NTNU – Trondheim
Norwegian University of
Science and Technology

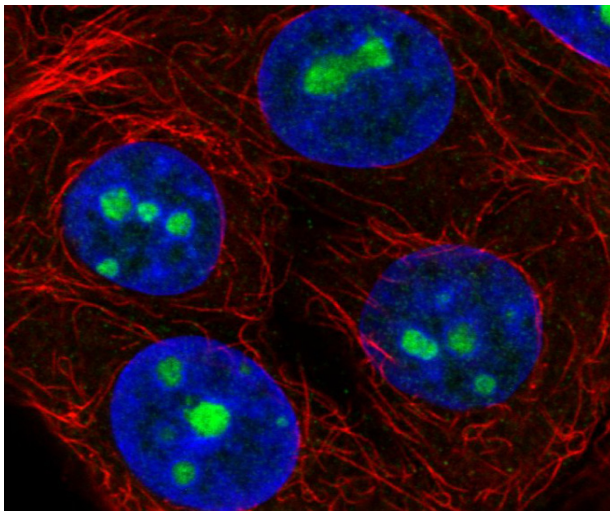
NTNU, Institutt for Fysikk

# TMA4320 vår 2022 - Biofysikkprosjekt

Niels Henrik Aase and Rita de Sousa Dias

January 28, 2022

## Impact of multivalency in the formation of cellular membrane-less organelles



Prosjektperiode: 31.01.22 - 13.02.22

## Praktisk informasjon

Innleveringsfrist: Mandag 13.02.22, kl 16.00
Innleveringsplattform: Inspera (Anonymt)
Språk: Dere kan velge om dere vil svare på engelsk eller norsk.
Innleveringsformat: Jupyter Notebook
Vurderingsansvarlig: Niels Henrik Aase, mail: `niels.h.aase@ntnu.no`.

# 1 Introduction

All living cells in multicellular organisms, such as ourselves, contain a number of specific structures called organelles ('little organs'). This allows the cells to physically separate (compartmentalize) function and components. Examples of organelles are the nucleus, which contains the genetic information, the rough endoplasmic reticulum, where many proteins are synthesized, and lysosomes, that contain enzymes that digest and degrade unwanted cellular components and pathogens (Fig. 1).

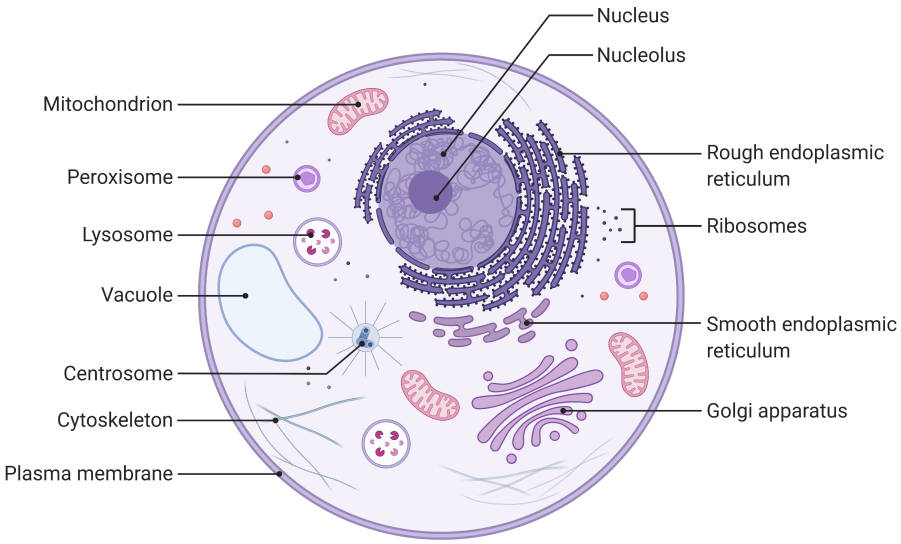



Figure 1: Schematic representation of an eukaryotic (animal) cell, highlighting the organelles. Adapted from "Structural Overview of an Animal Cel", by BioRender.com (2021). Retrieved from https://app.biorender.com/biorender-templates

All mentioned organelles are membrane-enclosed bodies, that is, a lipid membrane separates the cytosol of the cell from the interior of the organelles. Lipid membranes, with their hydrophobic interior, work as permeability barriers, allowing only the passage of small uncharged molecules. This allows the cells to control the transport of molecules across the membranes and their concentration within the different compartments.

While the best-known organelles are membrane-bound structures, cells also contain many organelles without membranes that have diverse biological functions. In this

project you will study some of the conditions necessary to form these structures. These membrane-free organelles can be described as liquid droplets formed by the spontaneous phase separation of some of the components present in the cell. Nucleoli, are an example of membrane-less organelles that appear in transcriptionally-active regions of the nucleus, that is, regions where the genes are transcribed to RNA (see figure in the cover page depicting cells observed using confocal microscopy. The DNA is marked with blue, the microtubules (skeleton of the cells) in red and the protein of interest in green [1].) Nucleoli are composed of RNA and a variety of proteins and formed when the concentration of certain proteins reach some threshold value [2, 3]. It is, however possible to induce the formation of liquid droplets *in vitro* using only two components [4].

One 'everyday example' of a liquid-liquid phase separation can be appreciated in mayonnaise. To make mayonnaise one mixes vinegar and other water-based components with, for example, olive oil and shakes it vigorously. The energy is used to break the oil into small droplets that mix in the water. These droplet are stabilized by the addition of egg. In vinegar/oil salad dressing without a stabilizer the oil and water quickly separate into two visible phases. While the interactions that are present in these examples are different to those forming the membrane-less organelles, the formation of liquid droplets (containing a larger concentration of some proteins than the surroundings, see cover figure) can be depicted in the same way.

