28th International Conference on Science and Technology of Complex Fluids

June 20-24, San Luis San Luís Potosí, San Luis San Luís Potosí, México.

Coarse-grained molecular dynamics simulations using GPUs: a mini-tutorial.

Instructor: Iván Guerrero

National Council of Science and Technology Research Fellow at the Institute of Physics
of the Autonomous University of San Luís Potosí, México.
http://www.ifisica.uaslp.mx/~givan

Molecular dynamics is a powerful simulation technique that is ubiquitous in science today. On the other hand, coarse-grained models are very useful nowadays to study systems that would be impossible to simulate in more detailed descriptions including explicit solvent particles, due to the astronomical number of particles that would be required. In this mini-tutorial, I would like to introduce and illustrate the use of the HOOMD-blue package to perform coarse-grained simulations of simple liquids in a hands-on approach in our biophysics lab at the Institute of Physics. HOOMD-blue is a highly optimized code that was developed originally to run natively in graphics cards or graphic processing units (GPUs).

Introduction

One of the most used potential in molecular dynamics simulations is perhaps the Lennard-Jones pair potential.

This pair potential has been used to model simple liquids such as noble gases since the 60s, and nowadays it is widely used to study the structure and macroscopic properties in complex fluids.

In this mini-tutorial, designed for non-specialists undergrad students, we will study the dependence of the some properties of identical particles interacting via a very simple Lennard-Jones pair potential as a function of the number of particles, the time, and the temperature using the HOOMD-BLUE package. This software and the documentation can be downloaded from the link:

http://glotzerlab.engin.umich.edu/hoomd-blue

Additional tools we will use include a text editor, the program gnuplot to display data. All this software is already installed in some marked PCs in the Biophysics Lab at the Institute of Physics of the UASLP.

To start, open two console terminals A and B in a marked PC. Terminal will be A used as the local terminal and terminal B as the remote terminal.

In terminal A type:

$raiz

This command will bring us to our working directory in the local PC at the Biophysics Lab.

In order to get connected to a GPU machine, type the following command in the terminal B:

$gpu

you have to answer **yes** to the question “Are you sure you want to continue connecting (yes/no)?”

and the password will be given by the instructor.

Then type

$raiz

This will bring us to our working directory in a remote GPU machine, which is placed in a cold room.

In teminal B you will find two directories.

To see the files and directories in the current directory you can type the linux command:

$ls

As an example, we will work in the directory **T_1.2**

To enter to the directory **T_1.2** just type

$cd **T_1.2**

Then, if you type

$ls

you will see several directories:

N_10_2 N_10_3 N_10_4 N_10_5 N_10_6

N_10_2 means 100 particles, N_10_3 means 10*10*10=1000 particles, etc.

To access a particular directory, e.g., N_10_2, just type

$cd N_10_2

In order to run the simulation with 100 particles at a reduced temperature 1.2 type

$time hoomdd lj_1.2_n_10_2

The command **time** will display the execution time, **hoomdd** is the executable of HOOMD-BLUE, and **lj_1.2_n_10_2** is the file where the parameters of the simulation are stored.

To see the contents of the file script, just type:

$more lj_1.2_n_10_2

A typical successful run will end with the execution time

real 9m45.528s
user 6m48.653s
sys 2m57.280s

We will consider the time “real” as the total execution time spent by the program.

In the script program the relevant parameters are

num_tot_par: total number of particles
vol_frac_par: volume fraction of the particles
tot_time_steps: total number of time steps
per= period of time

In order to come back to the previous directory type the following linux command:

$cd ../

You always can come back to the intial directory typing:

$raiz

In terminal A (local terminal) type the following commands:

$gedit time_1.2_N.txt

and write the number of particles and the execution time “real” in seconds in two columns:

100 time1
10000 time2
1000000 time3
…

When you finish to run all the scripts, copy your data from the GPU machines to your local PCs as follows:

In terminal A type:

$bajar

To see the results, type

$gnuplot

$plot 'temp_1.2_10_2.log' w l

or

$plot 'msd_1.2_10_2.log' w l

where the first file correspond to the temperature of the system and the second one the mean square displacement as a function of time.

You can compare the evolution of the temperature and the msd as a function of the number of particles entering to each directory and typing:

$raiz
$cd **T_1.2**
$cd **N_10_2**
$cp *log ../
$cd ../
$cd N_10_3
$cp *log ../
etc.

and plotting

$plot 'temp_1.2_10_2.log' w l,'temp_1.2_10_3.log' w l,'temp_1.2_10_4.log' w l,'temp_1.2_10_5.log' w l

and the same for the files with the msd.

Do not run N_10_6 because there will not be enough time in the assigned time.

Questions:

1.- How are the fluctuations in the temperature as a function of the number of particles?

2.- How is the behaviour of the msd as a function of the number of particles?

3.- What are the main differences in the system when the temperature is 1.2 and 0.6?

4.- Can you estimate how long it will take to simulate 100,000 time steps using 1,000,000 of particles, that is, the case **N_10_6**?