Tutorial 2

Introduction to GROMACS simulation of water

F. Dommert* and P. Košovan†

May 2, 2012
ICP, University of Stuttgart

In this tutorial the widely used simulation package GROMACS is introduced. At first, a standard procedure to setup a simulation is described. Afterwards different models of water are examined in terms of thermodynamic properties, the diffusion, and the static dielectric constant. Finally, the influence of finite-size effects is investigated.

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 an 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.

*dommert@icp.uni-stuttgart.de
†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/dommert/local/gmx4.6
```

To be able to use it, you first have to set environment variables to proper values, which is achieved with the command

```
source /home/dommert/local/gmx4.6/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`. However, be aware, that if you open a new shell or terminal, the information is not present anymore. Therefore, you can also consider, putting this line in the file `$HOME/.bashrc`, that it is always executed, if a new shell or terminal is opened.

Finally a compressed file, that includes the files which are required for this tutorial, has been prepared. Before you start, extract this file into a suitable location, e. g.your `$HOME` directory. **Pay attention, that no directory with name `water` is present in this directory, because otherwise its contents may be overwritten.**:

```
cd $HOME
tar xvzf /home/dommert/water.tgz
```

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`. Since we will deal only with solvent in this tutorial, step 2 and 3 are switched inhere, because we have to obtain a GROMACS topology file for a generated water box.

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_msd`, ...

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]. Most of the flags of programs, that come with the GROMACS package are capable of BASH completion. This means that if a flag is written to the command line, either pressing the Tab key will fill in the corresponding file type, in case only one match is possible, or pressing the Tab key twice shows the possibilities, if more than one is present.

2 Getting started with GROMACS

At first we want to introduce you to GROMACS with a simple example. We will guide you through the procedure to setup and analyse a simulation. In doing so, the main ideas and methods of GROMACS should become clear. However, questions will remain open, surely. Therefore, we refer you to another essential part of every software, its documentation. A reference manual for GROMACS is given at the manual page `http://manual.gromacs.org/current` and answers to many questions regarding parameters, methods and workflows, can be found there.

2.1 Generating a starting configuration

In order to obtain an initial configuration for our simulations, we want to prepare a box with 250 water molecules for two different water models, SPC/E and TIP4P. In order to do so, we will need the commands `genbox` and `pdb2gmx`. In the directory `water`, which has been created with the extraction of `water.tgz`, two subdirectories `spce250` and `tip4p250` contain solvent files with 500 SPC/E and TIP4P water molecules, respectively. The files are in the `gro` format, that includes coordinates and velocities of the atoms.

Homework 1 (2 points)

1. Which units are used for the coordinates and velocities in a `gro` file? Which standard units are used in GROMACS for time, coordinates, velocities, energies, and forces ?

2. Use the command `genbox -h` to learn more about the options of `genbox`. Create two boxes with 250 SPC/E and TIP4P water molecules each. *Hint*: The last line in a `gro` file contains the box vectors. Choose an appropriate size for the `-box` flag. If the box is too small the molecules will not fit in, but a too large box requires a long equilibration time.

3. Use the command `pdb2gmx -h` to see, which flags are necessary to create a topology file `top`. Generate corresponding topology files for the SPC/E and TIP4P boxes with 250 molecules. *Hint*: The choice of the force field does not really matter, only the choice of the water model. **However to assure, that everybody uses exactly the same parameters, use OPLS-AA as force field.**

4. Explain, why a topology file is required and which important information is contained. What is the most significant difference between the two water models ?

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 Simulating a system

After generating the initial configuration, almost everything is ready for the first simulations. However, as described in 1.3, three main steps are required: energy minimization, equilibration, and finally, the production. In the `spce` and `tip4p` subdirectories, the MD parameter files, `em.mdp`, `equi.mdp`, and `T300.mdp` are given. Each of them has been prepared for a certain task and the names already give much information about which. Now, a set of files is present, that is the minimal input for `grompp`, the GROMACS. This program checks the consistency of the input and prepares a `tpr` input file, which is required for `mdrun`, the actual simulation tool.

Homework 2 (2 points)

1. Compare the different `mdp` files, where are the differences ? Why are different `.mdp` files used for the equilibration and production, apart from the simulation time ? *Hint*: The command `diff` is very helpful here!

