MOLECULAR DYNAMICS SIMULATIONS

Objectives

After completing the reading of this chapter, you will be able to:
Realize the importance of computer experiments.
Write the equation of motion of an atomic/a molecular system.
Design a simple MD simulation program.
Write different forms of theVerlet algorithm.

Keywords

Molecular Dynamics, Potential, Force,
Equation of motion, Verlet algorithm.

Introduction

With the advent of the computers, chemists, physicists and material scientists had begun(since 1950s) to exploit the power of the computers for probing the properties of materials through simulations. Almost all the materials of the physical world can be probed for their properties by designing appropriate simulation algorithms. These include atomic and molecular systems, biomolecules, complex materials, nuclear materials, life processes and the like.

One can construct simulation schemes for the dynamics of the molecules present in the materials; i.e., the Molecular Dynamics (MD), where the constituents of the system are allowed to interact according to known laws of physics, over a period of time. Through the numerical solutions of the equations of motion (often described by the laws of Newtonian mechanics), one obtains the trajectories (position coordinates and/or velocities) of all the constituents of the system, under the influence of the interacting potential.

These trajectories are then analyzed in order to extract the desired properties such as pressure, stress, diffusion, viscosity, surface tension, dielectric constant, order parameter, autocorrelation functions, fluctuations, conformational changes etc. Since molecular systems generally consist of a vast number of particles, it is impossible to find the properties of such complex systems analytically. The MD simulation exercises circumvent this problem by using the numerical solutions of the equations of motion. Thus the MD simulation technique presents an interface between laboratory experiments and the theory. This often leads to the realization that 'computer simulations' are actually 'computer experiments'.

The MD technique

We begin with a system of particles (atoms, molecules, united atoms, species etc) which is governed by the equation of motion,

$$m_i\left(\frac{d^2r_i}{dt^2}\right) = f_i \quad (26.1)$$

where m_i is the mass of the i-th particle, f_i is the force on it and r_i represents its position coordinates. The computation of the force f_i involves the calculation of the derivative of the interacting potential, $U(r_1, r_2, \ldots, r_N)$,

$$f_i = -\left(\frac{\partial U(r_1, r_2, \ldots, r_N)}{\partial r_i}\right) \quad (26.2)$$

In each of the time step of the simulation, one needs to compute the force f_i and using this force, the position r_i gets updated. In order to solve the second order differential equation as in eq. (26.1), there are several numerical schemes available. These are based on finite difference methods and the integration algorithms include Gear predictor-corrector algorithm, Verlet algorithm and the Toxvaerd algorithm. The Verlet algorithm and its several variations are the most widely used by the practitioners of the trade and we describe this algorithm below.

The Verlet algorithm

Equation (26.1), when integrated using the Verlet integration algorithm involves the computation of the positions at different times using the Taylor expansion about $r(t)$, where Δt is the time step. Thus,

$$r(t+\Delta t) = r(t) + \Delta t v(t) + \frac{1}{2}(\Delta t)^2 a(t) + \cdots \qquad (26.3a)$$

$$r(t-\Delta t) = r(t) - \Delta t v(t) + \frac{1}{2}(\Delta t)^2 a(t) - \cdots \qquad (26.3b)$$

Making use of these two expressions, the next step position $r(t+\Delta t)$ is easily found out,

$$r(t+\Delta t) = 2r(t) - r(t-\Delta t) + (\Delta t)^2 a(t) \qquad (26.4)$$

Although the velocities $v(t)$ are not required to compute the trajectories, those are useful for the computation of kinetic energy (hence, total energy) and the velocity auto-correlation functions. Following eqs (26.3a) and (26.3b), one may write,

$$v(t) = \left(\frac{1}{2\Delta t}\right)[r(t+\Delta t) - r(t-\Delta t)] \quad (26.5)$$

In the Verlet integration scheme, the errors in calculated positions$r(t)$ are of the order of Δt^4 while those in calculated velocities $v(t)$ are of the order of Δt^2. The algorithm is exactly reversible in time and easy to program.