Studies on the composition of these droplets have highlighted the importance of electrostatic interactions in these systems [5]. Electrostatic interactions are long ranged but their intensity decreases in aqueous solution (high dielectric permitivitty) and in the presence of small ions (screening of the interaction). The ionization of molecular groups (*e.g.*, $-\mathrm{COOH} \Leftrightarrow -\mathrm{COO}^- + \mathrm{H}^+$, or $-\mathrm{NH}_2 + \mathrm{H}^+ \Leftrightarrow -\mathrm{NH}_3^+$) depend on the concentration of $\mathrm{H}^+$ (*i.e.,* pH), which contribute to control the strength of interaction between molecules by varying their charge. Furthermore, cells possess a number of proteins that catalyse reactions such as acetylation and phosphorylation that, in practice, switch off a positive charge or introduces a negative one, respectively. This allows a more precise control of the interactions in a biological setting.

Other modes of intermolecular interactions have been found to be relevant for the formation of membrane-less organelles [6] however, in this project, we will solely focus on the charge-charge interactions.

As mentioned above, nucleoli are composed of RNA and proteins. These are examples of biopolymers (or biomacromolecules), that is molecules that possess repeating units (monomers) that are connected by covalent bonds. If each monomer possesses one interaction point (a charge, in this context), such connectivity between monomers allows for multiple interaction points within the same molecule (multivalency), which will increase the interaction strength between the polymer and other charged particles.

Recall that these multiple interaction points can be regulated biologically by, for example, switching on and off the charge of the biopolymer, as mentioned above. This way the cell can amplify or suppress phase separation and, consequently, organelle formation [7].

Another interesting property of polymers is their flexibility, thus the ability to adapt

their shape to best interact with surfaces and other molecules. The covalent bonds between monomers allow, most often, for an almost free-rotation around the bonds which allows the polymer to adopt many different conformations that have about the same potential energy

In this project you will use Monte Carlo to assess the importance of monomer connectivity (multivalency) in the formation of membrane-less organelles (clusters, see below) in systems composed of oppositely charged polymers.

## 2 How to describe a polymer-system numerically

A polymer consists of many monomers, bound to each other by covalent bonds. In order to see how they can be represented on a computer, let us first consider a system where monomers are placed in a solvent. This monomer-system has $M$ positively charged monomers, and $M$ negatively charged monomers. We confine their movement to two dimensions, such that their positions can be described using two coordinates, $x$ and $y$. In order to ease the numerical implementation, we further restrict their movement, such the monomers can only be positioned at grid points (Norsk: oppdeling/kvadratisk gitter). Thus a general point on the grid, $\boldsymbol{r}_k$, can be written as $\boldsymbol{r}_k = a(i\hat{x} + j\hat{y}),\ i, j \in [1, N]$, with $i, j$ being integers, and where $a$ is the distance between adjacent grid points. It follows immediately that subtracting dimensionful vectors is equivalent to

$$\boldsymbol{r}_k - \boldsymbol{r}_l = a(i_k\hat{x} + j_k\hat{y}) - a(i_l\hat{x} + j_l\hat{y}) = a\big((i_k - i_l)\hat{x} + (j_k - j_l)\hat{y}\big). \tag{1}$$

Equation (1) might seem trivial, but the concept of transforming *real* coordinates to *discrete* coordinates is extremely useful in physics and engineering in general. There is an equal number of grid points in the horizontal and vertical direction, so the grid is essentially an $N \times N$ matrix.

The magnitude of the charge of the monomers is always equal to the elementary charge $e$, and we denote the charge as $q$. Thus, the charge of an arbitrary monomer, $q_i$, can be written as

$$q_i = w_i e, \tag{2}$$

so that the magnitude of the charge is contained in $e$, while $w_i$ is only the sign of the charge. Note that because there are no interactions between a monomer and the solvent, it is convenient to use $w_i = 0$ for the solvent. The possible choices of $w_i$ for a monomer is of course just $\{1, -1\}$.

Even though it would be perfectly fine to represent all monomers as a collection of 1s and –1s, we act with foresight to accommodate future implementations of polymers. We give all positively charged monomers a unique positive number. With $M$ such monomers, the obvious choice is to enumerate the monomers from 1 to $M$. Similarly, we enumerate the negatively charged monomers from $-1$ to $-M$. So to summarize: The magnitude of the charge of all monomers is always equal to $e$. Each monomer is represented by a unique number, where the sign of the number contains the information about the sign of the monomer charge.

We are now ready to discuss how to deal with the inclusion of polymers. The number of monomers making up a polymer is denoted by $L$. For simplicity, interactions between monomers belonging to the same polymer are considered to be zero. Similarly to the case of a monomer-system, we enumerate each polymer by a unique number (again, the sign of this number represents the charge of the polymer), and all monomers belonging to the polymer are assigned the same number. Note that putting $L$ equal to 1, the implementation reduces naturally to the monomer-system described previously. Then all "polymers" are just constituted by a single monomer.

In this project we will only study polymers that never break, which is a reasonable assumption. We define an unbroken polymer as a polymer where there exists a path between any pair of monomers belonging to the polymer, where the path is piece-wise horizontal or vertical (not diagonal!), and that the path only touches grid points where there are monomers belonging to the same polymer. An illustration of a broken polymer is given in Figure 2.

