Worksheet 5: Charge Distribution Around a Charged Rod

Kartik Jain, David Sean, Olaf Lenz and Mehmet Süzen

Institute for Computational Physics, University of Stuttgart

Contents

General Remarks

- In this worksheet, you can achieve a maximum of 20 points.
- The report should be written as though it would be read by a fellow student who attends the lecture, but doesn't do the tutorials.
- To hand in your report, send it to your tutor via email.
  - Christoph (`clohrmann@icp.uni-stuttgart.de`)
- Please attach the report to the email. For the report itself, please use the PDF format (we will *not* accept MS Word doc/docx files!). Include graphs and images into the report.
- The report should be 5–10 pages long. We recommend using LaTeX. A good template for a report is available online.
- The worksheets are to be solved in **groups of two or three** people.

1 Introduction

This tutorial is based on an article [1] by Deserno et al., and we will try to reproduce the plots in the article in this tutorial using computations. [2] is a more comprehensive reference for the tasks of this tutorial.

Task (0 points)

- Read the article [1] in detail. It is provided together with this worksheet.

2 Short Questions - Short Answers

Task (3 points)

Answer the following questions:

- Explain the concept of counterion condensation.
- What does the Bjerrum length describe?
- Describe the concept of a mean field theory.

3 The Simulated System

The system under consideration is a so-called *cell model* of a polyelectrolyte, i.e. a polymer that dissociates charges in solution (cf. 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 tackle the problem of obtaining the charge distribution, we will introduce two methods:

1. Poisson-Boltzmann theory, which is an analytical mean-field theory
2. Computer simulations using ESPResSo.

We will learn about the strengths and weaknesses of both methods.
The cell model is defined by the following parameters:

Bjerrum length l_B In water, the Bjerrum length is 7.1 Å under normal conditions.

Line charge density λ The line charge density of the rod is the number of charges per length unit. It is closely related 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 we are going to use are:

$$l_B = 1.0, \quad \lambda \in \{1.0, 2.0\}, \quad r_0 = 1.0, \quad R = 28.2, \quad v = 1$$

Note that you should do the following tasks for both values of λ.

4 Analytical Solution: Poisson-Boltzmann Theory

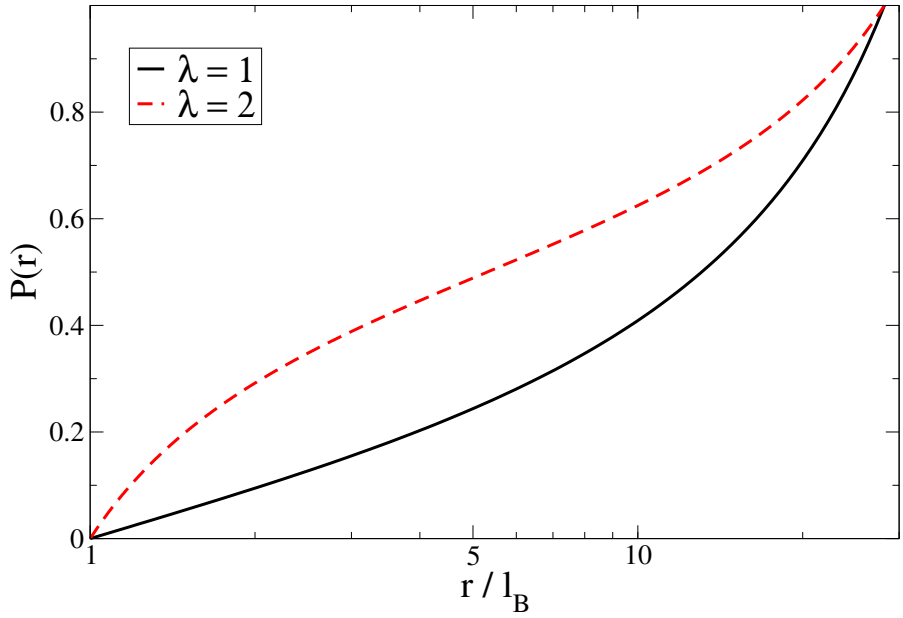

Figure 1: Poisson-Boltzmann solution for the charge distribution $P(r)$ over radius r for the default parameters.

