

Computer Assignment 1: Depletion Forces

Supervisor: Anna Bille
(+46 46 22 23492, anna.bille@thep.lu.se)
Deadline: October 15, 2015

This document and the java-program can be downloaded from
<http://cbbp.thep.lu.se/~anna/>

1 Introduction

Imagine a system consisting of N small balls of radius r and one large ball of radius R in a cavity of some sort, as illustrated in figure 1. A ball can not overlap with another ball or the boundary. Thus, each ball, and also the boundary, is surrounded by a shell of thickness r , in which the centers of the small balls cannot be. This shell is called a *depletion zone*. When the surface of the large ball comes within a distance $D < 2r$ from the boundary the two depletion zones will overlap which increases the volume available to the small balls, increasing their entropy. There will thus be an effective force driving the large ball towards the boundary. Analogously there will also be an effective attraction between any two large objects in the system. The better the surface-geometries of these objects match, the larger the possible depletion-zone overlap and the larger the attraction.

In a biological context the small balls could be small molecules like sugars and the large balls proteins or other macromolecules. Depletion forces are necessary in some biological processes to help macromolecules find each other [1]. In particular, depletion forces drive bodies of matching geometry together. It has been argued that this effect can also induce curvature in lipid membranes enhancing for example insertion of proteins into membranes [2].

We will study the system described above by two different approaches. First by finding the free energy of the system given that the small balls can be treated as an ideal gas. Second we will perform computer simulations of the system. Measuring free energy from simulations is however not always trivial, since free energy is an integrated quantity. We will therefore start by introducing the concept of *potential of mean force*, which we can measure in the simulations. We will then show that the potential of mean force is equivalent to the free energy. By using this equivalence we can compare our theoretical result with the simulations.

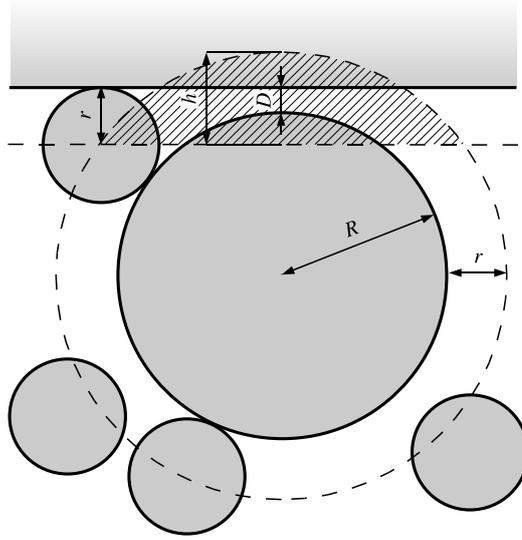


Figure 1: A system of large and small balls. Depletion layers are indicated by dashed lines. The hatched area marks an intersection of the overlap volume V_{ov} .

2 Potential of Mean Force

Consider a system with N particles and an interaction potential U . The probability density for a given microstate in such a system can be written (with particles at positions $\mathbf{r}_1, \dots, \mathbf{r}_N$)

$$\rho_N(\mathbf{r}_1, \dots, \mathbf{r}_N) = \frac{e^{-\beta U(\mathbf{r}_1, \dots, \mathbf{r}_N)}}{Q}, \quad (1)$$

where $\beta = 1/k_B T$ and Q is the partition function

$$Q = \int d\mathbf{r}_1 \dots d\mathbf{r}_N e^{-\beta U(\mathbf{r}_1, \dots, \mathbf{r}_N)}. \quad (2)$$

The marginal probability density for having one particle at a certain position can be obtained by integrating out all the other degrees of freedom

$$\rho(\mathbf{r}_1) = \frac{1}{Q} \int d\mathbf{r}_2 \dots d\mathbf{r}_N e^{-\beta U(\mathbf{r}_1, \dots, \mathbf{r}_N)}, \quad (3)$$

The average force on a particle, at a position \mathbf{r}_1 , is the negative average gradient of the potential

$$\mathbf{f}(\mathbf{r}_1) \equiv -\langle \nabla_1 U \rangle = -\frac{\int d\mathbf{r}_2 \dots d\mathbf{r}_N (\nabla_1 U) e^{-\beta U}}{\int d\mathbf{r}_2 \dots d\mathbf{r}_N e^{-\beta U}}. \quad (4)$$

