UPPSALA
UNIVERSITET

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 2015

Polymer and Protein Physics

Simulations of Interactions and Dynamics

ANNA SINELNIKOVA

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2021

ISSN 1651-6214
ISBN 978-91-513-1139-5
urn:nbn:se:uu:diva-434275

Dissertation presented at Uppsala University to be publicly examined in Polhemsalen, Ångströmlaboratoriet, Lägerhyddsvägen 1, Uppsala, Friday, 26 March 2021 at 13:30 for the degree of Doctor of Philosophy. The examination will be conducted in English. Faculty examiner: Professor Erik Lindahl (Stockholm University, Department of Biochemistry and Biophysics).

Abstract

Sinelnikova, A. 2021. Polymer and Protein Physics. Simulations of Interactions and Dynamics. *Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology* 2015. 126 pp. Uppsala: Acta Universitatis Upsaliensis. ISBN 978-91-513-1139-5.

Proteins can, without any exaggeration, be called the "building blocks of life". Their physical properties depend not only on the chemical structure but also on their geometric shape. In this thesis, I investigate protein geometry using several different methods.

We start with a coarse-graining model to study the general behavior of polymers. For this reason, we utilize an effective Hamiltonian that can describe the thermodynamic properties of polymer chains and reproduce secondary and tertiary structures. To investigate this model, I perform classical Monte Carlo simulations using my software package.

Another problem we address in this thesis is how to distinguish thermodynamic phases of proteins. The conventional definition of phases of polymer systems uses scaling laws. However, this method needs the chain's length to be varied, which is impossible to do with heteropolymers where the number of sites is one of the system's characteristics. We will apply renormalization group (RG) theory ideas to overcome this difficulty. We present a scaling procedure and an observable through which RG flow can define a certain polymer chain's phase.

Another part of the thesis is dedicated to the method of molecular dynamics. Our focus is on a novel experimental technique called Single Particle Imaging (SPI). The spatial orientation of the sample in this method is arbitrary. Scientists proposed to use a strong electric field to fix the orientation since most biological molecules have a non-zero dipole moment. Motivated by this, we investigate the influence of a strong electric field's ramping on the orientation of protein ubiquitin. For the same question of SPI and using the same protein, we study the reproducibility of unfolding it in a strong electric field. With the help of a new graph representation, I show different unfolding pathways as a function of the electric field's value and compare them with thermal and mechanical unfolding. I show that the RG flow observable can also detect the different ubiquitin unfolding pathways more simply.

The study described in this thesis has two types of results. One is a very concrete type, which can be utilized right away in the SPI experiments, like MS SPIDOC on the European XFEL. The other type of results are more theoretical and opens up a new field for further research. However, all of them contribute to protein science, an area vital for humanity's ability to protect us from threats such as the current COVID-19 pandemic.

Keywords: polymers, proteins, Monte Carlo, molecular dynamics, phase diagram, renormalisation group, SPI, polymer effective model, coarse-graining

Anna Sinelnikova, Department of Physics and Astronomy, Materials Theory, Box 516, Uppsala University, SE-751 20 Uppsala, Sweden.

ISSN 1651-6214
ISBN 978-91-513-1139-5
urn:nbn:se:uu:diva-434275 (http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-434275)

In loving memory of my father
Boris Sinelnikov (1935 – 2018)

List of papers

This thesis is based on the following papers, which are referred to in the text by their Roman numerals.

I **Phase diagram and the pseudogap state in a linear chiral homopolymer model**
A. Sinelnikova, A.J. Niemi, M. Ulybyshev
Phys. Rev. E ***92*** *(3), 032602 (2015)*

II **Multiple scales and phases in discrete chains with application to folded proteins**
A. Sinelnikova, A.J. Niemi, J. Nilsson, M. Ulybyshev
Phys. Rev. E ***97*** *(5), 052107 (2018)*

III **RG smoothing algorithm which makes data compression**
A. Sinelnikova
preprint arXiv:1806.01663 (2018)

IV **Reproducibility in the unfolding process of protein induced by an external electric field**
Anna Sinelnikova, Thomas Mandl, Christofer Östlin, Oscar Grånäs, Maxim N. Brodmerkel, Erik G. Marklund and Carl Caleman,
accepted for publication in Chemical Science (2020)

V **Orientation before destruction. A multiscale molecular dynamics study**
Anna Sinelnikova, Thomas Mandl, Harald Agélii Oscar Grånäs, Erik G. Marklund, Carl Caleman and Emiliano De Santis
preprint arXiv: 2102.03145;
submitted to Biophysical Journal (2021)

VI **NMR refinement and peptide folding using the GROMACS software**
Anna Sinelnikova, David van der Spoel
ChemRxiv. Preprint. https://doi.org/10.26434/chemrxiv.13637819.v1;
submitted to Journal of Biomolecular NMR (2020)

Reprints were made with permission from the publishers.

My contribution

I The idea and proof that it is necessary to take into account the van der Waals attraction to find the proper collapsed phase. Participation in all discussion. Developed the Monte Carlo simulation software (from scratch). All results and their representation for the paper.

II Ensured that we were using a theoretically sound definition for the RG transformation for the polymer scaling that is used to define the phases. Developed the scaling procedure and the software for performing the necessary calculations (from scratch). Participation in all discussion. All the results and their representation for the paper.

III Everything.

IV Performed the simulations for the results in Figures 1 and 2 in the paper. Came up with the idea of the graph representation used in Figures 1 and 2. Participated in all discussions related to this and wrote the corresponding part in the manuscript.

V All the classical simulations. Participated in all discussions related to this part. Produced the corresponding results and their representation for the paper. Described it in the manuscript.

VI Developed the software from a previously existing code. All the simulations and results. Equal contribution to writing the manuscript.

Contents

1. Introduction

Proteins are biological macromolecules that play a crucial role in Life. Our health, behavior and mood entirely depend on them. They build our bodies and regulate the processes in them. Almost all drugs produced by humans target one protein or another. Thus the protein research is not just important but vital. At the time when this thesis is written, the question of the importance of biophysics is obvious. In 2020, humanity met the new world pandemics COVID-19, maybe the biggest since the Spanish flu pandemic one century ago [1, 2]. The efficiency of the vaccine and how fast it was made will define the whole world's future for years or tens of years ahead. The better we understand proteins and can do computer simulations, the faster and cheaper we can produce new medicines when it is needed.

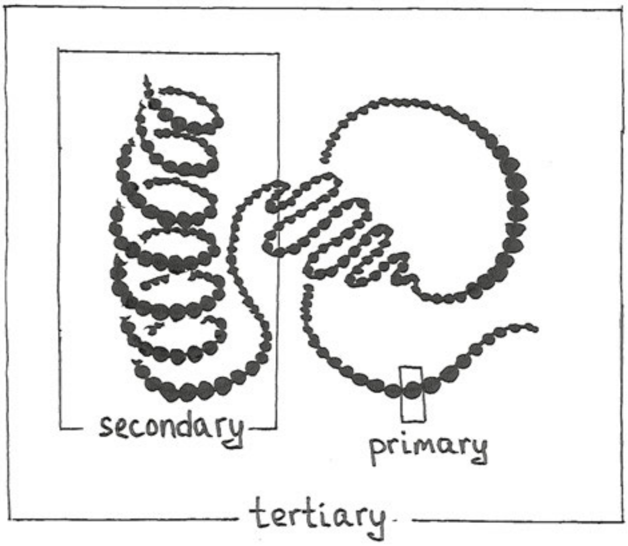


Figure 1.1. Structures of proteins.

The proper behavior of all living organisms on Earth depends on the appropriate conduct of all the proteins they consist of. And the latter depends not only on the chemical structure of the macromolecules but on their three-dimensional shape [3].

Scientists usually define four levels of structuring of proteins (Figure 1.1):

1. *Primary structure*: the sequence of amino acids. Every amino acid has one C_α atom with a specific side chain connected to it. There are only 20 different side chains, which define 20 amino acids responsible for all the variety in DNA encoded proteins.
2. *Secondary structure*: local geometrical patterns, which can be repeated along the protein. The most common are α-helices and β-sheets [4].
3. *Tertiary structure*: overall geometrical structure of the chain of amino acids (or polypeptide chain), which defines its functions.
4. *Quaternary structure*: a structure of several polypeptide chains if the protein has more than one chain.

The tertiary structure results from a process called *protein folding* [5]. The folding process is fully defined by the primary structure (and the environment) [3]. This means that by knowing the amino acid sequence, we should be able to predict the protein's shape. However, this problem appeared to be more complicated in practice, and the question of protein folding became a Holy Grail for tens of years for scientists until the 30th of November 2020, when it was finally found (maybe).

Before that historical date, the main tool for structure prediction was *molecular dynamics* (MD). It is the most "honest" way to simulate proteins. MD takes into account all physical forces in the system and thus provides the real dynamics. One of MD's problems is this "all physical forces". Strictly speaking, they are never possible to calculate exactly. The "art" of physics and the physicist is to choose the right model that considers everything that is important in a given problem and neglects everything that is not. The second problem with MD is the size of the systems. Proteins have thousands and millions of atoms plus thousands of atoms of water solution. So conventional MD is a so-called "brute force" algorithm. It relies on computation power and is limited by it.

Another way to simulate proteins is to use *coarse-graining methods* [6]. Instead of considering every atom, one can combine them into blocks and work with these blocks. It solves the problem with an enormous amount of particles to simulate but gives rise to an even more complicated question about which model to use.

Independently of what simulation tactics you use, you will meet another big conceptual problem. Apparently, a particular primary structure allows more than one possible chain shape, called *conformations*. *Native conformation* is a proper tertiary structure that provides the correct functioning of the concrete protein. Other conformations are the result of *misfolding* of the protein. Such molecule loses its native properties and can become toxic [7, 8]. There are many diseases caused by protein misfolding [9, 10]. The most famous are Alzheimer's and Parkinson's diseases [11], diabetes type II [12], cataracts [13], cystic fibroses [9] and some types of cancer [14].

The ultimate assumption made in MD is that the native conformation is a global minimum for the system. It seems very reasonable, but strictly speaking, do not have to be true. Of course, we know the general physical concept about the minimization of energy. But we also know that it must work only for closed systems, which one single protein cannot be. It can turn out that native conformation is a metastable state, with lifetime longer than the one for proteins. Indeed, let us not forget that biomolecules are parts of living organisms. They cannot live forever. Proteins are degraded continuously by the body and replaced by new ones. For example, the average lifetime of the proteins in a mouse brain is 9 days [15]. The protein with one of the longest observed lifetimes is human eye lens crystallin. Some crystallins live with us all our life and are never replaced [16]. At the end of our and their lives, crystallins are folded wrongly (that is why we all will have problems with lens transparency when we become old). More likely, this is a result of external factors. But we do not know this. Theoretically, the natural conformation of crystallin can be a metastable state with a lifetime of about 70 years.

When we do our simulations, we want to minimize the energy to find the global minimum, but why does Nature do the same? Maybe local minimum works well enough for her. Proteins are assembled in the body by the ribosome. And we do not know how exactly this process looks like and what laws ribosomes are governed by.

Coming back to the Holy Grail, which was found (maybe). What did happen on the 30th of November 2020? The two most influential scientific journals, “Science” and “Nature”, published news that Google’s artificial intelligence (AI) predicted a 3D structure of a protein from an amino acids sequence, and it was indistinguishable from the experimental result. In other words, it solved the protein folding problem for at least one specific protein. This is indeed “a game changer” as the authors claimed. But we have to be very careful with the results we get from AI. Any AI is trained on vast databases where the right answers are already known. After the training, the machine is ready to try to find an answer to an unsolved problem. AI works by recognizing similarities by its complicated neural network built at the training phase. But the machine does not “understand” any physics behind the phenomena. So there is no reason why we can claim that if one solution was found correctly, any other solution would be correct as well. This is a game of probabilities. AI cannot give us 100% assurance that every other answer will be correct, no matter how many were correct before. And here is a crucial difference with any physical method. If we found the correct model, then we have 100% guarantee that the answer is right. Because in the latter case, we found the underlying principles for the process, while in the former case, we compare the outcomes of those processes.

Regardless of what was said before, there is no question about developing AI methods for protein folding. They will have many practical applications. And together with experimental methods, it can become a robust tool for protein studies. However, as scientists, we do not only want to know the correct result but also to understand what leads to it. Thus no matter how good AI will fold the proteins, we should keep developing the physical methods in the first place.

In this thesis, we use different methods to study proteins.

We will start with a simple model of proteins: polymer chains of C_α atoms. For this simple model we present the effective Hamiltonian and investigate the phase diagram it provides by classical Monte Carlo simulations. Besides the collapsed phase and denatured phase, it can provide α-helices and several interesting crossovers. This is a topic of **Paper I** and Part II of this thesis. Part I is fully dedicated to classical Monte Carlo theory. We have to introduce it here to be able to talk about the main problem we encounter in Part II.

Further, in Part III we will try to solve the problem with phase definition for heteropolymers using renormalization group theory. This is the question we will naturally come to when we want to get the same phase diagram for heteropolymers instead of simple homogeneous chains. This work yielded **Paper II** and side **Paper III** about a new smoothing algorithm.

Both Parts II and III are based on original software written by me. It includes the main Monte Carlo code for polymer chains simulations PCMC (in C++) [17], the code for performing renormalization of polymer chains (in C++) [18], the plotter (in Python) [19] and the command line program for getting the data from the Protein Data Bank and converting it to another human-readable format which is used by all programs mentioned above (in C++) [20]. Everything is open-source and distributed under the Apache 2 license.

Part V of this thesis is dedicated to MD simulations. There we use the third-party software GROMACS [21] which is one of the most popular open-source tools for performing MD simulations. **Paper VI** will tell about our contribution to the GROMACS framework. We wrote a code in Python which converts experimental data from nuclear magnetic resonance to GROMACS format.

The last two studies focused on a concrete experimental technique, which is discussed in a separate Part IV among the other experimental methods used for protein structure recognition.

Paper IV is about the pathways of the unfolding of one concrete protein, ubiquitin, in a strong electric field. The last **Paper V** concludes the thesis with the research about the behavior of the same protein in a time-dependent strong electric field.

Part I:
Classical Monte Carlo

2. Games

The method I want to present in this chapter got its unusual name after the famous casino Monte Carlo in the Principality of Monaco. The idea belongs to Metropolis, who, together with Ulam, published the original paper in 1949 [22]. The crucial point of this algorithm is a random number generator. And probably the most widely known random number generator for all times is a roulette. And roulette is the most popular game in casinos. In Ulam's autobiographic book [23] he also mentions that Metropolis dedicated the name to his uncle, who liked gambling.

Anyway, the Monte Carlo method is closely connected to games; that is why we will also start with one described in Krauth's book [24].

It is remarkable that even though the method got its name 70 years ago, the game we will talk about is similar to experiments done by french mathematician Buffon in the XVIII century known as Buffon's needle problem [25].

2.1 Direct sampling

Let us start by drawing a closed figure with chalk on the ground. It can be any smooth shape like a circle, bean, two peanuts in a shell, etc. The rules are simple: use whatever you need, do whatever you want to measure the figure's area. At first sight, the problem seems to be very complicated. What would you need to perform the measurement? A ruler? A protractor? A flexible rod? How do you want to do it? Would you try to cover the area with squares and then try to approach the boundaries? Do you have enough will and time to integrate over the curve? Would you prefer to employ physics: you can try to build a small solid fence along the curve, pour a known amount of water, measure the liquid column's height, and finally find out the area. This method is messy but could work as well, see Figure 2.1.

Luckily, another solution exists, where you need only a ruler and one small stone like a pebble. The idea is presented in Figure 2.2. First, you should draw a square around the figure. It does not matter how you place the figure inside; the important thing is that the whole figure is within the square. Then you measure the square area, which is just a^2, where a is the side length. Now let us *randomly* throw the pebble into that square and count: 1) N = all the times it falls within the square, 2) M = all the cases when it falls within the drawn figure. The crucial point is that the probability of hitting the figure in this game

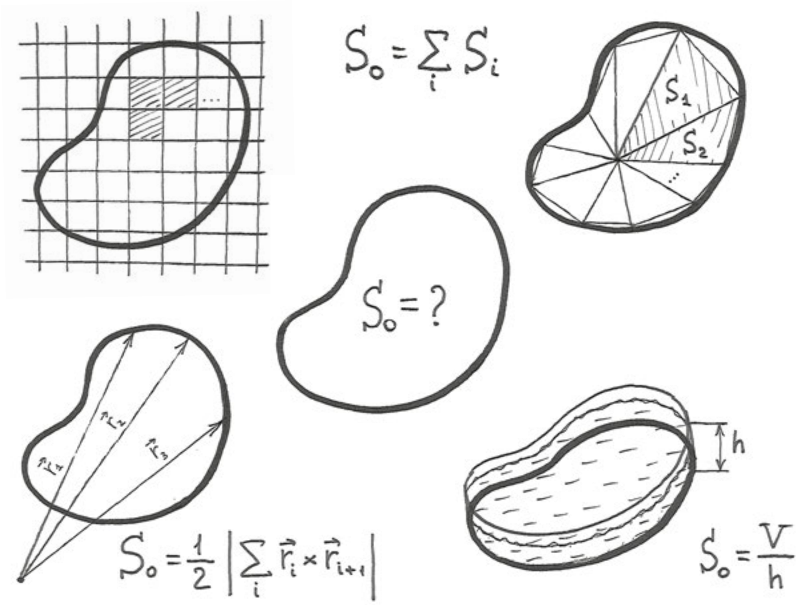


Figure 2.1. Different methods to measure the area of the drawn figure.

is proportional to its area. This means that the ratio M/N should be equal to the areas of the figures:

$$\frac{M}{N} = \frac{S_\circ}{S_\Box} = \frac{S_\circ}{a^2} \quad , \tag{2.1}$$

where the area of the drawn figure is denoted as $S_\circ$. Knowing M, N and the square's area a^2 we can easily find the unknown area of the drawn figure.