The leap-frog Verlet algorithm

In the Verlet algorithm, eq. (26.4), a small term (of the order of Δt^2) has been added to the difference of two large terms (of the order of Δt^0).This may bring in some numerical imprecision in calculating the trajectories. In order to avoid this, a half-step 'leap frog Verlet' scheme has been proposed. This takes the form,

$$r(t+\Delta t) = r(t) + \Delta t v\left(t+\frac{1}{2}\Delta t\right) \text{(26.6a)}$$

$$v\left(t+\frac{1}{2}\Delta t\right) = v\left(t-\frac{1}{2}\Delta t\right) + \Delta t a(t) \text{(26.6b)}$$

This modified Verlet algorithm needs current positions $r(t)$ and accelerations $a(t)$ and the mid-step velocities $v\left(t-\frac{1}{2}\Delta t\right)$. In the actual practice, eq (26.6b) is implemented first and the velocities leap over the coordinates to give the next mid-step values,$v\left(t+\frac{1}{2}\Delta t\right)$. The current velocities are calculated as,

$$v(t) = \frac{1}{2}\left[v\left(t+\frac{1}{2}\Delta t\right) + v\left(t-\frac{1}{2}\Delta t\right)\right] \qquad \text{(26.7)}$$

This is necessary as the total energy at time t can be calculated. Following this, eq. 26.6(a) updates the position, once more ahead of the velocities. The new accelerations $a(t)$ are then calculated, ready for the next step. The advantages of the 'leap-frog Verlet' algorithm are: we do not need to take difference of two large quantities to obtain a small one, thus no loss of precision; and, as the velocities appear explicitly, adjusting the simulation energy is achieved through appropriate scaling of the velocities at time t.

The velocity Verlet algorithm

In the leap-frog Verlet algorithm, the current step velocities are still calculated from the mid-step velocities $v\left(t+\frac{1}{2}\Delta t\right)$ and $v\left(t-\frac{1}{2}\Delta t\right)$ [see eq. (26.7)]. In order to avoid this, the 'velocity Verlet algorithm' has been proposed which utilizes positions $r(t)$, velocities$v(t)$ and accelerations $a(t)$, all at the same time. This is based on the following equations,

$$r(t+\Delta t) = r(t) + \Delta t v(t) + \frac{1}{2}(\Delta t)^2 a(t) \text{(26.8a)}$$

$$v(t+\Delta t) = v(t) + \frac{1}{2}\Delta t[a(t) + a(t+\Delta t)] \text{(26.8b)}$$

This algorithm requires storage of $r(t)$,$v(t)$ and $a(t)$. The actual implementation involves two stages. First, the new positions at time $t+\Delta t$ are calculated using eq. (26.8a), and the velocities at mid-step are computed using,

$$v\left(t+\frac{1}{2}\Delta t\right)=v(t)+\frac{1}{2}\Delta t a(t) \quad (26.9)$$

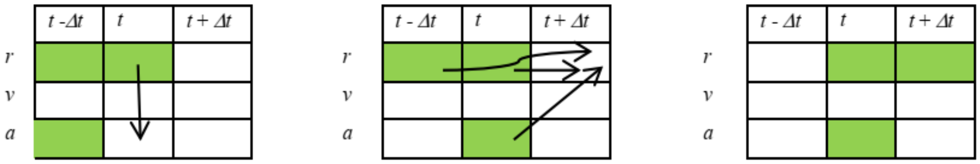
The forces and accelerations attime $t+\Delta t$ are then computed and the velocity move completed.

$$v(t+\Delta t)=v\left(t+\frac{1}{2}\Delta t\right)+\frac{1}{2}\Delta t a(t+\Delta t) \quad (26.10)$$

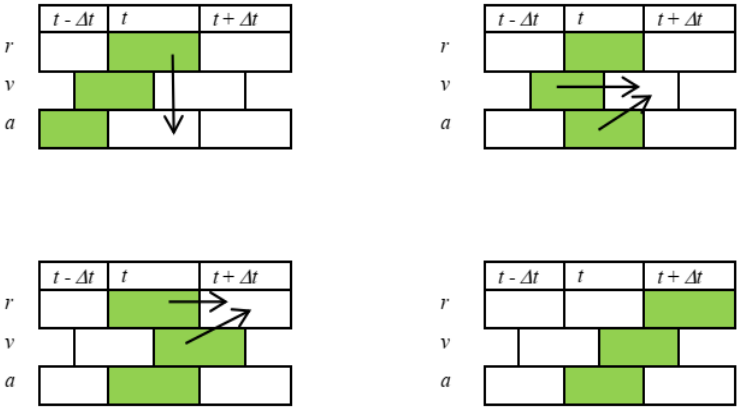
At this point, the kinetic energy at time $t+\Delta t$ is available. The potential energy at this time will have been evaluated in the force loop.

The three forms of the Verlet algorithm are compared in figure 26.1.

