

Worksheet 4: Charge Distribution Around a Charged Rod

Solutions

Johannes Zeman

June 16, 2016

Institute for Computational Physics, University of Stuttgart

Contents

1	Introduction	1
2	Short Questions - Short Answers	2
3	The Simulated System	3
4	Analytical Solution: Poisson-Boltzmann Theory	4
5	Computer simulations	7
5.1	Mapping the Cell Model onto a Simulation	7
5.2	First Runs	9
5.3	Equilibration and Sampling Time	10
5.4	Measuring the Charge Distribution	12
	Bibliography	13

Remarks

The Solutions provided here show possible approaches to solve the tasks from the corresponding worksheet and may not be exhaustive.

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

(3 points)

- Read the article [1].
You can access it under `/group/sm/2016/tutorial_04/deserno00a.pdf`.
If you have trouble accessing the article, write an email to your tutor!

Solution

- Well... just read it.

2 Short Questions - Short Answers

Task

(4 points)

Answer the following questions:

- What is counterion condensation?
- What does the Bjerrum length describe?
- Describe the concept of a mean field theory. (2 points)

Solution

- Counterion condensation describes the phenomenon of counterions condensing onto a strongly charged polymer in solution, effectively reducing its line charge density. According to Manning theory, this happens whenever the Bjerrum length l_B in the solvent is larger than the distance of charges l_c within the polymer. In water, where the Bjerrum length is $l_B \approx 7.1 \text{ \AA}$, this implies that counterion condensation is important for many standard polyelectrolytes, where l_c typically lies between 2 and 3 \AA .

- The Bjerrum length is the distance between two elementary charges at which their electrostatic interaction energy equals the thermal energy:

$$\frac{1}{4\pi\epsilon_0} \frac{e^2}{\epsilon_r l_B} = k_B T \quad \Rightarrow \quad l_B = \frac{1}{4\pi\epsilon_0} \frac{e^2}{\epsilon_r k_B T} \quad (\text{SI units})$$

- The concept of a mean field theory is to reduce the complexity of a statistical many-body problem by describing the interactions of single components (*e.g.* particles) with the rest of the system as an interaction with a mean field. This effectively reduces the many-body problem to a one-body problem. In statistical physics, this means to decouple an N -particle system's Hamiltonian into a sum of N one-body Hamiltonians with an effective mean field.

Note that usually, the application of a mean field theory means that effects arising from correlations between individual components or non-uniformity of the system will not be represented properly.

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 mobile 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 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 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 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

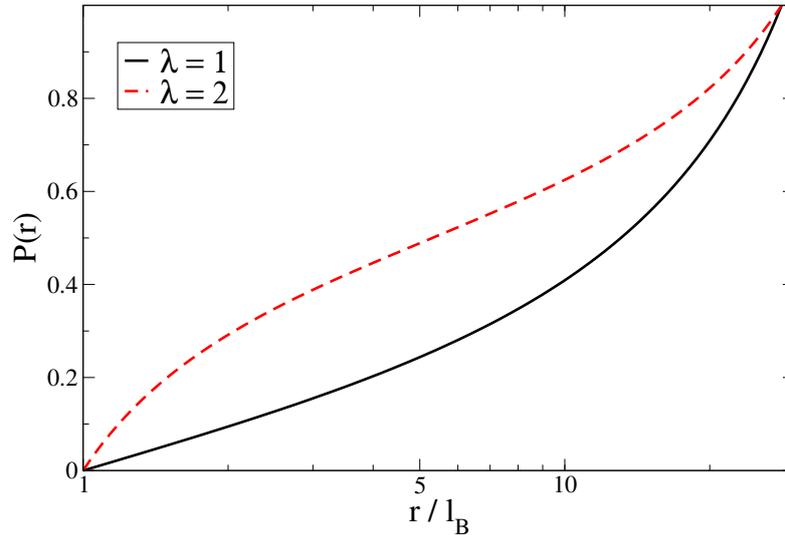


Figure 1: Poisson-Boltzmann solution for the charge distribution $P(r)$ over radius r for the default parameters.

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

Figure 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. It is given by equations (8) and (9) in the article. The equation contains two free parameters γ and R_M , which are defined by the equations (6) and (7) in the article.

