Homework 2: Molecular Dynamics

Please upload **one** PDF file for report and **one** archive file (.zip or .tgz) for code to Compass.

For this homework you will implement code to perform a molecular dynamics simulation of the Lennard-Jones fluid with periodic boundary conditions. To do this you will need subroutines that calculate minimum images, potentials, and forces. Please see PRAIRIELEARN for information and exercises on these subroutines.

The Lennard-Jones Fluid

Once you are comfortable with the PRAIRIELEARN exericses, you will implement a molecular dynamics code to model a Lennard-Jones liquid. In this model, particles interact via an isotropic pair potential

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

where ϵ and σ are model parameters. Shown on the second line is the form of the potential in reduced units, which is what we will use in the simulation. That is, we measure energies in units of ϵ and lengths in units of σ. For example, temperature will be in units of ϵ/k_B, and time will be in units of $\sigma\sqrt{m/\epsilon}$, where m is the mass of the particles.

Computing the Energy and Momentum

The total energy can be partitioned into kinetic and potential contributions

$$E = K + V = \left(\sum_{i=1}^{N} \frac{1}{2} m \vec{v}_i \cdot \vec{v}_i \right) + \left(\sum_{i=1}^{N} \sum_{j=i+1}^{N} v(r_{ij}) \right), \tag{2}$$

where $\vec{v}_i$ is the velocity of particle i and r_{ij} is the distance between particles i and j. For example, with 3 particles in the simulation box, the potential energy is

$$V = v(r_{12}) + v(r_{13}) + v(r_{23}). \tag{3}$$

The total momentum is

$$\vec{P} = \sum_{i=1}^{N} m \vec{v}_i. \tag{4}$$

Monitoring conserved quantities such as the total energy and momentum is an important way to ensure your simulation is physical.

Potential truncation

We will implement the "cut and shifted" Lennard-Jones potential in our small simulation cell. That is, what we will actually use is

$$v_{cs}(r) = \begin{cases} v(r) - v(r_c), & r < r_c \\ 0, & r \geq r_c \end{cases} \tag{5}$$

where v is the pair potential (1) and r_c is the cutoff radius of interaction. The reason for the shift is to remove the discontinuity at r_c.

Technically, we should add a "tail correction" back to the total energy

$$V_{tail} = \frac{1}{2}\rho N \int dr\ 4\pi r^2 g(r)\left[v(r) - v_{cs}(r)\right]. \tag{6}$$

However, we will ignore this correction to simplify the code. Since this correction is a constant shift of the total energy, it does not affect the dynamics in the microcanonical ensemble.

Initialization

We will simulate $N = 64$ particles at a number density of $\rho = 0.8442$. These values correspond to a box side length of approximately 4.232317. Set the initial temperature to $T = 0.728$, which will change over the course of the simulation, since there is nothing holding it fixed. Set the mass $m = 48$ and cutoff $r_c = 2.08$.

The initial velocities should be chosen such that the total momentum is zero and the kinetic energies follow equipartition:

$$m\langle v_{\alpha,i}^2\rangle = T \tag{7}$$

where the average is over all particles α and all components i. The simplest way to do this is to

1. Randomly set all velocities
2. Shift all velocities by a constant (find this constant!) such that the total velocity summed over all particles is zero
3. Calculate the temperature T from this set of velocities using (7)
4. Multiply all velocities by $\sqrt{T_t/T}$, where T_t is the target temperature

We will begin the particles on a cubic lattice. The choice of initial positions for the particles has two primary effects. First, it determines the initial potential energy, and in turn determines the total energy, which should be conserved throughout the simulation. Thus different initial positions can lead to different behaviors, since they can be on different energy manifolds. Second, a "bad" set of a initial positions, for example one in which two particles are sitting on top of one another, can lead to large forces and cause numerical routines to explode.

Advancing the particles

We will advance the particles with the velocity Verlet algorithm:

$$\begin{aligned}
v(t + h/2) &= v(t) + \frac{1}{2}a(t)h \\
r(t + h) &= r(t) + v(t + h/2)h \\
v(t + h) &= v(t + h/2) + \frac{1}{2}a(t + h)h
\end{aligned} \tag{8}$$

where h is a chosen time step. You can find a discussion of the algorithm as well as its various forms in chapter 3 of the textbook. In P1 you will examine the effect of the choice of time step on the simulation. In P2 you will examine the original Verlet algorithm. In P3 you will estimate σ and ϵ for a real material.