Exercise 1 (5%): Show that

$$\mathbf{f}(\mathbf{r}_1) = k_B T \nabla_1 \ln \rho(\mathbf{r}_1). \quad (5)$$

Equation 5 states that the function

$$w(\mathbf{r}_1) = -k_B T \ln \rho(\mathbf{r}_1) \quad (6)$$

acts as an average potential on our particle. If we replace all the other particles in our system by the potential $w(\mathbf{r}_1)$, the probability density function $\rho(\mathbf{r}_1)$ will be the same for the remaining particle¹. This potential is usually called the *potential of mean force* (PMF), or the reversible work. Furthermore, the energy difference

$$\Delta w = w(\mathbf{r}_1^{(f)}) - w(\mathbf{r}_1^{(i)}) \quad (7)$$

describes the work needed to translate the considered particle from $\mathbf{r}_1^{(i)}$ to $\mathbf{r}_1^{(f)}$ with all the other particles in equilibrium during the process. If we consider \mathbf{r}_1 an external parameter, the two states can be treated as two different macrostates. In this interpretation Δw is equivalent to the free energy difference ΔF between two given macrostates. From here on we will treat free energy and PMF equivalently. More generally we can define a PMF for any parameter, λ , describing a macrostate, as

$$w(\lambda) = -k_B T \ln \rho(\lambda). \quad (8)$$

Spherically symmetric systems

Suppose the probability density $\rho(\mathbf{r})$ is spherically symmetric, $\rho(\mathbf{r}) = g(r)$. The probability density as a function of r , $\rho(r)$, is obtained by integrating over a shell of width dr at distance r from the origin,

$$\rho(r)dr = dr \int g(r)r^2 \sin \theta d\theta d\varphi = 4\pi r^2 g(r)dr. \quad (9)$$

Correspondingly, the PMF can be studied as a function of r . This PMF is given by

$$w(r) = -k_B T \ln \rho(r) = -k_B T \ln(4\pi r^2 g(r)) = -k_B T \ln g(r) - k_B T \ln 4\pi r^2. \quad (10)$$

3 Free energy in an ideal gas approximation

Now return to the system introduced in section 1. For a given microstate, $\rho(\mathbf{r}_0, \mathbf{r}_1, \dots, \mathbf{r}_N)$ is either 0 or $1/Q$, depending on if there is an overlap or not. Given that the volume fraction of the small balls is small, such a system is described well as an ideal gas. Given a position, \mathbf{r}_0 , of the big ball, the free energy of the system is then

$$F = -Nk_B T \ln V' \quad (+ \text{ terms independent of volume}), \quad (11)$$

¹Because in such a system $\rho \propto \exp(-w(\mathbf{r}_1)/k_B T)$, which is basically an inversion of (6).

where V' is the volume available to the small balls, *i. e.*

$$V' = V - V_{\text{big}} + V_{\text{ov}}. \quad (12)$$

V is the available volume of the cavity, $(4\pi(R_S - r)^3)$, V_{big} the volume of the big ball (including its depletion layer), $4\pi(R + r)^3/3$, and V_{ov} the volume of the overlap between the depletion zones of the big ball and the cavity. Here we will mainly study a spherical cavity of radius R_S , which gives $V = 4\pi(R_S - r)^3/3$.

We want to determine how the free energy depends on the position of the big ball. Without loss of generality, we may assume that the ball is constrained to move in one dimension. The free energy is then a function of the distance r_0 between the center of the ball and the center of the cavity, or equivalently the distance D between the ball and the sphere ($D = R_S - R - r_0$). Denote this free energy $F(D)$. The free energy for an unconstrained big ball, moving freely in the cavity, is $F(D) - k_B T \ln 4\pi r_0^2$ (compare equation 10).

To calculate $F(D)$ it is convenient to first determine the entropic force

$$f(D) = -\frac{\partial F}{\partial D} = \frac{Nk_B T}{V'(D)} \cdot \frac{\partial V_{\text{ov}}}{\partial D}. \quad (13)$$

If one assumes $V'(D) \approx V$

$$f(D) \approx \frac{N}{V} k_B T \cdot \frac{\partial V_{\text{ov}}}{\partial D} = c k_B T \cdot \frac{\partial V_{\text{ov}}}{\partial D}, \quad (14)$$

where $c = N/V$ is the concentration of small balls.