<p>Task</p> <ul style="list-style-type: none"> • Reproduce the plot in figure 1. To complete this task, you can use whatever program suits you best. 	(3 points)
--	------------

Hints

- 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.
 - If you use Mathematica, the function `FindRoot[]` might be useful. In Python, look for `scipy.optimize.fsolve()`.
-

Solution

The following example Python code computes the Poisson-Boltzmann solution for the given parameters:

```
from numpy import *
from matplotlib.pyplot import *
from scipy.optimize import fsolve

l_B = 1.0 # Bjerrum length
r_0 = 1.0 # Rod radius
R = 28.2 # Cell radius
v = 1.0 # Counterion valency
e = 1.0 # Elementary charge

def eq1(gamma):
    '''This is eq. (6) minus eq. (7) from the Deserno paper.'''
    xi = lam * l_B / e # Manning parameter
    return gamma * log(r_0 / R) - arctan((1.0 - xi) / gamma) \
        + arctan(1.0 / gamma) # == 0

def manning_radius(gamma):
    '''This is eq. (7) from the Deserno paper solved
    for the Manning radius R_M.'''
    return R * exp(-1.0 / gamma * arctan(1.0 / gamma))

def P(r):
    '''This is eq. (8) plugged into eq. (9) from the
    Deserno paper.'''
    xi = lam * l_B / e
    return 1.0 - 1.0 / xi * (1.0 - gamma \
        * tan(gamma * log(r / R_M)))

lam = 1.0 # Line charge density
gamma = fsolve(eq1, 1.0)
R_M = manning_radius(gamma)
r = logspace(log10(r_0), log10(R), 100)
P1 = P(r)
```

```

lam = 2.0 # Line charge density
gamma = fsolve(eq1, 1.0)
R_M = manning_radius(gamma)
P2 = P(r)

figure()
subplot(1, 1, 1)
semilogx(r, P1, '-', label="\lambda=1")
semilogx(r, P2, '-', label="\lambda=2")
xlim(1.0, 30.0)
ylim(0.0, 1.0)
xlabel("$r$")
ylabel("$P(r)$")
legend(loc="best")
show()

```

The resulting plot looks as follows:

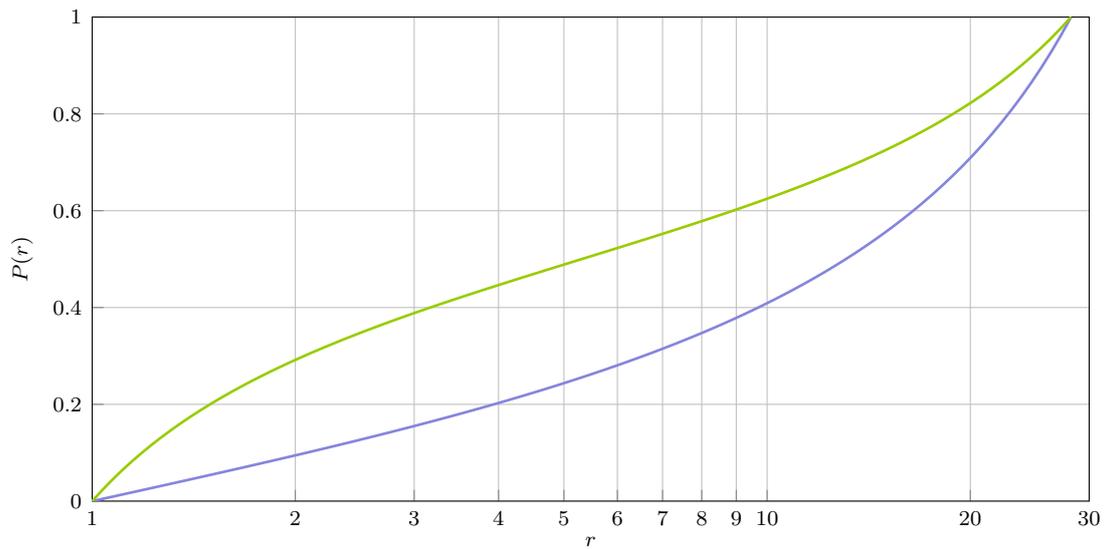


Figure 2: Normalized charge distribution P with respect to the distance r from the center of a charged rod with radius $r_0 = 1$ for different line charge densities $\lambda = 1$ (—) and $\lambda = 2$ (—)

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: 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.

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

(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?
- Download the ESPResSo template Tcl script `template.tcl` from the lecture's website. This script 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.

