Tutorial 4

ESPResSo 2: Charge distribution around a charged rod

Olaf Lenz* Mehmet Süzen

June 10, 2011
Institute for Computational Physics, Stuttgart University

Contents

*olenz@icp.uni-stuttgart.de

1 Introduction

This tutorial is based on an article by Deserno et al. [1]. Throughout the tutorial, you will try to reproduce some plots from the article. As further reading, you can refer to [2], which is probably a bit more comprehensive.

Task:
Read the article [1].

The system under consideration is a so-called *cell model* of a polyelectrolyte, i.e. a polymer that dissociates charges in solution (see in the lecture). In the cell model, a polyelectrolyte is modelled as a single, charged, infinite rod with its counterions and maybe some additional salt that is confined to a cylindrical cell. The observable of interest is the distribution of ions $P(r)$ around the rod.

To obtain the charge distribution, we will introduce two methods that can be used to tackle the problem. The first method is the Poisson-Boltzmann theory, an analytical mean-field theory, the second method is to carry out computer simulations. We will learn about the strength and weaknesses of both methods.

The cell model is defined by the following parameters:

Bjerrum length l_B The Bjerrum length is the distance at which the Coulomb energy of two elementary charges equals the thermal energy k_BT. Here, it defines the length scale of the simulation: all length are measured in units of the Bjerrum length. In water, the Bjerrum length is 7.1Å.

Line charge density λ The line charge density of the rod is the number of charges per length unit. It is closely coupled to the *Manning parameter* $\xi = \frac{\lambda l_B}{e_0}$.

Rod radius r_0 The radius of the charged rod defines the minimal distance between an ion center and the rod center.

Cell radius R The radius of the cylindrical cell defines the maximal distance between an ion center and the rod center.

Valency of the counterions v The valency of the counter ions.

The default values that we are going to use are:

$$\begin{aligned} l_B &= 1.0 \\ \lambda &= 1.0/2.0 \\ r_0 &= 1.0 \\ R &= 28.2 \\ v &= 1 \end{aligned}$$

2 Analytical solution: Poisson-Boltzmann theory

On the one hand, the cell model for infinite charged rods can be solved within the *nonlinear Poisson-Boltzmann theory* (PB).

Figure 1 on the following page shows a plot of the analytical solution of the Poisson-Boltzmann equation for the default parameters. Note that the x-axis of the plot of the distribution $P(r)$ is in logarithmic scale to stress the structure close to the rod. To create such a plot, you can use.

For the salt-free case, an analytical solution of the charge distribution $P(r)$ exists. It is given by equations (8) and (9) in the article. The equation contains two free parameters γ and R_M, that are defined by the equations (6) and (7) in the article.

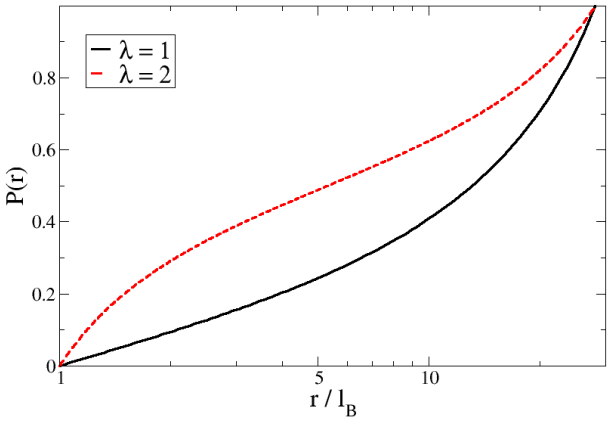

Figure 1: Poisson-Boltzmann solution for the charge distribution $P(r)$ over radius r for the default parameters.

Task: (3 points)
Reproduce the plot in figure 1. To complete this task, you can use whatever program suits you best.
First, solve equations (6) and (7) numerically and obtain values for R_M and γ. Then, substitute the values into equations (8) and (9) to obtain a solution for the distribution. Create a file that contains the solution $P(r)$ over r that can be used for plotting.

3 Computer simulations

Alternatively, the charge distribution can be obtained via computer simulations.

3.1 Mapping the cell model onto an ESPResSo computer simulation

It is not possible to simulate the full cell model, as it requires an infinite rod. However, we can simulate a quasi-infinite system by exploitation of periodic boundary conditions: we create a rod that spans the whole simulation box size and uses periodic boundary conditions in that direction. We will model the rod by a number of fixed charged particles on a line parallel to the z-axis in the center of the simulation box.