Flat boundary

The simplest case we can study is a system where the boundary is a flat wall (as in figure 1). The derivative of V_{ov} with respect to D is equal to the negative of the area where the depletion zones meet (the base area of the spherical cap which V_{ov} measures, see figure 2)

$$\frac{\partial V_{\text{ov}}}{\partial D} = -A(D). \quad (15)$$

Exercise 2 (5%): Show that for the flat wall system, the depletion force

$$f(D) = -ck_B T \pi [4rR + 2(r - R)D - D^2] \quad (16)$$

when $0 \leq D \leq 2r$ and 0 otherwise

Exercise 3 (5%): Using the previous result and the convention that $F(2r) = 0$, show that

$$F(D) = -ck_B T \pi (2r - D) \left[2r \left(R + \frac{r}{3} \right) - D \left(R - \frac{r}{3} \right) - \frac{D^2}{3} \right], \quad (17)$$

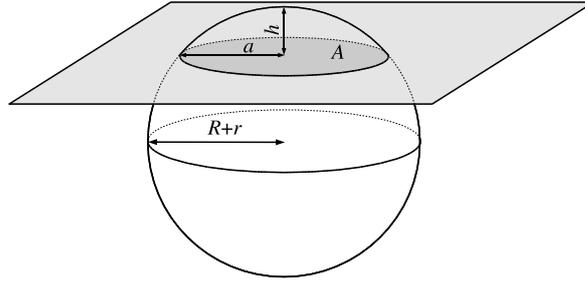


Figure 2: 3D view of the overlap volume V_{ov} . The base of the spherical cap of height h has radius a and area A .

by calculating

$$F(2r) - F(D) = - \int_D^{2r} f(D') dD'. \quad (18)$$

Spherical Cavity

We will do computer simulations of a somewhat more complicated system where the boundary is a large spherical cavity of radius R_S . The depletion force and free energy can be calculated in the same way as for the flat wall-system (see figure 3 for a picture of the geometry).

The surface area of a spherical cap is $A = 2\pi hr$, where h is the height of the cap (as in figure 3) and r is the radius of the sphere.

Exercise 4 (5%): For the spherical boundary, find the full expression for $A(D) = 2\pi h(D)(R_S - r)$. To determine $h(D)$ one can use the fact that X (see figure 3) is the cathetus of two different triangles. Then show (by using (14) and (15))

$$f(D) = -ck_B T \pi \frac{(2r - D)(2R + D)(R_S - r)}{R_S - R - D}. \quad (19)$$

Exercise 5 (5%): Integrate the result in (19) (as in Exercise 3) to obtain

$$F(D) = -ck_B T \pi (R_S - r) \left[(2r - D) \left(R_S + R - r + \frac{D}{2} \right) + (2r - R_S + R)(R_S + R) \ln \frac{R_S - R - D}{R_S - R - 2r} \right]. \quad (20)$$

Hint: do a polynomial long division of $f(D)$ before integrating.

Exercise 6 (5% bonus): Show that in the limit $R_S \rightarrow \infty$ the result in (20) is the same as in (17). You will need to do a Taylor expansion of the logarithm in (20) to the third order in $1/R_S$. *This exercise is optional, but gives bonus points.*

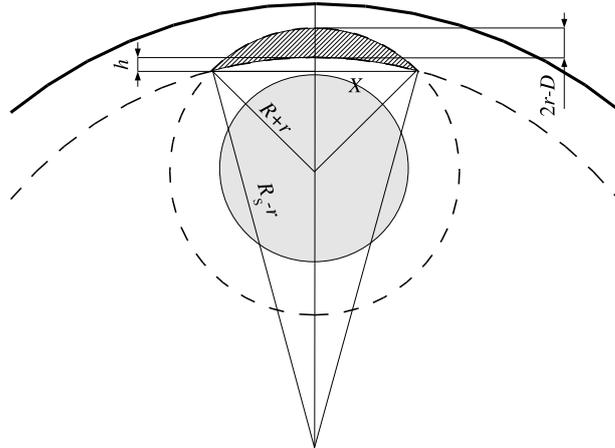


Figure 3: Overlap between the big ball and a bounding sphere. The overlap volume is hatched.