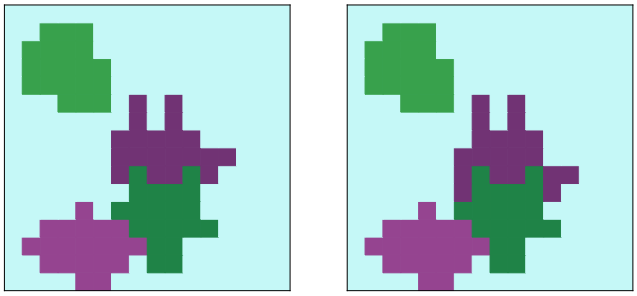


Figure 2: Illustration of a broken polymer. The light blue, the green, and the violet represent the solvent, positively charged monomers, and negatively charged monomers, respectively. Different shades of violet and green represent different polymers. In the left panel, the original system with $N = 16$, $M = 2$, $L = 20$ is illustrated. In the right panel, the violet polymer is moved downwards, causing three monomers to loose their connection to the rest of the polymer.

## 2.1 Pseudo-code for initializing a system of polymers

An algorithm for generating an $N \times N$ grid with $M$ positvely charged, unbroken, polymers, with $L$ monomers constituting each polymer, is given in Algorithm 1. A placement is illegal if is attempted to place a monomer on an already occupied grid point. In the algorithm, we have used for-loops, but in practice you should use while-loops. A visualization of the output of this function can bee seen in the left panel in Figure 2.

## 2.2 Energy

As stated in the introduction, we will only consider electrostatic interactions between the monomers. Using Coulomb's law together with equation (2), we can write the interaction

**Algorithm 1** Initialization of system with only positively charged polymer

```
grid ← N × N array of zeros
for i = 1 to M do
    randomPosition ← Random position on grid
    if illegalPlacement then
        Try again with new random position
    else
        grid(randomPosition) ← i
        monomerPositionsArray ← L × 2 array of zeros
        monomerPositionsArray(0) ← randomPosition
        for j = 1 to L − 1 do
            randomMonomer ← position of a random monomer
in monomerPositionArray
            if no legal placement of neighbors then
                Try again with new randomMonomer
            else
                randomNeighbor ← position of a random grid point filled with solvent,
adjacent to randomMonomer
                grid(randomNeighbor) ← i
                monomerPositionsArray(j) ← randomNeighbor
            end if
        end for
    end if
end for
```

between two monomers as

$$V_{kl} = w_k w_l \frac{e^2}{4\pi\varepsilon_0\varepsilon_r a^2} \frac{a^2}{|\boldsymbol{r}_k - \boldsymbol{r}_l|} \equiv w_k w_l \alpha \frac{a^2}{|\boldsymbol{r}_k - \boldsymbol{r}_l|}, \tag{3}$$

where $V_{kl}$ denotes the interaction between the monomers situated at $\boldsymbol{r}_k$ and $\boldsymbol{r}_l$, and $\varepsilon_0$ and $\varepsilon_r$ denote the vacuum and the relative permittivity, respectively. and $\alpha$ is a constant with unit Joule.

Now we will make a rather crude approximation. We will put the interactions between any monomer pair to zero, unless they are situated directly next to each-other on the grid, i.e. they have to be *nearest neighbors* [1]. Assuming that $\boldsymbol{r}_k$ and $\boldsymbol{r}_l$ are nearest neighbors (NN) and using equation (1), it follows that the interactions between monomers situated

[1] While this may seem like an oversimplification, it can be justified by the fact that the solvent screens the electric field emitted from the monomers, effectively quenching any long range interactions. This approximation has applications in many areas in physics, e.g. a similar effect can be derived in the context of solid state physics using quantum field theory, and is used to predict the conductivity of metals.

at $\boldsymbol{r}_k$ and $\boldsymbol{r}_l$ can be written as

$$V_{kl} = \begin{cases} w_k w_l \alpha \text{ if } \boldsymbol{r}_k \text{ and } \boldsymbol{r}_l \text{ are NN,} \\ 0 \text{ otherwise} \end{cases} . \tag{4}$$

Note that on our grid, each grid point has four NNs.

**NB!** One caveat with equation (4) is that it does not distinguish between monomers belonging to the same or to different polymers. Recall that we want to restrict the interactions to monomers that belong to different polymers, as it is less computationally demanding. Thus, all interactions between monomers belonging to the same polymer are always zero. The total energy of a polymer-system is then given by

$$E = \sum_{k,l} V_{kl}, \tag{5}$$

where the sum is over all NN pairs of monomers that belong to different polymers.

### 2.3 Boundary conditions

Because the system is represented on a $N \times N$ grid, one might ask the question, what are the neighboring sites of grid points situated on the edge of the grid? We will make use of periodic boundary conditions (PBCs), where the grid points situated to the right of the rightmost column is simply the leftmost column. Analogously, south from the bottom row, is the top row. This can be imagined as a torus/donut, as illustrated in Figure 3. If you make two cuts along the two red lines with a scissor, you would get back your original grid, without PBCs.

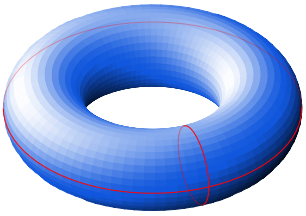


Figure 3: Illustration of the geometry arising from periodic boundary conditions.

We stress that even though the geometry (more specifically, the topology) might seem complicated, you should think of this is a "programming trick" that will ease the numerical implementation. In real life, polymers do not live on the surface of such complex structures! On the other hand, having hard walls along the edges of the grid induces 'surface effects' (not present in a 'real' solution), which may strongly affect the polymer conformation (that is, the relative position of the monomers in the polymer).

## 3 Monte Carlo methods

Monte Carlo (MC) methods is a term that incorporates a wide variety of algorithms. The defining trait of these algorithms is that they make use of random sampling. In

this project we will use the Metropolis algorithm, which is a Markov chain Monte Carlo (MCMC) method. It is instructive to understand a Markov chain using an example, which we borrow from Haakon Thømt Simensen.

