Lab 4: Molecular Dynamics

Due date: 4/18/2019

Problem 1. Integrator accuracy (40 pts)

Our first exercise will focus on timestep selection as well as the stability and accuracy of MD integration methods. In class, we discussed Verlet algorithms and briefly mentioned some predictor-corrector algorithms. In this exercise, we will examine the accuracy and stability of the velocity Verlet integrator as they apply to an EAM model of Al. Use the LAMMPS setup and documentation from Lab 1 on how to set up the structure and refer to the samples provided and notes below.

A. Plot the time-averaged total energy as a function of timestep size (0.001 - .02 ps) using the default velocity Verlet integration algorithm in LAMMPS. Do these calculations with the NVE ensemble starting with a temperature of 300K, and a 3x3x3 supercell. You should also plot the energy as a function of time to decide how to best take the average. Comment on your results with respect to which time step is appropriate for the most accurate and most computationally efficient MD simulation task.

B. For the timestep of 0.001 ps plot the instantaneous temperature as a function of time. Can you explain the trend that you see, using other available thermodynamic quantities, and suggest a strategy to improve the behavior? You can try setting different initial temperatures to gain more intuition.

C. Repeat part D for bigger system sizes with the same integration and timestep. Explain why fluctuations of the temperature change with the system size. Derive a mathematical relation between the standard deviation of the temperature and the system size (i.e. number of atoms). Verify this with your data.

Problem 2. Melting temperature of bulk Al (50 pts)

In this problem we will find the melting temperature of bulk Al using MD with the same EAM potential. Use the NVT ensemble with a Nose-Hoover thermostat in LAMMPS. Make sure to record all of your simulation parameters: time step, potential, cell size, equilibration time, sampling time, ensemble, and so on.

A. First, you will need to determine a suitable timestep and supercell size necessary to simulate the phase transition. Can you use the timestep and supercell size determined in Problem 1? It's often more efficient to look for the regime of interest and investigate trends using less accurate but faster calculations.

B. How can you determine if the material is melted? Think of properties you could measure from MD: energy, structural correlation, and atom kinetics. Show results of at least 3 methods that can be readily implemented using the capabilities of LAMMPS. How do your results compare with experiments? Would this method overestimate or underestimate the melting temperature?

C. What other ways of calculating melting can you think of? You should come with as many quantitative ways as possible.

Problem 3. Scalability of MD Systems (10 pts)

Consider a bulk material simulation using periodic boundary conditions and large supercells.

A. For an fcc material, how many atoms are contained in a supercell consisting of NxNxN unit cells?

B. How many force calculations are required for each timestep if all atoms are considered? How many force calculations are required if a potential cutoff is implemented? How many times must you calculate the distance between atoms?

C. Assume you have a computer that can perform 35 x 10^{12} floating point operations per second. If each distance calculation uses 1 floating point operation and each force calculation also uses 1 floating point operation, what is the maximum number of particles such that a single timestep calculation would take 1 second?

D. Based on your answer to part E, it should be apparent that billion atom calculations utilize several tricks in order to gain a very significant speed advantage. For simple forms of empirical potentials, much of the computation time is spent calculating the distance between atoms. Can you think of any ways of reducing the amount of time spent calculating distances?

E. Rank the following force calculations in terms of the CPU time required: Hartree-Fock methods, Lennard-Jones pair potentials, EAM pair functional potentials, and Density Functional Theory. Explain your choices.

Notes

To set up the environment, follow the same steps as in previous labs. Pull the lab files using `git pull` in the ~/Software/labutil directory and copy the Lab 4 files to your working directory.

Let us examine at the LAMMPS input file provided as a sample for this lab. For more detailed information please read the official code documentation.

First, we define the initial parameters of the simulation and provide the path for the structure data file, that we generate using ASE, as before

```
# ---------- Initialize simulation ---------------------
units metal
atom_style atomic
dimension  3
boundary   p p p
read_data /home/<...>/lammps.data
```

Similar to Lab 1, we tell the code what interatomic potential we are using and where it is

```
pair_style eam/alloy
pair_coeff * * /home/<...>/Al_zhou.eam.alloy  Al
```

Set up initial velocities for all atoms, according to Gaussian random distribution, which corresponds to Maxwell-Boltzmann statistics. The first integer is the temperature (here 300K), and the second is just a seed for the random number generator.

```
velocity  all create 300 87287 dist gaussian
```

Specify thermodynamic properties to be output:
pe = potential energy, ke = kinetic energy, etotal = pe + ke
temp = temperature, press = pressure, density = density

```
# ---------- Describe computed properties ------------
compute msdall all msd
thermo_style custom step pe ke etotal temp press density c_msdall[4]
thermo 100
```

We also ask the code to compute mean square displacement (MSD), averaged over all atoms and print the value as a function of time along with the other thermodynamic properties. The third line asks to report instantaneous thermo values every 100 steps.

Next we specify which thermodynamic ensemble is used.
fixid = 1, atoms = all, ensemble = nve or nvt

In the case of the NVT (canonical) ensemble LAMMPS uses the Nose-Hoover thermostat (comment out the line for NVE).
The following parameters are needed:
initial temperature = 300K, final temperature = 300K, thermostat controller gain = 0.05 (units of time in ps, bigger is more gentle control, recommended ~ 50-100 timesteps)

```
    # ---------- Specify ensemble  --------------------
    #fix  1 all nve
    fix  1 all nvt temp 300 300 0.05
```

Calculate the radial distribution function on a grid of 100 distance points. A separate file 'lammps.rdf' will be written in the run path directory. RDF is computed on the entire structure at each timestep and then averaged over frames of 250 timesteps. In the Python script you will notice that it generates 4 frames, thus giving you 4 RDF curves for each simulation. To make your life simpler, a parser is implemented for in the updated LAMMPS Python plugin. You can change the scripts and plot these as you like.

```
    # --------- Compute RDF ---------------
    compute rdfall all rdf 100 1 1
    fix 2 all ave/time 1 250 250 c_rdfall[*] file /home/<...>/lammps.rdf
mode vector
```

Finally, define the simulation time step (in units of ps) and specify how many MD steps the code should take.

```
    # --------- Run ------------
    timestep 0.001
    run 1000
```