

# CHM 579 LAB 2B: MOLECULAR DYNAMICS SIMULATION OF WATER IN GROMACS

In this lab you will learn the basics of using the GRONingen MACHine for Chemical Simulations (GROMACS) package. The first system to study is liquid water by using a simple SPC/E model<sup>1</sup>. You will learn how to obtain thermodynamic observables, such as the potential energy, as well as structural and dynamic properties of the system. This is arguably the most important lab in the course. Much of the information needed to complete this lab is available in the GROMACS online manual. Refer to the manual first, before asking technical questions.

First, take some time to familiarize yourself with the GROMACS manual:

<http://manual.gromacs.org/documentation/>

You may also find useful information in some of the GROMACS tutorials:

<http://www.gromacs.org/Documentation/Tutorials>

Scholar's User Guide may also be convenient:

<https://www.rcac.purdue.edu/compute/scholar/guide/>

GROMACS unit system:

Length:	$nm = 10^{-9} m = 10 \text{ \AA}$
Mass:	$u = 1.6605402 \times 10^{-27} \text{ kg}$ (1/12 the mass of a $^{12}\text{C}$ atom)
Time:	$ps = 10^{-12} s = 1000 \text{ fs}$
Charge:	$e = 1.60217733 \times 10^{-19} \text{ C}$ (electron charge)
Temperature:	$K$
Energy:	$\text{kJ mol}^{-1}$
Force:	$\text{kJ mol}^{-1} \text{nm}^{-1}$
Pressure:	$\text{kJ mol}^{-1} \text{nm}^{-3} = 16.6054 \text{ bar}$
Velocity:	$\text{nm ps}^{-1} = 1000 \text{ m/s}$
Dipole Moment:	$e \text{ nm}$
Electric Potential:	$\text{kJ mol}^{-1} e^{-1}$
Electric Field:	$\text{kJ mol}^{-1} \text{nm}^{-1} e^{-1}$

## Lab Procedure:

The first part of this procedure will serve as an introduction to the use of GROMACS. On the second part you will carry out your first Molecular Dynamics simulation using GROMACS!

### Step 1

From the `/scratch/carter/g/gchopra/class/CHM579` directory on Scholar copy `lab2b/` to your `CHM579/` folder located in your home directory.

**To copy a folder and all its contents:** `cp -r /scratch/carter/g/gchopra/class/CHM579/lab2b ~/CHM579/lab2b/`

**IMPORTANT:** Once you copied all the required files, go to the new `~/CHM579/lab2b/` directory using the `cd` command. Remember, you can find out where you are in the file system by using the `pwd` command. Make sure you end up at `~/CHM579/lab2b/`.

**To print the current working directory:** `pwd`

### Step 2:

Before running any GROMACS commands, you need to *load* the GROMACS *module* on the *node*<sup>2</sup> that you are using. By loading a module you will add *paths* to GROMACS into your *environment settings*. In other words, as GROMACS is not a conventional

<sup>1</sup> See: [http://en.wikipedia.org/wiki/Water\\_model](http://en.wikipedia.org/wiki/Water_model)

<sup>2</sup> See: [http://en.wikipedia.org/wiki/Node\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Node_(computer_science))

program that you use every day, you need to tell the supercomputer that you want to run GROMACS, so that it will prepare all the things it needs to do it.<sup>3</sup>

**To load the GROMACS module:** `module load gromacs`

**IMPORTANT:** You can print on the terminal all the programs, or *modules*, that are available in the supercomputer by introducing the `module avail` command. You can verify if you did load the module by using the `module list` command.

### **Step 3:**

Similarly to the Monte Carlo program of Lab 2, GROMACS uses an input file (`.tpr`). However, the input file (`.tpr`) does not exist and has to be generated. In order to prepare GROMACS' input file (`.tpr`) you will need to execute the `grompp script`, which needs three files to run: the topology file (`.top`), molecular structure file (`.gro`), and the parameters file (`.mdp`).

**IMPORTANT:** It is strongly recommended that you go over the details of each of these files on the GROMACS *manual webpage*: [http://manual.gromacs.org/current/online/getting\\_started.html#files](http://manual.gromacs.org/current/online/getting_started.html#files)

For this lab, the topology file (`.top`) and molecular structure file (`.gro`) are provided so you do not need to worry about them for now. Nevertheless, you will need to modify the parameters file (`.mdp`) at various steps of this procedure.

