MD software : LAMMPS

LAMMPS is a C++ code for classical molecular dynamics with a focus on material modeling. It is an acronym for Large-scale Atomic/Molecular Massively Parallel Simulator It is also a free software optimized for OPENMP and OPENMPI libraries. The code is flexible in order to simulate various physical situations and can perform simulation with many different algorithms. The price to pay is to write an input script file which defines all parameters as well as the quantities to be monitored. All possibilities can be find in the software manual at https://lammps.sandia.gov/doc/Manual.pdf. The last version of the manual (31/07/2021) contains 2442 pages, which illustrates the complexity of the software. To start with LAMMPS is not always easy, and if you are a beginner you must follow the basic instructions and understand the meaning of the lines of a script file. The goal of this session is to illustrate the versatility of the software and the quality of the parallelization.

The binary file is called lmp (or lammps, it depends on your installation).

In order to have a brief description of lammps. type **lmp -help**. You will learn how the software has been compiled and all methods are available. If some option are missing, you can download the source tarball and compile the software with the options you can use with your computer. See appendix for an installation.

You can start a simulation by using the command

lmp -in in.system

where in. system is a script file defining the system (particle types, ensemble, forces, boundary conditions,...), the different observables to be calculated and different correlation functions (static and dynamic).

1 Lennard-Jones system : performance analysis

The script file **in.ljM2** is

```
# 3d Lennard-Jones melt

variable t index 5000
units  lj
atom_style atomic

lattice  fcc 0.8442
region  box block 0 10.0 0 10.0 0 10.0
create_box 1 box
create_atoms 1 box
mass  1 1.0

velocity all create 4 87287 loop geom

pair_style lj/cut 2.5
```

```
pair_coeff 1 1 1.0 1.0 2.5

neighbor 0.3 bin
neigh_modify delay 0 every 40 check no

#fix  1 all nve
fix 1 all nvt temp 300.0 300.0 100.0
thermo 200
thermo_style custom elapsed pe ke etotal press temp
run  $t
```

The script file corresponds to a Molecular Dynamics in the NVE ensemble for a monodisperse Lennard-Jones model in three dimensions. The number of steps is equal to 5000. The thermodynamic quantities are the mean potential energy the mean kinetic energy the mean total energy per atom, the pressure and the temperature (for this last quantity, check the ratio between ke/temp)

By using the script file **in.ljM2** compare the user time by running the program sequentially (by using one core) by using 4 cores with MPI. The version of LAMMPS provided by Ubuntu does or does not implement the OPENMP library, it depends on the version you have.

If you have a compiled version with OPENMP you can run lammps by using 4 cores with OPENMP. (In this case, first type OMP_NUM_THREADS=4 and type lmp -sf omp -in in.ljM2)

1. First run the program sequentially by typing

   lmp -in in.ljM2

   A log file is created which contains a lot of information concerning the physical quantities obtained as well as the performances of the simulation.

2. Some parameters are set in the code and can be modified by adding a line in the script file

   The timestep can be modified by adding the line (before the last line of the script)

   timestep 0.001

   Rerun the simulation and analyze the evolution of the thermodynamic quantities.

   Increase the timestep to 0.01 and rerun. What do you observe ?

3. For using OPENMPI, it is necessary to run by using **mpirun**.

   By using the script file **in.ljM2**, type

   mpirun -n 4 lmp -in in.system

   Compare the user time by running the program sequentially (by using one core) and by using 4 cores with OPENMP by using 2 and 4 cores with MPI. How to run the program for the best efficiency.

2 Lennard-Jones system : NVT ensemble

1. Comment the line which set the NVE ensemble and add the line
   fix 1 all nvt temp 300.0 2000.0 200.0 What are the differences with the simulation in nve ensemble ?

3 LJ system : Radial distribution function

1. Add two lines to the lammps script

```
compute  myRDF all rdf 100
fix 2 all ave/time 1000 1 $t c_myRDF[*] file tmp.rdf mode vector
```

   Run the simulation again. At the end, a new file is created tmp.rdf. In order to display the radial function, you must extract the relevant quantities from the file tmp.rdf
   You can use this python file for plotting the figure by using the data in the file tmp.rdf.

```
import numpy as np
import matplotlib.pyplot as plt

fi=np.loadtxt("tmp.rdf",skiprows=106)
plt.plot(fi[:,1],fi[:,2],'r-')
plt.xlabel("r",fontsize=15)
plt.ylabel("g(r)",fontsize=15)
plt.xlim(0.5,2.5)
plt.tick_params(labelsize=15)
plt.tight_layout()
plt.show()
```

2. Modify the reduced density defined in the line lattice, run the simulation again. How does the radial distribution function evolve when the density decreases ?

4 LJ system : A simulation movie

1. In compiled lammps version provided by the distribution, the movie creation is not available , but it is possible to write a sequence of jpg files wich can be combined by using the opencv library. The last step can done by using a python script named here movie.py

```
import numpy as np
import cv2
import matplotlib.pyplot as plt
toto=np.arange(0,10000,200)
files=[]
```

```
for tt in toto:
  files.append("image."+str(tt)+".jpg")

img_array = []
for fi in files:
  img=plt.imread(fi)
  img_array.append(img)

height,width,layers=img_array[0].shape

out = cv2.VideoWriter('simu.mjpg',cv2.VideoWriter_fourcc(*'MJPG'),10,(height
    ↪ ,width))

for i in range(len(img_array)):
out.write(img_array[i])
out.release()
```

In order to create the jpg files, add the line in the lammps script

dump 2 all image 200 image.*.jpg type type center d 0.5 0.5 0.5

Rerun the simulation, run the python script and watch the movie by using vlc (or any video reader)

2. Add the parameter **zoom 1.2** in the previous line and create a new movie. What are the changes with the previous movie ?
3. Increase the number of images during the simulation and modify the python file, accordingly. Rerun the simulation and the python script.

A How to install Lammps

1. Lammps is available in repository of many Linux distributions. For instance, for Ubuntu, you can obtain lammps for a repository by typing

   sudo apt install lammps lammps-data lammps-doc lammps-examples

   For Windows, go to https://lammps.sandia.gov/doc/Manual.html and/or http://packages.lammps.org/windows.html. If you need to run by using MPI library, you need to install the Mpich package (see the instructions)

   For MaxOs, go to https://lammps.sandia.gov/doc/Install_mac.html and follow the instructions.
2. If you need the latest stable version, you can download on the web site https ://-lammps.sandia.gov/

   This team developing this software is very active. The last stable version of Lammps dated 22 October 2020.