Tutorial 2

Introduction to GROMACS – simulation of water

Peter Košovan[*]

May 12, 2011
ICP, University of Stuttgart

In this tutorial we introduce one of the most widely used simulation packages: GROMACS. We first go through the standard demo and standard tutorial provided by the package. Afterwards we use it to simulate different models of water and examine their properties as well as the influence of box size on the obtained results.

1 Introduction

For a short simulation task, one typically writes a short problem-specific code. However, especially Molecular Dynamics simulations are all very much alike in the sense that they integrate the equations of motion and differ mainly in the potentials which are used. Therefore it is effective to have a universal tool which can perform the simulations using various potentials, thermostats, integration algorithms, etc. Developing such a tool is a task far beyond what a single person can do within a reasonable time. Fortunately, a handful of efficient and universal simulation programs are available. For an overview of the most common ones, you can look at wikipedia [1]. A bit different situation is in the world of Monte Carlo, where one needs to implement problem-specific smart moves in order to achieve efficiency. Therefore self-written MC programs are much more common. When using a universal tool, one has to remember that although it is robust and can prevent a lot of mistakes, in the end it is the user who has to do the thinking and has to understand what he is doing and why. The program is just a tool which does the tiresome work.

[*] kosovan@icp.uni-stuttgart.de

In this tutorial, we introduce GROMACS, one of the most widespread simulation programs when it comes to atomistic simulations, especially those of biomolecules. These often involve aqueous environments and GROMACS is particularly efficient here, because it has highly optimized algorithms for simulating water. Common force-fields for water are included in the GROMACS distribution. It is not a single program, but rather a collection of utilities which are used in series to set up the molecules, the interactions, perform the simulation and finally the analysis of the output.

1.1 Obtaining GROMACS

GROMACS is a free software available online at `http://www.gromacs.org`. In addition to the source code, the homepage provides the actual user manual and online reference as well as some tutorials. If you should be working further with GROMACS, it is worthwhile reading the manual. The online reference for the latest release can be found at the url `http://manual.gromacs.org/current` and provides a web-based documentation which is also available as `man` pages in the GROMACS installation. It is important to make sure that you are using the installation of the same version as the documentation.

1.2 GROMACS on CIP pool computers

For the purpose of this tutorial, GROMACS has been installed in the directory `/home/kosovan/gromacs-4.5.4`. To be able to use it, you first have to set environment variables to proper values. This can be done using the command

```
source /home/kosovan/gromacs-4.5.4/bin/GMXRC
```

Once you have done that, all programs of the GROMACS package can be executed by calling their name and also the corresponding `man` pages are available. You can try it out by typing `man mdrun`.

1.3 Steps in running a simulation with GROMACS

In the following, we will suppose a typical simulation scenario for GROMACS: simulating one or several big molecules (e. g.a protein) surrounded by small solvent molecules, typically water.

1. *Obtaining initial structure of your molecule(s).* This usually comes from a PDB file. You can download files containing the known structure of your molecule in the pdb format from the Protein Data Bank [2]. Although we will not encounter such a problem in this tutorial, be aware that sometimes pdb files do not contain all information you need and you may have to fix missing parts.

2. *Conversion of the topology* to GROMACS native format. This can by done using program `pdb2gmx`.

3. *Addition of the solvent*, i. e. filling up the remaining space in the box by solvent molecules using the program `genbox`.

4. *Initial energy minimization* to avoid close contacts which would blow up your system if you start integrating now. This is done using programs `grompp` and `mdrun`.

5. *Equilibration run* to let things settle down which could not be done by simple energy minimization again using `grompp` and `mdrun`.

6. *Production run* using `grompp` and `mdrun`.

7. *Analysis* using a variety of tools such as `g_analyze`, `g_rdf`, `g_dipoles`, ...

Performing a simulation requires this sequence of steps, perhaps slightly modified to fit the particular task. Each of the programs feeds on the output of its predecessor and produces output for the follower. To get a schematic view of the sequence, look at the flow chart in the online reference [3]. All GROMACS utilities take input files with default names and produce output files with default names. If you follow this naming convention then you may call them with fewer arguments. To override it, you have to specify the name of your input file among the arguments.