**IMPORTANT:** On the GROMACS *manual webpage* you will find a thorough explanation of the parameters file format and of its general use. Refer to:

<http://manual.gromacs.org/current/online/mdp.html>

[http://manual.gromacs.org/current/online/mdp\\_opt.html](http://manual.gromacs.org/current/online/mdp_opt.html)

Take a look at these files. They are located at the `~/CHM579/lab2b/` directory: `topol.top`, `conf.gro`, and `grompp.mdp`. It is suggested that you use either a text editor, such as `vi` or `nano`, or just the `more` command.

Find explanations of all the options you see in your parameters file (`.mdp`). Understanding these options is the *key* to realize what this lab is all about.

### **Step 4:**

Now that we understand what the simulation is supposed to do, let us prepare the input file by executing the `grompp script`.

**To generate the input file:** `grompp -f grompp.mdp -c conf.gro -p topol.top -o step04.tpr`

After the script is executed several lines of information will come up on the terminal, a *funny quote* should be the last line. Take a look at the notes, they always provide interesting information. More importantly, if the execution was successful, two files will be produced: `step04.tpr` and `mdout.mdp`. The latter contains a summary of parameters of the simulation you prepared.

Do *not* try to open the input file with a standard visualizer or text editor, it is a binary file.

**IMPORTANT:** The structure of the previous command starts with the name of the script, `grompp`, followed by a series of parameters of execution, or options, which are denoted by the hyphen symbol (`-`). These *options* must be specified in case your files have non-default names. For example, the parameter `-f` means that an explicit parameters file (`.mdp`) will be used, `-c` is to indicate the name of the molecular structure file (`.gro`), `-p` is to use a specific topology file (`.top`), and `-o` is to give a name to the *input* file that *is going to be generated*. There are other options that may be useful for you in this lab, check the manual.

### **Step 5:**

Let us take a look at input file (`.tpr`). It is a binary, which means that you will not be able to use neither your text editor nor the `more` command. Instead, you can check what is inside using the `gmxdump` script:

**To check the input file:** `gmxdump -s step04.tpr | more`

**IMPORTANT:** The pipe "`|`" character is usually typed using `<Shift> + <\>`.

<sup>3</sup> See: <https://www.rcac.purdue.edu/compute/scholar/guide/>

## **Step 6:**

In order to run the simulation you just prepared in the input file (. tpr) we will use the mdrun script.

This will submit a GROMACS Molecular Dynamics simulation as a *front-end application*, meaning that we are going to carry out the simulation on the same *front-end node* that is in charge of handling all the instructions we input and projecting the terminal on our screen. This may seem trivial, but think of how many people is doing similar things on the same *front-end node* at the same time. That is why, we will not be allowed to run any program or script as a *front-end application* for more than 30 minutes. We will learn how to run longer GROMACS jobs in a while,<sup>4</sup> for now:

**To run a Molecular Dynamics simulation:** `mdrun -s step04.tpr` (*Be patient!*)

After the completion of the script, you will find several new files.

<code>confout.gro</code>	<i>New molecular structure file:</i>	A text file with the final structural configuration of the system.
<code>ener.edr</code>	<i>Energy file:</i>	It is a binary. Used by the <code>g_energy</code> script.
<code>md.log</code>	<i>Log file:</i>	A log file that includes information about the simulation.
<code>state.cpt</code>	<i>Checkpoint file:</i>	It is a binary. to restart the simulation from the point it was stopped.
<code>traj.xtc</code>	<i>Trajectory file:</i>	It is a binary. Used by many <code>g_*</code> scripts, e.g., <code>g_msd</code> .

When the simulation finishes, the new configuration information –basically coordinates and velocities– is saved to the `confout.gro` file. The default name `confout.gro` will be used unless you specified another name via the `-c` option when executing `mdrun`.

## **Step 7:**

Now we are going to use some of the GROMACS utilities for trajectory analysis. The `g_energy` script analyzes the `ener.edr` file and can provide a lot of useful information.

**To run the `g_energy` script:** `g_energy -f ener.edr`

**IMPORTANT:** When the `g_energy` script is executed a series of options will come up on the terminal, on the prompt type “4” (potential), and press `<Enter>` twice. The results will be printed and saved into an `energy.xvg` file.

Take a look at the new `energy.xvg` file using the text editor or the `more` command. This (. xvg) file has the proper format to be plotted with the plotting program Grace.

**IMPORTANT:** It is recommended that you plot your graphs with this utility. You can find all the information you need about Grace in the following links.

