Sommersemester 2019, Struktur und Simulation

Übung: A Molecular Dynamics Program

Assignment due date: 3.6.2019

Introduction

The goal of this exercise is to get a small molecular dynamics (MD) program running. We will simulate a Lennard-Jones system in the NVE (microcanonical) ensemble. As mentioned in the lectures, a simulation in the NVE ensemble should conserve N: the number of particles, V: the volume of the system, and E: the total energy (the sum of potential and kinetic energy).

There is a small skeleton MD program with some missing parts that can be found in the directory `/home/petersen/teaching/sus/4_md/`. Your assignment in this exercise will be to fill in the missing parts of the program. Copy the whole directory (with `cp -r`) to your home directory so you can edit the files. The functions that you need to implement are in the file `md.c` and marked with an `[ASSIGNMENT]` in a comment above the function.

There is a makefile supplied, so you can compile the program by typing `make`.

Outline of the program

```
input: number of atoms, density, starting temperature, timestep

setup simulation box: volume, density
setup atom positions (FCC lattice)
setup velocities (random, scaled to starting temperature)

calculate energy at start of simulation (potential and kinetic)
for each integration step:
    update positions, velocities with leapfrog integrator
    calculate kinetic energy, temperature
    print step, time, energy, temperature, energy drift
    abort if system has exploded (large change in energy)
```

Notation

- N_a: number of atoms

- N_{dim}: number of dimensions

- $N_{\text{dof}} = N_{\text{dim}} N_a$: number of degrees of freedom (dof) in the system

- k_B: Boltzmann's constant

- x, v, f, m: atom positions, velocities, forces, and masses

Leapfrog integrator

$$x_{n+1} \quad = x_n + v_{n+1/2}\Delta t \tag{1}$$
$$v_{n+3/2} \quad = v_{n+1/2} + a(x_{n+1})\Delta t \tag{2}$$

The acceleration for atom i is given by the force f_i acting on atom i and its mass m_i

$$a_i(x_{n+1}) = f_i(x_{n+1})/m_i$$

In pseudocode this looks like this

```
update positions x
calculate potential energy and forces for new positions
update velocities
```

Energy function: Lennard-Jones potential

We will use a purely repulsive form of the Lennard-Jones potential which is sometimes also called a Weeks-Chandler-Andersen (WCA) potential.

The potential function is

$$U(r_{ij}) = \begin{cases} 4\epsilon((\frac{\sigma}{r_{ij}})^{12} - (\frac{\sigma}{r_{ij}})^6) + \epsilon & \text{if } r_{ij} \leq r_{\text{cut}} \\ 0 & \text{otherwise} \end{cases}$$

The cutoff distance $r_{\text{cut}} = \sqrt[6]{2}\sigma$ is chosen so that the force vanishes at the cutoff, as the potential attains a minimum there (the global minimum actually). The energy is shifted by ϵ, which causes the energy to be zero at the cutoff as well. This means that we can use a simple cutoff instead of having to use a more elaborate switching procedure.

The forces are given by

$$F_i(r_{ij}) = \begin{cases} -24\epsilon(\frac{\sigma^6}{r_{ij}^8} - 2\frac{\sigma^{12}}{r_{ij}^{14}})\vec{r}_{ij} & \text{if } r_{ij} \leq r_{\text{cut}} \\ 0 & \text{otherwise} \end{cases}$$

with $r_{ij} = r_i - r_j$.

When calculating distances between atoms, we must take periodic boundary conditions into account. This can be done with the `rvec_sub_pbc` function that can be found in `rvec.h`.

Kinetic energy and temperature

The definition of the kinetic energy is

$$E_{kin} = \frac{1}{2} \sum_{i=1}^{N_a} m_i v_i^2$$

and the instantaneous temperature is then given by

$$T = \frac{2E_{\text{kin}}}{N_{\text{dof}} k_B}$$

Hint: you can compute the term v_i^2 with the `rvec_dot` function that can be found in `rvec.h`.

Energy conservation

The total energy of our system is given by

$$E = E_{\text{pot}} + E_{\text{kin}}$$

As we are simulating in the NVE ensemble, the total energy should (approximately) be conserved.

Assignment

Implement the missing parts in the molecular dynamics program: calculating the energy and forces arising from the Lennard-Jones potential, integrating the equations of motion with the Leapfrog integrator, and calculating the kinetic energy and the temperature as well as the total energy.

Show that the simulation approximately conserves energy. This will not work for every combination of density and timestep, but it should eventually work for sufficiently low density and a small timestep.

In the previous exercise we performed a Metropolis Monte Carlo simulation using the same Lennard-Jones energy function. Should the averages from this MD simulation be comparable to the averages of the Metropolis Monte Carlo simulation? Think about the ensembles that were simulated.

Bonus assignment

Some ideas if you get bored, ask me or google for more information to get started on these things:

- implement the Berendsen thermostat to keep temperature constant, compare simulation to NVT simulation of Lennard-Jones system in Metropolis Monte Carlo exercise

- implement cell lists for $O(n)$ force evaluation