So with this very simple game of throwing a stone, we can solve the problem, which is very complicated to solve otherwise. As one can see, the crucial

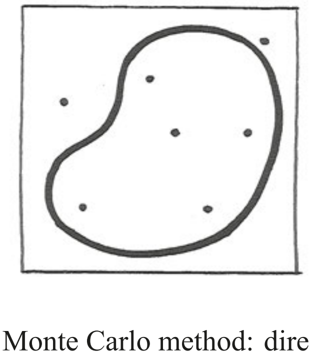


Figure 2.2. Monte Carlo method: direct sampling.

point here is the randomness of tosses and their equality. Only in this case, the probabilities are proportional to areas, and Eq. (2.1) is valid.

The procedure we performed is the Monte Carlo algorithm. More precisely, we should call it *direct sampling*, in contrast with the method we will present next. It is essential for direct sampling that all our tosses were independent of each other.

2.2 Markov Chain sampling

Now imagine another situation. Somebody drew some closed, smooth figure around you, which is so big that you can freely walk inside. You even cannot see the overall shape because of the size. The goal is the same: whatever you need and do, just measure the drawn figure's area.

This task is more difficult than the previous one because of the larger size. Our former ideas from Figure 2.1 was hard to implement for the first version of the game, but now we can expect even more troubles because the complexity of the problem grows with the size of the figure's area. What about our last successful solution: a simple Monte Carlo algorithm with tossing a pebble? Let us try to do the same trick again.

We again draw a square (or any figure with a known area) containing the whole drawn figure. You start to throw a stone and soon will realize that the tosses will be biased toward the spot you stand on. It happens because the field is too big. Even if you are physically able to throw a stone from one side of the playground to the other, you cannot reproduce the uniform distribution of tosses among the whole field. You would be prone to make "easier" throws, which require less energy from you. And if you think you can overcome this by keeping in mind the previous statement, you are wrong again. Humans are very bad at randomization and understating randomness. And as we understood from the previous part, the proper random distribution is crucial for Monte Carlo to work. Okay, your next idea more likely would be to somehow move around the field to not always stand on the same sport. That is an excellent thought which brings us closer to the correct solution!

The Monte Carlo algorithm we will use in this case is called *random walk*. It works as follows. You throw a pebble inside the square with closed eyes (the latter we need for the reason we mentioned: humans are imperfect at randomness, especially when thinking about it). Then you go to the place where it landed, pick it up, close your eyes and throw the pebble again in a random direction. The rules now are this: count 1) N = number of all throws of the stone, 2) M = events when the stone lands within the original figure. Continue doing this a lot more times.

There is, however, a small but essential nuance here. Since you throw the stone with closed eyes and in a random direction, it may land outside the square. The first idea would be to exclude these cases and pretend that it has never happened. That is what we did in the previous game. There we only counted the cases when the stone landed within the square and ignored the opposite outcome. But we cannot do the same here! For the reasons we will explain later, you should count this case as a throw to the same spot you are standing now. So you should remember your spot and bring the pebble back there, increasing N by one and M if you are inside the figure. This sounds counter-intuitive, but in the next section, we will make sense of it.

After playing this game for a long time, you can again use Eq. (2.1) and find out the desired area.

Notice, this time, the trials are not independent. Each throw depends on the previous one and only on the previous one. Such a sequence of events is called a *Markov chain* after Russian mathematician Andrey Markov (1856 — 1922). This is a very important notion in statistics. Markov chains have interesting properties, which we will discuss further. This name will appear in this thesis many times, even beyond the Monte Carlo framework.

Monte Carlo algorithms with Markov Chains sampling are called *Markov Chain Monte Carlo (MCMC)*

2.3 Boundary problem

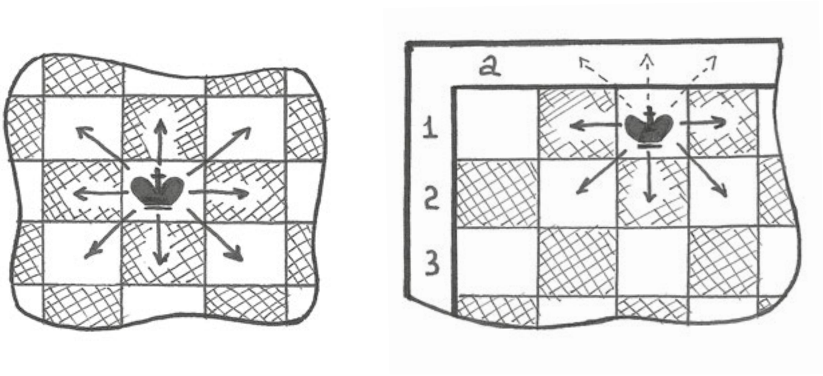


Figure 2.3. Boundary problem.

In the previous section, we had a special rule when the stone is thrown outside the square. Now we will explain why we had to count those cases as a throw on the same spot. The problem we have here is a problem with boundary conditions. Let us take a look at another game: chess with a single king on the board Figure 2.3. If we randomly move the king (according to the chess rules: one step a time in any direction, and in our case, we allow to take a step outside the chessboard), then we come to the discrete version of our previous game.

If the king stands on, say, the c3 square, it has 8 possible moves. Or the other way around: there are 8 ways how the king can come to the c3 square. The same we can say about the squares d2, f5, ... and any of the squares of the board. The different situation happens when we talk about the edge or *boundary* squares. There we have always less than 8 neighbor squares. C1, for example, has only 5 neighbors. It means that there are only 5 possible moves that can lead the king to this square. But for the c3 there are 8 such moves. This means that we are biased toward the inner squares now. But as we mentioned above, the crucial thing for us in the game is equally distributed sampling. The solution we already described, but let us repeat it once more. If the king stands on c1 then there are $8 - 5 = 3$ moves, which lead the king outside the board. So if we convert these 3 moves into moves that lead to the c1 square itself, then we come to 8 possible ways to come to this square. It means that the king will be in c1 with the same probability as in any inner square. This rule should work for all boundaries: if the king moves outside the chessboard, we should consider it as a move from the square to the same square.

That was how we solved the boundary problem and made the probability distribution equal within the whole playing area.

3. Games over

In this chapter, we will see that everything we did in chapter 2 has a wonderful mathematical description.

3.1 The probability density function

In the game from the previous chapter, we aimed to calculate the area of the figure. This is a two-dimensional problem. Let us for the simplicity consider a one-dimensional case first. And soon I will show that all our reasoning works for 2D as well. From a mathematical perspective, our game in 1D means to calculate an integral

$$S = \int_a^b f(x)dx \; ; \qquad b > a \quad , \tag{3.1}$$

where $f(x)$ is a function of the boundary curve[1]. To evaluate this integral we randomly tossed pebbles. Whenever we talk about a random process, we always can associate the *probability density function* $p(x)$ with it. A probability density function by definition should satisfy two conditions on the segment where it is defined: 1) to be not negative on the interval, 2) the full probability to find the value on the interval should be 1:

$$p(x) \geq 0; \tag{3.2}$$

$$\int_a^b p(x)dx = 1. \tag{3.3}$$

In the case of our simple game, this probability distribution is a flat function: all the outcomes were equally possible: $p(x) = const$.

The area integral in Eq. (3.1) can be rewritten as

$$S = \int_a^b f(x)dx \equiv \int_a^b \frac{f(x)}{p(x)} p(x)dx \quad , \tag{3.4}$$

which is by definition *the expectation value* for the value $f(x)/p(x)$.

[1] For example, let a vertical line with the given x-coordinate cut the boundary from $y = -\infty$ to $y = \infty$. Let $\{y_m\}$ be the collection of points where the line crosses from the inside to the outside of the boundary, while $\{y_n\}$ are the collection of points where the line passes from the outside to the inside. Then $f(x) = \sum_m y_m - \sum_n y_n$ is a valid choice of such function.

Summarizing what we said we can formulate our problem as following:

$$z(x) \equiv \tfrac{f(x)}{p(x)} \tag{3.5}$$

$$S \equiv \int_a^b f(x)dx = \int_a^b z(x)p(x)dx \equiv \langle z \rangle \tag{3.6}$$

$$p(x) : \{p(x) \geq 0; \int_a^b p(x)dx = 1\} \text{ on } [a, b]. \tag{3.7}$$

3.2 The Law of Large Numbers

We just have learned that instead of taking the integral, we can calculate some quantity's expectation value. But how can we do it in practice? When we calculated the area in the game, we counted all the throws as N and then successful throws (when the stone landed within the drawn shape) as M and took their ratio M/N. Let us introduce the function z_i which is

$$z_i = \begin{cases} 1\ , & \text{if } i\text{-th stone is within the figure} \\ 0\ , & \text{otherwise} \end{cases} . \tag{3.8}$$

If i is a trial's number, then M can be expressed as a sum over i of our new function

$$M = \sum_{i=1}^{N} z_i. \tag{3.9}$$

And the area of the figure $S_\circ$ can be found as

$$\frac{S_\circ}{S_\square} = \frac{M}{N} = \frac{1}{N}\sum_{i=1}^{N} z_i \equiv \bar{z} \quad , \tag{3.10}$$

where according to the rules of the game, $S_\square$ – the area of the square around the figure – is known. Thus in the game, we calculated the arithmetic mean value of the function z. And the throws were sampled from the uniform distribution $p(x)$.

Now we come to the central point part of the whole Monte Carlo idea – the *Law of Large Numbers* [26] which says: *in a limit of infinite number of samples N, the arithmetic mean value almost always converges to the expectation value*

$$\lim_{N\to\infty} \bar{z} = \langle z \rangle. \tag{3.11}$$

"Always almost" stands there to stress that there are some very special cases when this will not work, but this is out of the scope of this thesis.

Finally, the Law of Large Numbers binds together the integral in Eq. (3.1) which we want to calculate and the arithmetic mean in Eq. (3.9), which we can obtain by throwing random numbers according to a certain distribution.

The last step we will need to do for completing the example with the game is to show that z_i in Eq. (3.9) is indeed the same function as $z(x)$ in Eq. (3.5).

3.3 Why did it work?

Let us come back to two dimensions. Without loss of generality, we will consider a circle as the figure of interest. The area of a circle with diameter 1 can be found as

$$S = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} F(x,y)dxdy \equiv \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \theta(\frac{1}{4} - x^2 - y^2)dxdy = \pi^2 \ , \quad (3.12)$$

where we used Heaviside step function defined as

$$\theta(x) = \begin{cases} 1\ , & \text{if}\, x \geq 0 \\ 0\ , & \text{if}\, x < 0 \end{cases} . \quad (3.13)$$

The probability density function in Eq. (3.7) for tosses of stones was independent of the coordinates: $p(x,y) = p$. The value of the constant should be found from normalization

$$\int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} p\, dxdy = 1 \Rightarrow p = 1. \quad (3.14)$$

Our $z(x,y)$ function defined in Eq. (3.5) is now just Heaviside function

$$z(x,y) \equiv \frac{F(x,y)}{p(x,y)} = \theta(\frac{1}{4} - x^2 - y^2). \quad (3.15)$$

How can we calculate the arithmetic mean of $z(x,y)$? According to the Heaviside function definition, it should be equal to 1 when $x^2 + y^2 < 1/4$ and 0 otherwise. This condition exactly describes the rule for the game we consider: we counted M as a number of events when the stone landed within the circle. By dividing M over the total number of events, we come to the arithmetic mean for function $z(x,y)$, which is equal to its expectation (due to the Law of Large Numbers), which is equal to the original integral S

$$\overline{z(x,y)} = \frac{1}{N} \sum_{i=1}^{N} z(x_i, y_i)$$

$$\qquad \| \text{ by the Law of Large Numbers}$$

$$\langle z(x,y) \rangle = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} z(x,y)p(x,y)dxdy = \int_{-\frac{1}{2}}^{\frac{1}{2}} \int_{-\frac{1}{2}}^{\frac{1}{2}} \theta(\frac{1}{4} - x^2 - y^2)dxdy \equiv S. \quad (3.16)$$

And that is what we wanted to get.

We showed the pictorial example of the Monte Carlo process and explained why it worked. Now we can talk about the optimization of the method and consider the sampling procedure more carefully.

3.4 Precision

In the formulation of the Law of Large Numbers, Eq. (3.11), we say that it works in a limit of an infinite number of samplings. This can never be the case in practice: our N is always finite. Then the question is: how big N we should take to have the correct answer. And the answer is: we will never get a correct answer, but we can approach the correct answer with any given precision.

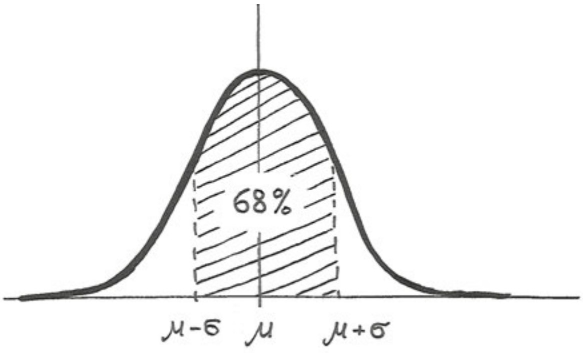


Figure 3.1. Almost 68% of the area under a normal distribution belongs to the one-sigma region.

We can claim this because of another very important law in statistics: the *Central Limit Theorem* [26]. It can be formulated like this: *The averages of a big number of independent random values almost always tends to a normal distribution.* No matter from what distribution the random values are picked from, the averages tend to a normal distribution anyway. The phrase "almost always" again indicates that it is not an absolute law, and there are some peculiar cases when it can be violated.

This gives us the estimation of the error of $\overline{z}$ as *the standard deviation of the mean*

$$\sigma(\overline{z}) = \sum_{i=1}^{N} \frac{\sigma(z_i)}{\sqrt{N}} \equiv \sum_{i=1}^{N} \sqrt{\frac{1}{N(N-1)}(z_i - \overline{z})^2}. \qquad (3.17)$$

One sigma gives the confidence of 68%, while 3 sigma gives 99% and 4 sigmas 99.99%. So the interval $[\overline{z} - \sigma(\overline{z})\,, \overline{z} + \sigma(\overline{z})]$ contains the true answer with probability almost 70% (Figure 3.1).

With increasing N, the value of the standard deviation $\sigma(z)$, according to the Central Limit Theorem, tends to a constant value, which is a dispersion of some normal distribution. Hence the error of our mean value $\overline{z}$ drops as $\sqrt{N}$. So if you want to decrease the error by two times, you should increase the number of samples by four times.

3.5 Importance sampling

We always considered the uniform distribution $p(x)$ so far, because the game is a trivial example of the Monte Carlo technique. In most other cases, the choice of $p(x)$ to be uniform is not optimal. Thus we come to the idea of *importance sampling*, where we try to use more samples for more "important parts" of the region and less samples for less important. Let us see how it works in practice. Our integral of interest is

$$I = \int_a^b f(x)dx. \tag{3.18}$$

Let $f(x)$ be factorizable like

$$f(x) \equiv \varphi(x)\psi(x). \tag{3.19}$$

We can again multiply and divide the integral by some $p(x)$ as we did in Eq. (3.4) or we can let one of the subfunctions in $f(x)$ be a probability density function. This is kind of a "natural way" of doing things.

It becomes more clear if we consider thermodynamic problems, where you have to find the average parameters over some ensemble. It can be pressure, energy, temperature or any specific value. Any thermodynamic value $\mathcal{A}$ is defined as an expectation value

$$\langle \mathcal{A} \rangle = \int \mathcal{A}(x)p(x)dx. \tag{3.20}$$

The concrete form of the probability density function $p(x)$ depends on the system. For example, a gas in a thermostat is described by the *canonical Gibbs distribution*. The probability that the system will be found in a microscopic state with energy E_i is

$$p_i = \frac{1}{Z}e^{-E_i\beta} \quad , \tag{3.21}$$

where we use a common notation for $\beta = 1/k_BT$; k_B is the Boltzmann constant, and T is the temperature. The normalization factor Z is called the *partition function*

$$Z = \sum_i e^{-E_i\beta} \equiv \sum_{E_i} W_i e^{-E_i\beta}. \tag{3.22}$$

W_i is a weight function for the state with energy E_i, i.e. how many microscopical states provide the given energy. This is the statistical meaning of entropy:

$$S_i = k_B \ln W_i \quad , \tag{3.23}$$

then

$$Z = \sum_{E_i} W_i e^{-\beta E_i} = \sum_{E_i} e^{-\beta E_i + \ln W_i} = \sum_{E_i} e^{-\beta(E_i - TS_i)}. \tag{3.24}$$

Using the definition of free energy for the system with a fixed number of particles

$$\mathcal{F} = E - TS \quad , \tag{3.25}$$

we can finally rewrite the partition function as

$$Z = \sum_{E_i} e^{-\beta \mathcal{F}_i}. \tag{3.26}$$

Converting the summation into an integral the thermodynamic value in Eq. (3.20) can be expressed as

$$\langle \mathcal{A} \rangle = \frac{\int \mathcal{A}(x) p(x) dx}{Z} = \frac{\int \mathcal{A}(x) e^{-\beta E(x)} dx}{\int e^{-\beta E(x)} dx} = \frac{\int \mathcal{A}(x) e^{-\beta E(x)} dx}{\int e^{-\beta F(E)} dE}. \tag{3.27}$$

And as far as we already know, Monte Carlo will estimate this integral as the average for the value $\mathcal{A}$

$$\overline{\mathcal{A}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{A}_i \quad , \tag{3.28}$$

where $\mathcal{A}_i$ is distributed according to Eq. (3.21).

This is actually a great result! We have not only calculated the average. We got our system in the equilibrium state (at average!). Indeed, from Eq. (3.21) one can see that the most likely states are ones with lower energies. However, it is very important to remember that not every configuration is in the energy minimum. The statement is valid only from a stochastic point of view. What can it give to us? Once you have found the configurations with the correct distribution, you can calculate not only Eq. (3.28), but any thermodynamic value you want, without the need to run a new Monte Carlo simulation like you would do for calculating regular integrals. This fact makes Monte Carlo a very powerful tool for physical calculations.

3.5.1 Ergodicity

Let us come back to the game from the previous chapter.