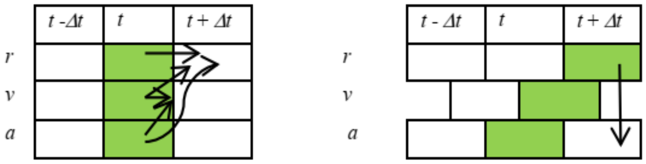
(A) The original Verlet method



(B) The leap-frog Verlet method



(C) The velocity Verlet method



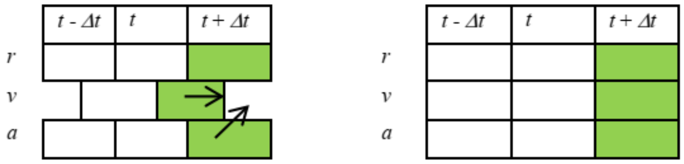


Figure 26.1: Three different forms of the Verlet algorithm. (A) Original Verlet method (B) Leap-frog Verlet method (C) Velocity Verlet method. The variables are: r (position), v (velocity) and a (acceleration). In each algorithm, the successive steps in the implementation are as shown in the blocks. The stored variables are indicated in the light green boxes.

Design of a Molecular Dynamics program

A typical MD simulation program works according to the following scheme:

The size of the simulation system is to be chosen such that the number density in the simulation box corresponds to the experimental density of the system.

1. Give initial positions $r(t=0)$ to the particles, choose an appropriate time step, Δt.

2. For the chosen potential function $U(r)$, calculate the force $F(r)$ for all the particles (eq. 26.2).

3. Move the atoms using the Verlet algorithm

4. Move time forward: $t+\Delta t$

Repeat procedure 2,3,4 as long as desired. During these operations, all the properties of interest, viz., temperature, pressure, total energy etc. need to be monitored for the subsequent analysis.

Analysis of the trajectories

In the MD simulations, we save the (x,y,z), (v_x,v_y,v_z), (f_x,f_y,f_z) of the particles at regular intervals (say once in 10 steps). The chronological array of the (x,y,z) data is called as the trajectory of the particles. We now illustrate how to make use of the trajectory data by calculating a structural quantity of the system; say, for example, the radial distribution function (RDF). The RDF is represented by the function,$g(r)$ and is defined as,

$$g(r) = \frac{V}{N^2}\langle \sum_i \sum_{j\neq i} \delta\left(r - r_{ij}\right)\rangle \qquad (26.11)$$

where, N is the total number of particles present in a volume V, r_{ij} is the distance between a pair of particles, $\delta(x)$ is a delta function and the angular brackets represent ensemble average. In the MD simulation, we compute the r_{ij}s of all the particle pairs from the trajectories and prepare the histogram of all pair separations. This is then used to calculate the RDF.

The RDF of a typical Lennard-Jones fluid [characterized by two parameters, viz., σ (interparticle distance) and ε (well depth in the potential energy curve); example, liquid argon] has the following structure.

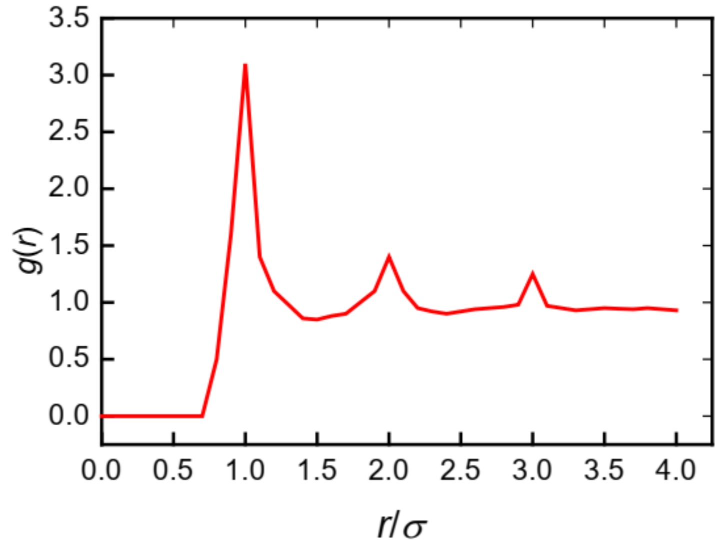


Figure 26.2: Radial distribution function for a typical Lennard-Jones fluid.

Questions

1. Discuss "a computer simulation is a mimicry of the physical world."

2. What is more important to start a MD simulation, potential energy or force?

3. Compare the three different Verlet integration schemes.

4. Is it possible to do a MD simulation by not using the Newton's equation of motion?