2 First steps: the *Getting started* tutorial of GROMACS

GROMACS provides a simple introductory tutorial intended for people who have not been using it before. It can be found in the online reference manual at the manual page `http://manual.gromacs.org/current`. We will only go through first two parts of it.

2.1 The Demo

In the online reference, follow the links to *Getting started* and the to *Demo*. Copy the tutorial files to your working directory:

```
cp -r /home/kosovan/gromacs-4.5.4/share/gromacs/tutor/gmxdemo ./my_playground
```

Then enter the directory and execute `demo`. It explains step by step what is being done and executes the programs to perform a short simulation.

Homework 1 (1 point)

1. Explain what GROMACS demo is doing. *Use your own words! Copy-paste from the manual produces anti-points.*

2. Why do the steps have to be performed in the given sequence?

3. List all the programs involved in the demo and briefly say what is the task of each of them.

The demo gives a quick start-up introduction but does not provide all the necessary information. You may need to consult the manual or man pages for more information. We recommend that you open a man page of each of the programs and read at least the description in the first paragraph.

2.2 The Water tutorial

After you have finished the demo, click on *Go to the next step* and proceed to the water tutorial. This is the second and last step of the standard GROMACS tutorial which we will do here. Of course, you are free to continue through the rest and learn more. Copy the water tutorial files to your working directory and give it a name `water_spc` after the force field for water which it uses:

```
cp -r /home/kosovan/gromacs-4.5.4/share/gromacs/tutor/water ./water_spc
```

Follow the instructions in the GROMACS tutorial and go through the water simulation. Inspect the output after each step and make sure that new files were created. The tutorial is outdated, so some things will not work exactly as described:

- If you type `luck`, nothing happens. Try `g_luck` instead.
- Tell `grompp` to ignore one warning using option `-maxwarn 2` or comment-out the line containing the error.
- You do not have to use `xmgrace` to do the plots. Use any software with which you feel comfortable. You may need to comment-out some lines from the output files containing `xmgrace` commands.
- To view the trajectory, you can use either GROMACS-native `ngmx` viewer, or `vmd` which you already know. When using `vmd`, do it in the following steps:
    1. Load the corresponding pdb file by calling `vmd myfile.pdb`
    2. Select the molecule by clicking on it in the `vmd` main window.
    3. Select `Load Data Into molecule` and load the trajectory file `traj.xtc`.
    4. To get a better idea about the shape of the molecules, select the `VdW` drawing style.

Homework 2 (2 points)

1. Explain in your own words what kind of information is contained in files with extensions `.gro`, `.mdp` and `.top`. Check the online reference and the manual.
2. Which line in the file `conf.gro` defines the force field that we want to use?
3. What information does `grompp` need at the input and what kind of output does it produce? What are its default file names?
4. Which program preforms the actual simulation and what kind of input and output does it need and produce?
5. What is a trajectory?
6. What is the index file and groups in GROMACS? Can we run a simulation without an index file?

3 Various models of water: small boxes

Although it consists only of 3 atoms, water is not a simple liquid. Its behaviour is rich in anomalies and desgining force fields which would be able to reproduce them is a science for itself. Several models of water for molecular simulations have been developed which differ in complexity and also in quality of reproducing the experimentally observed propertiess. In this section we force fields for different water models and look at the results they produce. We will be repeating the steps of the basic water tutorial but with different models: SPC/E, TIP3P and TIP4P. They are all standard models and the force fields are included in the GROMACS distribution. We describe in detail the procedure for the SPC/E model.

- To begin with, copy the water tutorial from the system directory and give it a suitable name, for example

  ```
  cp -r /home/kosovan/gromacs-4.5.4/share/gromacs/tutor/water ./water_spce
  ```

  Do *not* copy the directory where you have done the basic water tutorial. It contains some output files with the default names which may cause trouble if something goes wrong.