Imagine that now we have a big obstacle, maybe a lamp post in the center of the game field like on the left side of Figure 3.2. The cross-section of the obstacle is known (which means we know its area). Now, if we perform the same algorithm of random walk form section 2.2, sometimes the stone will bounce from the post. This kind of event should be calculated in a special way that we already mentioned and will discuss in more detail in the next section. The point we want to make now is the following. Look again at Figure 3.2. If your pebble is in the point a, then there is no way to come to point b in one step, even if it was possible to do without the lamp post.

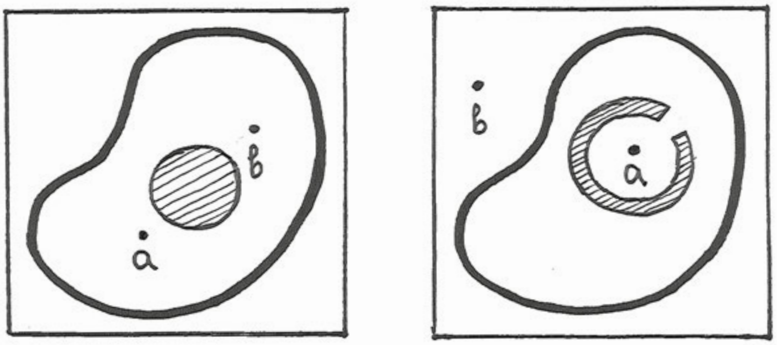


Figure 3.2. Obstacles on the game field.

The situation can be even worse if the obstacle has a more complicated shape, for example, as on the right side of Figure 3.2. If both obstacles to the left and to the right are of the same area, then the drawn picture's resulting areas should also be the same. However, in the case of the wall, it is hard to get inside, and once you are there, it is hard to go out. Now the trajectory of the random walk from point a to b can be very long.

Thus instead of having a uniform distribution like it is supposed to be, the probability will be biased, depending on the obstacle shape. The weight of states inside the obstacle will be artificially increased because we are stuck there, but not because it reflects any physics.

There are no problems from the theoretical point of view: the statement about correct stationary distribution for a MCMC is made for an infinite time. Everything will work for us in infinite time as well. But in a real experiment, we can only approach infinity. It is intuitively clear that in the last example, the time we have to spend performing Monte Carlo is longer than in the situation with the round obstacle.

Here we met the difference between theory and practice, which may lead to an incorrect answer. This is the most dangerous problem of MCMC called the *ergodic problem*. Unfortunately, there is no universal and straightforward solution to it, so each case should be considered separately.

Finally, we can summarize: *on the one hand, MCMC allows you to get the answer without knowing the system's overall profile, but on the other hand, you have to know the details of your configurational space to not run into ergodic problems.*

3.5.2 Detailed balance

When we talk about the random walk, we always face the boundary problem. In section 2.3 we showed with a simple example that a mistreatment of boundaries would lead to a wrong stationary distribution for MCMC and hence to the wrong answer. There we talked about the edges of the game field or chessboard.

In the previous section, we met another kind of boundaries: obstacles. From the mathematical point of view, there is no difference between those two cases. Both kinds of boundaries define the domain of the function. Nevertheless, practically they are different: one is a drawn line, and the other is a solid obstacle. How should we treat the situation when the stone bounced from the lamppost? We can do the same as we did with the playground boundaries: take the stone and put it in the same position as before the last throw and increment the number of throws by one. However, it is not the only correct solution. Another way is to do nothing! It might seem strange initially, especially because we proved that this tactic was wrong for our flat boundaries. However, this time, it will lead us to the correct stationary distribution, which means the correct answer. This happens because of the amazing physical principle: the angle of incidence is equal to the angle of reflection. What we want to do, after all, is to *balance* the distribution. The way how exactly we will do it is not important. For instance, if in the example in Figure 2.3 every time when the king moves onto the square c0 (i. e., outside the chessboard) from c1, we would put it on b1 instead of returning it to c1 and other way around, then there would be no problems.

The formal rule for what we just discussed is called the *detailed balance condition*: *The probability to come to the state B through the state A should be equal to the probability to come to the state A through the state B.* In the case of the Markov chain, when the probability depends only on one previous step, the detailed balance condition can be written as

$$P(A)P(A \to B) = P(B)P(B \to A), \tag{3.29}$$

where $P(A)$ is the probability of the system to be in state A and $P(A \to B)$ is the probability of the transition. Notice that the probability of transition is important, but the full probability also includes the probability of appearing in the initial state. This is very clear in our example with chess in Figure 2.3. The probability of transition c2→c1 is the same as vice versa: c1→c2, so in Eq. (3.29) right term from each side will be canceled out. But the probability of the king to be in the square c1 is not the same as to be in the square c2, as we showed in section 2.3 and Figure 2.3. To make these probabilities equal, we have to set special rules for the boundaries.

Any solution which satisfies the detailed balance condition is a correct one.

3.6 Algorithms

We will present two MC algorithms of the importance sampling for solving thermodynamic integrals which we discussed in section 3.5

$$\langle \mathcal{A} \rangle = \frac{1}{Z} \int \mathcal{A}(x) e^{-\beta E(x)} dx. \tag{3.30}$$

By solving this problem we will also get the thermodynamic equilibrium states of the system as ones that provide the lowest energy and hence are more likely to appear.

3.6.1 Heat Bath

The first MC algorithm we consider is a trivial (from a theoretical, but not practical point of view) example of importance sampling (see sec. 3.5). The probability of every new stage b is calculated from the probability

$$P(b) = \frac{1}{Z} e^{-\beta E(b)} \quad , \tag{3.31}$$

just the same as we discussed in section 3.5. Physically generating that process leads to the system's thermodynamic equilibrium, and that is how the algorithm got its name.

This described process is not Markovian in the sense that all the steps are completely independent of each other (strictly speaking, it is an extreme case of the Markovian process). This is the reason for the most significant advantage of the Heat Bath algorithm: no problems with ergodicity. The disadvantage is that you should know your potential's full profile, i.e., $E(x)$ for the whole domain of x. For most real physical problems it is not the case, so this method is not possible to use.

3.6.2 Metropolis

Another solution was introduced by Metropolis *at el.* in 1953 [27] and got his name. Metropolis algorithm is a basic MCMC algorithm. The idea is to prefer the states with lower energy at every MC step. Thus, if a new state's energy is lower than the current state's energy, the new state is accepted with probability 1. And if the energy of the new state is higher than that of the current state, then the new state is accepted with the probability proportional to the energy difference between the new state and the current one. If we denote the current state as a and the new randomly generated state as b then

$$P(a \to b) = \begin{cases} 1 \, , & \text{if } E(b) \leq E(a) \\ e^{-\beta(E(b)-E(a))} \equiv e^{-\beta \Delta E} \, , & \text{if } E(b) > E(a) \end{cases} . \tag{3.32}$$

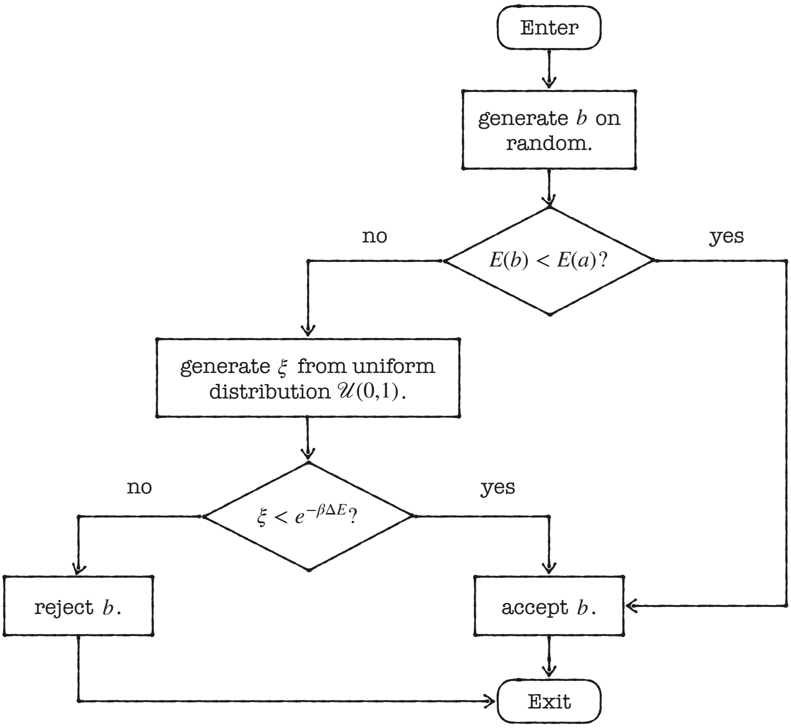

Enter
generate b on random.
$E(b) < E(a)$?
no
yes
generate ξ from uniform distribution $\mathscr{U}(0,1)$.
$\xi < e^{-\beta\Delta E}$?
no
yes
reject b.
accept b.
Exit

Figure 3.3. Acceptance-rejection algorithm.

From the algorithmic point of view, this leads to the famous *acceptance-rejection sampling* which is shown in Figure 3.3.

N.B. 1 You could note that there is no normalization factor in the probability of transition. It is the great advantage of the Markovian character of the process. All we want from our probability is to be positive, not greater than one, and to fulfill the detailed balance condition in Eq. (3.29). The probability we are working with here is a transition probability $P(a \to b)$, because every new state is generated from the previous one. So if we now express this probability from Eq. (3.29)

$$P(a \to b) = \frac{P(b)P(b \to a)}{P(a)} \quad , \tag{3.33}$$

it is clear that normalization factors of the transition probabilities will be cancelled out since they are the same.

We also do not need the normalization factor to keep the probability within the range $[0, 1]$. This is fulfilled automatically because ΔE in the exponent in

Eq. (3.33) is always positive (when it is negative, you accept the configuration with probability 1) as well as temperature in β.

N.B. 2 At the first step of the algorithm in Figure 3.3 one should generate b at random. There is no rule for what the distribution $P(b)$ should be. In general, you can use any distribution you like. But for practical reasons for finding the minimum, the normal distribution is the most convenient, unless you have some insight about your system, which can be useful here. For example, a uniform distribution will converge too slow. Just imagine that you are already close to the minimum, then if you generate a new b at random from a uniform distribution, the acceptance rate (the ratio of accepted configurations to all tries) will be very low, so the configuration will be almost never updated, and hence approach the minimum very slowly. Some more complicated distribution than a normal one can slow down the performance without giving any gain.

N.B. 3 You might notice that the normalization for $P(b)$ is also not needed because it will be canceled out in Eq. (3.33), but this time you might want to keep it to force $P(b)$ falls in the range $[0, 1]$.

N.B. 4 Another thing to pay extra attention to is that in the flowchart in Figure 3.3 when the new state b is rejected, you do not go into the loop where you generate new b and go through the whole algorithm again until b finally is accepted. Rejection of b meant that the new state is your old state; in our case, the new state is a. This is a reflection of the detailed balance condition. In this case, rejection is the same thing as throwing the stone outside the game field in our example from section 2.2.

N.B. 5 Metropolis algorithm is not really a random walk algorithm. It can tunnel through the obstacles if the dispersion of $P(b)$ is large enough to allow the stone to appear from the other side of the barrier.

N.B. 6 Thus, MCMC does not describe the real dynamics of the system. Real physical objects in classical physics cannot tunnel through potential walls higher than their energies. The thermalization process in MCMC is not physical. The only thing we can rely on in this sense is the final stage – the system in the thermal equilibrium state.

3.7 Simulated annealing

There exists a special technique for a more efficient search of the thermodynamic minimum. The physical observations show that the slow cooling down of the system brings it closer to the equilibrium state than instant cooling. The idea to use this fact for optimization problems was proposed in the early 1980s [28]. The analyses show that the system will come to the minimum if the temperature steps are small enough.

The ground state will be found with probability one only in a limit of infinite time.

With this, we conclude the dive into Monte Carlo theory and are ready to move on. After presenting the necessary physical background in the next chapter, we will see how powerful MC is. And at the same time, we will see how dangerous MCMC can be.

Part II:
Phase Diagram for Homopolymers

4. Theory

This thesis is dedicated to simulations of proteins. Let us start from approaching the proteins with a polymer model and investigating its thermodynamic properties. For this we will use everything we just learned about Monte Carlo method in the previous chapter.

4.1 Geometry

The polymer chains we will consider is a freely joined model with rigid solid links. For describing this kind of model one needs two parameters for each site according to two degrees of freedom of each monomer. The most trivial way is to use coordinates of the sites with constraints for the connections between the vertices. This is not a convenient solution because it leaves us with three parameters {x,y,z}, while we know that there should be only two independent values. Traditionally for the description of polymer chains, one uses two Euler angles. One is an angle between two adjacent links. The other fixes the rotation degree of freedom left: the rotation in the plane perpendicular to the adjacent link.

However, we will go another way.

4.1.1 Frenet frame

The Frenet formulas are used for describing the kinetics of a particle moving along a smooth continuous trajectory in space [29]. In our case, we will use the same formulation to describe multiple particles along a chain at a single instance in time, rather than the position of one particle over time. However, we will meet an issue: polymer chain is a discrete curve, but the Frenet description is developed for the continuous case. This problem is solved in the paper by Hu *et. al* [30], and we will follow it to introduce a discrete version of the Frenet formulas.

First of all, we will associate an orthonormal frame with every chain site in the following way. Let $\mathbf{r}_i$ be a radius vector of the i-th vertex of the chain. Then the first vector of the frame for the i-th site is a regular *tangent vector*

$$\mathbf{t}_i = \frac{\mathbf{r}_{i+1} - \mathbf{r}_i}{|\mathbf{r}_{i+1} - \mathbf{r}_i|}. \tag{4.1}$$

The second vector is called the *binormal vector*

$$\mathbf{b}_i = \frac{\mathbf{t}_{i-1} \times \mathbf{t}_i}{|\mathbf{t}_{i-1} \times \mathbf{t}_i|}. \tag{4.2}$$

The third vector is defined as a vector product in a way that $\{\mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i\}$ constructs a right-hand frame:

$$\mathbf{n}_i = \mathbf{b}_i \times \mathbf{t}_i \tag{4.3}$$

and has the name *normal vector*.

We can introduce two angels like shown in Figure 4.1. The discrete *bond angle* or *curvature*

$$\kappa_i \;\equiv\; \kappa_{i+1,i} \;=\; \arccos\left(\mathbf{t}_{i+1} \cdot \mathbf{t}_i\right) \quad , \tag{4.4}$$

and the discrete *dihedral angle* or *torsion*

$$\tau_i \;\equiv\; \tau_{i+1,i} \;=\; \mathrm{sgn}[(\mathbf{b}_{i-1} \times \mathbf{b}_i) \cdot \mathbf{t}_i] \times \arccos\left(\mathbf{b}_{i+1} \cdot \mathbf{b}_i\right). \tag{4.5}$$

The bond angle κ is the same as one of the Euler angles: the angle between two adjacent links. The physical meaning of both angles is the following:

- *curvature* κ shows how the chain deviates from being straight;
- *torsion* τ shows how the chain deviates from being planar.

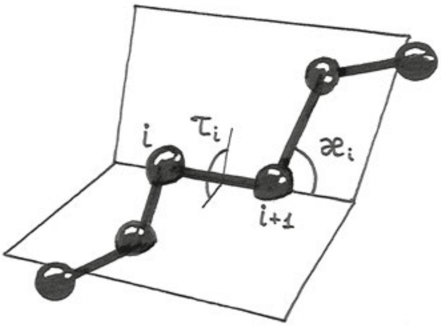


Figure 4.1. Two degrees of freedom for each link of the polymer chain: bond angle κ and torsion angle τ.

If we know all the angles $\{\kappa_i, \tau_i\}$ we can iteratively find all the frames $\{\mathbf{t}_i, \mathbf{n}_i, \mathbf{b}_i\}$ (except the first one, which can be chosen arbitrarily) through the *discrete Frenet equations* [30]

$$\begin{pmatrix} \mathbf{n}_{i+1} \\ \mathbf{b}_{i+1} \\ \mathbf{t}_{i+1} \end{pmatrix} = \begin{pmatrix} \cos\kappa\cos\tau & \cos\kappa\sin\tau & -\sin\kappa \\ -\sin\tau & \cos\tau & 0 \\ \sin\kappa\cos\tau & \sin\kappa\sin\tau & \cos\kappa \end{pmatrix}_{i+1,i} \begin{pmatrix} \mathbf{n}_i \\ \mathbf{b}_i \\ \mathbf{t}_i \end{pmatrix}, \tag{4.6}$$

and the polymer chain can be built from tangent vectors

$$\mathbf{r}_k = \sum_{i=0}^{k-1} |\mathbf{r}_{i+1} - \mathbf{r}_i| \cdot \mathbf{t}_i. \tag{4.7}$$

4.1.2 Gauge invariance

As one could note, there is no information about normal vectors $\mathbf{n}_i$ and binormal vectors $\mathbf{b}_i$ goes in Eq. (4.7) which describes the geometry of a chain. The only physical part is the tangent vectors $\mathbf{t}_i$. Two normal vectors can be freely rotated in the plane perpendicular to the tangent vector. So the transformation

$$\begin{cases} \mathbf{n}_i \to -\mathbf{n}_i \\ \mathbf{b}_i \to -\mathbf{b}_i \end{cases}, \tag{4.8}$$

which corresponds to the rotation by 180° will keep the chain intact.

It can be shown [30] that this transformation

$$\begin{cases} \mathbf{n}_k \to -\mathbf{n}_k & \text{for all } k \geq i \\ \mathbf{b}_k \to -\mathbf{b}_k & \text{for all } k \geq i \end{cases} \tag{4.9}$$

will act on the angles in the following way:

$$\begin{cases} \kappa_k \to -\kappa_k & \text{for all } k \geq i \\ \tau_i \to \tau_i - \pi \end{cases}. \tag{4.10}$$

Note that torsion should be changed only once at the i-th site while curvature changes the sign of all the following sites.