"A Markov chain can be defined as a sequence of events, in which the next event only depends upon the present event. That is, for any event there is a probability distribution for the next events. There is a finite amount of possible events, and all events must somehow be within the reach from any other event. However, there is no requirement that all events must be within one step from any given event. A simple example of a Markov chain is if you decide to buy a random plane ticket (from a pool of all available tickets) at Værnes Airport, and fly to whatever destination you got. At your arrival, you immediately buy another random ticket, and fly to your next destination. During your journey, you keep record of which airports you visit. This is a Markov process, as the next possible destinations only depend on where you are, that is, on the tickets available in your current location, and not where you have been (we neglect difference in weekdays, time of day and such complicating factors). If you look at your record of airports after having travelled for a long time, you will probably see that airports such as Frankfurt, Dubai, Chicago O'Hare, Los Angeles, Beijing, Heathrow etc. are well represented, while you have been lucky if you have had the opportunity to visit e.g. Funafuti International Airport. Now, if we make a bar graph of airports vs. number of visits, what exactly are we looking at? We are looking at an empirical sample of the probability distribution of where to find you after a long time of travelling in the way described above! The law of large numbers states that if we have sampled over a sufficiently long time, the sample distribution will approach the real equilibrium distribution of the Markov chain.

If you don't like flying that much, we are able to reproduce the distribution above only by knowing the plane ticket distribution at each airport, and then simulate the ticket lottery on a computer with the use of random numbers. Instead of travelling for years, we only need a couple of seconds to obtain a good approximation of the real probability distribution. Now, that's an improvement in computation time! This is an example of a Markov chain Monte Carlo method."

If we know the mechanisms that govern a physical system, we can simulate the system in a similar way as with the airport example. Let us now turn our attention towards the polymer systems we introduced earlier. The polymers interact with each other through electrostatic interactions, and by equation (5), we know how to calculate the energy $E$ of a system configuration $S$. In the case of polymer-systems, $S$ represents how the polymers are arranged in the solvent. In our numerical implementation, the grid describes the system configuration, as it contains all information about the placement of the polymers. Say we want to calculate some interesting real physical property of the system (e.g. how many clusters the polymers form) denoted $x$, and that for any system configuration, we know how to calculate $x(S)$. The expression for the mean value of $x$, denoted $\langle x \rangle$, is then given by

$$\langle x \rangle = \sum_{S \in \text{ states}} f(S)x(S), \tag{6}$$

where $f(S)$ is the probability that the system has a system configuration $S$. The sum is

over all possible states the system can find itself in. Thus, in order to calculate $\langle x \rangle$ we need an expression for $f(S)$.

Using thermal and statistical physics (which we will not go into detail in this project, it is covered in the course TFY4230), one can derive that in physical systems, $f(S)$ only depends on the energy of the configuration $S$, and the inverse temperature of the system (often denoted as $\beta = \frac{1}{k_B T}$). **NB!** In this project we always assume the temperature to be in Kelvin, not Celsius. The expression for $f(S)$ is equal to

$$f(S) = \frac{1}{Z} \mathrm{e}^{-\beta E(S)}, \tag{7}$$

where $Z$ is a quantity named the partition function. The partition function is much used in theoretical physics, but for our purposes we can treat it a constant that normalizes the probability distribution $f(S)$.

In general, $Z$ is hard to calculate. However, consider what the ratio between the probability of the system being in state $S_n$ and $S_i$

$$\frac{f(S_n)}{f(S_i)} = \mathrm{e}^{-\beta(E(S_n) - E(S_i))}. \tag{8}$$

In order to find the probability ratio between two states, we thus only need to evaluate the change of energy $\Delta E = E(S_n) - E(S_i)$, the scary quantity, $Z$, cancels!

### 3.1 The Metropolis algorithm

To numerically find an approximation for $f(S)$, we will use equation (8) as a guiding principle. We study a system, initialized in some random state, denoted $S_0$. Imagine that a polymer is chosen at random, we move it in a direction (horizontally or vertically), also chosen at random. The new, imagined, system configuration where the polymer has been moved is denoted $S_n$. In the Metropolis algorithm, there are two ways the system can actually change from $S_0$ into the new imagined configuration. Let's consider them in detail:

1. As most of you already have an intuition of, a physical system seeks to minimize its energy. Thus, if $\Delta E = E(S_n) - E(S_0) < 0$, the imagined move is physically realized, the polymer moves, and the system configuration thus changes.

2. Even if $\Delta E > 0$, the polymer may still move due to thermal fluctuations. We account for this possibility by comparing $\exp(-\beta \Delta E)$ with a random decimal number between 0 and 1, denoted $p$. If $\exp(-\beta \Delta E) > p$, the imagined move is realized, even though the move is energetically unfavorable.

In this project, we define the process above as one MC time step, denoted by $t$. A large scale Monte Carlo simulation is often just a matter of carrying out a large amount of MC steps. For the interested reader, we refer to [8] for more details regarding the Metropolis algorithm.

Before you start solving the exercises, it might be instructive to consider how you expect a high temperature simulation will be different from a low temperature simulation. Think of how the inequality $\exp(-\beta\Delta E) > p$ depends on the temperature, and how the system may get stuck at local energy minima at low temperatures.

### 3.2 Pseudo-code for Monte Carlo algorithm

In Algorithm 2, we have provided a possible pseudo-code for an MC simulation of a polymer system. As will be clear from the exercises, the necessary functionality to run the simulations should be developed before you implement the algorithm in full.