- Edit the `topol.top` file to replace `spc.tip` with `spce.tip`.
- In case of the TIP4P model, you will need to replace the `conf.gro` file with `tip4p.gro` from `/home/kosovan/gromacs-4.5.4/share/gromacs/top`
- Follow the steps of the basic water tutorial.

3.1 Analysis: dielectric constant and diffusion coefficient

Besides the simulation tools, GROMACS provides a number of analysis routines which feed on its standard output files. Two of them: `g_rdf` and `g_hbond` were introduced in the basic tutorial.

Diffusion coefficient

To compute the mean square displacement (msd) of molecules in the simulation, use the program `g_msd`. If you have output files with the default names, it does not need any additional arguments. Its output is written to file `msd.xvg` and it automatically computes the diffusion coefficient by fitting the expression

$$\lim_{t\to\infty}(\mathrm{MSD}(t)) = 6D\Delta t \tag{1}$$

to the computed msd. You might want to use options `-b` to specify when to begin the sampling (exclude equilibration) and `-noxvgr` to suppress `xmgrace` commands in the output file.

Dielectric constant

To compute the dielectric constant from the simulation, use the program `g_dipoles`. It uses the following formula to obtain it

$$\epsilon = 1 + \frac{1}{3Vk_\mathrm{B}T\epsilon_0}\left(\left\langle \vec{M}^2 \right\rangle - \left\langle \vec{M} \right\rangle^2\right) \tag{2}$$

Similar to other programs, when called without arguments, it uses default input files and assumes a default temperature.

Homework 3 (3 points)

1. Perform simulations of the following water models: SPC/E, TIP3P, TIP4P.

2. Analyze msd from each of the simulations. Plot msd from all simulations in one graph. Why are there steps in the plots and how could one avoid them?

3. Analyze dielectric constant from each of the simulations. Take the box volume, $\left\langle \vec{M}^2 \right\rangle$ and $\left\langle \vec{M} \right\rangle^2$ from the output of `g_dipoles` and use Equation 2 to compute ϵ. Be careful with units.

4. How sensitive are the computed values to the amount of time you exclude with the `-b` option?

5. Put the computed values of D and ϵ for different models in a table. Compare them between different models and with the experimental values for water at 300 K: $D = 2.6 \times 10^{-5}\,\mathrm{cm}^2\,\mathrm{s}^{-1}$ and $\epsilon = 77.6$. In case of ϵ include both your result and that automatically computed by `g_dipoles`.

6. Why does the TIP4P model require new file `tip4p.gro` while in other cases GROMACS is happy with the original `conf.gro`?

7. Why are there so many models of water and what are the major differences between them? Why in protein simulations people often use SPC/E water while people studying properties of water use TIP4P or more complex models?

8. Which model would you expect to give values closest to experiments?

4 Various models of water, big boxes

There are several reasons why the computed values from the previous section were far from the experimental results:

1. The models are not strictly exact.

2. We were using Berendsen thermostat which is not sampling the canonical ensemble.

3. We were using cutoff method for electrostatic interactions which produces artifacts.
4. We were using too small boxes.

To partly eliminate the last three, we will try to perform the simulation in a bigger box and with the Nosé-Hoover thermostat and Particle-Mesh Ewald algorithm for electrostatics.

4.1 Making a larger solvent box

We will utilize the program genbox to do so. Get the files conf.gro and topol.top from the tutorial directory and type the following command:

```
genbox -cs conf.gro -box L -o big_box.gro
```

where L is the box length. This generates a box with randomly placed solvent molecules. Try L = 2,3,5 and then choose one which produces a box with about 1000 molecules. In the file topol.top change the number of molecules from 216 to the new value.

4.2 Energy minimization and the run

Now we need to get rid of close contacts, so the next step is energy minimization. To save time, we have prepared the file em_commands.mdp with the instructions on how to perform the minimization. Just execute

```
grompp -v -f em_commands.mdp -c big_box.gro -o em_input.tpr
```

to produce the binary input file em_input.tpr and then

```
mdrun -v -s em_input.tpr -o em.trr -c em_final_conf.gro -g em.log
```

to perform the minimization. We also prepared for you MD parameter file for the actual run which you may directly use with the output of the energy minimization to perform the simulation

```
grompp -v -f T300.mdp -c em_final_conf.gro -o T300.tpr
mdrun -v -s T300.tpr -o T300.trr -c T300_final_conf.gro -g T300.log
```

4.3 Analysis

To perform the analysis, we will need to modify the index file. Program make_ndx reads the big_box.gro file to do it for us

```
make_ndx -f big_box.gro -o index.ndx
```

After this, we can perform the analysis in the same way as before.