Originally in the description with $\{\kappa, \tau\}$ angles, all bond angles κ are supposed to be positive. However, the Frenet equation (4.6) does not forbid negative values for κ. Extension of κ to the whole region $[-\pi, \pi]$ will lead to double-counting of every direction if $\tau \in [0, 2\pi]$. It means that there exists $\mathbb{Z}_2$ symmetry between positive and negative values of κ. The transformation in Eq. (4.10) describes exactly that symmetry.

When we come to the description of the model's Hamiltonian, this gauge invariance will play a crucial role.

4.2 Physics

We will follow the description given in Grosberg's book [31] and show how one can get different thermodynamic phases in the model of the polymer chain we presented with van der Waals forces.

4.2.1 Ideal chain

The most straightforward physics of the freely jointed chain one can think about is just a chain without any interactions. This model is almost the same as the ideal gas model, except that there are fixed distances between neighboring particles. That is why we shall call this case an *ideal chain*.

What is the ground state of the ideal chain? At first sight, one can think that it is a straight line. Is it true?

To answer this and any other question about the chain's physics, we need to introduce a macroscopic parameter. All we know about the chain is only its geometry, so it is reasonable to use some geometrical variable. The size of the chain or how much space it needs can be associated with the *radius of gyration*

$$R_g = \sqrt{\frac{1}{2N^2}\sum_{i,j}^{N}(\mathbf{r}_i - \mathbf{r}_j)^2} \quad , \tag{4.11}$$

where $\mathbf{r}_i$ is a radius vector of the corresponding atom.

Another quantity for measuring the "size" of the chain can be the average end-to-end distance. It must be the statistical mean because you can imagine, for example, the chain banded into the circle. In this case, the end-to-end distance will be equal to zero while the chain occupies non-zero space. However, statistically, this state is very unlikely.

In all situations we will consider in this thesis, the end-to-end distance would give the same results as the radius of gyration [31]. However, the latter is more stable in extreme cases since there is already an averaging along the chain itself. Nevertheless, we will drop the index g in this chapter to stress that it is not essential how the size is calculated as far as it is a statistical mean.

Now we are ready to come back to the question about the ground state. The situation when the chain is entirely straight provides the maximum possible end-to-end distance. There is no other state which will give the same value for it. In other words, the entropy of this state is minimal. But as we know from thermodynamics, the entropy should tend to its maximum when in the equilibrium state. So even if we do not know the ground state yet, with very general ideas, we just proved that it is not the straight line.

The ideal polymer chain is a Markov chain. This notion we brought up in section I. Indeed, the position of the next segment depends only on the position of one previous link. Using the Central Limit Theorem, one can show that the probability distribution for R_g has the Gaussian shape when the number of segment $N \gg 1$. And then one can find the expectation value for the radius of gyration [31]

$$\langle R^2 \rangle = Nl^2, \qquad \text{for } N \gg 1, \tag{4.12}$$

where l is the segments' length. Eq. (4.12) thus describes the ground state of the ideal chain. This state is called a *Gaussian coil*. The converse statement

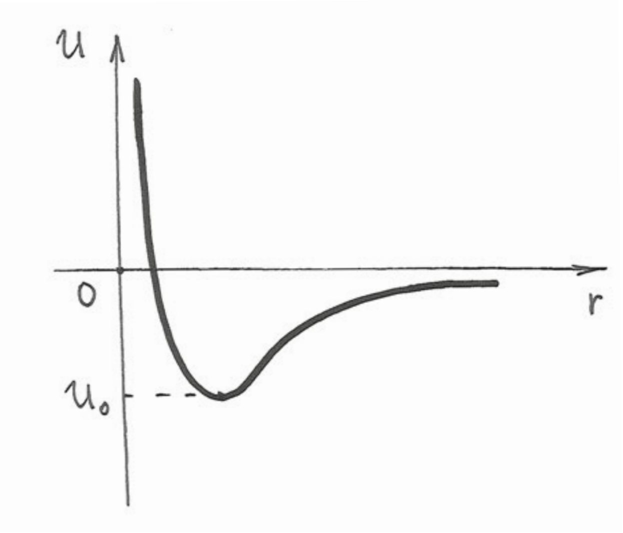


Figure 4.2. The profile of the van der Waals potential.

is also correct: if the system's equilibrium state is a Gaussian coil, then the system is an ideal polymer chain.

4.2.2 Real chain

The real polymer systems are more complicated than ideal chains. The main difference is the presence of volume interactions between particles. It makes the physics of the chains more interesting and complicated, giving rise to different phases.

The volume forces act as an attraction at large distances between atoms and as repulsion at small ones and called the *van der Waals forces*. The van der Waals potential is presented in Figure 4.2.

Although the physics is more complicated now, it can still qualitatively be described by the radius of gyration or a similar size quantity R. Let us introduce dimensionless *swelling parameter*

$$\alpha = \frac{R}{R_0} \ , \tag{4.13}$$

where R_0 is the size of the Gaussian coil, and R is the size of a given chain. This parameter specifies the role of volume interactions. If $\alpha > 1$ the coil is swollen compared to the Gaussian coil, and when $\alpha < 1$, the coil is compressed.

We can expect, but cannot be sure yet, that there can be different phases of the chain depending on the swelling parameter. When α is equal to 1, we come to the trivial case of the ideal coil. The scaling law (i.e., the rule for how the size R changes depending on the number of links) is $N^{1/2}$. If for other values of α this law changes, it indicates another phase. In the next section, we will try to find scaling laws for the extreme α.

4.3 Phase diagram

4.3.1 Virial expansion

An ideal coil is not a high-density system [31] which can be also seen from the scaling law in Eq. (4.12) (comparing to the dense 3D sphere which scaling law would be $N^{1/3}$). For that kind of systems, as we know from real gas theory, one can use *virial expansion* for thermodynamic functions [32]. The idea is to expand the function into a power series of the number of particles in a unit volume n.

The interaction part of the free energy can be presented as

$$\mathcal{F}_{\text{int}}(\alpha) = NT(nB + n^2C + ...) \ , \tag{4.14}$$

where T is temperature, N is the number of particles, n is the number of particles per volume and B, C are the second and the third expansion coefficients correspondingly (the first one has vanished).

The terms in the virial expansion have exact physical meanings. The one which is proportional to B corresponds to a double collision of particles, i.e., a collision of two particles. The term proportional to C is a contribution of triple collisions, i.e., a simultaneous collision of three particles and so on.

The form of interaction defines the coefficient of the expansion. For example, the second virial coefficient has quite a simple form

$$B(T) = \frac{1}{2}\int_0^\infty \left(1 - \exp\left\{-\frac{U(\boldsymbol{r})}{T}\right\}\right) dr^3. \tag{4.15}$$

It gives the main contribution to thermodynamic functions so let us look at it a bit closer, trying to get some information about the physics of the system.

For the van der Waals potential the integral in Eq. (4.15) splits into two: repulsion, where $U(\boldsymbol{r}) < 0$ ($r < a$) and attraction with $U(\boldsymbol{r}) > 0$ ($r > a$) (see Figure 4.2). Strictly speaking, attraction and repulsion are defined not by the sign of the potential itself but by the sign of the potential's derivative. But to make the derivation more readable, we will neglect the small part of the negative potential where the derivative is still positive, which corresponds to repulsion. Let us symbolically mark the positive part of the potential as U^+ and the negative as U^-. Then the integral in Eq. (4.15) can be expressed

$$2B(T) = \underset{\text{(repulsion)}}{\int_0^a \left(1 - \exp\left\{-\frac{U^+(\boldsymbol{r})}{T}\right\}\right) dr^3} + \underset{\text{(attraction)}}{\int_a^\infty \left(1 - \exp\left\{-\frac{U^-(\boldsymbol{r})}{T}\right\}\right) dr^3}. \tag{4.16}$$

When the temperature is low, $T \to 0$ then e with positive exponent tends to $+\infty$, while e with negative exponent tends to 0. This turns the attraction term into $-\infty$, but repulsion stays finite. Thus we can conclude that:

- For low temperatures attraction dominates over repulsion, and the second virial coefficient B is negative.

For small but non-vanishing temperatures $0 < T < \epsilon$, the attraction term turns to zero. Since the repulsion energy tends to infinity at zero distance, it will suppress any finite and non-zero temperature in the exponent. The integral for the second virial coefficient then becomes

$$2B(T) = \int_0^a 1 dr^3 + 0 \equiv \upsilon \ \ , \tag{4.17}$$

where υ is called an *excluded volume*. Now we can say:

- For high temperatures repulsion dominates over attraction, and B is positive.

From the condition of continuous B, there is a finite and non-vanishing temperature (which is called *θ-point*) where B turns into 0. The repulsion and attraction are balanced.

4.3.2 Scaling laws

Following Flory [33] let us write the free energy of the chain as a function of the swelling parameter

$$\mathcal{F}(\alpha) = \mathcal{F}_{\substack{\text{ideal}\\\text{chain}}}(\alpha) + \mathcal{F}_{\text{int}}(\alpha). \tag{4.18}$$

The two terms on the right hand side correspond to the free energy of the ideal chain and the free energy of the interaction. Here we used the same trick as for a real gas: decompose the thermodynamic potential into ideal and interactive parts.

It can be shown [31] that the first term

$$\mathcal{F}_{\substack{\text{ideal}\\\text{chain}}}(\alpha) \sim T\left(\alpha^2 + \frac{1}{\alpha^2}\right) \ \ , \tag{4.19}$$

where α^2 dominates when $\alpha > 1$ and α^{-2} dominates when $\alpha < 1$.

The interacting part of the free energy $\mathcal{F}_{\text{int}}(\alpha)$ as we discussed above will be expanded into a power series for unit volume in Eq. (4.14). Unlike the volume of a gas, the volume of a chain is not defined. But from very general geometrical ideas we can say that it should be proportional to the cubic size of the chain

$$n = \frac{N}{V} = \frac{N}{R^3}. \tag{4.20}$$

Substituting this into the virial expansion we have in Eq. (4.14)

$$\frac{1}{T}\mathcal{F}_{\text{int}}(\alpha) = \frac{N}{R^3}NB + \left(\frac{N}{R^3}\right)^2 NC + ... = \left(\frac{N}{R^3}\right)^2 R^3B + \left(\frac{N}{R^3}\right)^3 R^3C + ... \tag{4.21}$$

We should express R via the swelling parameter in Eq. (4.13). And then get rid of R_0 using the scaling law for the Gaussian coil in Eq. (4.12)

$$R^3 = (\alpha R_0)^3 = \alpha^3 N^{\frac{3}{2}} l^3 \tag{4.22}$$

Then the expression for the full free energy becomes

$$\frac{1}{T}\mathcal{F}(\alpha) \sim \left(\alpha^2 + \frac{1}{\alpha^2}\right) + \left(BN^{\frac{1}{2}}\frac{1}{l^3}\alpha^{-3} + \frac{C}{l^6}\alpha^{-6}\right). \tag{4.23}$$

The equilibrium value of α can be found from the minimization of free energy (where we drop all numerical coefficients)

$$\frac{d\mathcal{F}}{d\alpha} = 0 \Rightarrow \alpha^5 - \alpha \sim \left(BN^{\frac{1}{2}}\frac{1}{l^3}\right) + \left(\frac{C}{l^6}\right)\alpha^{-3}. \tag{4.24}$$

4.3.3 Three phases

Let us now define and discuss the properties of three distinct phases for the polymer based on the swelling parameter α.

- **Self avoiding random walk $\boldsymbol{\alpha \gg 1}$**

We can neglect the second term in Eq. (4.24) in both parts:

$$\alpha \sim \left(BN^{\frac{1}{2}}\frac{1}{l^3}\right)^{\frac{1}{5}}.$$

Then the scaling law becomes

$$R = R_0\alpha \sim lN^{\frac{1}{2}}N^{\frac{1}{10}}\left(B\frac{1}{l^3}\right)^{\frac{1}{5}} = \left(B\frac{1}{l^3}\right)^{\frac{1}{5}}N^{\frac{3}{5}} \ ,$$

from which we obtain the famous *Flory formula*

$$\boxed{R \sim N^{\frac{3}{5}}} \tag{4.25}$$

The coefficient B have to be positive since $\alpha > 0$. As we know from the previous section, the repulsion dominates in case when $B > 0$, and this happens at high temperatures, $T > \theta$ (see sec. 4.3.1).

This is in line with general physical reasoning: at high temperature the particles have high kinetic energy and tend to collide chaotically with each other. Then strong short-range repulsion plays a major role compared to long-range weaker attraction, which has to compete with the particles' kinetic energy.

The name of this phase reflects the fact that the repulsion is a major force here.

• **Random walk $\alpha = 1$**

This case corresponds to the ideal chain, so we just repeat the equation for the Gaussian coil

$$\boxed{R \sim N^{\frac{1}{2}}} \tag{4.26}$$

The ideal chain has no interactions at all, so integral in Eq. (4.15) vanishes. By definition $T = \theta$.

The name "random walk" means Brownian motion. In the real chain, the interaction cannot just disappear. So random walk is not possible; particles always feel each other. But from a macroscopic point of view, there is no difference between the situation with pure random walk and the state where all the interactions are balanced because both give the same values of the thermodynamic functions (R in our case). The word "phase" has meaning only in the macroscopic sense, so there are no assumptions in calling the phase "random walk".

This is a good example of the difference between the microscopic and macroscopic picture. Two qualitatively different systems from the microscopic perspective are considered the same from the macroscopic perspective.

• **Collapsed phase: $\alpha \ll 1$**

We can neglect the left part of Eq. (4.24)

$$\left(\frac{C}{l^6}\right)\alpha^{-3} \sim -BN^{\frac{1}{2}}\frac{1}{l^3}.$$

So R can be expressed

$$R = R_0\alpha \sim lN^{\frac{1}{2}}\left(-\frac{B}{C}l^3\right)^{-\frac{1}{3}}N^{-\frac{1}{6}} \sim \left(-\frac{B}{C}N\right)^{\frac{1}{3}}.$$

And the scaling law has the form

$$\boxed{R \sim N^{\frac{1}{3}}} \tag{4.27}$$

The interesting point is that this scaling law is the same as it would be for a dense three-dimensional sphere because its volume is proportional to r^3.

The attraction should dominate over repulsion to keep the chain in this high-density state. This domination leads to a negative B which means the chain is in the region of low temperatures $T < \theta$ (see sec. 4.3.1).

Another thing we want to highlight is the fact that the third virial coefficient C does not vanish in this case. It makes sense if we remember the physical meaning of it: collisions of three particles. Indeed, when a chain becomes a high-density system with the domination of attraction, the triple collisions become more likely.

4.3.4 Cheat sheets

Here we summarize the main results in a compact way. In all three phases the scaling law has the form

$$R \sim N^{\nu} \quad , \tag{4.28}$$

where the three phases only differ in their value of the exponent ν.

• SARW

Self avoiding random walk (or coil swelling).

$$\boxed{\begin{array}{lcl} T \gg \theta & \Rightarrow & B > 0 \\ \alpha \gg 1 & \Rightarrow & \nu = \dfrac{3}{5} \end{array}} \tag{4.29}$$

The repulsion dominates over the attraction.

• RW

A random walk (or the Gaussian coil or an ideal coil or θ-regime).

$$\boxed{\begin{array}{lcl} T = \theta & \Rightarrow & B = 0 \\ \alpha = 1 & \Rightarrow & \nu = \dfrac{1}{2} \end{array}} \tag{4.30}$$

Repulsion balances attraction.

• Collapsed phase

(or globular phase).

$$\boxed{\begin{array}{lcl} T \ll \theta & \Rightarrow & B < 0 \\ \alpha \ll 1 & \Rightarrow & \nu = \dfrac{1}{3} \end{array}} \tag{4.31}$$

Attraction dominates over repulsion.

4.4 Energy

We have already discussed the geometry of the polymer chains we will use (see sec. 4.1). Now it is time to talk about the physical model. In the present work, we are focused on the geometrical properties of polypeptide chains. We want to be able to form secondary (α-helix) and tertiary structures.

The secondary structures are local and should be formed by short-range interaction, while the overall tertiary structure should be the result of long-range forces. So if we want to simulate both structures, we should combine both these mechanisms in our energy function.

4.4.1 Short range

As we said in section 4.2.2 the phase transitions of a polymer chain is defined only by its geometry. And the geometry can be fully described by the curvature and torsion angles (see sec. 4.1).

Using the Frenet equations for the geometry gave us a $\mathbb{Z}_2$ symmetry (Eq. (4.10)) for κ: positive and negative curvature angles should provide the same physical state. This property is described with a double-well potential. For τ, it can be just a single minimum potential.

The Hamiltonian we are using in paper I and II was considered in several papers [34, 35, 36, 37]

$$\begin{aligned} H = &- \sum_{i=1}^{N-1} 2\kappa_{i+1}\kappa_i + \\ &\sum_{i=1}^{N} \left\{ 2\kappa_i^2 + q(\kappa_i^2 - m^2)^2 + \frac{c}{2}(d\kappa_i^2 + 1)\tau_i^2 - a(b\kappa_i^2 + 1)\tau_i \right\}. \end{aligned} \tag{4.32}$$

The second term in the last sum gives the desired double well profile.

The minimum of the Hamiltonian in Eq. (4.32) should be in alpha helix, which geometry is described by the curvature and torsion angles [36] as

$$\begin{cases} \kappa = \pm\frac{\pi}{2} \\ \tau = 1 \end{cases}. \tag{4.33}$$

With this, we fix the coefficient m to be 1.5. The parameter q is set to 3.5 and b will be considered as 0, so the term $\kappa^2\tau$ vanishes. From the rest a, c, and d parameters, not all are independent. Let us find the ground state for τ_i

$$\frac{\partial H}{\partial \tau_i} = 0 \Rightarrow \tau_i = \frac{a}{c}\left(\frac{b\kappa_i + 1}{d\kappa_i + 1}\right). \tag{4.34}$$

So if we want the minimum to be given by the expressions in Eq. (4.33) (and set $b = 0$) then

$$\tau_i = 1 = \frac{a}{c}\left(\frac{1}{\pm d\frac{\pi}{2} + 1}\right) \Rightarrow \frac{a}{c} = \pm d\frac{\pi}{2} + 1. \tag{4.35}$$

This gives us the relation for the parameters a, c and d.