Grace Home Webpage:

<http://plasma-gate.weizmann.ac.il/Grace/>

QtGrace: native Grace for Windows, Linux, and Mac OS. This is a good option if you plan on using your personal computer:

<http://sourceforge.net/projects/qtgrace/files/>

Similarly to GROMACS, you need to *load* the Grace *module* on the *node* that you are using.

**To load the Grace module:** `module load xmgrace`

**WARNING:** The following instructions will only work if you are connected to Scholar through a terminal that supports X11. SecureCRT, Cygwin-X, or Xming are recommended.

**To plot the energy.xvg file using grace:** `xmgrace energy.xvg`

**IMPORTANT:** If you get a message saying:

<sup>4</sup> See: [https://www.rcac.purdue.edu/compute/scholar/guide/#run\\_pbs](https://www.rcac.purdue.edu/compute/scholar/guide/#run_pbs)

Can't open display  
Failed initializing GUI, exiting

It means your terminal does not support X11. You can either ask for help or skip the following instructions and go to Step 8.

Hopefully, Grace's main window will come up and show your plot. This window may respond slowly, so be patient. Maybe the most convenient way to save your plots would be to press the <Print Screen> key and then paste and crop it in an image that you can save or edit with a standard image editor. You could also save it as a Grace project file (`.agr`), if you figure out how to manipulate those files to put them in your report.

### **Step 8:**

In GROMACS, the diffusion coefficient can be obtained from mean-square displacements using the Einstein relationship.<sup>5</sup> The `g_msd` is the tool for that. When executed it takes information from both, the trajectory file and the input:

**To run the `g_msd` script:** `g_msd -f traj.xtc -s step04.tpr`

**IMPORTANT:** Again, a series of options will come up on the terminal. Type either "0" system or "1" water on prompt. The results will be saved in the `msd.xvg` file. This has the same Grace format and you can plot it just like you plotted the `energy.xvg` file.

### **Step 9:**

Using `g_rdf` you can obtain pair correlations functions, or *radial distribution functions* (RDF)<sup>6</sup>, for the different types of atoms present in your system, i.e O...O, H...H, and O...H. In order to use `g_rdf`, you need to create an *index file* with *groups* first:

**To create a new index file with groups:** `make_ndx -f conf.gro`

**IMPORTANT:** When prompted type "a OW", then "a HW\*", and then "q" to exit. The "\*" character represents a *wild card* matching all characters, in this case 1 and 2 (there are two hydrogen atoms in water!). The information is saved in the `index.ndx` file.

After successful execution a new file `index.ndx` will be produced. Explore the information inside with a visualizer or a text editor. Sometimes it is convenient save previous index files. If you have already made an index file and wish to read in previous groups:

**To use an existing index file with groups:** `make_ndx -f conf.gro -n index.ndx`

Now you can use this information to build radial distribution functions:

**To run the `g_msd` script with an existing index file:**<sup>7</sup> `g_rdf -n index.ndx -o RDF_${atoms} -f traj.xtc -s step04.tpr`

**IMPORTANT:** Specify different `RDF_${atoms}` for O...O, O...H, and H...H RDF. On the prompt, pick the groups between which RDF will be calculated. For instance, "4" and "4" will prepare an H...H RDF, which should be named `RDF_HH.xvg`.

Plot the RDF for H...H, O...O, and O...H. Make sure you understand them.

### **Step 10:**

It is possible to extend or continue a simulation that has completed, and even those that have crashed. This is a good technique for the handling of longer simulations reducing the time lost due to crashes of the computer being utilized. It is only possible to continue seamlessly from the point at which the last full coordinates and velocities are available for the system.

<sup>5</sup> See: [http://en.wikipedia.org/wiki/Einstein\\_relation\\_\(kinetic\\_theory\)](http://en.wikipedia.org/wiki/Einstein_relation_(kinetic_theory))

<sup>6</sup> See: [http://en.wikipedia.org/wiki/Radial\\_distribution\\_function](http://en.wikipedia.org/wiki/Radial_distribution_function)

<sup>7</sup> Whenever you see the `${}` notation in the instructions, it means that you have to replace it by inputting characters of your own.

Now you will learn how to extend a simulation, with the exact same conditions and parameters, by restarting it from a previous point in the trajectory.

**IMPORTANT:** It is strongly recommended that you go over the details of this procedure on the GROMACS *manual webpage*: [http://www.gromacs.org/Documentation/How-tos/Extending\\_Simulations](http://www.gromacs.org/Documentation/How-tos/Extending_Simulations)

**WARNING:** When following any of these directions, be careful to not *overwrite* files when executing `mdrun` for the second time. Make sure you have a consistent method to rename, save, and organize backup files.

