CHM 579 LAB 2A: EQUATION OF STATE OF A LENNARD-JONES FLUID

The goal of this lab is to investigate the equation of state of Lennard-Jones fluids by using a Monte Carlo program. This is the first realistic computer simulations we run in this class. Thus, we need to learn to recognize and respect important details: periodic boundary conditions, truncation of potential, tail corrections, and the equilibration phase of the simulation.

Lab Procedure:

The program `mc-016-ru.f90` that you should download from the class web page (or from the `/scratch/carter/g/gchopra/class/CHM579/lab2a` directory on Scholar) is a complete Monte Carlo simulation code for a Lennard-Jones fluid. It is a slightly more developed version of the program `mc-015.f90` we considered in Lab 1. A sample input (`input`) is also provided on the website.

Step 1

Copy `lab2a/` to your `CHM579/` folder located in your home directory. Take a look at the `mc-016-ru.f90` code. It is suggested that you use either a text editor, such as `vi` or `nano`, or the `more` command. When you review a long file using `more` you may need to press the *space bar* several times to go down to the end of the file. If you want to quit the review mode, press *Control + C*. This will immediately cancel the execution of any command (not just `more`).

The `mc-016-ru.f90` program has been written using reduced units and it calculates the energy and pressure of the system and their correction due to the tail cutoff. Compile and run the program on the cluster.

To compile: `gfortran mc-016-ru.f90 -o mc-016-ru.out`

To run the code and stream the output to a file: `./mc-016-ru.out | tee T2.0_d0.4_mc-run.log` *(Be patient!)*

Step 2

Note that since the program has an `input` file, you *do not need to recompile it* each time you change the initial conditions. After modifying the `input` file, you just have to perform the run step. The program will take the information you provided in the `input` file and use it to setup the calculation. Now you will see how it works.

IMPORTANT: Make sure that the `input` files you use do not have any extensions such as `.txt`, `.doc`, etc.

Review the `input` file using the `more` command. You will see a list of numbers. The `input` file – the list of numbers – has the following structure. Things to the right of the (`!`) sign are comments:

`rcut` ! Cut-off radius.
`np` ! Number of particles.
`density` ! Density of the system. The number of particles and density will determine volume and dimensions of the (cubic) box.
`temp` ! Temperature.
`istart` ! `0` or `1`.
`maxd` ! Maximum displacement (Monte Carlo step size).
`mceq mcsteps` ! `mceq` is the number of steps to equilibrate the system. `mcsteps` are the number of steps in the production run.
`nsample` ! Frequency of sampling of the energy and pressure.
`nadjust` ! Frequency to adjust the maximum displacement `maxd`.

IMPORTANT: The variable `istart` may have the value `0` to start a simulation with the particles located at random positions, or `1` to start a simulation reading the initial conformation from a `conf.gro` file.

Step 3

When the program is executed it generates four files: `energy.dat`, `pressure.dat`, `confout.gro`, and `output`.

The first two are the potential energy and pressure of the system as a function of the Monte Carlo step. The file `confout.gro` contains the `x`, `y`, `z` coordinates of the particles in the *final* configuration. The `output` file has the format of the `input` file, and includes the *last value* for the `maxd` variable (optimized maximum displacement).
Use the `more` command to open the `output` file. You may take a look at the others too!

Step 4

To restart a simulation from the final conformation of a previous run you need to copy the file `confout.gro` to `conf.gro`. Also, it is convenient to copy `output` to `input`. You may use the `cp` command. In addition, you *must* change the `istart` value from `0` to `1` in the `input`, so that the simulation starts with the particles located where they were at the end of the previous run (`conf.gro` file) and not at random positions. Try `vi` or `nano` to edit the `input` file.

IMPORTANT: The program *will rewrite* `energy.dat`, `pressure.dat`, `confout.gro`, and `output` when it is started again. Make sure to copy these files to a different directory or rename them. You may need them later. You can use the `cp` command for that.

For example, if you want to save the energy file for a simulation that run at a temperature of 2.0 and a density of 0.4:

To copy a file:
```
cp energy.dat T2.0_d0.4_energy.dat
```

IMPORTANT: Keep in mind that there are certain characters that you should not (or cannot) use in file names, including the *blank space*.

If you want to rename a file instead of making a copy, you can use the *move* command, for example:

To rename a file using the `mv` command:
```
mv pressure.dat T2.0_d0.4_pressure.log
```

IMPORTANT: Both the `cp` and `mv` commands *will overwrite files without asking for confirmation*, so be careful when picking the new file name.