4.4.2 Long range

As we learned earlier, the long-range interaction (or volume forces) should consist of two parts: attraction on long distances and repulsion on short distances. The phase diagram does not depend on the details of the interaction [31], so we are entirely free to choose the concrete profile of the potential $U(r)$.

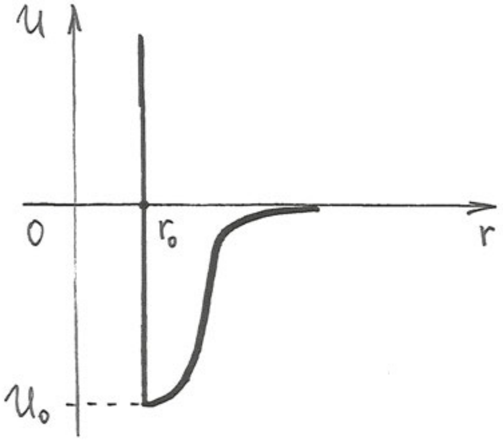


Figure 4.3. The profile of the potential we use in our model (Eq. (4.38)).

The asymptotic behavior for the repulsion is

$$U(r \to 0) \to +\infty. \tag{4.36}$$

The simplest model of repulsion is *hard core repulsion*. In this description, the particles cannot come closer to each other than some fixed value r_0. It forms an *excluded volume* $v \sim \pi r^3$ around each particle, which is impenetrable for all other particles. The potential $U(r)$ for this situation looks like an infinite wall at $r = r_0$; thus, all the states $r < r_0$ are forbidden.

For the attraction we need to have a small well with asymptomatic behavior

$$U(r \to +\infty) \to 0. \tag{4.37}$$

We chose to use hyperbolic tangent so the full potential have the form

$$U(r) = \begin{cases} +\infty, & 0 < r \leq r_0 \\ U_0(\mathrm{tanh(r - r_0)} - 1), & r_0 < r < +\infty \end{cases}, \tag{4.38}$$

and is presented in Figure 4.3. The parameter r_0 defines the range of the attraction and U_0 the depth of the well.

In the next chapter we will simulate the model we just discussed using the Monte Carlo method from Part I.

5. Simulations

In this chapter we will summarize everything we learned about the polymers and the Monte Carlo method from Part I of the thesis to find the phase diagram for the polymer model we described in the previous chapter. The total Hamiltonian is a composition of the effective Hamiltonian considered in section 4.4.1 and the van der Waals forces from section 4.4.2

$$H_{\text{tot}} = H + V_{\text{vdW}}. \tag{5.1}$$

The work is presented in Paper I. Here we will discuss only the essential parts, dropping out unnecessary details, which can be found in the original article.

The main goal is to investigate the phase diagram with respect to temperature, short-range interaction H and long-range interaction V_{vdW}. The most interesting parameter in the Hamiltonian in Eq. (4.32) is the chirality a. The van der Waals potential has only one energetic parameter U_0. So the phase diagram we will get is $R_g = f(T, a, U_0)$. Here R_g is the radius of gyration (see Eq. (4.11)); its exponent is an order parameter for the phase transition (see sec. 4.3.2).

5.1 Repulsion only

We started by following the paper by Chernodub *et al.* [36] and considered the model with only Hamiltonian in Eq. (4.32) and hard core repulsion. Although the authors used only Metropolis there, we will compare three different algorithms to simulate the system based on Metropolis and Heat Bath (see sec. 3.6).

All the algorithms share the same following features. The angles κ_i and τ_i are generated independently to each other and to all other chain sites. Since we will use MCMC[1] the order of updates should not influence the result. For finding the equilibrium states, we use simulated annealing (see sec. 3.7). The self-avoiding condition is taken into account as an additional condition for every update: whenever any pair of sites come closer than one-link-length distance, the configuration is rejected. This is actually a conventional Metropolis accept-reject algorithm for vanishing probability (see Eq. (3.32)), so the condition of detailed balance is not violated.

We will try 3 different algorithm which only differ in how they find new curvature and torsion angles:

[1] Heat bath is also MCMC in the sense that the next step does not depend on the history of steps at all.

1. conventional Metropolis to generate κ and conventional Metropolis to generating τ.
2. “Mixed” algorithm: Heat Bath for τ updates and conventional Metropolis for κ updates.
3. Heat bath for generating τ and Heat bath for generating κ.

Heat bath methods require us to generate random numbers according to the given distribution defined in the Hamiltonian in Eq. (4.32). This question is considered in the details in appendices A and B.

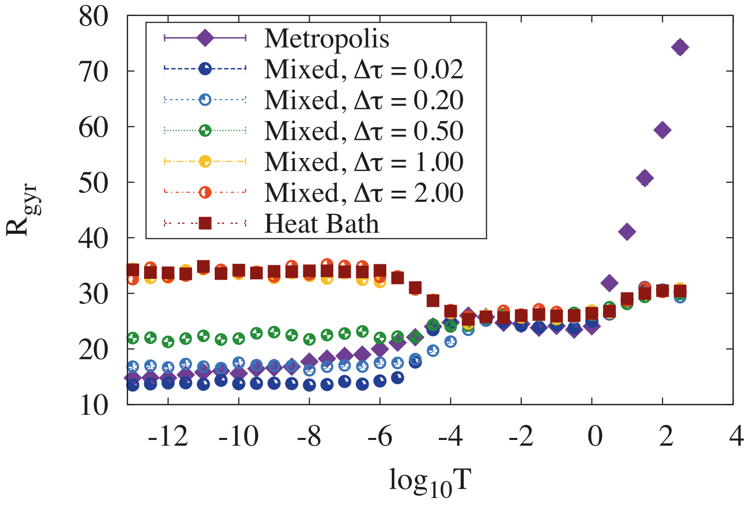


Figure 5.1. Comparison between the three different algorithms.

The equilibrium distribution cannot depend on the method we use, this is an essential property of MCMC and the only reason why we can assign a physical meaning to the answer. So as far as we do everything correctly, we should have the same results. However, one can see that it is not the case in Figure 5.1 where results of different algorithms are presented. $\Delta\tau$ is a dispersion of a proposal distribution which one uses in Metropolis. In our case, it is a Gaussian distribution. You should read the figure from the right to the left because we use simulated annealing. We start from the high temperatures and then decrease the temperature in an adiabatic way, while all the other parameters of the model are fixed. It means that every data point in the figure corresponds to the equilibrium state at a certain temperature.

How should we understand this deviation of different MC methods?

This is exactly the problem, with ergodicity, which we talked about in section 3.5.1 as the most dangerous misuse of MCMC algorithms. Metropolis is stuck in local minima and cannot get out of them. Of course, the profiles for the potentials in the Hamiltonian in Eq. (4.32) are simple: one peak for torsion and two peaks for curvature. It seems there is no place to get stuck there. However, the Hamiltonian in form presented in Eq. (4.32) does not take into account repulsion. The self-avoiding condition, which we have as an extra restriction, can drastically change the picture, creating a huge amount of local minima. Heat bath is freed from ergodic problems because the new state does not depend on the previous one. Although our Heat Bath is not a pure Heat Bath, but Heat Bath + Metropolis, the ability to overcome trapping of this algorithm is much higher.

Another thing that indicates that we meet the ergodic problem is that by changing the dispersion of the distribution for the torsion angle $\Delta\tau$ in Metropolis, we change the result. Increasing the dispersion increases the probability to tunnel through a broader potential barrier. Then the results start to converge with Heat Bath results.

Finally, we should check the most crucial thing that will prove or disprove our guess about what algorithm is correct: the observable dependence on the number of MC updates. In the limit of infinite time, all algorithms will converge to the same result as Markov chains theory predicts. In real simulations, the time is finite, and the question is: is the thermalization length we used enough for coming to the limit distribution in MCMC?

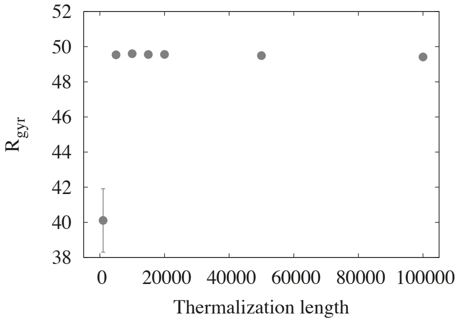


Figure 5.2. Mixed algorithm.

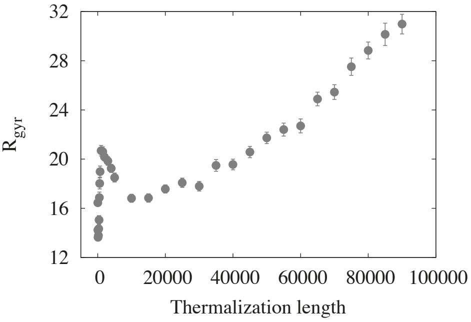


Figure 5.3. Metropolis algorithm.

The dependence of the radius of gyration on the number of MC steps (thermalization length) is presented in Figure 5.2 for the Mixed algorithm with large dispersion and Figure 5.3 for Metropolis.

The first plot shows that the Mixed algorithm comes to saturation very fast. All the configurations at the plateau are thermalized, i.e., the system is in the equilibrium state. The Metropolis algorithm in the second figure shows the opposite behavior: even for a sufficiently long time, the algorithm still does not provide the equilibrium state.

Based on three facts we just discussed:

1. the Mixed algorithm with large dispersion thermalizes, while Metropolis does not;
2. larger dispersion of the Mixed algorithm provides better convergence;
3. general knowledge about the principles of MC algorithms,

we can conclude without any doubt: pure Metropolis gives non-physical results, and we should choose either the Mixed algorithm or Heat Bath algorithm. In all further calculations, we will use the latter one.

5.1.1 The ground state

Look again at Figure 5.1. The "bad algorithms" at the low temperatures show some compact state since the radius of gyrations becomes smaller than at the beginning. This situation can be mistaken as a collapsed phase like in paper [36] which we followed. Now we already know that it is an artifact of the wrong usage of the MCMC algorithm, but there is even one more argument for why it is not a collapsed phase.

As we discussed in the theory section 4.3.1 for the existence of the collapsed phase, there should be domination of long-range attraction forces over the repulsion. There was no long-range attraction in the model we considered, but only repulsion, so no wonder that the "good" algorithms came to some not compact conformation. Because from the physical point of view, we cannot have a collapsed phase under such conditions.

Thus, we now have both physical and methodological proof that the collapsed phase cannot be obtained as presented in papers [35, 36].

The reasonable question now: what is the ground state for the model we have considered? The ground state there is an α-helix. One can see this by plotting the structure at the low temperature, and getting the scaling law for those states. The compactness index $\nu \approx 1$, which means that the structure scales as a straight rod. For more details, one should read paper I.

Interestingly, if there were only the Hamiltonian without any self-avoiding condition, the ground state would be the same, just because our Hamiltonian was constructed to generate α-helix in the minimum. It seems our excluded volume condition did not have any effect there. This is not surprising if we remember that the radius of the repulsion is equal to one link length and the α-helix is defined as in Eq. (4.33). So the distance between i and $i+2$ vertices is in $\sqrt{2}$ times larger than the repulsion radius.

5.2 Phase diagram

Using the correct algorithm and full potential in Eq. (4.38) presented in Figure 4.3 we were able to obtain the final three dimensional phase diagram presented in Figure 5.4.

The three-dimensional diagram is not easy to read so we will consider the most interesting cross-section: at fixed a [2], Figure 5.5.

[2] More cross-sections one can find in Paper I.

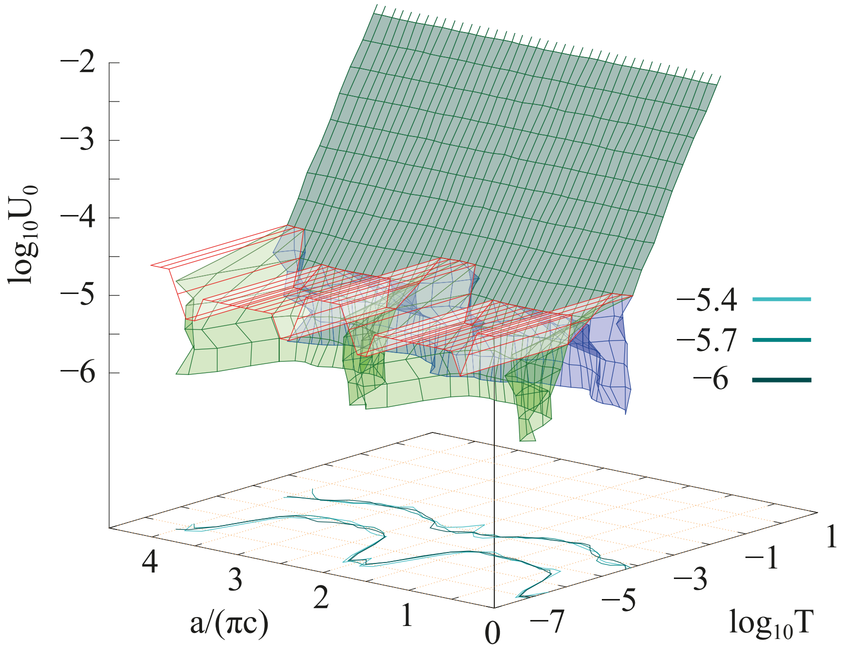


Figure 5.4. Full 3D phase diagram for the interaction described by Eq. (5.1).

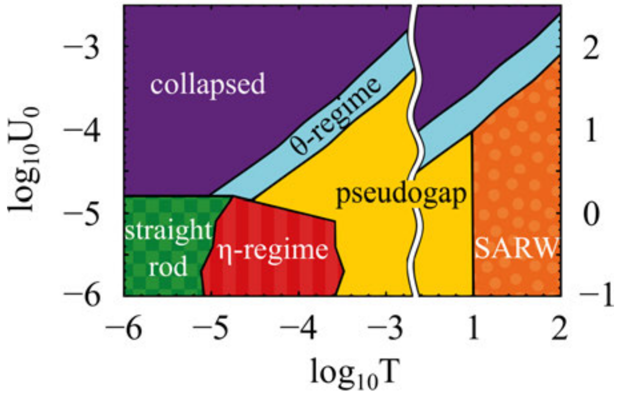


Figure 5.5. Cross-section of the phase diagram presented in Figure 5.4 at $a = 10^{-4}$.

When the attraction potential is strong enough, we have precisely the picture we expected (see sec. 4.3.2): self-avoiding random walk at high temperature, collapsed phase at low temperature and the θ-region in the between. When the attraction is suppressed by the short-range forces (the Hamiltonian and the repulsion), the ground state is a helix phase, as we know from the previous section. We called this situation the “straight rod” phase because it scales exactly like a straight rod.

Two other regions in Figure 5.5 do not appear if exponent ν from Eq. (4.28) is taken as an order parameter of phase transition: they both scale as SARW phase. But those phases are seen in the radius of gyration, so they have a character of cross-overs. Pseudogap phase you can see as the step at $\log_{10} T \approx 1$ in Figure 5.1. The η-regime is named as an analogy for the θ-regime: a region of coexistence of several phases. It seems like a split of the plane in Figure 5.4. More information can be found in Paper I.

5.3 Heteropolymer

The real structures do not exist in the pure coil state or the pure helical state. They have different regions with different structures. We can do this with our Hamiltonian if we set different parameters for different parts of our chain such that for some parts the ground state is a straight rod, and for other parts it is a collapsed phase. This type of chain, with different parameters for different parts we will call *heteropolymer*. The resulting configuration of the simulation of the heteropolymer is presented in Figure 5.6. After the thermalization, we got an α-helix for just a part of the chain, while the rest is a coil, exactly like we assumed.

The natural question comes: in what phase the structure in Figure 5.6 is? And here we face a new problem. We distinguish the different phases according to their scaling laws. If we want to follow the same procedure as we did for homopolymer, we have to make the same simulations for chains with various lengths and then find the exponent ν in Eq. (4.28) as a fitting parameter. This procedure cannot be implemented to heteropolymers because we do not know how we should vary the length in this case. Should we also change the length of the α-helix, i.e., the part with special parameters of Hamiltonian?

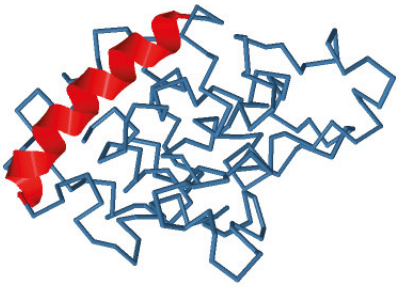


Figure 5.6. Simulated heteropolymer.

We will meet the same problem if we want to use the same definition of phases for real protein structures. There is no way to vary the length of the concrete polypeptide chain.

There is no answer to this question yet. But we will find the way in the next part

Part III:
Implementation of Renormalization Group Theory to Polymers

6. Theory

6.1 Introduction

In chapter 4, we found a scaling law for polymer chains

$$R \sim N^{\nu} \quad , \tag{6.1}$$

where R is the radius of gyration and N is the number of sites of the chain. Different values of the critical exponent ν allowed us to define different phases (see sec. 4.3.4). However, to use this method in practice, one should be able to vary the given polymer's length. This is possible to do only with homopolymer chains. Because for heterogeneous systems, the procedure of varying N is not defined. This is especially easy to see with real proteins. The number of sites, i.e., the number of amino acids the chain consists of, is one of the given protein characteristics. This number is fixed and cannot be different.

We have to find another way to distinguish phases having only one concrete chain.

6.2 Simple scaling procedure

The task is: finding a scaling law that can be used to define the phase for a given heteropolymer chain The natural idea which comes up is to *rescale* the given chain. The simplest way to do this is to connect every second site, as shown in Figure 6.1. The original chain is drawn with bold lines, while the new chain is drawn with thinner lines. We can do the same procedure once more for the new chain and get the chain shown as a dashed line in the same Figure. We could have more iterations if our initial chain were longer.