2. Perform an energy minimization of the two systems which you have created in Homework 1. In order to generate a `tpr` file with `grompp`, use the files `em.mdp`, the topology file, and the box configuration file. Afterwards use `mdrun` to perform the simulation. *Hint*: Use an intuitive name for the `tpr` file, such as `em.tpr` and run the simulation with the command

   ```
   mdrun -deffnm em -v -pd
   ```

   This will name all output files `em` with the corresponding suffix. The flag `-v` sets a higher verbose level, which means more output and information for the user, e. g. remaining runtime. The flag `-pd` is **required** for the energy minimization, but not necessary for the equilibration and production runs.

3. Use `equi.mdp` and the resulting coordinates of the energy minimization to equilibrate the systems.

4. Run production simulations with the parameters given by `T300.mdp`. The simulation will keep your computer busy for a while. Therefore, continue with the tutorial and start the simulation of the other water model, when the first one has finished.

2.3 Analyzing a system

After the simulations are finished, we have to analyse them. A big advantage of atomistic MD is the accessibility of the atom coordinates, which allows to calculate a variety of properties. For the most common task, GROMACS provides corresponding tools, usually named with a prefix `g_`. The next task is to inspect our simulations. Therefore a script `analyse.sh` has been prepared, which is located in the directory `water`. It is able to perform different analysis tasks, depending how the script is executed. Let your BASH know, where this executable can be found. If the extracted directory is in your `$HOME`, this is the appropriate command:

```
export PATH=$HOME/water:$PATH
```

Variables, which are required for the script `analyse.sh`, **have to be provided before the actual command**, e. g.:

```
BEGIN=0 TPR=T300.tpr analyse.sh
```

Always be careful with the variable BEGIN. Choose it properly, that some time for equilibration is considered, also in the production run. The parameter above is only an example how to set a variable!

Homework 3 (2 points)

1. Open the script `analyse.sh` with your editor of choice and find out, what happens, if the script is executed with `DEFD` set to:

   ```
   DEFD=‘pwd‘
   ```

   Which variables are necessary, that `g_energy` is used to calculate different thermodynamic properties ? Explain what is happening in this case, step-by-step. Perform the corresponding analysis for both water models. *Hint*: A nice overview for BASH scripting is given in [4]. Perhaps the section **Tests** provides enough information to understand the whole script.

2. Now you have calculated the densities, written to the file `thermo.log` in the directory, which you set before. Compare the densities, given by the different models. How large are the deviations from the experimental value of $996.6\,\mathrm{kgm}^{-3}$?

3. To calculate the mean-square-displacement, an index file, which is an important concept for GROMACS, is required. It is generated with `make_ndx`, which is an interactive tool. Create an index file for the water models with the following commands:

   ```
   make_ndx -f em.gro
   splitres 2
   del 0-2
   q
   ```

   Explain the four steps of the procedure. **You will need such an index file for every analysis of the MSD, therefore become familiar with the procedure.** Why are different index files required for SPCE and TIP4P water ? With the created index files, use `analysis.sh` to obtain the MSD for both water models. *Hint*: Check again, which variables are necessary to enable the calculation of the MSD.

4. For the processing of the MSD, a further script `procMSD.py` is provided in the main directory. Use it, to generate an averaged MSD for all molecules. Plot the MSDs for both water models on a **double-logarithmic** scale in a **single** graph. Describe the common behaviour of the models and the difference. *Hint*: `procMSD.py -h` shows which flags are possible. For the averaging only an input and an output file name is necessary.

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 designing 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 properties. In this section we study the already introduced force fields SPC/E and TIP4P for water in more detail. In this regard, the diffusion and the static dielectric constant are very interesting properties and can be calculated comparably easily. However, the size of the box as well as the simulation time has to be appropriate. To shorten the waiting time, a system with 500 SPC/E and another with 500 TIP4P water molecules has been prepared and the input and output files of the simulation, which are necessary for the analysis are present in the subfolders `spce500` and `tip4p500`. **No equilibration time for the analysis has to be considered, because the `xtc` trajectory and the `edr` energy file have already been truncated.**

3.1 Analysis: dielectric constant and diffusion coefficient

Diffusion coefficient

You already computed the mean square displacement (MSD) of the molecules, with the GROMACS tool `g_msd`, which is executed within `analyse.sh`. Its output is written to the file `msd.xvg`, which contains the MSD of the single molecules in each column. The tool `procMSD.py` allows to average the MSD and finally 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 have to set the options `-b` to specify where to begin the fit and `-e` to define the end in order to fit this expression in the linear regime of the MSD.**

Dielectric constant

To compute the static dielectric constant ϵ from a simulation, the dipole moment M of the systems has to be extracted and ϵ is obained by:

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

For this task, the part of the script `analyse.sh` has to be enabled, which uses `g_current`. Apart from the standard input parameters, such as input files and directories, the variable `NSHIFT` is **important**, which determines, how much samples are left out for averaging, in order to access only statistically independent snapshots.