**Algorithm 2** MC simulation of interacting polymers and their total energy

```
grid ← N × N array of zeros
grid ← initializeGrid(grid)
ϵ ← N_s array of zeros
E ← energy of grid
ϵ(0) ← E
for i = 0 to N_s − 1 do
    E ← energy of grid
    Pick a random polymer
    Pick a random direction
    if illegalMove then
        Do nothing
    else
        newGrid ← Copy of grid but with the polymer moved
        E_new ← energy of newGrid
        if E_new < E then
            grid ← newGrid
            E ← E_new
        else if RAND(0, 1) < exp(−β(E_new − E)) then
            grid ← newGrid
            E ← E_new
        else
            Do nothing
        end if
    end if
    ϵ(i) ← E
end for
```

### 3.3 A crash course in statistical aspects of Monte Carlo simulations

Monte Carlo simulations are often used to calculate some interesting physical property in parameter regimes that are inaccessible through brute-force numerics. However, as

the systems we study increase in size, the number of Monte Carlo steps (which we will often describe as time) required to obtain accurate results grows rapidly. Let's use our polymer system as an example. Again, let $x$ be some physical quantity that we know how to calculate if we know the system configuration (i.e. we can calculate $x(S)$).

Firstly, as will become clear in the exercises, if the system is initialized in a random state, it will take some time before the system finds itself in a state that is representative for the system's natural state. Typically, if the energy of the system is plotted, it will start out being close to zero, but as time passes, the energy will be lowered, until it reaches and fluctuates around a mean value. These fluctuations may be quite large, but we still say that the system has reached its equilibrium. The number of MC steps required to reach equilibrium from a random state is denoted as $t_{\text{equil}}$. Note that because MC simulations are stochastic in nature, $t_{\text{equil}}$ will not always be the same, so it is often a good idea to overestimate it, such that you are certain that the system has indeed reached its equilibrium after $t_{\text{equil}}$ MC steps. Typically, $t_{\text{equil}}$ increases with system size (i.e. $N$ for our system).

Once we have done $t_{\text{equil}}$ MC steps, we are ready to calculate $\langle x \rangle$. This will be done by calculating $x$ as a function of $grid(t)$ at different $t$, and then taking the average $\bar{x}$. However, if $x$ is calculated for the grid at $t = t_{\text{equil}}$, it is superfluous to also calculate $x$ at $t = t_{\text{equil}} + 1$. Why? Because the system at $t = t_{\text{equil}} + 1$ is essentially the same as at $t = t_{\text{equil}}$ as, at most, they differ by having moved one polymer. You might recall from statistics that the average of some stochastic variable is a good estimator for the true value of the variable, but this is only true if the average is made up by *independent* stochastic variables. Clearly, $x(grid(t = t_{\text{equil}}))$ and $x(grid(t = t_{\text{equil}} + 1))$ are not independent, so one should not use both when calculating $\bar{x}$. This is remedied by waiting $t_r$ MC steps between measurements of $x(grid)$. One can optimize how long one should wait between measurements, see e.g. [8], but to keep it simple, we will always wait $t_r = 1000$ between measurements of physical quantities (unless we calculate the energy, in which case we will keep all measurements).

Lastly, as you also might recall from statistics, the accuracy of using $\bar{x}$ as an estimator for $x$ increases with the number of measurements used to calculate $\bar{x}$. We denote this number by $n$. You should choose $n$ such that the run time of your code is reasonable, but be aware that increasing $n$ will increase the validity of your results.

### 3.3.1 Summary of useful variables for a Monte Carlo simulation

- $t_{\text{equil}}$: The number of time steps done before measurements of the systems starts.
- $t_r$: The number of time steps between each measurement.
- $n$: The number of measurements made after the equilibration of the system.
- $N_s$: The total number of time steps of the Monte Carlo simulations. Note that $N_s = t_{\text{equil}} + t_r n$.

# 4 Exercises

The exercises follow below. The exercises where you are supposed to give an explicit figure or answer a specific question are underlined. For your convenience, you may refer directly to specific questions (such as e.g. 1e)) in your text, but still try to write a cohesive text. In particular, include a small introduction of the project which does not overly rely on the project description. In some exercises you are asked to implement a function, but you will need more functions than the ones we explicitly ask for. It might be useful to quantify your uncertainty by calculating the standard error and standard deviation when appropriate.

## Exercise 1: Monomers

In order to acquaint ourselves with Monte Carlo simulations, we will start with the simplest physical system, namely a system of monomers in solvent. Thus, you work with a multivalency, $L$, equal to one for the entirety of exercise 1. However, a lot of the functionality you develop here can be reused for the polymer-systems, so try to keep your code flexible when possible.

This system also serves as a control for assessing the impact of monomer connectivity (i.e. the polymers we will study in the next exercise) in the formation of membrane-less organelles (here referred to as clusters).

### Parameters

In the monomer-part of the exercise, as parameter for your systems, use relative permittivity $\varepsilon_r = 78$, which is the relative permittivity of water at room temperature. For the distance between grid points, use $a = 23\,\mu\text{m}$. Because the distance between grid points is essentially the length of the monomer, this value of $a$ is orders of magnitude larger than realistic values for $a$. We close our eyes to this problem, as using realistic values of $a$ would result in a dramatic increase in required computational run time. However, the physics in the problem remains the same. Look up the tabulated values for the other necessary parameters.

**1a)** Write a function that takes as input the grid size, $N$, and the number of monomers we wish to consider $M$. The function should return a grid with the monomers placed randomly. Monomers can only be placed in the solvent, not on top of other monomers. Check that the grid has $2M$ non-zero values. $M$ of these values represent positively charged monomers, the other $M$ represent negatively charged monomers.

**1b)** Use `plt.pcolormesh` (or a similiar function) to visualize a system with grid size $N$. Include a figure of the system you created in **1a)**. It should be clear from the figure what $N$ and $M$ you have chosen. *Tip:* If you decide to use the function `plt.pcolormesh`, you might notice that the rows are interchanged in the plot and in matrix form. An easy fix for this is to use `grid[::-1,]` as input to `plt.pcolormesh`.