Solution

- In order to obtain a neutral system, the total charge of the counterions $|q_{ci}|$ must equal the charge of the rod $|q_{rod}|$. With rod length L and line charge density λ , the absolute value of the rod's total charge q_{rod} is

$$|q_{rod}| = \lambda L \equiv |q_{ci}| .$$

The volume V_{cyl} of a cylindrical cell with radius R around a rod with radius r_0 accessible to counterions is

$$V_{cyl} = \pi L (R^2 - r_0^2) ,$$

and thus, the charge density in the accessible volume is

$$\rho_{cyl} = \frac{\lambda}{\pi (R^2 - r_0^2)} .$$

For a cubic box, the corresponding Volume is

$$V_{cub} = L (L^2 - \pi r_0^2) ,$$

yielding a charge density of

$$\rho_{cub} = \frac{\lambda}{(L^2 - \pi r_0^2)} .$$

Requiring $\rho_{cub} = \rho_{cyl}$ finally leads to

$$L = \sqrt{\pi} R .$$

For $R = 28.2$ this yields $L \approx 50$.

The number of counterions is simply charge per valency:

$$n_{ci} = \frac{q_{ci}}{v} = \frac{\lambda L}{v}$$

With $L = 50$, this means that one has to simulate $n_{ci} = 50$ ($n_{ci} = 100$) counterions for $\lambda = 1$ ($\lambda = 2$).

- The only line in the template Tcl script requiring adjustments is **`set L 25.0`** \rightarrow **`set L 50.0`**