This method was presented in the paper [38]. The authors suggest keeping track of the flow of an average curvature α (called *folding angle*) over the chain which they assume to converges to a fixed point $\kappa^{\star}$

$$\begin{aligned} \alpha = \langle \cos\kappa^{(p)} \rangle &\equiv \langle \boldsymbol{t}_i^{(p)} \cdot \boldsymbol{t}_{i+1}^{(p)} \rangle \\ \cos\kappa^{\star} &\equiv \lim_{p\to\infty} \langle \cos\kappa^{(p)} \rangle, \end{aligned} \tag{6.2}$$

where p is the number of iteration steps in the procedure described above and $\boldsymbol{t}_i$ and $\boldsymbol{t}_{i+1}$ are tangent vectors of two adjacent links.

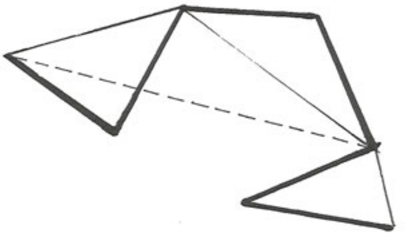


Figure 6.1. The simple scaling procedure. Bold line – Initial conformation. Thin line – First iteration. Dashed line – Second iteration.

The authors argue that when the limit in Eq. (6.2) exists and it is unique, then the folding angle and the critical exponent ν from the scaling law have the following relation

$$\cos\kappa^\star \approx 2^{2\nu-1} - 1. \tag{6.3}$$

And thus

$$\cos\kappa^\star = \begin{cases} -0.21 & \nu = 1/3 \\ 0 & \nu = 1/2 \\ 0.11 & \nu = 3/5 \end{cases}. \tag{6.4}$$

We can see three problems here:

1. Eq. (6.3) is not exact, hence the result in Eq. (6.4) is also not. But the difference between values of ν is pretty small, so we should be concerned about whether the result is correct at all.
2. It is not clear if the fixed point exists at all and what it could depend on.
3. If it exists, it should appear at the number of iteration steps p equal to infinity. In this thesis we have already talked about infinity in simulations. MCMC gives the correct answer only in infinite time. The simulating annealing technique also requires infinite time. And we know that having a tends-to-infinity condition was not a problem there. However, now this condition is a big problem. For MCMC, we could approach the infinity with any accuracy. So if the infinity was not enough “infinite”, we could always take more steps or smaller increments. Now the situation is different because p is limited by the length of the chain N

   $$p \leq \log_2 N. \tag{6.5}$$

   And even for the longest polypeptide chains in the Protein Data Bank [39], p would be not greater than 10. More likely, it will never be a good approach to infinity.

6.3 Renormalization Groups

The idea to apply Renormalization Group Theory to polymer chains was proposed by the French physicist Pierre-Gilles de Gennes[1] [40]. The concept was adopted from Kadanoff block spin transformations for describing a magnet [41]. There, the author proposed recombining the lattice sites into blocks and then treating the blocks as new sites. This should be done recursively, and the corresponding relations written at each step yields the fixed point in the rescaling procedure.

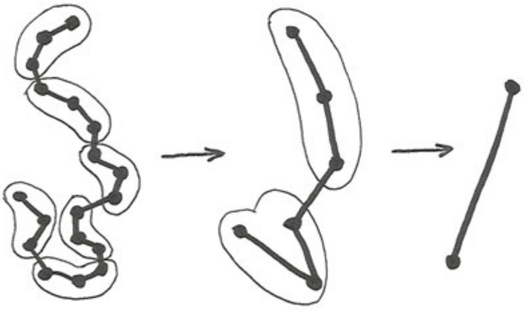


Figure 6.2. Renormalization group transformations following the Kadanoff block spin transformation adapted for proteins.

The same idea can be used for polymers. Schematically the procedure of rescaling is presented in Figure 6.2. Assume that each new block will consist of three old ones. Then we will say that the *scaling parameter* $s = 3$. If the original chain had N links, the new chain in the first step has $N_1 = N/s$ links. In the next step $N_2 = N_1/s = N/s^2$ and so on. Mathematically these transformations form a *renormalization group*[2]. Any observable as a function of iterative step $\mathcal{A}(p)$ form a *renormalization group flow*.

6.3.1 RG flow

When we derived the scaling law in the previous part and took into account volume interactions, we got that the main contribution to the microscopic structure was given by the dimensionless term $B/l^3 \equiv \beta$ (Eq. (4.24)), where B is the second virial coefficient and l is the link length.

[1] Nobel Prize in physics 1991: for discovering that methods developed for studying order phenomena in simple systems can be generalized to more complex forms of matter, in particular to liquid crystals and polymers.

[2] This is not a group in a strict mathematical sense.

Instead of the virial expansion, one can use a power series of β for the right part of Eq. (4.24). As we mentioned before, B (and hence β) corresponds to double collisions and also to excluded volume in Eq. (4.17). So you can think about the expansion in two ways: we expand the volume forces using perturbation theory with excluded volume as a parameter, or that we express all the collisions (like triple, quadruple, and so on) as double collisions.

We want to find the renormalization group flow for the parameter β, i.e., its evolution during the rescaling transformations. Instead of using the number of iteration steps p as an argument, we will use g – the number of initial monomers lumped into a single monomer in the current step. For the first step of rescaling procedure $g_1 \equiv s$, because it was how we defined the scaling parameter s. But for the second step $g_2 = s^2$. For the third $g_3 = s^3$ and so on.

First, we need to find recursive relations for β using the perturbation theory from the previous section. We will not present the derivation here because it is quite lengthy, but it can be found in [31]. Second, we want to convert that relation to differential equations.

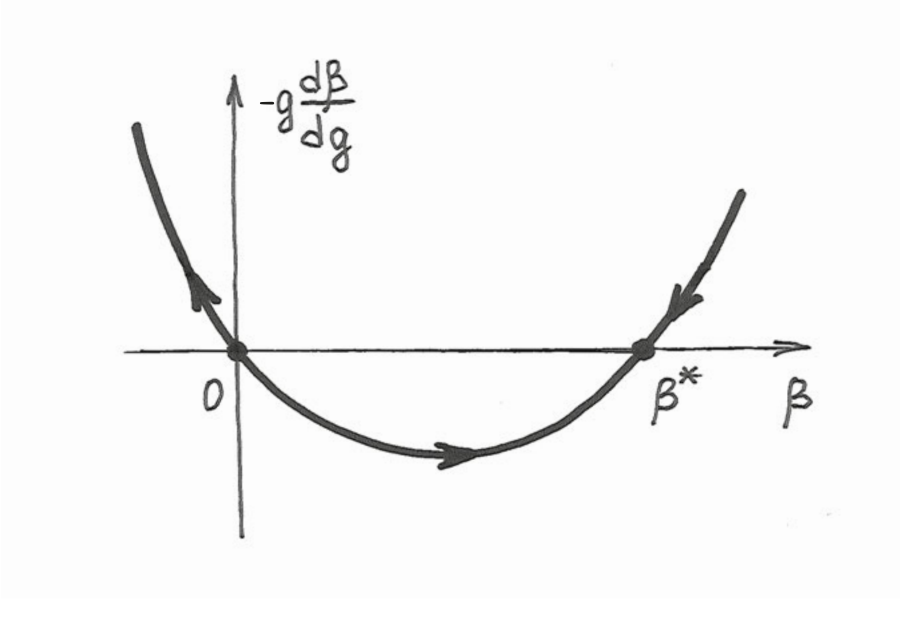


Figure 6.3. Phase portrait of the renormalization group equation.

The number of initial links contained in a new link in the $(p + 1)$-th step of the scaling procedure is

$$g_{p+1} = g_p + \Delta g. \tag{6.6}$$

And scaling parameter can be expressed as

$$s = \frac{g + \Delta g}{g} = 1 + \frac{\Delta g}{g}. \tag{6.7}$$

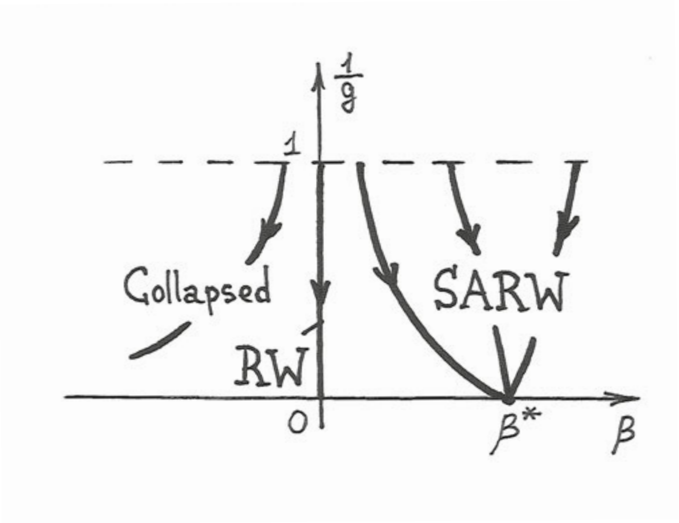


Figure 6.4. The renormalization group flow.

The virial coefficient β now can be written as a function of g. So at the p-th step, it is just $\beta_p \equiv \beta(g)$. And at the $(p+1)$-th

$$\beta_{p+1} \equiv \beta(g + \Delta g). \tag{6.8}$$

If we make the assumption that $s \to 1$ then

$$\frac{\Delta g}{g} \ll 1 \Rightarrow \Delta g \ll g. \tag{6.9}$$

And Eq. (6.8) can be expanded into Taylor series

$$\beta(g + \Delta g) \approx \beta(g) + \frac{d\beta}{dg}\Delta g. \tag{6.10}$$

This assumption yields a differential equation, which phase portrait is presented in Figure 6.3. There are two stationary points: the stable one $\beta = \beta^*$ and unstable one at $\beta = 0$.

When $\beta > 0$, the polymer is in the SARW phase, and the flow goes to the fixed point β^* which does not depend on β_0. If $\beta < 0$ then the chain in the collapsed phase and $\beta(g) \to -\infty$. The θ-regime which corresponds to $\beta = 0$ gives zero value for $\beta(g)$ at any scale. Summarizing what we said we can draw the plot of the renormalization group flow, Figure 6.4.

6.4 The observable

The dimensionless second virial coefficient β is not possible to calculate directly from the simulations. Thus we have to find some other observable which represent the RG flow as β and at the same time can be calculated just from the geometry of the chain.

The first candidate is the folding angle in Eq. (6.2). The expression for this observable in the first order of perturbation theory requires some calculations which we have skipped here, but the answer would be

$$\langle\alpha\rangle \sim \frac{\beta}{N}\sum_{i<j}(j-i)^{-3/2}. \tag{6.11}$$

As you can see it is a function of the parameter β which is a good point. However, the sum in Eq. (6.11) cannot be transferred safely to the integral because $\int_a^b x^{-3/2}dx$ diverges. This means that the folding angle is under a strong influence of the details of the potential and the perturbation theory we build cannot be trusted.

Instead of the folding angle we can consider another observable

$$\gamma = \sum_{i<j}\frac{(\boldsymbol{t}_i\cdot\boldsymbol{t}_j)}{|\boldsymbol{t}_i||\boldsymbol{t}_j|}. \tag{6.12}$$

The summation of cosines of angles between all possible couples of links. The new observable in the first order of perturbation theory gives the following expression

$$\langle\gamma\rangle \sim \beta\sum_{i<j}(j-i)^{-1/2}. \tag{6.13}$$

This sum can be converted into an integral in the limit of large N

$$\sum_{1\leq i<j\leq N-1}(j-i)^{-1/2} \overset{N\gg1}{\longrightarrow} \int_0^N dx\int_0^x dy\frac{1}{\sqrt{x-y}} \sim N^{3/2}. \tag{6.14}$$

And for the expectation of the observable γ then we get

$$\langle\gamma\rangle \sim \beta N^{3/2}. \tag{6.15}$$

Our new observable is proportional to β. And it is exactly what we wanted to get; now we can distinguish phases by a sign of RG flow according to the diagram in Figure 6.4.

However, before we can use this result, we should define the rescaling procedure because the theoretical description we gave was very abstract and impossible to use in practice.

6.5 The rescaling procedure

In the simple rescaling procedure (see sec. 6.2), the scaling parameter s was equal to 2: every new link contained two old ones. Now for RG theory, we need s to be close to 1. This means that it cannot be integer any longer. The question of a non-integer scaling parameter is complicated and allows for more than one solution. We propose the following [3].

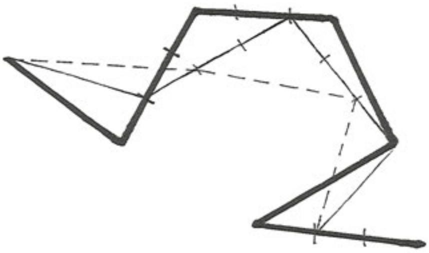


Figure 6.5. The scaling procedure for scaling parameter $s = 1 + 1/3$.

Let us first consider $s = 1 + 1/3$, Figure 6.5. The first vector of the first scaling step should be built as 1 full segment plus $1/3$ of the next segment

$$\boldsymbol{t}_1^1 = \boldsymbol{t}_1 + \frac{1}{3}\boldsymbol{t}_2. \tag{6.16}$$

The second link of a new chain consists of $2/3$ of the second segment plus $2/3$ of the third segment of old chain

$$\boldsymbol{t}_2^1 = \frac{2}{3}\boldsymbol{t}_2 + \frac{2}{3}\boldsymbol{t}_3. \tag{6.17}$$

The new third link takes the remaining part of the old second link and the full fourth link

$$\boldsymbol{t}_3^1 = \frac{1}{3}\boldsymbol{t}_3 + \boldsymbol{t}_4. \tag{6.18}$$

The new fourth segment can be built in a similar way as the first one.

This method has a disadvantage: truncating the end. The more steps we make, the more significant loss we have. And the proper RG flow will require many repetitions. Another minor thing is the asymmetry of the procedure. The left and the right ends of the chain are not treated equally, and there is no physical reason for it. To prevent both issues, we will use the following trick.

The loss of the ends does not happen always. If the number of segments N/s is an integer number, then the last site of the new chain matches the last site of the old chain. For instance, if the chain has 4 links and the scaling parameter s is chosen to be $4/3$ then the next iterative chain will have $4/(4/3) = 3$, as you can see in Figure 6.5. So the new chain has one link less than the previous one and has the same last site. The requirement we expressed is

$$\frac{N}{s} \in \mathbb{Z}. \tag{6.19}$$

[3] The extensive description of the rescaling procedure one can find in Papers II and III.

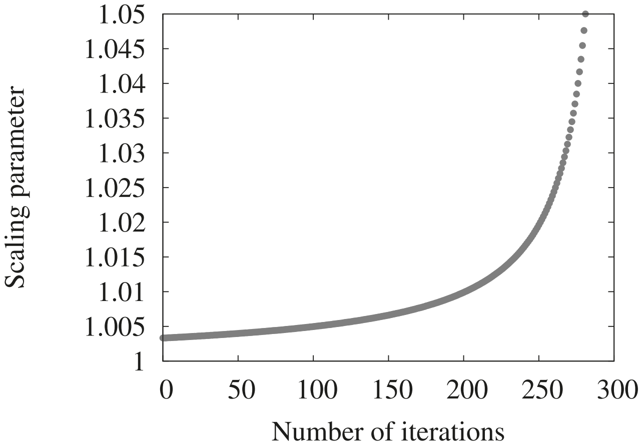


Figure 6.6. The dependence of the optimal scaling parameter s^{opt} on the number of iteration step for a chain with 300 links.

On the other hand, we want s to be very close to unity to get the picture of RG flow (section 6.3.1).

The optimal scaling parameter for us is the smallest s, which does not lead to the chain's truncation. It can be defined as

$$s^{\mathrm{opt}} \equiv \frac{N_p}{N_p - 1} = s(p) \ , \tag{6.20}$$

where p is the iteration step. The scaling parameter now depends on the iteration step but from the theoretical point of view it should be constant during the whole procedure. However, if we take a chain of 300 vertices, which is quite typical in the case of a protein backbone, we estimate that after $\sim$ 200 iteration steps, the optimal scaling parameter is changed by less than 0.7% as shown in Figure 6.6.

In addition, this scaling procedure treats both ends of the chain in the same way. It does not matter what end one starts the scaling procedure with; the same renormalization group will be produced.

So hereafter, we always will use s^{opt} as our scaling parameter.

7. Simulations

We will apply the RG ideas we expressed to the simulations from chapter 5 for homopolymers, where we have well-defined phases to verify our new theory. We are going to implement the procedure described in section 6.5 and calculate the value in Eq. (6.12) at every rescaling step. Then we want to build an RG flow of the observable and see whether we have the different phase behavior that we saw in Figure 6.4.

7.1 The first result

Let us take the homopolymers from the previous part of the thesis and calculate the RG flow for the rescaling procedure we just discussed to get the first results which is presented in Figure 7.1.

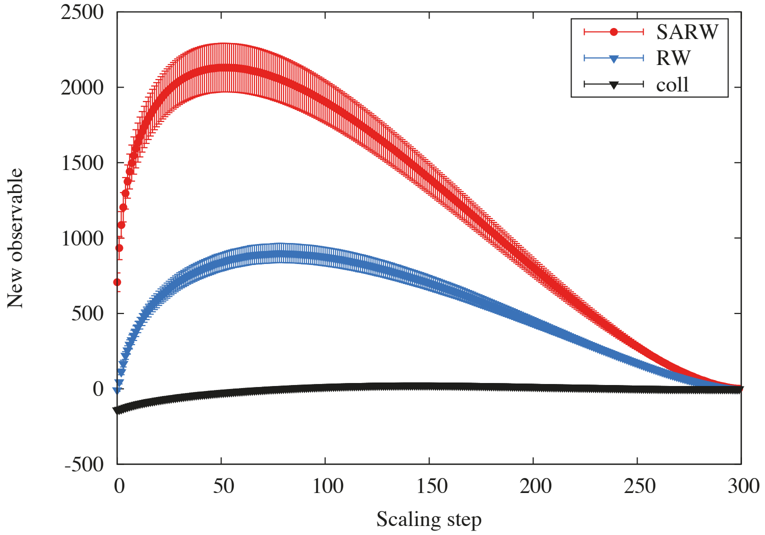


Figure 7.1. The first attempt to calculate the RG flow for three different phases of homopolymers.