**1c)** Write a function that takes the size of the grid $N$, and the matrix indices $i$ and $j$ as input, and returns an array containing the coordinates of the four nearest neighbors. Remember to account for the periodic boundary conditions as described in subsection 2.3.

**1d)** Write a function that takes the grid as input, and returns the total energy of the system. *Tip:* For all monomers, use equation (4) to calculate the energy between the monomer and all of its nearest neighbors. The energy of the system is the sum of all the monomer energies (divided by some integer that account for the overcounting of interactions). You may introduce a dimensionless quantity $E/\alpha$ if you wish. Calculate the energy of the grid you presented in **1b)**.

**1e)** Implement the Monte Carlo algorithm described in Algorithm 2. The input parameters should be the number of Monte Carlo steps in the simulation, $N_s$, as well as $N$, $M$, $T$ and any other parameters you find necessary. It will be convenient to also have a grid as an argument to the function. This grid is the initial system configuration before the simulation starts. For monomer-systems, the only illegal move is trying to move a monomer on top of another monomer.

**1f)** Run two simulations, both with, $N_s = 50000$, $N = 15$ and $M = 25$, but with differing $T$. Use $T = 200\,\mathrm{K}$ and $T = 500\,\mathrm{K}$. We note that while such temperature variations are not realistic from a biological point of view, it is still interesting to consider its impact. After the last MC step is complete, save the grid configuration from the low $T$ simulation (Using e.g. `np.savez`). Plot how the energy develops as a function of $t$ for both simulations in the same figure. Determine visually how many MC steps are needed for the system to reach its equilibrium[2] for the two temperatures. We denote this value as $t_{\text{equil}}$. Explain why $t_{\text{equil}}$ is different for the two temperatures. *Hint:* You might want to relate this to the concept of local energy minima.

We will now turn our attention to the formation of aggregates, that is, the clustering of monomers. We define a cluster of monomers as a group of monomers where there exists a continuous path between any monomer belonging to the group. The path can only be piece-wise horizontal or vertical, and crucially, the path can only be along other monomers belonging to the same group (i.e. the path cannot be across any solvent). In Figure 4 we have provided an example for the different clusters of the system.

In order to create a data structure that keeps track of where the clusters are, it is convenient to create a new grid/matrix, of equal size with the grid keeping track of the monomer-positions. Let's call this grid for the cluster grid. At any position in the original grid where there is a monomer (represented by a non-zero value), there will be a positive number at the same position in the cluster grid. Each cluster is represented by a unique positive number, so all monomers belonging to a cluster will have the same positive number in the cluster grid.

[2]Here, equilibrium means that the energy fluctuates around a stable mean value.

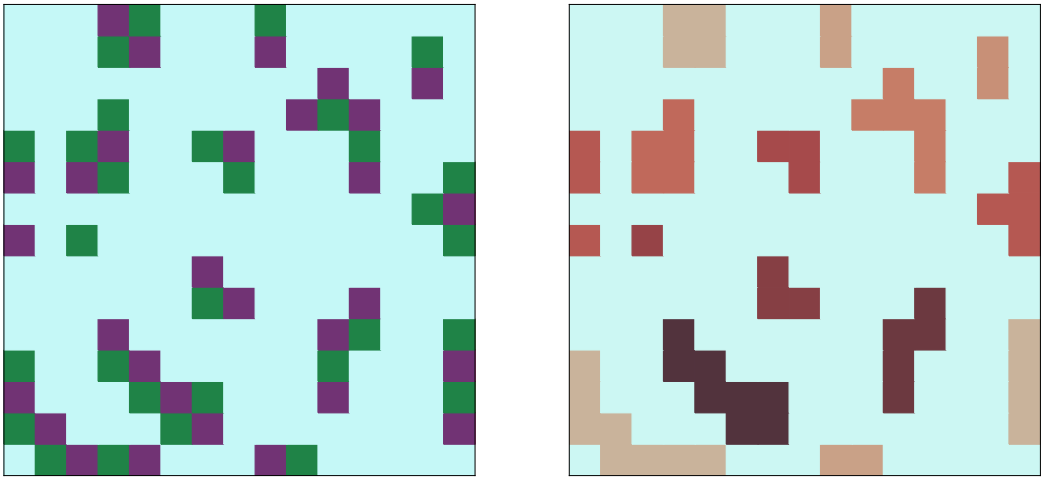


Figure 4: *Left:* A system consisting of monomers situated on a grid. The light blue, the green, and the violet represent the solvent, positively charged monomers, and negatively charged monomers, respectively.
*Right:* The corresponding grid cluster of the system. Different clusters are represented by different colors. Note that clusters may cross the boundaries, due to the periodic boundary conditions.

**1g)** Write a function that takes a grid as input, and returns the corresponding cluster grid. *Hint:* Using recursion in combination with the function you wrote in **1c)** may be a good idea here. Use the data you saved in **1f)** to visualize the grid and the corresponding cluster grid side-by-side (i.e. a similar figure to Figure 4).

You are now ready to do your first large scale Monte Carlo simulation. The physical property of interest is the mean cluster size, $\langle d \rangle$, where the size of a cluster is defined as the total number of monomers belonging to the cluster. In the next simulation, use $N = 15$, $M = 25$. Consider 10 evenly spaced temperatures between $T_l = 100\,\mathrm{K}$ and $T_h = 1000\,\mathrm{K}$. As discussed in subsection 3.3, it is necessary to let the system reach its equilibrium before you start making measurements. Choose $t_{\mathrm{equil}}(T) = t_{\max} \exp(-s(T - T_l)) + C$ with $t_{\max} = 100000, s = 1/200, C = 10,000$, as this will result in reasonable run times. Note that this is a slightly optimistic estimate for how many MC steps that are required to reach equilibrium for low temperatures, so if the run time is manageable, you might consider to increase $t_{\max}$. However, if the run time of your code with $t_{\max}$ is too long, you may use a lower value. Use $t_r = 1000$, and choose a suitable $n$. Use different initial grids at different temperatures.