At this point you should know everything you need to manipulate and organize your files. Figure out your own way to do it.

Step 5

Later in this lab, you may calculate the average energy and pressure and save only these values instead of the whole files. The program `avg.f90` is included for that purpose. This program will read `energy.dat` and `pressure.dat`, compute and print the averages. However, you must have an equilibrated system to do this averaging correctly (see discussion below).
Review the code, it will help you to understand the following paragraphs.

Compile the program `avg.f90` (it is also available on the website):

To compile the code:
```
gfortran avg.f90 -o avg.out
```

You can run the program to see what it does on the screen. You will realize that the program needs the user to provide the number of lines in the `energy.dat` and `pressure.dat` files. Can you figure it out?

To run the program:
```
./avg.out
```

It may be convenient to write the averages that the program prints on the screen to a log file, which can be reviewed again at any other time.

IMPORTANT: When you run the program streaming to a log file it still needs the user to provide the number of lines in the `energy.dat` and `pressure.dat` files, even though nothing but a blank space with the cursor appears on the screen. After introducing the command, write the number and press *Enter*.

To run the program *on the terminal* and stream the output to a file:
```
./avg.out | tee T2.0_d0.4_average.log
```

IMPORTANT: In addition to the averages, you will need the energy and pressure *corrections* that are printed in the end of each *log file*. They look like this:

```
Energy correction: -214.173240817818
Pressure correction: -0.171572851901320
```

Step 6: YOUR TURN

(A) Equilibration

If you *did not* run the `mc-016-ru.f90` program in **Step 1**, run it now. Remember to stream the output to a log file.

Use your favorite plotting program (i.e. GNUPlot, Origin, Excel) to plot Energy vs. Monte Carlo step and also Pressure vs. Monte Carlo step.

1. Describe in detail what you observe in each plot.

2. What is the step number when the Energy and Pressure become approximately constant (system at equilibrium)? Explain.

Save the `confout.gro`, and `output` files for the following questions. Create a folder and backup each of the generated files there.

Restart the simulation from the previous run (see instructions above).

3. Plot the Energy and Pressure against the Monte Carlo step. What do you see now? Compare and explain.

Restart the simulation again, but now increase temperature to 4.0 and density to 0.8 in the `input` file.

4. Again plot the Energy and Pressure vs. the Monte Carlo step. What is different now? Compare and explain.

5. How long does it take now to equilibrate the system?

6. Make conclusions about how long your equilibration should be, what should be the value of the variable `mceq`. Explain and give some *margin of safety* to this number. It is certainly much better to over-equilibrate the system than under-equilibrate it!

(B) Investigating the Monte Carlo algorithm

Set up the Monte Carlo program to restart from the final position of question **(A) 2.**, as explained in **Step 4**.

We want to generate a random distribution of energies. In the input file, make sure you set the value of `istart` to `1`, reduce the `mcsteps` value from `100000` to `10000`, set `nsample` to `1`, `maxd` to `2.5`, and `nadjust` to `50000`.

Now run the calculation, and plot Energy vs. Monte Carlo step.

Make a list of every step in which the energy increased and find the maximum of this list. Find your own way to do this.

7. What is the probability that such a step would be accepted? Is this reasonable? Explain.

(C) Investigation of equation of state

In this step you are interested in average values of energy and pressure at different initial conditions. However – I cannot emphasize enough – make sure that your *system is well equilibrated* before you do a production run. In other words, make sure `mceq` is large enough.

8. Calculate and plot several isotherms (Pressure vs. Density) for the temperature interval `0.7 ≤ T ≤ 6.0` and the density interval `0.1 ≤ ρ ≤ 1.2`. Each isotherm should consist of at least of 6 points. Include the tail correction for the pressure. You probably want to consider the temperatures both above and below the critical temperature, `T = 1.31`. For example, you could run using `0.70`, `1.15`, `2.00`, `4.00`, and `6.00` for temperature; and `0.1`, `0.3`, `0.5`, `0.7`, `0.8`, and `0.9` for density.

9. Compare your results with the tabulated values by J. K. Johnson, *et al*. (Mol. Phys. v.78, p591, 1993), particularly, with Table 2 in the paper.

10. Discuss observed differences or similarities, compare conditions of these simulations and estimate uncertainty of your numerical results.

11. In the report, make sure you carefully explain what you are simulating with this Monte Carlo program. Hint: check Lab 1.