We can see a problem: the RW phase, which we expect to be 0 like in Figure 6.4 has a bizarre behavior. The collapsed phase is also not negative as we predicted. But, for the latter, we made much more assumptions during the theoretical derivation. However, the RW phase is an ideal case, and the theoretical description there should be very accurate. For this study, we made a new RW state simulation as a pure Markov chain, i.e., without any interactions. Thus, we can be sure that if everything is correct, the RW phase should give a trivial solution for RG flow.

The only thing that could lead to the wrong result is the scaling procedure itself. Something like: during the RG transformation, we get some artificial correlation. Apparently, this is what indeed happens.

Let me explain. If you look at the scaling procedure once more in Figure 6.5, you can see that because the new link contains parts from several old links, we get some extra dependencies. In the RW case, only adjacent links correlate with each other. Let us consider Figure 6.5 together with corresponding equations (6.16-6.18). The link $\boldsymbol{t}_1$ correlates with $\boldsymbol{t}_2$. The link $\boldsymbol{t}_2$ correlates with $\boldsymbol{t}_3$. But $\boldsymbol{t}_1$ and $\boldsymbol{t}_3$ are not dependent on each other. This is what the RW phase is. After the first step of rescaling $\boldsymbol{t}_1^1$ and $\boldsymbol{t}_3^1$ will not be independent any longer! They both share the same old vector $\boldsymbol{t}_2$. Thus our RW state is not RW as soon as we start the renormalization procedure. This correlation will spread along the chain with increasing iteration steps number. But luckily, it is a local effect, so it decays along the chain.

7.2 The corrected result

What can we do then? We know where this artificial correlation comes from. What if we will exclude from the summation in Eq. (6.12) the terms that correspond to close links. Instead of making a summation over $i < j$ we will do it over $i < j + k$, where k is some fixed number

$$\gamma = \sum_{i<j+k} \frac{(\boldsymbol{t}_i \cdot \boldsymbol{t}_j)}{|\boldsymbol{t}_i||\boldsymbol{t}_j|}. \tag{7.1}$$

This subtraction should not significantly influence the theoretical result because the number of terms is N^2, and we will reduce only kN. Thus for large N and small k the theory should still work, but we will reduce the artifacts we got just from the scaling procedure.

In addition, k should not depend on the chain length. This parameter corresponds to the propagation of the correlation along the chain with the number of iterative steps of the rescaling procedure. So we can find it empirically once and use it for all chain lengths.

The RG flow for the new observable with $k = 10$ is shown in Figure 7.2. Finally, three different phases show three different behavior as we wanted. However, instead of different fixed points at a large scale, we got convergence to 0 for all three curves. Zero value corresponds to the RW state, i.e., a chain

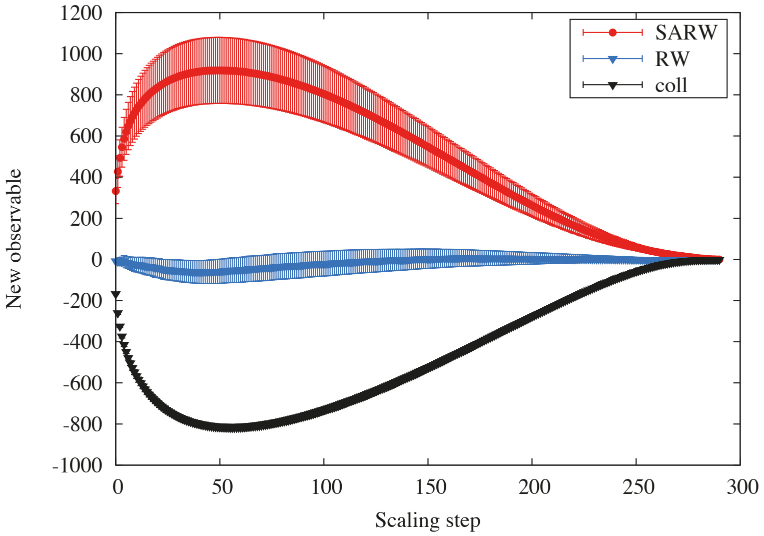


Figure 7.2. The calculation of the RG flow for three different phases of homopolymers.

without any interaction. So when rescaling the chain, where we select bigger and bigger blocks, the information about volume forces vanishes, and the chain behaves as an ideal one. It goes in line with the RG idea: to get rid of small scale effects to see the large scale phenomena.

You may understand this easily if you think that the chain is reduced to just two segments at the last step of the rescaling procedure. And two segments should always show the RW behavior.

The fact that the all three phases converges to zero for large number of scaling steps does, however, not signal a failure of our method. Rather, the problem we have accounted is that for large scaling step number the scaling parameter can no longer be considered constant, which is important, as pointed out in section 6.5. When applying the procedure, we should therefore pay attention to the behavior in the intermediate regime.

7.3 Scaling

For the new observable, we got the scaling law expressed by Eq. (6.15). This result is not solid, we used a lot of assumptions for derivation, but it is interesting to see how good it approaches the result of our simulations.

We fitted the results for homopolymer in the SARW phase with the function

$$\langle \gamma(n) \rangle = an^b + cn \tag{7.2}$$

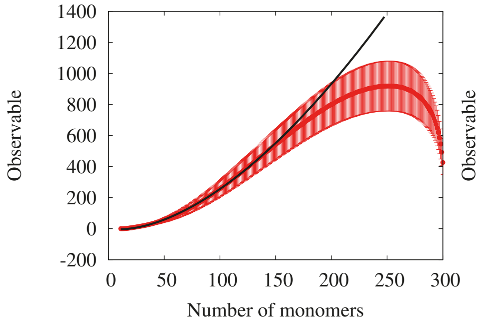


Figure 7.3. Fitting (Eq. 7.3).

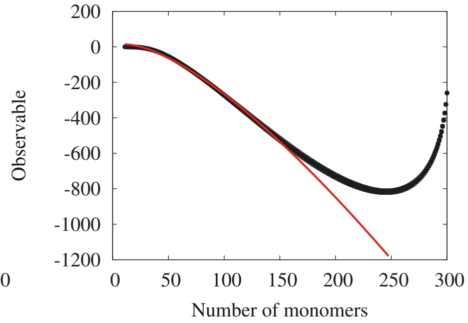


Figure 7.4. Fitting (Eq. 7.4).

via parameters a, b and c, where n is the number of chain monomers at the certain rescaling step. In Figure 7.3 one can see the result of the fitting. The parameters we found using least square approximation are:

$$\begin{aligned} a &= 0.24 \quad \pm 0.03 \\ b &= 1.61 \quad \pm 0.02 \\ c &= -1.45 \quad \pm 0.12. \end{aligned} \tag{7.3}$$

In Figure 7.3, the abscissa now shows the number of monomers which is inversely proportional to the iteration step since we use the optimal scaling parameter defined in Eq. (6.20). From both the figure and the value for b, we can conclude that we are in good agreement with scaling law $N^{1.5}$ even though it lies outside the one-sigma interval.

It can be easily shown that in the first order of the perturbation theory, the scaling law for the collapsed phase appears to be the same. The result of fitting the collapsed phase with the function in Eq. (7.2) is presented on Figure 7.4. The parameters are

$$\begin{aligned} a &= 6.22 \quad \pm 2.05 \\ b &= 1.16 \quad \pm 0.03 \\ c &= -10.5 \quad \pm 2.4. \end{aligned} \tag{7.4}$$

However, in this case the deviation b from the desired 1.5 is much bigger. The reason is that for the collapsed phase, the first order of perturbation theory is a too rough approach. In section 4.3.2 we showed that triple collisions play a more significant role in collapsed phase than in SARW, which is depicted through the non-vanishing third virial coefficient C. So the collapsed phase requires the next term in the expansion in Eq. 6.15 to obtain γ.

Another interesting question is why our fitting works better for a smaller number of links or at the end of the iteration procedure. The answer could be the following. As we said above, all three phases converge toward the RW one. So the closer we come to the end of the rescaling procedure, the closer we come to the ideal chain behavior. And our theory was built as a perturbation theory in the vicinity of the RW state. That is why it works better at the ends of the RG transformation procedure.

7.4 Multiple phases

At the introduction to the thesis, we talked about different structures of proteins. These structures are different not because some of them are more “chemical” and some are more “physical”. This is a very conditional division as conditional the division between physics and chemistry itself. The real difference is the *scale* where these structures are found. The primary structure is a sequence of amino acids that represents single vertices in our chain. This is the scale of one chain site. The secondary structure is a conformation of several amino acids, so several sites of our chain. For example, the one loop of α-helix is constructed with four vertices. Tertiary structure is an overall geometrical structure of the chain. So this happens on the scale of the chain size. And finally, the quaternary structure combines several polypeptide chains together; the scale is larger than one chain size.

We can think about different phases as different scales. We can approach this problem using the method we discussed but now varying parameter k in the observable in Eq. (7.1). The discussion of this idea can be found in Paper II, so we will not repeat it here but rather send the reader to the article.

8. Smoothing algorithm

The idea of rescaling polymer chains led to the idea of smoothing polygonal curves. And from smoothing curves, one can come to the idea of smoothing scattered plots.

Here we will just leave the link to Paper III, because it contains an exhaustive description of the new smoothing algorithm, which is based on the RG theory explained in this thesis.

Part IV: Experiment

9. Experiment

In this chapter we will refer to the biggest international open access digital data base Protein Data Bank or RSCB PDB [39] which stores the experimental data for biological structures. The database grows daily and during its 50-year history it has collected the information of over 171 000 entries (171 588 at the moment when this text is written). The price of the database is estimated at $16 billion (USD) [42] counting all the expenses of all the contributors including the work hours cost.

There are three main experimental techniques for resolving molecular structures: X-ray crystallography, NMR spectroscopy and electron microscopy. All of them need some extra data to build a full atomic picture. Usually it is information about the chemical structure of the molecule, since this is what scientists have before they are interested in the molecular geometry.

9.1 X-ray crystallography

The idea of X-ray crystallography is based on the Bragg's law of diffraction discovered in 1913 [43]. In 1915 The Braggs, father and son, got a Nobel Prize in Physics "for their services in the analysis of crystal structure by means of X-rays". They discovered that the elastic scattering of a beam of coherent and monochromatic photons on a crystal lattice produce a characteristic diffraction picture which provides information about the distribution of electrons in the sample. This information can be used to determine atoms positions in the crystal.

In 1958 Sir John Cowdery Kendrew published his famous paper in Nature where he presented the first successful result of the implementation of Braggs's law to protein structure imaging [44]. In 1962 he shared the Nobel Prize in Chemistry with Max Perutz, who was the PhD student of Bragg Junior "for their studies of the structures of globular proteins" [45]. And from that time until today X-ray crystallography remains to be the most popular method for protein geometry recognition (80% of all PDB entries).

X-ray crystallography can provide high resolution all-atoms protein picture together with the ligands and other molecules. However, it has a set of serious flaws. The first and the most obvious is the need to crystallize the sample. Proteins are the most interesting for us in their native state, i.e. like they are presented in our body for example. But there and everywhere they exist in liquid solvents, but never like solid crystals. So the structure we get from

X-ray crystallography can be very different from the structure of our interest without us even knowing about it.

The second problem of X-ray crystallography is that not all proteins can be crystallized and not all parts of one protein can be crystallized equally well. The former leads to the problem that some proteins cannot be investigated by this method at all. And the latter leads to the problem of missing atoms: when the most flexible parts of the protein cannot be localized with a given accuracy.

One more complexity of the method is the distinguishing between the crystal symmetry and the structure symmetry.

The long time of preparing the big enough crystal for many proteins (1 - 12 month) can be also considered as a drawback of the method [46].

9.2 Nuclear magnetic resonance spectroscopy

Nuclear magnetic resonance or NMR was discovered in the middle of the previous century. In 1944 Isidor Rabi got a Nobel Prize in Physics "for his resonance method for recording the magnetic properties of atomic nuclei" [47]. And eight years later Felix Bloch and Edward Purcell shared the 1952 Nobel Prize in Physics "for their development of new methods for nuclear magnetic precision measurements and discoveries in connection therewith" [48]. The scientists showed that if the sample has a nucleus with non-zero spin and is placed in a magnetic field with a certain NMR frequency (radio waves) the sample will produce the characteristic radiation spectrum. This spectrum not only depend on the atom, but also on its surroundings.

NMR spectroscopy can give us the information about different atoms restraints in the protein: the limitations for atoms distances and angles. To get the full molecule model one has to do computer simulations where the experimental restraints for atoms' distances and angles are taken into account. This is an obvious drawback of the method.

Another disadvantage is that NMR spectroscopy can only work with small molecules. The spectra for large proteins cannot be resolved due to overlap of the peaks. This, however, can be improved with further development of the technology.

NMR spectroscopy takes away the main disadvantage of X-ray crystallography: the proteins now are examined in their native phase. It also opens the door to study proteins dynamics.

9.3 Electron microscopy

This name can refer to different methods, we will talk about the most popular for protein structure recognition – cryo-EM.

The first electron microscope (EM) was built in 1932 by Ernst Ruska and in 1986 the author was awarded with half a Nobel Prize in Physics “for his fundamental work in electron optics, and for the design of the first electron microscope” [49]. Unlike the optic microscope the electron one uses an electron beam instead of a light beam. Since the wavelengths of electrons in an electric field can be much shorter than the wavelength of visible light, EM gives much better resolution. EM allow us to “see” on the atomic level. But to use EM to study protein structures one had also to solve the problem of sample preparation. This breakthrough happened in 1984 and was published in Nature [50]. The group of scientists found out that by vitrification (creating an amorphous solid state) of samples one can fix the protein and at the same time preserve its native structure. Jacques Dubochet, Joachim Frank and Richard Henderson were awarded with the Nobel Prize in chemistry in 2017 “for developing cryo-electron microscopy for the high-resolution structure determination of biomolecules in solution”.

Cryo-EM is a new fast growing field. According the PDB statistics the number of entries discovered by EM grows exponentially every year. Cryo-EM can provide the same resolution as X-ray crystallography but does not change (significantly) the native protein structure.

9.4 Single Particle Imaging (SPI)

The technique we are going to talk about is a new approach to protein imaging.

As we already know, X-ray crystallography needs a crystal. This crystal have to be big enough to produce the visible diffraction pattern. The bigger crystal is the brighter the image. But the brighter diffraction image one can also get by instead increasing the X-ray beam intensity. And then we come to a question: can we generate an intense enough X-ray beam to obtain a visible diffraction pattern from just one protein, not a protein crystal?

The positive answer came only 10 years ago together with a new era of lasers: X-ray free electron lasers (XFEL) , like US LCLS [51] or European XFEL [52]. The first free electron laser (FEL) was made in 1971 by John Madey [53] by utilizing Hanz Motz’s *undulator* developed 20 years earlier [54].

An undulator is the device which force electrons to move in a sinusoidal trajectory. On every turn of the sinusoid electrons radiate photons according to the laws of classical electrodynamics. This radiation forms a beam of a FEL. By modifying the speed of electrons one modify the photon frequency. And to get the X-ray radiation the speed of electrons has to be close to the speed of light. This explains why the way from the first FEL to XFEL took 40 years.

Apparently, XFEL can be utilized for molecular structure recognition. Several scientific groups predicted the possibility of using this technology for Single Particle Imaging (SPI) of biomolecules [55, 56] way before it became manageable in practice. They showed that just one protein can produce a bright enough diffraction pattern which can be used for reproduction of its 3D model (by collecting a lot of statistics).

But here a new problem appears: the intensity of the laser beam should be so high that this will unavoidably lead to the sample explosion. What will we get on the diffraction picture in this case?

Let us consider the easy analogy. You want to take a photo of a watermelon, but every time you press the button on you camera it will explode (we assume the speed of light is infinite here). Is it possible to make a watermelon photo in this settings? It is clear that if you do a "normal" photo it will be blurred. This happens because the watermelon explodes so fast compared to the time the diaphragm is open, that you collect several events on one photo and they overlap each other. But if you could make the exposition time sufficiently small, then you could record one time event at the beginning when the watermelon is still untouched. This will give you a clear picture of the watermelon.

Coming back to laser and protein, we can say that the photo from the analogy above is our diffraction picture, and the exposition time is... the length of the laser pulse. Luckily, the modern XFELs work in a pulse regime to provide high energies. So if the laser pulse is short compared to the time of protein explosion we can get a clear diffraction picture from the original protein. This conception got the name *diffraction before destruction* [57]. The very short femtosecond pulse length of modern XFELs makes SPI a new potential tool for protein structure recognition [58].

With sample destruction at every measurement comes an obvious need to prepare many of them. This creates a new difficulty. The sample delivery for SPI is done with aerosol spray: a droplet with a single molecule injected in vacuum to be shot with the laser beam. The particle is orientated randomly in this droplet. So to every new laser shot we deliver a new droplet with the same protein but unknown orientation. This leads to increasing the number of sampling and hence increasing the experiment cost. But it works!

The first proof of concept was done in 2011 [59]. The authors struggled with a low hit rate (there were no synchronization between sample delivery and laser pulses) but they managed to make a 2D image of a virus. The full 3D image of the same virus was done several years later by another scientific group [60]. There are more successful examples with virus imaging [61, 62] as well as with full cells and organelles [63, 64].

To improve SPI technology a smart way around the orientation problem was proposed recently [65]: to use a strong electric field to orient the proteins and reduce the number of degrees of freedom from three to just one. The actual experiment is ongoing now under the MS-SPIDOC Horizon2020 project [66] where soon we hope to have the first experimental results to see whether the idea with orientation works or not.