**1h)** Plot $\langle d \rangle$ as a function of $T$. Why is $\langle d \rangle$ larger at small $T$? Discuss if your choice of $n$ will yield reliable results. Are any of your results surprising? In order to observe how the initial conditions of the grid affects the clustering size, redo the simulation, and compare the results of the two simulations. Why is the discrepancy between the simulations larger at lower temperatures?

## Exercise 2: Polymers

We will now turn our attention to polymers, and observe how the introduction of monomer connectivity (i.e. covalent bonds between monomers) affect aggregate formation/clustering. Furthermore, you will be asked to evaluate how the polymer's flexibility affects this clustering.

### Parameters

Continue to use $\varepsilon_r = 78$, but change the value of $a$ to $a = 91\,\mu\text{m}$.

**2a)** Write a similar function as you did in **1a)**, but now you should also take the multivalency $L$ as input. This is an extension of Algorithm 1 where you place $M$ positively and $M$ negatively charged polymers. The function should return a grid with $2M$ polymers, each with multivalency $L$. Include a figure of a system with $L > 20$.

**2b)** Extend the energy function you wrote in exercise **1d)**, so it can calculate the energy of polymer-systems. As mentioned in the introduction, covalent bonds bind monomers within the same polymer, thus, for simplicity, you may put interactions between monomers belonging to the same polymer to zero.

There are many ways to computationally mimic how polymers move. In the interest of keeping the programming complexity manageable, in this project we will avoid twisting the polymers. In the following exercises, you are asked to write two functions that can move polymers. The different moves will give the polymers a varying degree of flexibility. See Figure 5 for illustrations of different types of moves. One of these types of moves, is called a *rigid* move. This move has the following simple rules (We use moving to the right as an example): Every monomer belonging to the polymer is moved one step to the right. If any of these monomers collides with a monomer belonging to a different polymer, the move is prohibited, and the entire polymer remains in its original position.

**2c)** Implement a function that rigidly moves an input-specified polymer in an input-specified direction. In order to specify direction, one possible choice is to specify the direction by integers: $[\text{Right}, \text{Left}, \text{Up}, \text{Down}] \iff [0, 1, 2, 3]$.

**2d)** Implement Algorithm 2 and use it to run a Monte Carlo simulation of a system with $L > 10$, $2 < M < 6$, $T = 200\,\text{K}$. Choose $N$ freely, and use the rigid move from **2c)** to move the polymers. Include a figure of the grid after 30000 MC steps. Comment on why the system looks the way it does. Explain, using both quantitative results from your simulation and from an "algorithmic" point of view, how and why the run time requirements have changed in comparison with the monomer-systems you studied in exercise **1)**. *Tip:* It might be convenient to have the function that moves the polymers as an optional argument to the Monte Carlo function. In this way, you can reuse your MC function when you introduce new move functions.

Other types of moves are more challenging to implement, but will give more realistic simulations. You are now asked to implement the *medium flexibility* move, illustrated in the bottom left panel in Figure 5. It has the following rules: Every monomer belonging to the polymer is moved one step in the same horizontal (vertical) direction. If any of these monomers collide with a monomer belonging to a different polymer, the entire row (column) where this occurs is not moved, it remains in place. Collision-free rows (columns) move as normal. In order to highlight the difference between this type of move, and a *full flexibility move* (which you are not asked to implement!), an illustration of a full flexibility move is also included in Figure 5.

**2e)** Implement the medium flexibility move. *Tip:* It might be useful to do this exercise in tandem with **2f)**.

If the medium flexibility move is used, there is a chance that the polymers breaks. A broken polymer is defined using our definitions of clusters in exercise **1g)**: If the path (fulfilling the same requirements as in exercise **1g)**) between any two monomers belonging to the polymer is broken, so is the polymer. An illustration of this is given in Figure 2.

**2f)** Write a function that checks if a polymer is broken. *Hint:* This function will be almost identical to the one you wrote in exercise **1g)**. Improve your implementation of the medium flexibility move, such that it checks if the polymer is broken after the move is completed. If it is broken, the function should instead return the original grid (i.e. the grid where the polymer is not moved).

**2g)** Redo the simulation in exercise **2d)** with the same parameters, but using the medium flexible move. Compare the grid produced with flexible polymers to the one you presented in 2d). Plot the energy as a function of MC steps from the two simulations, and comment on your results. How does the energy of the system relate to the physical placement of the polymers?

As mentioned in the introduction, liquid-liquid phase separation drives the formation of membrane-less compartments in eukaryotic cells. The droplet-like compartments possess a large concentration of biopolymers. Now you are asked to study how the size of polymers (which is closely linked to multivalency) impacts the clustering, and thus also the formation of organelles.

In the last large scale simulation, you are asked to calculate two quantities related to phase-separation, as a function of $L$. The first being the average cluster size divided by $L$, $\langle d \rangle / L$. The second quantity is the average number of clusters, denoted $\langle m \rangle$. Each cluster represents an organelle/compartment.

**2h)** Use $T = 300\,\mathrm{K}$, $t_r = 1000$, $N = 30$ and $M = 5$ to calculate $\langle d \rangle / L$ and $\langle m \rangle$ as a function of $L$. For $L$, use 13 evenly spaced values between 3 and 39. Use the medium flexibility move. You decide yourself the values of $t_{\mathrm{equil}}$ and $n$, but justify your

choices. Plot $\langle d \rangle / L$ and $\langle m \rangle$ as a function of $L$. Discuss your results. How and why do you think your results would change if you used the rigid move? Because the initial grid in each system is randomly generated, redoing the simulations might yield quite different quantitative results. For which values of $L$ do you expect the results to vary the most? Lastly, choose a different system parameter[3] than $L$, and discuss qualitatively how changing this parameter will induce/destroy aggregate formation.