Let us go over a bit of the theory first. There are several ways to prepare equilibrated configurations consistent with a given temperature. One way is to run two subsequent Molecular Dynamics simulations at the same temperature and to use the first simulation for equilibration purposes only (no data analysis), and the second for data analysis (production run).

In order to determine if the system is equilibrated, check the average temperature (and pressure, if appropriate) using `g_energy` as described above to make sure the system has the desired values.

A simulation that has completed can be extended using the scripts `tpbconv` and `mdrun`, and *checkpoint* files (`.cpt`). First, the number of steps or time has to be changed in the `.tpr` file, then the simulation is continued from the last checkpoint with `mdrun`. This will produce a simulation that will be the *same* as if a continuous run was made.

**To prepare an extended simulation:** `tpbconv -s step04.tpr -extend 200 -o step10.tpr`

Here, `step04.tpr` should be the name of the `.tpr` file of the completed run. `200` should be the time of the new simulation in picoseconds (*ps*) and `step10.tpr` is the new file for the second Molecular Dynamics simulation.

**To run the extended simulation:** `mdrun -s step10.tpr -cpi state.cpt`

**IMPORTANT:** The second simulation will start as a continuation of the first one. Be careful to specify these new files in analysis scripts, using `-f`, `-s`, etc. options.

### **Step 11:**

If you desire to make some changes to your run parameters, such as changing ensembles or temperatures, you will need to start by making a new parameters file (`.mdp`). Then you will need to rerun the `grompp` script.

**IMPORTANT:** the `gen_vel` keyword in the new parameters file (`.mdp`) should be set to `no` in all consecutive calculations as `yes` resets velocities and destroys equilibrated trajectory.

**WARNING:** The following script may *overwrite* your files; make sure you have a backup.

**To create a new parameters file:** `grompp -f ${new}.mdp -c ${old}.tpr -o ${new}.tpr -t ${old}.cpt`

Files named `${old}` correspond to those from the *finished* run and those named `${new}` correspond to those named for the *new* run.

**To execute `mdrun` with the *new input* file (`.tpr`) file:** `mdrun -s ${new}.tpr`

Now you have learned all the elements you need to run your own simulations.

### **Step 12: YOUR TURN**

#### **(A) Temperature-dependent dynamics**

Perform Molecular Dynamics simulations, in the canonical ensemble (NVT), fixing the temperature at 300 K, 280 K, and 260 K. You need to be sure that the simulations start from a conformation consistent with the chosen temperature. The production run must be of at least 200 ps. Use the Nosé-Hoover thermostat to keep the temperature fixed. Calculate the average potential energy, diffusion coefficient, and pair correlation functions for the O...O, H...H, and O...H pairs.

1. Present your results using tables and figures.

2. At 260 K, the temperature is lower than the freezing point for water. Do you observe formation of ice? Explain what happens with the system as you lower the temperature.

3. Compare the value you get for the diffusion coefficient at 300 K with the value reported by Mahoney and Jorgensen (J. Chem. Phys. 114, p 363, 2001).

See also: [http://www1.lsbu.ac.uk/water/water\\_models.html](http://www1.lsbu.ac.uk/water/water_models.html)

4. Are the conditions of your simulation similar to those of Mahoney and Jorgensen?

### (B) Diffusion experiments

Prepare an input file (`grompp.mdp`) to simulate water at 25 °C and 1 atm, in an isothermic-isobaric ensemble (NPT). For that you need to use pressure coupling.

You can do that by adding the following lines to your `grompp.mdp` file:

```
pcoupl           =Parrinello-Rahman
pcoupltype       =isotropic
tau_p            =1.0
compressibility  =4.5e-5
ref_p            =1.0
```

Figure out the meaning of each of those lines. You can use the online manual.

Additionally, you may need to adjust the interactions cut-off of your system. Figure out how to do it.

**IMPORTANT:** The cut-offs of your simulation *must* be shorter than half the length of your simulation box! Why?

5. Calculate the diffusion coefficient

6. Compare with the published value.

7. Why are you using Molecular Dynamics instead of Metropolis Monte Carlo?

For your lab report explain the methodology you followed to perform your simulations. Include all your tables and figures with their corresponding discussion. Answer the questions providing your reasoning. Do not forget your concluding remarks.