M. Troyer, P. Hadjidoukas
ETH Zentrum, ML H 44
CH-8092 Zürich

Fall semester 2016

Set 11 - Molecular Dynamics and Cell Lists

Issued: December 9, 2016

In this exercise, we will write a two-dimensional molecular dynamics solver for N particles interacting based on the Lennard-Jones potential V_{LJ} (see Figure 1). Position $\vec{x}_i$ and velocity $\vec{v}_i$ of each particle i are governed by Newton's equation of motion

$$\frac{\mathrm{d}\vec{x}_i}{\mathrm{d}t} = \vec{v}_i, \tag{1}$$

$$\frac{\mathrm{d}\vec{v}_i}{\mathrm{d}t} = \vec{a}_i = \frac{1}{m_i}\vec{F}_i, \tag{2}$$

where $\vec{F}_i$ is the sum of all forces acting on a particle with mass m_i.

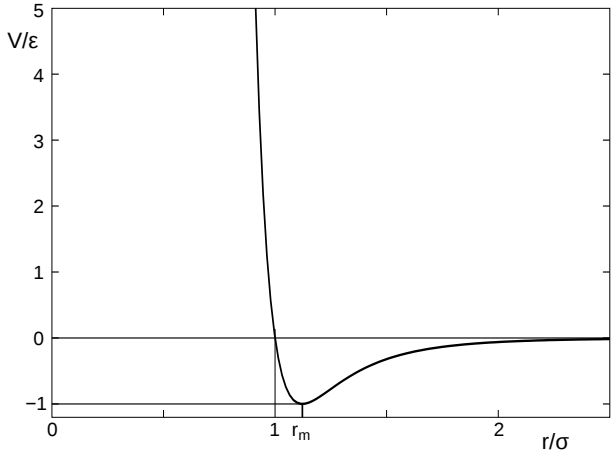

Figure 1: Lennard-Jones potential (taken from `http://en.wikipedia.org/wiki/Lennard-Jones_potential`).

To integrate these coupled ordinary differential equations in time, we use the Verlet scheme, which reads as

$$\vec{x}_i(t+\Delta t) = \vec{x}_i(t) + \Delta t\vec{v}_i(t) + \frac{(\Delta t)^2}{2}\vec{a}_i(t), \tag{3}$$

$$\vec{v}_i(t+\Delta t) = \vec{v}_i(t) + \frac{\Delta t}{2}\left(\vec{a}_i(t) + \vec{a}_i(t+\Delta t)\right). \tag{4}$$

We will calculate the forces in a simple N^2 loop and concentrate on optimizing the force calculation for the individual particle pairs in the first part of the exercise. In the second part, we will improve the scaling of the loop by introducing cell lists.

Question 1: N-Body Problem in Molecular Dynamics

Here, we will implement the Verlet algorithm for the N-body problem. For simplicity, we assume that all particles i have identical unit masses $m_i = 1$. Use the provided skeleton code `nbody_skeleton.cpp` as a starting point for your implementations.

The skeleton code already contains the initialization of the particles. The particles are initially positioned on a square lattice. Their velocities are determined at random according to a given mean kinetic energy.

The Lennard-Jones force is given by

$$\vec{F}_{LJ}(\vec{x}_i, \vec{x}_j) = -\nabla V_{LJ}(\vec{x}_i, \vec{x}_j) \tag{5}$$

based on the potential V_{LJ} and the positions $\vec{x}_i$ and $\vec{x}_j$ of two particles i and j. The Lennard-Jones potential V_{LJ} only depends on the distance $r = \|\vec{r}\| = \|\vec{x}_i - \vec{x}_j\|$ between the two particles:

$$V_{LJ}(r) = \varepsilon \left(\left(\frac{r_m}{r} \right)^{12} - 2 \left(\frac{r_m}{r} \right)^{6} \right), \tag{6}$$

where the parameters ε and r_m control the minimum and the width of the potential. We truncate the potential at radius $r_c = 2.5 \cdot 2^{-1/6} \cdot r_m$ and assume periodic boundaries.

a) Implement a function that computes the distance between two particles first. Then using this function, complete the functions which calculate the Lennard-Jones potential and force. See TODOs 1a in the provided skeleton code. As your code will spend most of the time in these functions, try to realize them in a way such that they run as fast as possible. Explain the optimizations you implemented.

b) Implement time stepping based on Verlet integration for a system of N point particles. Incorporate the force calculation of question 1a. See TODOs 1b in the provided skeleton code.

Potential and kinetic energy of a specific configuration are given by

$$E_{\text{pot}} = \sum_{i \neq j} \left(V_{LJ}(\vec{x}_i, \vec{x}_j) - V_{\text{shift}} \right), \tag{7}$$

$$E_{\text{kin}} = \frac{1}{2} \sum_i m_i \vec{v}_i^2. \tag{8}$$

The cut-off at radius r_c introduces a discontinuity into the potential $V_{LJ}(\vec{x}_i, \vec{x}_j)$, which is compensated by shifting the potential by $V_{\text{shift}} = V(r_c)$.

c) Implement a function that calculates potential, kinetic and total energy $E_{\text{tot}} = E_{\text{pot}} + E_{\text{kin}}$ of the current configuration (i.e., at each instant of time) and then writes the data to a file for further post-processing. See TODO 1c in the provided skeleton code.

d) In order to test your code, choose the following set of parameters:

$$\begin{aligned} &\Omega = [0,1]^2, \quad N = 100, \quad r_m = 0.05, \quad \varepsilon = 5.0, \quad \Delta t = 10^{-7}, \\ &n_{\text{steps}} = 100000, \quad n_{\text{dump}} = 1000, \quad E_{\text{kin}} = 10^3, \end{aligned} \tag{9}$$

where Ω denotes the domain, Δt the time-step length, n_{steps} the number of time steps, n_{dump} the time step for writing output data and E_{kin} the initial kinetic energy. Based on

these values, you have to execute:
`./nbody_serial_nocells 1.0 100 0.05 5.0 1e-7 100000 1000 1e3`.
Plot the particle positions for several time steps and check that the time evolution looks physically reasonable. Monitor kinetic and potential energy and ensure that the total energy $E = E_{\text{pot}} + E_{\text{kin}}$ is conserved during the simulation. Adapt the skeleton for the Makefile to compile your program using `make`.

e) Measure the run time of your serial code depending on the number of particles. Provide a diagram of your results.

Question 2: Cell Lists

a) Enhance your serial molecular dynamics solver from the previous question by using cell lists. Comment on (theoretical) advantages and measure the run time of the algorithm depending on the number of particles. Compare your results to the ones obtained in question 1e. Adapt the skeleton for the Makefile to compile your program using `make`.

b) Parallelize your code using OpenMP and explain your parallelization strategy. Adapt the skeleton for the Makefile to compile your program using `make`.

c) Show strong and weak scaling behavior of your code for up to 24 cores.