5.2 First Runs

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

```
$> /<espresso_install_path>/Espresso template.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 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 containing all the data required to restart the simulation from the point where the simulation ended. *xxx* is the elapsed simulation time.

Task	(2 points)
<ul style="list-style-type: none">• Run the previously adapted script <code>template.tcl</code> and plot the energies over time. Do you notice anything strange?• Take a 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.	

Hint

- Remember that you have to run the simulations for two different values of λ ! Consider running the simulations in two different subdirectories, so that they do not overwrite each others' files.

Solution

- The energies obtained from the first runs are plotted below:

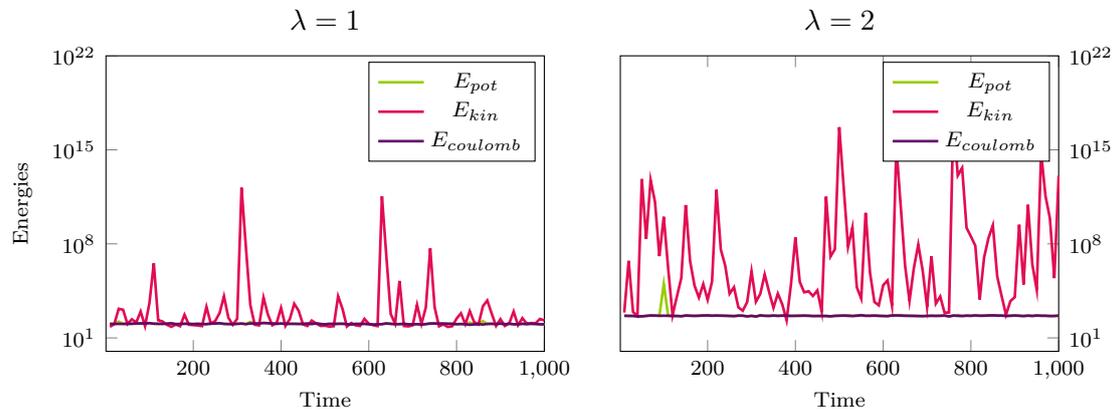


Figure 3: System energies obtained from first simulation runs. For both values of the line charge density λ , the kinetic energy fluctuates strongly.

- For both values of the line charge density, the kinetic energy fluctuates strongly. This is due to the time step of 0.1, which is too large. In simulations with a too large time step, particles can come very close to each other, causing the kinetic energy to become excessively large. The only reason the system doesn't crash is the frequent coupling of the thermostat. In order to obtain a well-behaved simulation, the time step should be reduced to 0.01.
- Script modification:
`set timestep 0.1` \rightarrow `set timestep 0.01`

5.3 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 hundred frames that average out the fluctuations and zoom the resulting curve. The program `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)
<ul style="list-style-type: none"> • Run the simulation. Monitor the energies. Increase the number of frames done in the script (<code>max_frames</code>) 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 10 and you have the number of frames that your sample should minimally encompass. 	

Solution

- With a time step of 0.01, the energies look as follows:

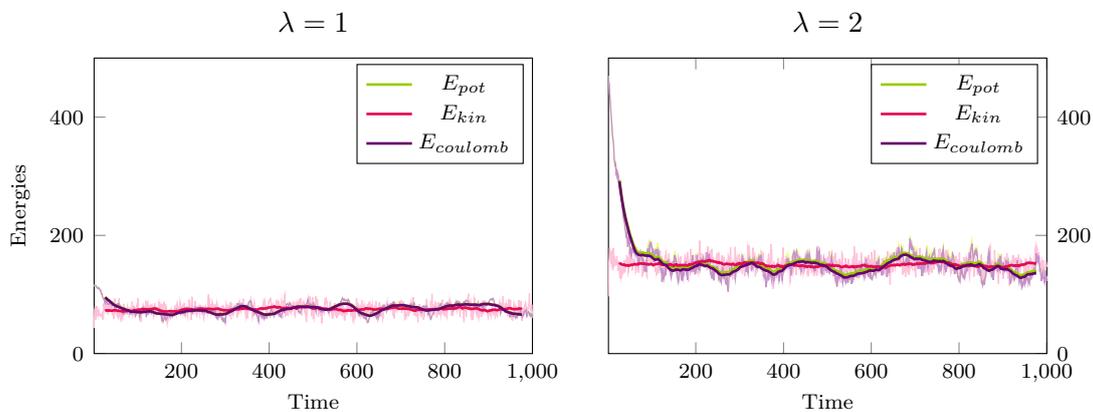


Figure 4: System energies (pale colors) obtained from simulation runs with reduced time step (0.01) with running averages (bin size 50, intense colors). All energies are now well-behaved. The system seems to be equilibrated after $t \approx 200$ and the slowest fluctuations span roughly 200 frames.

- The slowest fluctuations span roughly 200 frames, so a measurement should span at least 2000 frames.

5.4 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

(4 points)

- Measure the charge distribution around the rod. Make sure that you restart the simulation from an equilibrated checkpoint!
- Plot the resulting charge distribution and compare it to the distribution obtained from the both Poisson-Boltzmann solutions. Do they match? If not, try to explain what might be the reason.

Solution

- The measurements were run for 100000 frames. The resulting charge distributions are plotted in figure 5. For comparison, the analytical solutions obtained from Poisson-Boltzmann theory are also shown.

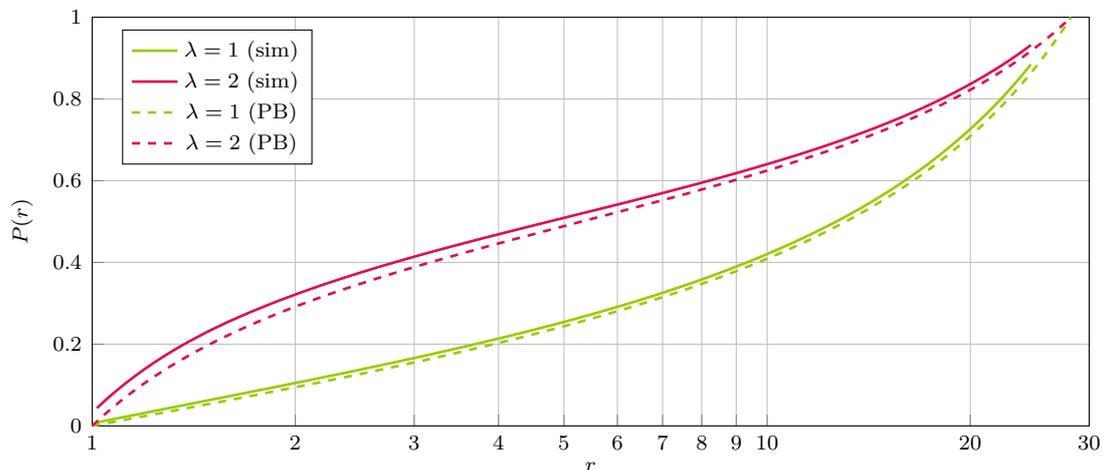


Figure 5: Normalized charge distribution P with respect to the distance r from the center of a charged rod with radius $r_0 = 1$ for different line charge densities $\lambda = 1$ (—) and $\lambda = 2$ (—). Solid lines: Simulation. Dashed lines: Poisson-Boltzmann theory.

- The simulation deviates from the analytical solution in the way that the simulated curves always lie slightly above the analytical ones. Since in the analysis of the simulation data the distribution is obtained by integrating over the distance r and the shift of the curves stays almost constant, the deviation is likely to be originated in the region close to the rod. Due to the fact that the particles in the simulation are “soft” Lennard-Jones particles, the probability of occurrence at $r = r_0$ is non-zero. This offset then causes a shift in the probability distribution. It is also plausible that for the higher line charge density the shift must be larger because of the stronger Coulomb attraction.

The curves obtained from the simulation don't start exactly at r_0 since the analysis relies on binning, and they end at $r = \frac{L}{2} = 25$, since the charge distribution can only be analyzed up to this threshold due to the cubic geometry.

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.