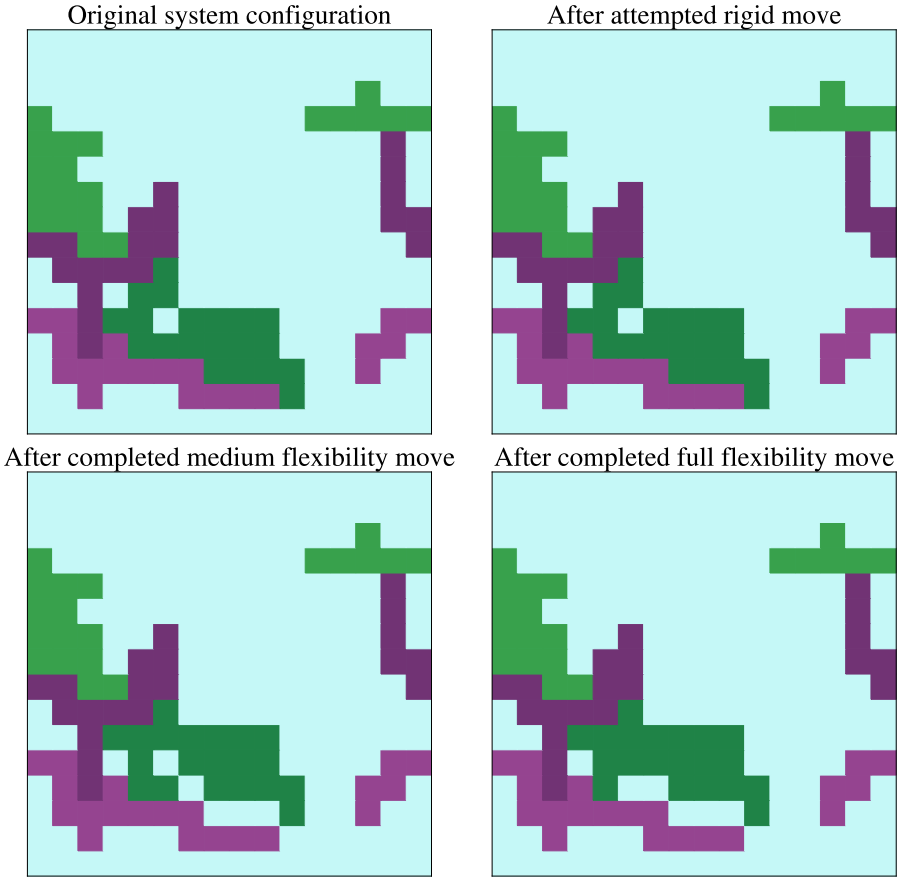



Figure 5: A system configuration with $N = 16$, $M = 2$, and $L = 20$ is demonstrated in the upper left panel. The results of three different ways of moving polymers are illustrated in the other panels. The dark green polymer in the center of the grid tries to move upwards. If the move is rigid (upper right panel), the move is prohibited due to the collision between the dark green polymer and the violet polymer. In the case of the two flexible moves, the polymer moves in a similar manner, expect in the column where there is a collision and solvent below the collision. With medium flexibility, the entire column remains in place. With full flexibility, only the two upper monomers in the column remains in place, the rest of the column moves upwards.

[3]Some system parameters: $N$, $M$, $T$, $\alpha$. Recall that $\alpha$ control the strength of interactions. Physically, changing $\alpha$ can be accomplished by changing the pH of the solvent, which subsequently changes $\varepsilon_r$. This can be done in real biological systems.

## References


[1] L. Stenström, D. Mahdessian, C. Gnann, A. J. Cesnik, W. Ouyang, M. D. Leonetti, M. Uhlén, S. Cuylen-Haering, P. J. Thul, and E. Lundberg, "Mapping the nucleolar proteome reveals a spatiotemporal organization related to intrinsic protein disorder," Molecular Systems Biology **16**, 9469 (2020).

[2] J. Berry, S. C. Webber, N. Vaidya, M. Haataja, and C. P. Brangwynne, "Rna transcription modulates phase transition-driven nuclear body assembly," Proc. Natl Acad. Sci. USA **112**, 5237–5245 (2015).

[3] S. C. Weber and C. P. Brangwynne, "Inverse size scaling of the nucleolus by a concentration-dependent phase transition," Curr. Biol. **25**, 631–646 (2015).

[4] D. Bracha, M. T. Walls, and C. P. Brangwynne, "Probing and engineering liquid-phase organelles," Nat. Biotechnol. **37**, 1435–1445 (2019).

[5] N. Srinivasan, M. Bhagawati, B. Ananthanarayanan, and S. Kumar, "Stimuli-sensitive intrinsically disordered protein brushes," Nature Commun. **5**, 5145 (2014).

[6] C. P. Brangwynne, P. Tompa, and R. V. Pappu, "Polymer physics of intracellular phase transitions," Nature Physics **11**, 899–904 (2015).

[7] P. Li, S. Banjade, H.-C. Cheng, S. Kim, B. Chen, L. Guo, M. Llaguno, J. V. Hollingsworth, D. S. King, S. F. Banani, P. S. Russo, B. T. N. Qiu-Xing Jiang, and M. K. Rosen, "Phase transitions in the assembly of multivalent signalling proteins," Nature **483**, 336–340 (2012).

[8] M. Newman and G. Barkema, *Monte Carlo Methods in Statistical Physics* (Oxford University Press, 1999).