The cell model for infinite charged rods can be solved within the context of the *nonlinear Poisson-Boltzmann theory* (PB).
Figure fig. 1 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.
For the salt-free case, an analytical solution of the charge distribution $P(r)$ exists, and is given by equations (8) and (9) in the article [1]. The equation contains two free parameters γ and R_M, which are defined by the equations (6) and (7).

Task (5 points)

- Reproduce the plot shown in fig. 1. To complete this task, you can use whatever program suits you best
- Explain what you see in the plot and report the time taken by the program.

Hints

- First, solve equations (6) and (7) numerically and obtain values for R_{M} and γ.
- Then, substitute these values into equations (8) and (9) to obtain a solution for the distribution.
- If you use Python, you might want to use `scipy.optimize.fsolve()`.

5 Computer simulations

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

5.1 Mapping the Cell Model onto a 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 i.e. we create a rod that spans the whole simulation box size and use 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. For this worksheet, you will need to enable new features for ESPResSo. It is easiest to follow the installation steps from the previous worksheet, but this time when editing `myconfig.hpp`, enable the following features:

```
ELECTROSTATICS
LENNARD_JONES
```

by removing the comment symbols '`//`'.
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 directions.
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 (2 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?
- Study the script `espresso_charged_rod.py` provided with this worksheet. Adapt the script to the parameters you calculated.

5.2 Warmup Runs

Now that the script is prepared for the first simulations, you can execute the script with the command:

```
$> /<espresso_install_path>/pypresso espresso_charged_rod.py
```

Upon completion, the simulation will produce output files for postprocessing and analysis. It is too expensive to measure observables after each time step 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 three files:

`rod-energy.dat` contains the total coulomb energy over time.

`rod.vsf` will contain the structure header of particles for every time frame (see `vtf,vsf,vcf` format).

`rod.vcf` will contain the positions (x, y, z) of particles for every time frame (see `vtf,vsf,vcf` format).

Task (1 point)

- Run the previously adapted script `espresso_charged_rod.py` and visualize the trajectory with VMD. Do you see ions crowd near the rod?

Hint

- You may combine the `vsf` and `vsf` files to a `vtf` or keep them separate. In which case you will need to specify both to `VMD`:

```
$>vmd rod.vsf rod.vcf
```

- Remember that you have to run the simulations for two different values of λ! Consider automatically naming the output files based on λ, so that you don't overwrite.

5.3 Equilibration and Sampling Time

Now we can start with the real simulations. First, we need to make sure that the systems are 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 aforementioned trend might well hide within the fluctuations. Therefore, it is useful to create *running averages* of the observables over a few frames that average out the fluctuations and smooth out the resulting curve.
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 (4 points)

- Execute the simulation. Monitor the energies. Increase the number of frames in the script (`max_frames`) until the energy settles. 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 for the equilibration phase
- Now analyze the fluctuations. How many frames do the slowest fluctuations span? Make sure that the simulation runs many of those spans to produce good statistics.

Hints

- The program `xmgrace` can create running averages (choose *Data > Transformations > Running Averages...* from the menu).

5.4 Measuring the Charge Distribution

Finally, the charge distribution can be measured. Modify the the script to make sure the system is equilibrated before taking data. Make sure the number of steps executed between writing output is large enough that the frames can be considered uncorrelated.

Task (5 points)

- Measure the charge distribution around the rod for both values of λ.
- Plot the resulting charge distribution and compare it to the distributions obtained from the Poisson-Boltzmann theory. Do they match? If not, try to explain what might be the reason.

Hint

- `vcf` files can be nontrivial to handle when reading in using python. Consider writing your own file that holds just the coordinates of the counterions.
- Be careful while generating the histograms! Use `numpy.logspace` to generate logarithmically spaced bins.
- For the histogram, use the ergodic hypothesis to stack up the frames to generate one large system.

Bibliography

[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. URL `http://archimed.uni-mainz.de/pub/2000/0018`.