Homework 4 (2 points)

1. Calculate the thermodynamic properties, the MSD, and the dielectric constant for the systems with 500 SPC/E and 500 TIP4P water molecules. Choose a value of NSHIFT, such that every 10 ps a sample is used for the average. *Hint*: The tool `gmxcheck` gives you information, how often the coordinates are saved. You have to do this just once, because the `xtc` files for 500 and 5000 water molecules contain the same number of frames. Furthermore, `analyse.sh` is capable to perform several task after each other, if the corresponding variables are set properly. The analysis will keep your computer busy for some minutes, hence do not wait, but continue with the tutorial.

2. Average the MSD and fit the results to 1 with the script `procMSD.py`. **Take care, where you begin and end the fit and explain why you chose this values. Think about the results of the simulations with 250 water molecules.** Compare the results to the experimental measurement of $D = 2.6 \times 10^{-5}\,\mathrm{cm^2\,s^{-1}}$ and put them in a table.

3. In the output directory for the analysis of ϵ, the file `current.jlog` contains the result important to us here. **Pay attention, two values for ϵ are given, but only one of them corresponds to 2.** Compare ϵ for the different water models to the experimental value of $\epsilon = 77.6$ and put the them in the table with the diffusion constants.

4 Various models of water: big boxes

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

1. The models are not strictly exact.

2. We were using only 500 molecules, which perhaps introduces artefacts of the periodic boundary conditions (PBC).

While tackling problem 1 by improving the model would be a very time consuming and complicated task, we can try to get rid of possible PBC artefacts by using a larger number of molecules. Corresponding simulations were performed for 5000 water molecules with the SPC/E and the TIP4P models and the required files for the analysis are present in the subdirectories `spce5000` and `tip4p5000`. **Also in this case the files are filtered, such that no equilibration time has to considered for the analysis.**

Homework 5 (2 points)

1. Analyse the thermodynamic properties of the simulations with `analyse.sh`. Compare pressure and density for the **two models** and **system sizes of 500 and 5000 molecules** in a table. Where do you observe finite-size effects?

2. Calculate the diffusion coefficients with `procMSD.py` from the given MSDs and analyse ϵ with `analyse.sh`. Append the results to the table with D and ϵ of the small boxes. Did the agreement with the experiments improve in comparison with the small box runs?

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. During the tutorial we have been using 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 see the parallel speedup, we will use the big box setup and perform timing runs with a single and two CPUs. The `mdp` files, required for this benchmark, are given in subdirectories `benchSPCE` and `benchTIP4P` as well as `benchSPCEs` and `benchTIP4Ps`.

Optional homework (2 points)

1. In order to perform timings for the SPC/E and TIP4P models, generate a run input file with `grompp` and skip the first 100 steps for the timing with `mdrun`. This is achieved, if the flag `-resetstep` is set, e. g.:

   ```
   mdrun -deffnm benchSPCE -resetstep 100 -v
   ```

   In the last lines of the corresponding `log` file, produced by `mdrun`, the simulation time, which can be achieved in 24 h, is given. Compare the results for SPC/E and TIP4P. Why are the first 100 steps skipped? What might be the reason for the different performance?

2. Repeat the same timings without multithreading. Therefore add the flag `-nt 1` to `mdrun`, e. g.:

   ```
   mdrun -s benchSPCE.tpr -deffnm benchSPCE_nt1 -resetstep 100 -nt 1 -v
   ```

   Compute the speedup factor due to multithreading for the simulations with SPC/E and TIP4P. As speedup σ, we use the ratio of the possible simulation time per day between a run with a single and two threads. Hence, if $\sigma = 2$, the work is optimally balanced between the threads. Is the factor close to the number of cores?

3. Repeat the same with the results of the water boxes with 250 molecules. For this reason, copy the topology file and `gro` file, resulting from your production simulations of SPCE and TIP4P to `benchSPCEs` and `benchTIP4Ps`, respectively. Perform step 1 and 2 with this systems. Compare σ for the systems with 250 and 5000 water molecules. What might be the reason for the different speed up ?

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://tldp.org/LDP/abs/html/`