Homework 3 (2 points)

1. Perform big box simulations with SPC/E, and TIP3P models.
   Hint: The simulation keeps your computer busy for a couple of minutes. Use free computers in the pool to start the second run in the meantime:
   - Prepare the input for your simulation.
   - Go to the CIP wiki page [4] and find the name of a computer which is free.
   - Connect to it via `ssh`.
   - Execute the `top` command to see that none of your friends is running GROMACS there.
   - Execute the simulation with the other model.
2. Analyze msd and ϵ in the same way as before and append them to the results of homework 2.
3. Did the agreement with the experiments improve in comparison with the small box runs?
4. Describe how the input file `T300.mdp` differs from the previously used mdp files. In particular, look at variables `dt`, `nsteps`, `Tcoupl`, `Pcoupl` and `coulombtype`. How should they affect the result (agreement with experiment) and the speed of computation? Would the agreement improve with a longer simulation run?

5 Running in parallel

Until several years ago the CPU clock frequency of computers doubled each two years, following the Moore's law. Since then the frequency has not changed much but a similar law applies to the number of CPU cores that the modern machines have. If we want to perform long simulations of many-atom systems, it is important to make them run efficiently on many CPUs in parallel. GROMACS release which we have been using supports multithreading which is one of the parallelization strategies for multicore CPUs. Theoretically parallelization can provide an increase in speed by a factor of N where N is the number of cores. In practice the factor is always smaller because some time is spent on communication between the cores. GROMACS is very efficient in running parallel computations and you have been taking advantage of that all the time. To illustrate the effect, we compiled also a GROMACS version without multithreading support. To see the parallel speedup, we will use the big box setup. Before you do this part, make sure that you have performed the analysis of the big box results or make a backup. The simulation will overwrite the old output files.

Homework 4 (2 points)

1. Edit the `T300.mdp` file to shorten the run to 20 ps. Process the input file and run the simulation with the following commands:

```
grompp -v -f T300.mdp -c em_final_conf.gro -o T300.tpr
mdrun -v -s T300.tpr -o T300.trr -c T300_final_conf.gro -g T300.log
```

2. While the simulation is running, open another shell window and execute the `top` command. How many % of CPU time is used by `mdrun`?

3. Inspect the end of the log file `T300.log` to see how much time GROMACS needed to run. How much time was spent waiting due to load imbalance?

4. Repeat the same simulation without multithreading. Open a new shell window and go to your big box directory. Source the startup script for the GROMACS binaries without multithreading

   ```
   source /home/kosovan/gromacs-4.5.4_no_threads/bin/GMXRC
   ```

   and follow the same procedure as before.

5. How many cores does your computer have? How many cores are there on machines `palisander` and `uranus`? Execute `cat /proc/cpuinfo` to find out.

6. Compute the speedup factor due to multithreading. Is the factor close to the number of cores?

7. Repeat the same with the basic water tutorial. Do the speedup and load imbalance differ from the big box system?

Optional homework (2 points)

1. Try setting some of the variables `dt`, `nsteps`, `Tcoupl`, `Pcoupl` and `coulombtype` in the big box system to the same values as we had in the small box case.

2. Perform the big box simulation with modified parameters.

3. How do different parameter combinations affect the results on msd and ϵ?

References

[1] `http://en.wikipedia.org/wiki/List_of_software_for_molecular_mechanics_modeling`

[2] `http://en.wikipedia.org/wiki/Protein_Data_Bank`).

[3] `http://manual.gromacs.org/current/online/flow.html`

[4] `http://wiki.cip.ica.uni-stuttgart.de/mediawiki/index.php/Raumplan`