Furthermore, we would like to be able to use the fast P3M method for computing the electrostatics. Therefore, our system has to be cubic (i.e. $L = L_x = L_y = L_z$) and it has to employ periodic boundary conditions in all three dimensions.

How can we map the cylindrical cell with a radius R onto a cubic simulation box with a box length L while still retaining the correct charge distribution? The trick is to use the same *ion density* in both systems. When the total ion density is the same in the cell model and in the simulation, we expect them to show the same charge distribution.

Note, that the box length L defines the length of the simulated segment of the rod, and consequently the charge of this segment. Since the whole system should be neutral, this also predefines the number of counterions in the system.

Task: (1 points)
Map the default cell model parameters onto a cubic simulation box, i.e. compute L for the given value of R. How many ions need to be simulated?
In the sources, you will find the ESPResSo*-script* `tutorial4.tcl`*. This scripts sets up a system in the described way, but for a different set of parameters. Study the script and understand how it works. Adapt the script to the given default parameters.*

3.2 First runs

Now the script is prepared for the first simulations. You can run the script via

```
Espresso tutorial4.tcl [CHECKPOINT]
```

If you omit `CHECKPOINT`, the system will be set up randomly, otherwise, the script will start from the given checkpoint. At the end of each run, the script will write such a *checkpoint* `checkpoint_`*time*`.chk`. This can be used to continue a simulation from the point where it left off before.

The `timestep` is the time step, measured in simulation units. It is too expensive to measure observables after each timestep and results would be correlated. Instead, observables are measured only after a fixed number of timesteps `steps_per_frame` has passed (which is referred to as a *simulation frame*). The number of frames that are done in a single simulation run is defined by `max_frames`.

The script will create two measurement files and a checkpoint file:

`rod-energy.dat` contains the total energy and its components (kinetic, coulomb and LJ energies) over the time.

`rod-dist.dat` will contain the average integrated charge distribution $P(r)$ over r. The distribution will only be measured, when the parameter `measure_distribution` is set to 1 in the script.

`rod_xxx.chk` is a checkpoint file that contains all the data required to restart the simulation from the point where the simulation ended. *xxx* is the simulation time that has passed.

Task: (1 points)
Run the script `tutorial4.tcl` *and plot the energies over time. Do you notice anything strange? Look at the different components of the energy. Which component behaves strange? What is the reason of this problem? Modify the script such that the problem disappears.*

3.3 Equilibration and sampling time

Now we can start with the real simulations. First, we need to make sure that the systems is equilibrated, and we need to get an idea how many simulation frames we will need to sample to get good statistics.

To do that, you should monitor the slowest observable that you can find. In general, the energies and the different energy components are a good starting point.

During equilibration, you will notice that the observable has a trend, i.e. it grows or drops. Only after you can not observe any trend anymore, the system can be assumed to be equilibrated. Usually, the values themselves fluctuate very strongly, so that the trend might well hide within the fluctuations. Therefore, it is useful to create *running averages* of the observables over a few hundered frames that average out the fluctuations. `xmgrace` can create running averages (choose `Data - Transformations - Running Averages...` from the menu).

Now you need to find out for how many frames you need to sample your simulation. To get useful statistics, the sample should encompass at least several of the slowest fluctuations.

Task: (2 points)
Run the simulation. Monitor the energies. Increase the number of frames done in the script (`max_frames`) and rerun the simulation from the last checkpoint several times if necessary. When you think you do not see a trend anymore, let the simulation run again for at least the number of steps performed so far as a safety margin.
Now analyze the fluctuations. How many timesteps do the slowest fluctuations span? If you have a number, multiply it by 2 *and you have the number of frames that your sample should minimally encompass.*

3.4 Measuring the charge distribution

Finally, the charge distribution can be measured. In the script, you need to set `measure_distribution` to 1 and set the number of frames to be performed to the number of frames that you got out above.

Task: (3 points)
Measure the charge distribution around the rod. Take care that you restart the simulation from an equilibrated checkpoint.
Plot the achieved charge distribution and compare it to the distribution obtained from the both Poisson-Boltzmann solutions. Do they match?

References

[1] Markus Deserno, Christian Holm, and Sylvio May. The fraction of condensed counterions around a charged rod: Comparison of Poisson-Boltzmann theory and computer simulations. *Macromolecules*, 33:199–206, 2000.

[2] M. Deserno. *Counterion condensation for rigid linear polyelectrolytes.* PhD thesis, Universität Mainz, February 2000.