Problem Set 8

Exercises for the lecture Fundamentals of Simulation Methods

Prof. Dr. Friedrich Röpke, Prof. Dr. Cornelis Dullemond

Tutors: Xudong Gao (gao@mpia.de), Theodoros Soultanis (theodoros.soultanis@h-its.org), Peter Rodenkirch (Rodenkirch@stud.uni-heidelberg.de)

Tutorial on Thursday/Friday 13.12./14.12. 2018

Improvement and application of a simple Molecular Dynamics code

We provide the complete implementation of the `md.c` *according to last week's homework problem set on the course website. You can use this version (or, of course, your own solution from last week) as a basis for solving the new problems.*
The two problems are independent of each other and it is possible to solve only one of them without the other. If you solve Problem 1 first, however, the code will be more efficient in the application in Problem 2.

1. Accelerating the MD code: neighbor lists

Up until now, we have used direct sums to compute the Lennard-Jones interaction. This inefficient and we will attempt to reduce the cost of calculating the interactions by using *neighbor lists*. Remember from the lecture that the procedure is the following:

- Each particle constructs a *neighbor list* of particles that are within r_{nb} of it (in this exercise, let us use $r_{\text{nb}} = 10\sigma$). This neighbor list is updated only every N_{freq} time steps.

- Between two neighbor-list updates, each particle only computes distances/forces with particles *in its neighbor list*. We will use a cutoff of $r_{\text{cut}} = 5\sigma$ here (last week, there was a cutoff of $r_{\text{cut}} = 10\sigma$).

(a) What do you think is a good value of N_{freq} (give an approximate formula)? What are possible trade-offs in choosing it?

(b) Implement the code using the parameters: $N_{\text{1d}}^3 = 512, T = 100K, \Delta t = 0.01, \rho = \frac{1}{125}$. First, turn off velocity rescaling to go back to the NVE ensemble. Up to which values of N_{freq} do you observe conservation of energy (no drift)? How does that compare to the estimate from point a)?

(c) Let us now quantify the time we save using the neighbor list. To make our comparison sharper, comment out all the "unnecessary" functions, especially outputs:

  - all the output to the screen and files
  - all the calls to the function `update_rdf`

Fit the performance of the code with the power-law:

$$t_{\text{execution}} = t_{\text{overhead}} + a * N_{\text{freq}}^{b} \tag{1}$$

and give the resulting parameters t_{overhead}, a, b (use only data where you know from point b. that the energy is still conserved). Plot all your timing data (best is a log-log scale where you subtract t_{overhead} in the y axis), also including a line for the execution time with a "no output version" of the initial problem (i.e., no neighbor lists at all). At what frequency N_{freq} does the neighbor list outperform the original program?

2. Application: phase diagram of Lennard-Jones particles

In case you worked on Problem 1 first, switch on all functions that were disabled for the performance test 1(b). Also turn on the velocity rescaling so that you simulate NVT ensemble again. Note that we have introduced velocity rescaling after every 100 steps to (approximately) conserve the temperature.

(a) For determining the state of matter of the simulated system, we need to calculate the *radial distribution function (RDF)* $g(r)$, which corresponds to the probability density that two particles are within r of each other. We compute this on a grid between $0 \ldots L/2$ and store the frequency of all the observed distances during all the time steps (without double counting). Implement the determination and print out of the radial distribution function (RDF).

By convention, we normalize $g(r) = 1$ for an ideal, homogeneous gas, without any volume exclusion. Another way of checking whether the normalization is correct is that $g(r) = 1$ holds for very long r. Complete the normalization function in the code (right at the end of the program), and plot $g(r)$ for the starting temperatures $T = 200K$ and $T = 1K$. Which state(s) of matter do they correspond to?

(b) Construct an approximate phase diagram of the Lennard-Jones fluid. Use **25 points** on a grid: $T = (10; 20; 50; 100; 200)$ in Kelvin and $\rho = (0.001; 0.005; 0.02; 0.1; 0.5)$. Plot the phase diagram on a log-log scale for both the temperature and the density.

Use the radial distribution function output by the program to see the state(s) of matter (some of the RDF's might not be clear-cut, in those, you can label them unclear or just decide on one of the two/three possibilities).

Note: we reduced the total time steps to 10000, to reduce the waiting time before the calculation finishes. if you implemented Exercise 2, you can use neighbor lists as well, of course.