4 Monte Carlo Simulations

A simple way to study a system of hard balls is by Monte Carlo (MC) simulations. Monte Carlo is a collective name for methods that use random numbers to study different physical and mathematical systems.

The simplest method we could use would be to generate a number of totally random configurations. This would give equal weight to all allowed microstates, which would allow us to measure the probability for different macrostates. Here, we will employ another, also relatively simple (and hopefully more efficient), protocol:

1. The system will be initialized randomly, *i. e.* the balls will be placed randomly within the cavity in such a manner that they don't overlap with each other or the boundary (this can be difficult if the density of balls is high).
2. A ball i will be chosen randomly and moved in a random direction, $\mathbf{r}_i \rightarrow \mathbf{r}_i + \mathbf{X}$, where \mathbf{X} is chosen randomly from a small cube $x, y, z \in [-L/2, L/2]$. Here, L is chosen to be 3 times the radius of the small balls.
3. If the ball does not overlap with any other ball or the boundary, in the new position, the update is accepted. If there is an overlap, the update will be rejected and the old state will be counted twice.
4. Return to 2.

It can be shown that this protocol ensures that the correct probability density, $\rho(\mathbf{r}_0, \dots, \mathbf{r}_N)$, is sampled. Hence a histogram, $H(\lambda)$ (of any quantity λ), measured from the simulations, will be an estimate of the corresponding probability density $\rho(\lambda)$.

In our simulations the distance, r_0 , from the center of the big ball to the origin, will be measured and collected in a histogram $H(r_0)$. This histogram can be used to check the free energy estimate $F(D)$ obtained above under the ideal gas assumption. If this assumption holds, $H(r_0)$ and $F(D)$ will be related by

$$-k_{\text{B}}T \ln H(r_0) = F(D) - k_{\text{B}}T \ln 4\pi r_0^2, \quad 0 \leq D \leq 2r_0. \quad (21)$$

Running the program

The program you will use is a java-program that can be downloaded from the web. There will be supervised sessions in the computer rooms at the astronomy department, but if you prefer working at home the program should run on Windows and Mac machines as well (neither has been tested though), provided you have Java 1.6 installed. You will also need some software that can plot curves and data, like gnuplot. After you have logged in at the astronomy department you should first create a directory with your name. You can do this by opening a terminal and using the linux-command *mkdir*, *i. e.* type

```
mkdir yourname
```

Then download the program *depletion.jar*, using a web browser, to that directory. To enter the directory write

```
cd yourname
```

and to run the program type

```
java -jar depletion.jar
```

After you have finished the exercise, and have made sure that you have copies of all the files you need for writing the report, remove your directory with the command

```
rm -r yourname
```

The program has several options, you can set the number of balls, the size of the small ball, the big ball and the cavity. You can choose if you want to simulate a two- or three-dimensional system, and if you want the cavity to be spherical or ellipsoidal. You can also decide how long the simulation will be, by two parameters: cycle length and number of cycles. Cycle length is the number of individual MC steps that are performed in one cycle. Dividing the simulation into cycles is simply for convenience. The number of steps in a cycle is roughly the same as the number of balls, thus ensuring that after a cycle the position of most balls has been changed, *i. e.* a significantly changed state has been reached.

During the simulation, two files are written, one corresponding to the PMF (called *PMF file*), which contains $-\ln H(r_0)$, and one with run-time snapshots of the position of the large ball (*Runtime data file*). You can set the sample rate for the

histogram $H(r_0)$ and how often the files should be updated. All the settings that were used in the simulation will be written to both files.

When you have selected the settings the simulation can be started. The time needed per MC step increases approximately linearly with the number of balls N . If you double the number of balls you will need a twice as long simulation for the same number of steps. But the complexity of the simulations increases a lot for dense system, the number of MC steps needed to obtain adequate sampling increases rapidly with N when N gets large.

5 Problems

Problem 1 (18%): The ideal gas approximation

Run a simulation with the default values. The simulation will have statistical errors, but is not biased by any approximation. Compare the PMF obtained from the simulation with the free energy we found under the ideal gas assumption (using (21)). Also check how the omission of V_{big} in (14) affects your result (we also omitted V_{ov} , but that has a smaller effect). Do this for some different values of the parameters. Remember that the comparison is only valid in the range $0 \leq D \leq 2r$. The histogram is not normalized so an additive constant should be included in (21).

