06-640: Molecular Simulations

Homework 3

Due Date: Thursday 2/22

This aim of this homework is for you to run a Molecular Dynamics code. We will use the Case Study codes provided with our textbook, so before doing the homework, you must work through the steps outlined below to install the codes and learn how to run them. Commands that you need to type and screen output are shown in `Courier font`.

Installation of the Case Study Source Codes

1. Log in to your andrew account on one of the unix.andrew.cmu.edu machines. While in your home directory, execute the following commands

```
cp /afs/andrew.cmu.edu/course/06/640/www/Casestudies.tar.gz .
gunzip Casestudies.tar.gz
tar xvf Casestudies.tar
rm Casestudies.tar
```

These commands copy a complete set of the source codes for Frenkel & Smit's case studies into your account under a newly created directory called `Frenkel_Smit`. Have a look at the directory structure that has been set up by moving into that directory,

```
cd Frenkel_Smit
```

and listing the contents of the directories by typing `ls` .

2. Change directories into the subdirectory for Case Study 4 (the first one we will use) and list the contents of the directory:

```
cd CaseStudy_4
ls
```

The structure of this directory is typical for all the case studies. The subdirectory `Source` contains the source codes and a script for compiling them. The subdirectory `Run` contains a script for running the executable code and various input and output files for the code.

3. As an example of how to compile the case study codes, we will compile Case Study 4, a code that performs MD simulations of a Lennard Jones system. First, move into the directory containing the source codes and clean up the existing object files:

```
cd ~/Frenkel_Smit/CaseStudy_4/Source
rm *.o
```

The latter step is necessary because the codes come precompiled for a specific Linux operating system and we need to make sure the executable is compiled for the specific operating system we are using. To compile the code, simply type

```
make
```

This will sequentially compile all the necessary codes and link them together. It may also generate a few warnings from the FORTRAN compiler. In the end, you should get a very brief message telling you the compilation was complete (typically it will say `done`).

4. To successfully run Case Study 4, we also need to compile the codes used for block averaging of the data. To do this, execute the following series of commands:

```
cd ../../Appendix
rm *.o
make
```

Introduction to Case Study 4

Frenkel & Smit's case studies are not set up in a particularly user friendly format. Before we can learn about MD simulations by running some simulations, we need to learn about the input and output files and how to run the code.

1. First we will simply run the code in the form that was provided. Move into the appropriate directory:
   ```
   cd ~/Frenkel_Smit/CaseStudy_4/Run
   ```
   and then type `./run` . This command executes a script that runs the main code and moves around some of the output files. A line of output should appear on your screen that contains information about how long it took to run the code, such as
   ```
   1.6u 0.0s 0:03 54% 0+0k 0+1io 0pf+ 0w
   ```
   The main useful information from all of this is the first number, how many total seconds of CPU time were used (1.6 s in this case), and the third number, the total real time elapsed while the code ran (3 s in this case, since the computer was also busy with other things).

2. The run you performed above has created two output files, `out` and `lj.gr`. The first is a general purpose output file, the second is a data file containing the radial distribution function, g(r), computed by the simulation. Note that if you run the code again, the older version of `lj.gr` is replaced by a new one, while the new output file is appended to the end of the older one.

3. For future reference, it is a good idea to make copies of the original input and output files so you can always recreate the state of the original distribution. To do this:
   ```
   cp out out.original
   cp run run.original
   ```

4. Read through the supporting documentation that has been prepared to describe the input script, `run`. This information will allow you to change the input conditions for the code and run the MD simulation for various conditions. This documentation is available at www.andrew.cmu.edu/course/06-640/CaseStudy_4.input.pdf

5. Read through the supporting documentation that has been prepared to describe the output file, `out`. This information will allow you to interpret the results of your simulations. This documentation is available at available at www.andrew.cmu.edu/course/06-640/CaseStudy_4.output.pdf

Exercises for Case Study 4

The objective of this exercise is to familiarize you with some of the practical details of running MD simulations. It would be a good idea to start by rereading the description of the case study in the text, p. 98-100. You must hand in a report describing your results.

1. Using the same temperature, density, time step and system size as the original case study, plot the radial distribution function, g(r), for the following run lengths: (a) t = 0.5 with no equilibration, (b) t = 2.5 with no equilibration, (c) t = 2.5 after 0.5 time units of equilibration, (d) t = 10 after 0.5 time units of equilibration. Explain your results. Specifically, list the differences between the four results, and give physical reasons for these differences. Also briefly explain the physical significance of the result that you believe is the most accurate. Note that the code is written so that the velocities are rescaled every 20 time steps during the equilibration period.
2. Again using the same temperature, density, time step and system size as the original case study, run the simulation for (a) t = 5.0 with no equilibration and (b) t = 5.0 after equilibrating for 1.0 unit of time. In each case note the average temperature and potential energy for the run and examine the run's energy conservation. Describe and explain your results.
3. The initial velocities in the simulation are assigned randomly using a random number generator. The sequence of random numbers that is generated by the code is defined by the random number seed (defined in the input file). If you run the code twice with the same seed, you will get identical results. Run the code 5 times using the conditions from part 2(b) and different random number seeds. Note the reported T, P, and U from each run. Are the variations in these values from run to run consistent with the error estimates provided by the code?
4. If an MD algorithm integrates Newton's equations perfectly, the total energy of the system should be exactly conserved. Run the code for the same conditions as part 2(b) using time steps of 0.001, 0.005, 0.01, 0.02, and 0.025. For each run, record the change in total energy between time 1.2 and time 6.0 as listed in the output file. Note that the total energy is not conserved between time 0.6 and 1.2 because the velocities are rescaled every 20 time steps during the equilibration period. Explain the results you observe.
5. For a system containing a single chemical species in a single phase, the equation of state is an expression relating the pressure, temperature, and density. One well known example is the ideal gas law, PV = nRT. Compute a partial equation of state for the Lennard-Jones system by computing P at ρ = 0.5, 0.6, 0.7, 0.8, and 0.9 at T = 1.2. You must choose an appropriate way to run the Case Study code to obtain these results. You should report how you ran the code (i.e., run length, equilibration, time step etc.) and why you believe these conditions give accurate results. For the purposes of this exercise, any run resulting in an average temperature between 1.19 and 1.21 can be considered to have an acceptable temperature. Compare your results with the predictions of the ideal gas law for methane, which has Lennard-Jones parameters ε = 148 K and σ = 0.373 nm. Is the ideal gas law a reasonable approximation under the conditions you have simulated?