P1: Molecular Dynamics Time Step

Implement a molecular dynamics simulation as just detailed. A template code is provided on the homework page of the course website. Make sure the total momentum is conserved.

a) Using a time step of 0.01, run your simulation for at least 2000 steps. Plot a trace of the energy (the total energy at each time step in the simulation) and include it in your report.

b) Plot the standard deviation of the total energy σ_E against time step. In other words, you are trying to see how the energy fluctuates with respect to the choice of time step. For this you need to run different simulations for, say, 10 different time steps.

c) Plot the ratio of the standard deviation of the total and kinetic energies (σ_E/σ_K) against time step. From this determine the largest time step one can use and still get energy conservation better than 1% of the kinetic energy fluctuations.

d) What happens if you pick a very large time step?

P2: The Verlet Algorithm

Consider the classic Verlet algorithm that utilizes the current positions $r(t)$ and previous positions $r(t-h)$:

$$r(t+h) = 2r(t) - r(t-h) + a(t)h^2, \tag{9}$$

using acceleration at the current time step $a(t)$.

a) Show that (9) is time-reversal invariant. That is, show that if we use $r(t+h)$ and $r(t)$ as inputs we get $r(t-h)$ (up to round-off errors).

b) Suppose at a certain step we have $r(t) = 1.0001, r(t-h) = 1.0000, a(t) = 1.0000$ and a time step $h = 10^{-4}$, and that we are working with single precision, i.e. assume floating point numbers are accurate to 7 decimals places. How would this finite precision affect the algorithm?

c) Show that (9) is equivalent to velocity Verlet. Instead of the form (8), it may be easier to do this with an equivalent form

$$\begin{cases} r(t+h) = r(t) + v(t)h + \frac{1}{2}a(t)h^2 \\ v(t+h) = v(t) + \frac{1}{2}(a(t) + a(t+h))h \end{cases}. \tag{10}$$
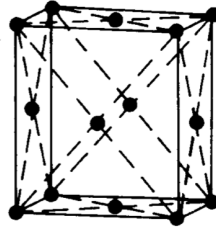
for velocity Verlet.

Figure 1: Conventional cell of the FCC structure, from Ibach & Luth (2009)

P3: Modelling FCC Copper

In ambient conditions, Copper is an FCC solid with lattice constant $a = 3.6$ Å and cohesive energy $u = -3.5$ eV per atom. Supplemental information for this problem is provided below the problem statements.

a) Using the lattice constant and cohesive energy, fit a Lennard-Jones model i.e. find σ and ϵ.

b) Fit a Morse potential model

$$u_M = D_e \left(e^{-2\alpha(r_n - r_e)} - 2e^{-\alpha(r_n - r_e)} \right), \tag{11}$$

where D_e, α, and r_e are model parameters, r_n is the nearest neighbor separation, and u_M is the cohesive energy. In order to fit this you will also need to know the bulk modulus, which is 134 GPa.

c) In the Lennard-Jones model, what is the bulk modulus?

d) Plot the cohesive energies (not the pair potential!) for both of the models on the same figure.

The FCC structure

The conventional cell of the face-centered cubic (FCC) structure is shown in Figure 1. The lattice constant a describes the side length of this cube. It can be seen here that the nearest neighbor separation is between an atom on a given corner and the atom in an adjacent face. There are *four* particles per conventional cell.

Fitting the models

Given a pair potential $v(r)$, the cohesive energy per particle in a lattice can be calculated by

$$u = \frac{1}{2} \sum_{\boldsymbol{R} \neq 0} v(R) \tag{12}$$

where the sum is over all lattice vectors $\boldsymbol{R}$ except 0. In the case of the LJ pair potential (1) this infinite sum can be performed and the result is

$$u_{\text{LJ}} = 2\epsilon \left[A_{12} \left(\frac{\sigma}{r_n} \right)^{12} - A_6 \left(\frac{\sigma}{r_n} \right)^{6} \right] \tag{13}$$

where r_n is the nearest neighbor separation in the lattice (not the lattice constant a!) and $A_{12} = 12.13, A_6 = 14.45$ for the FCC lattice. To find σ, we assume that the lattice constant a minimizes the cohesive energy. Given σ, ϵ can be solved for given the cohesive energy.

The bulk modulus

The bulk modulus B (at zero temperature) can be calculated with

$$B = \nu \frac{\partial}{\partial \nu} \left(\frac{\partial u}{\partial \nu} \right) \tag{14}$$

where u is the cohesive energy per particle, and ν is the volume per particle. Of course, neither of our models are written in terms of this volume. The two options are to either rewrite $u(r) \rightarrow u(\nu)$, or to find an expression for $\partial/\partial\nu$ in terms of $\partial/\partial r$; the latter is recommended!