Problem 2 (16%): Dense systems

In the text-book by Nelson [1] an experimental study of this type of system is presented [2]. In that study $R = 474$ nm, $r = 42$ nm, $R_S \approx 1700$ nm and the volume fraction of small balls, $\phi_V = Nr^3/R_S^3$, was 0.3. Simulating such a dense system in 3D is outside the scope of this exercise however (too many balls are needed). But you can try to simulate a 2D system with a similar area fraction of small balls, $\phi_A \sim 0.1 - 0.3$. Even in the 2D system you will need 400–500 balls if you take the numbers directly from the experiment. Therefore, use a smaller R_S so that the number of balls is less than 200. You will still need longer simulation times than the default value to get clear results. Our ideal gas approximation doesn't apply to these dense systems, the free energy is more complicated.² In particular there is more than one minimum in the PMF, why?

²When you plot $-\ln H(r_0)$ for a two-dimensional system, the geometric correction term will be $\ln(2\pi r_0)$ instead of $\ln(4\pi r_0^2)$.

Problem 3 (16%): Curvature

In the experiment the dependence of the depletion force on the boundary curvature was studied [2]. An ellipsoid cavity is a simple example of a system with varying curvature along the boundary. Simulate a 2D elliptical system for some different densities. The program uses the value from the option *Size of cavity* for the semiminor axis, and sets the semimajor axis of the ellipsoid to two times that value. $F(r_0)$ is not displayed in these simulations because r_0 is not a good coordinate in an ellipse, instead you should look at the runtime snapshots to get a qualitative picture of what is going on. Explain what you see!

6 Report (25%)

The report should consist of four main parts, it's up to you how you want to organize your text, but the following should be included in one way or another:

- Introduction – describe the problem in general terms, give background information, etc.
- Theory – discuss the theory that describes the problem, including answers to exercises. To get the highest grade this part should be a continuous text, not just a list of answers to the exercises.
- Results and discussion – describe your results from the different simulations, discuss them and try to explain them as far as possible. If you prefer, results and discussion can be two separate sections.
- Conclusions – summarize what you have done and present any general discussion or conclusions you want to include.

The reports should preferably be submitted in printed form to the supervisor's mailbox at the department. If they are submitted electronically, send them as .pdf or .ps files (not .doc). The corrected reports can be found in the bookshelf outside lecture hall D.

You are encouraged to collaborate when doing the exercise but the reports should be individual. In the report you are allowed to use information from any source you like, but a reference should be given for each statement. Any material that is not originally yours, *i. e.* text from the internet or other students, and not correctly referenced, will be recognized and reported to the university disciplinary council. Past experience has shown that the vast majority of students understand this, but there have been exceptions.

References

- [1] Nelson, Philip. *Biological Physics - Energy, Information, Life*. 2004. Freeman, New York.
- [2] Dinsmore, A. D.; Wong, D. T.; Nelson, Philip; Yodh, A. G. *Hard Spheres in Vesicles: Curvature-Induced Forces and Particle-Induced Curvature*. 1998. *Physical Review Letters* 80: 409–412.
- [3] Chandler, David. *Introduction to Modern Statistical Mechanics*. 1987. Oxford University Press, Oxford.
- [4] Dickman, Ronald; Attard, Phil; Simonian, Veronika. *Entropic forces in binary hard sphere mixtures: Theory and simulation*. 1997. *Journal of Chemical Physics* 107: 205–213.

A Gnuplot

gnuplot is a useful and popular program for generating plots. To start gnuplot you simply type `gnuplot` in a terminal window, and you should see a copyright notice followed by a gnuplot prompt, like this one:

```
gnuplot>
```

Actually you do not always have to start gnuplot, you can save all your plot commands in a file and type `gnuplot filename` directly in the terminal window, but more about that later.

Here follows some examples of useful gnuplot commands.

