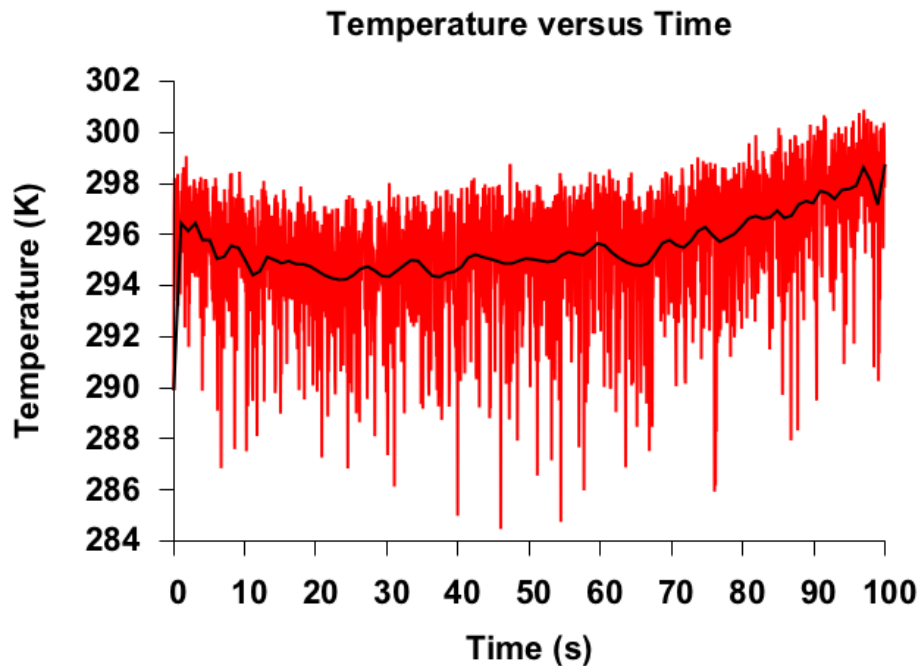


Figure 5. Temperature changes with the simulation time.

4.3. Diffusion coefficient

The script is tested by the default configuration with Matlab 2015a on AHPCC. The following figure is MSD versus time. The diffusion coefficient of the particle is about $1.25 \times 10^{-13} \text{cm}^2/\text{s}$. Compared with the experimental diffusion coefficient of water at 25°C ($\sim 2.30 \times 10^{-5} \text{cm}^2/\text{s}$)⁴. The particles move much slower than atoms in the liquid.

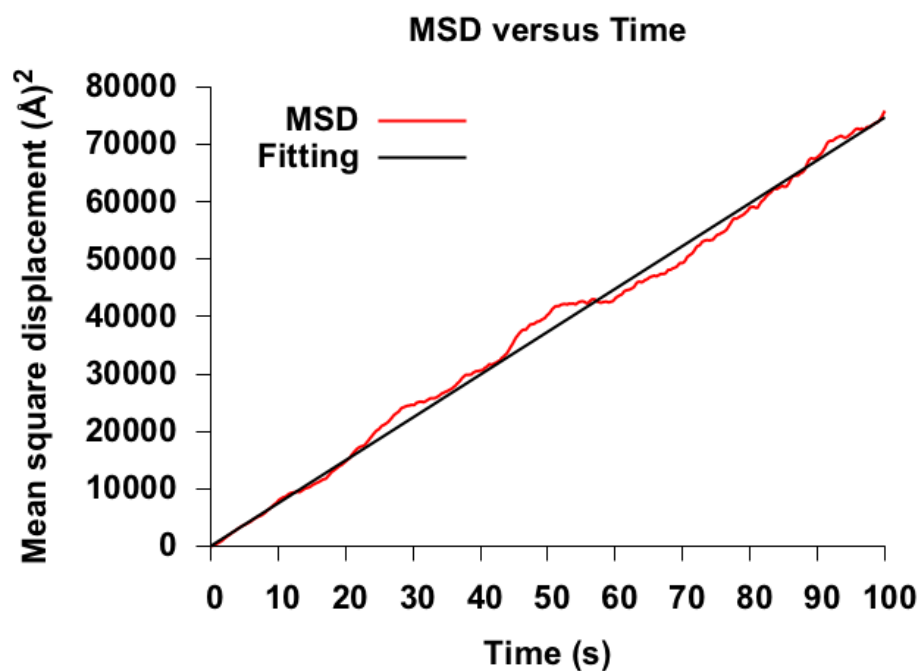


Figure 6. MSD change with time.

5. Reference

1. Understanding Molecular Simulation, Second Edition: From Algorithms to Applications. (Academic Press, 2001).
2. Jones, J. E. On the Determination of Molecular Fields. II. From the Equation of State of a Gas. Proc. R. Soc. Lond. Math. Phys. Eng. Sci. 106, 463–477 (1924).
3. Swope, W. C., Andersen, H. C., Berens, P. H. & Wilson, K. R. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. J. Chem. Phys. 76, 637–649 (1982).
4. Krynicki, K., Green, C. D. & Sawyer, D. W. Pressure and temperature dependence of self-diffusion in water. Faraday Discuss. Chem. Soc. 66, 199–208 (1978).