Plot

Plotting simple functions:

```
plot x * 5 + 1
plot sin(x)
```

Defining and plotting a function:

```
f(x,a) = x+a
plot f(x,1) title "a=1", f(x,2) title "a=2"
```

Setting ranges for the axes:

```
set xrange [0:1]
set yrange [0:*] (only lower bound set)
```

There is a `replot` command, which will let you add more plots in an existing window:

```
replot cos(x)
```

3D Plots

To generate 3D plots you use the command `splot` instead of `plot` otherwise everything works the same way as in the 2D case, with `set zrange` setting the range of the z-axis. For example:

```
splot x*x + y*y
```

Plotting data from a file

It is of course also possible to plot data from a text file. Suppose you have a file called `out.data` that looks like this:

```
1 1.21 2.22
2 1.10 2.50
3 1.05 2.75
4 1.30 2.66
5 1.26 2.88
```

This is typically how the outputs from the programs used in the exercises will look like.

To plot the data from this file, you can type

```
plot "out.data" using 1:3 with linespoints
```

This will create a plot where x coordinates are taken from column 1 and y coordinates from column 3. If you don't specify the columns (i.e., you remove "using 1:3"), the first columns will be used. `linespoints` means that the data points will be plotted as points connected with lines. Some other useful styles are `lines`, `points`, `dots` and `yerrorbars`. With `yerrorbars` you need to have a column with error values, and the using clause should specify three columns (e.g. `using 1:3:5`). To set the plotting style for all subsequent plot commands, use `set data style`, e.g.:

```
set data style linespoints
```

Of course you are also able to `splot` from a text file, simply type

```
splot "out.data" using 1:2:3 with linespoints
```

You can transform the data in a column before plotting it, for example if you want to plot the square of the data in the second column, this will do the job

```
plot "out.data" using 1:($2**2)
```

The operator `**` means "to the power of".

Title and labels

To give the diagram a title and set labels on the axes, you can type:

```
set title "Title goes here"  
set xlabel "My x label"  
set ylabel "My y label"
```

Log scale

Once in a while you may need to plot something with a logarithmic scale on either axis. For instance, if you want a logarithmic y axis:

```
set logscale y
```

And to set the scale back to linear:

```
unset logscale y
```

Save your plot as postscript file

To print a plot, you first need to save it to a postscript file. (Postscript is a language understood by printers.) This is done by telling gnuplot to output in postscript, rather than to the screen. The commands needed for this are:

```
set terminal postscript  
set output "yourfilename.ps"
```

You will then have to re-run the plot command (or simply replot). The postscript plot will now be in the file `yourfilename.ps`. To print or view this file, open another terminal window. There, you can start the postscript viewer `gv`:

```
gv yourfilename.ps
```

It's possible to print from `gv`, or you can do it directly from the shell using `lpr yourfilename.ps`. Depending on what computer you are using and what printer you want to use, a different command (or a `-P` argument to `lpr`) may be needed.

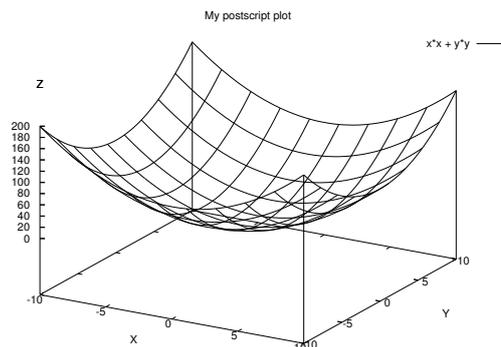
After generating a postscript file you'll probably want to go back to plotting to the screen: `set terminal X11`.

Putting it all together

It is often easier to save all plot commands in a file and type `gnuplot file` directly at the terminal (without first starting `gnuplot`). For example suppose you create a file called `myGnuplotFile.gnplt` that looks like this:

```
set terminal postscript
set output "myOutput.ps"
set title "My postscript plot"
set xlabel "X"
set ylabel "Y"
set zlabel "Z"
splot x*x + y*y with lines
quit
```

If you type `gnuplot myGnuplotFile.gnplt` in a terminal, a file called `myOutput.ps` that looks like this will automatically be created:



Getting help

`gnuplot` has a `help` command which lists all of the available commands and their options, and provides some useful examples. It may be a bit cumbersome to use, but once you get used to it it's not that bad. To get a list of all help topics, type `help`. If you want a list of the available commands: `help commands`. If you want help with a specific command:

```
help plot
help set key
help set term post
```

There is also help to be found online, see for example www.gnuplot.info.