Part V:
Molecular Dynamics

10. Theory

This chapter will tell about a more “natural” and straightforward way of simulating molecules than Monte Carlo (MC) – classical molecular dynamics (MD). As follows from the name, this method allows us to reproduce the system’s dynamics, unlike MC (see part I), which only provides the equilibrium states. Instead of playing a probability game as we did with MC, now we will honestly solve a system of Newton’s equation of motion.

The first crucial work in this field was done in 1957 by Alder and Wainwright, who calculated a phase transition for a hard sphere system using MD [67]. The solution for this problem, however, already existed (with slightly different results however). It was obtained by the MC method 3 years earlier by one of the co-authors of Metropolis algorithm [68]. We see that even though MD is conceptually easier than MC, it was implemented later. Soon we will understand why.

10.1 The basic idea

Let us imagine that we have a system of N particles of masses m_i. If we know the forces which acts on every particle $\mathbf{F}_i$ then, given boundary conditions, we can find the new state of the system at any time by solving a system of Newton’s equations of motion

$$m_i \frac{d^2 \mathbf{r}_i}{dt^2} = \mathbf{F}_i, \quad i \in \{1, \dots N\}. \tag{10.1}$$

After the configuration of the system is changed the conservative forces has to be recalculated, since they depend on the atomic positions

$$\mathbf{F}_i = -\frac{\partial V(\mathbf{r}_i, \dots, \mathbf{r}_N)}{\partial \mathbf{r}_i}, \quad i \in \{1, \dots N\} \ , \tag{10.2}$$

where $V(\mathbf{r})$ is a potential interactions between atoms.

Repeating Eq. (10.1) and Eq. (10.2) for small enough time step Δt one can numerically find the trajectory of the system, i.e. the position of all atoms as a function of time $\mathbf{r}(t)$.

10.2 Implementation: GROMACS

Now we will look close to MD's concrete implementation on the example of the most popular open-source software: GROMACS [21]. It utilizes many ideas and algorithms designed by different scientific groups over decades of scientific research. Combined with the newest ideas and best programming practices makes GROMACS a universal tool for MD simulations of a different kind.

In this section we will only touch on the basic mechanisms of the software. For more information please go to GROMACS manual [69]. There the authors describe all the concepts and tricks they implemented.

10.2.1 Force field

The set of forces in Eq. (10.2) are called the *force field*. It defines the concrete view of the additive potential functions $V = \sum_a V_a$ and the parameters' value in those equations for every type of atom.

The force field design is a complicated business done by different scientific groups around the world. There exist many solutions, and they are constantly improved. Force fields we have now are empirical. They are tuned to fit the experimental data or more accurate quantum calculations. So different force fields work better for different systems, depending on what fitting data was used for their parameters. Also note, that even though the potentials used in MD are classical, they carry the information about the quantum nature of atoms because of the fitting process.

Here is the top 3 popular fields:

1. CHARMM (Chemistry at HARvard Macromolecular Mechanics) developed in 1983 by the world wide collaboration together with the Nobel Laureate in Chemistry 2013 Martin Karplus at Harvard University [70].
2. OPLS (Optimized Potential for Liquid Simulations) developed in 1988 by William L. Jorgensen at Purdue University and later at Yale University [71]. And its most popular version OPLS-AA (All Atoms) designed in 1996 [72].
3. AMBER (Assisted Model Building and Energy Refinement) developed in 1995 at the University of California by Peter Kollman's group [73].

The potential function of any force fields can be split into 2 terms:

1. Non-bonded: van der Waals forces and Coulomb (-like) interaction.
2. Bonded: covalent, angle and dihedral bonds.

In addition GROMACS allows to apply extra restraints for position, angle, distance, dihedral or orientation.

Now we will look closer at the potential functions in the example of the AMBER force field.

Non-bonded interaction

This type of interactions are “physical” pair forces

$$V(\mathbf{r}_1, \ldots, \mathbf{r}_N) = \sum_{i<j} V_{ij}(\mathbf{r}_{ij}) \ , \tag{10.3}$$

$$\mathbf{F}_k(\mathbf{r}_{ij}) = -\sum \frac{dV_{ij}(r_{ij})}{dr_{ij}} \frac{\mathbf{r}_{ij}}{r_{ij}} \ , \tag{10.4}$$

where $V_{ij}(\mathbf{r}_{ij})$ is a potential of atom i and atom j separated by $\mathbf{r}_{ij}$.

The first contribution is the Coulomb force to model electrostatics

$$V_C(\mathbf{r}_{ij}) = k_C \frac{q_i q_j}{r_{ij}} \ , \tag{10.5}$$

where q_i and q_j the charges of the corresponding atoms, r_{ij} is the distance between them and k_C is a constant which depends on the unit system. In SI k_C can be expressed through the vacuum permittivity ϵ_0 as $k_C = 1/4\pi\epsilon_0$.

The second part in non-bonded interactions is the van der Waals interaction in the form of a Lennard-Jones potential

$$V_{vdW}(\mathbf{r}_{ij}) = \epsilon_{ij} \left(\left(\frac{r'_{ij}}{r_{ij}} \right)^{12} - 2 \left(\frac{r'_{ij}}{r_{ij}} \right)^{6} \right) \ , \tag{10.6}$$

where r'_{ij} is an equilibrium distance between i and j atoms and ϵ_{ij} is the depth of the potential well.

Non bonded interactions can be multiplied by some parameter f_{ij} for better fit to the results. For example, it allows to exclude interaction between the 3 neighbor atoms, to describe their behavior by only bonded interactions.

Bonded interaction

Bonded interactions are “chemical” interactions that can emerge between 2 atoms (covalent bond), 3 atoms (bond angle), or 4 atoms (dihedral angle or torsion).

A covalent bond is modeled as a harmonic oscillator

$$V_r(r_l) = k_{ij}(l_{ij} - l'_{ij})^2 \ , \tag{10.7}$$

where k_l is a coupling strength, l_{ij} is a bond length between two atoms i and j and l'_{ij} is an equilibrium distance between them.

The bond between 3 atoms are also represented as a harmonic potential for the θ angle between ij-bond and jk-bond

$$\begin{cases} \theta_{ijk} = \text{arccos}\left(\frac{\mathbf{r}_{\mathrm{ij}} \cdot \mathbf{r}_{\mathrm{kj}}}{\mathrm{r}_{\mathrm{ij}} \mathrm{r}_{\mathrm{kj}}} \right) \\ V_\theta(\theta_{ijk}) = k_\theta(\theta_{ijk} - \theta'_{ijk})^2 \end{cases} \ , \tag{10.8}$$

where k_θ is a strength constant for the triple interaction and θ'_{ijk} is an equilibrium angle.

The final 4 body interaction is a rotation of 2 planes: the plane of the first 3 atoms and the last 3 atoms' plane. The angle between 2 planes is denoted as ϕ. The dihedral potential is periodic

$$V_\phi(\phi_{ijkl}) = k_\phi(1 + \cos(n\phi - \phi')) \ , \tag{10.9}$$

where k_ϕ is a interaction constant and ϕ' is an equilibrium angle.

10.2.2 Leap frog integrator

When the force field is set, one can find all the forces in the system (Eq. (10.2)) and numerically solve the equations of motion (10.1). The responsible algorithm is called an *integrator*. There are several algorithms in GROMACS for doing this, but we will talk about one we used in our simulations: *leap frog integrator*. The integrator got its name because of the visualization: the velocity and the coordinate are always defined with a half time grid shift drawing the picture of frogs leaping over each other.

$$\begin{cases} \mathbf{v}(t + \frac{1}{2}\Delta t) = \mathbf{v}(t - \frac{1}{2}\Delta t) + \frac{\Delta t}{m}\mathbf{F}(t) \\ \mathbf{r}(t + \Delta t) = \mathbf{r}(t) + \Delta t\mathbf{v}(t + \frac{1}{2}\Delta t) \end{cases} . \tag{10.10}$$

The trajectory is solved up to the 3rd order

$$\mathbf{r}(t + \Delta t) = 2\mathbf{r}(t) - \mathbf{r}(t - \Delta t) + \frac{1}{m}\mathbf{F}(t)\Delta t^2 + O(\Delta t^4). \tag{10.11}$$

The algorithm is time reversible, hence preserve the time symmetry of Newton's equations.

The generic MD algorithm is presented in Figure 10.1.

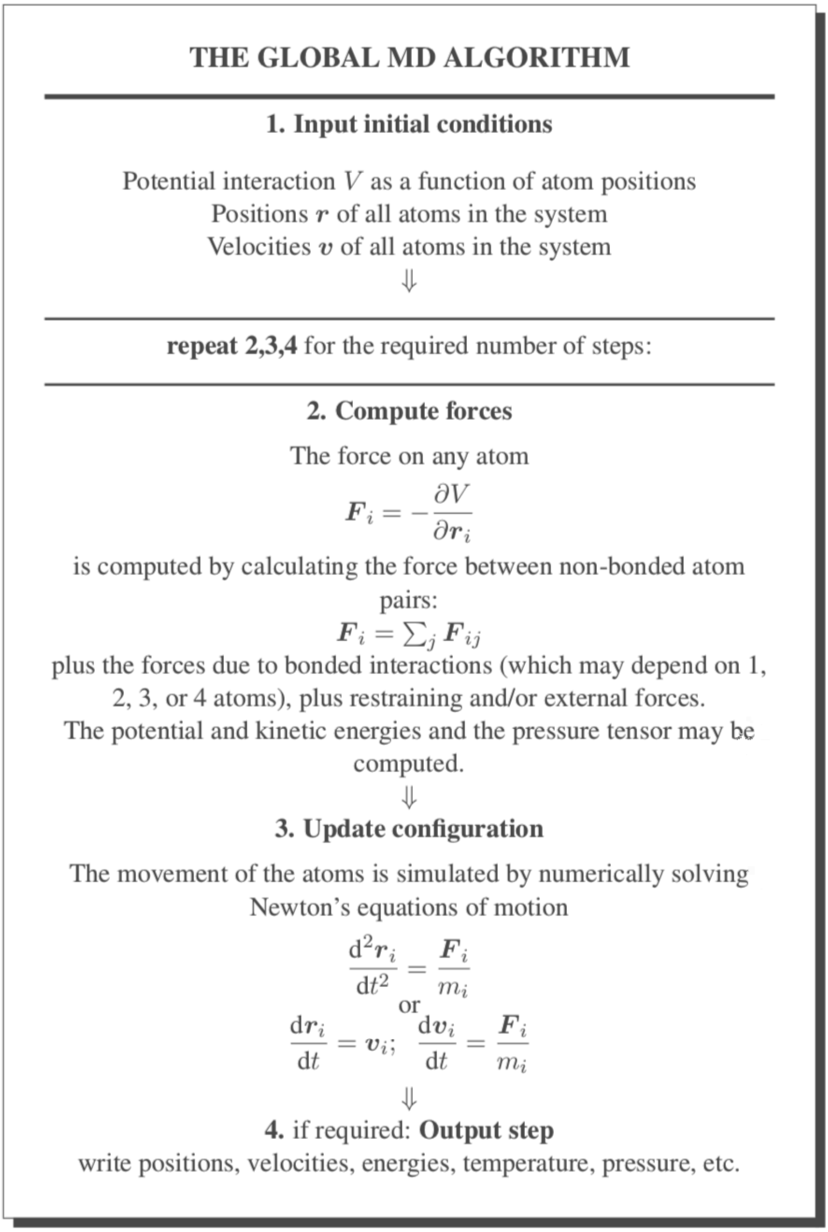
10.2.3 Thermostat

When we simulate a real thermodynamic system, we often want to keep the temperature fixed. It can be the NVT ensemble (fixed number of particles, volume and temperature) or the NPT ensemble (fixed number of particles, pressure, and temperature). GROMACS has more than one solution for keeping the temperature constant. All have their pros and cons, so they should be chosen based on the concrete problem. One of the simplest and most popular solutions is the Berendsen thermostat [74].

The method's idea is to couple the system to an external heat bath of a desired temperature T_0. Then the temperature of the system can be corrected

$$\frac{dT}{dt} = \frac{T_0 - T}{\tau} \ , \tag{10.12}$$

where τ is some time constant. Depending on the concrete needs, these corrections can be done more or less often. Practically it leads to scale the velocities by some factor, which is calculated from the system properties.



THE GLOBAL MD ALGORITHM

1. Input initial conditions

Potential interaction V as a function of atom positions
Positions $\boldsymbol{r}$ of all atoms in the system
Velocities $\boldsymbol{v}$ of all atoms in the system
⇓

repeat 2,3,4 for the required number of steps:

2. Compute forces

The force on any atom

$$\boldsymbol{F}_i = -\frac{\partial V}{\partial \boldsymbol{r}_i}$$

is computed by calculating the force between non-bonded atom pairs:

$$\boldsymbol{F}_i = \sum_j \boldsymbol{F}_{ij}$$

plus the forces due to bonded interactions (which may depend on 1, 2, 3, or 4 atoms), plus restraining and/or external forces.
The potential and kinetic energies and the pressure tensor may be computed.
⇓

3. Update configuration

The movement of the atoms is simulated by numerically solving Newton's equations of motion

$$\frac{\mathrm{d}^2\boldsymbol{r}_i}{\mathrm{d}t^2} = \frac{\boldsymbol{F}_i}{m_i}$$

or

$$\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} = \boldsymbol{v}_i; \quad \frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = \frac{\boldsymbol{F}_i}{m_i}$$

⇓

4. if required: **Output step**

write positions, velocities, energies, temperature, pressure, etc.

Figure 10.1. The generic molecular dynamics algorithm. The picture is taken from GROMACS user's manual [69].

11. Simulations: Unfolding pathways

Let us come back to the paper Marklund *et al.* [65] which we have mentioned in section 9.4 where we talked about the Single Particle Imaging (SPI) technique for imaging proteins. In that paper, the authors proposed a solution for keeping the sample orientation fixed. They showed that putting the protein in a strong electric field (the order of 10^4 V/nm) could align the molecule and hence fix 2 out of 3 degrees of freedom. If one increases the field even more, they can observe the protein's unfolding as the paper shows. And this is the subject of the study reported in Paper IV.

We were interested in learning how the unfolding process occurs. If the unfolding undergoes the same pathway every time (or most of the time), it is possible to make a video of this process! Indeed, the SPI technique requires many samples of the same structure since every single observation ruins the molecule (see sec. 9.4). If the unfolding happens in the same way, one can prepare a lot of samples of every state of the unfolding. And then, by doing imaging of every single step, one can make a video of the whole process.

11.1 Simulations

In our study, we use the protein ubiquitin (1UBQ [75]). This protein is very well studied in various papers, and what is most important for us, there is a study about mechanical [76] and thermal [77] pathways of the unfolding of ubiquitin. This allows us to compare our result of unfolding in the electric field with other unfolding methods.

Ubiquitin has 76 residues and 1280 atoms to simulate. We simulated it with the MD method in a vacuum without temperature coupling. We used 4 different electric fields, and for every field, we collected the statistics of 100 independent simulations. The evolution of radius of gyration (Eq. (4.11)) during 50 ns of simulations is presented in Figure 11.1 together with one standard deviation of average. The growing radius of gyration, as we remember from section 4.3.2, tells us about swelling of the molecule or unfolding (since we know it was folded before). Our lowest field (dark blue in the Figure 11.1) does not provide full thermalization on the simulations' length, but even there, we see the tendency. For the other 3 fields, 50 ns was enough for the radius of gyration to come to a plateau. Thus we see that the general idea of unfolding the protein with electric field works. What about the pathways of unfolding, i.e., in what order does the different parts of the protein break apart?

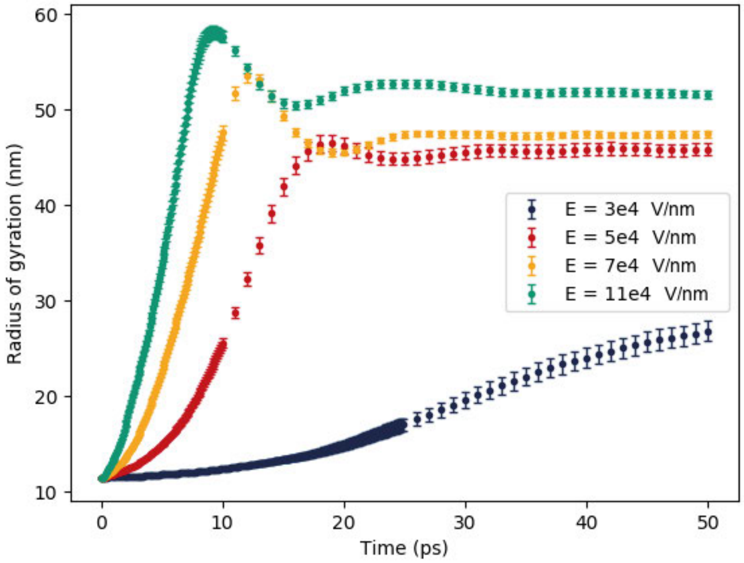


Figure 11.1. Dependence of the radius of gyration of ubiquitin on the time for different values of applied electric fields.

11.2 Pathways of unfolding

To answer this intriguing question, we have to look close at ubiquitin's secondary structure. Ubiquitin has one α-helix and one β-sheet formed by 5 β-strands. Following the notation suggested by Irback *et al.* [77] we will label the strands with Roman numerals as shown in Figure 11.2. The pair bonds between the strands are named as B C D and E (A stands for α-helix). The pathway of unfolding can be determined by the order of breaking BCDE couples.

To study the different pathways of the unfolding, we suggest an original idea: to build a graph. Every coupling between 2 strands can be either broken or not. This leads us to a binary description where 0 will denote not broken bond, and 1 is a broken bond. The 4 bonds of interest give us a 4-bit number and $2^4 = 16$ different states. Then we can build a graph of states, as shown in Figure 11.3, grouping the states with an equal number of broken bonds in one row. The source of the process is always 0000 because we start with the native conformation where, by definition, all the bonds are not broken. The next row of the map has 1 broken bond, hence 4 different states. The next row has 2 broken bonds: $C_4^2 = 6$ states. The 4th row has 3 broken (or 1 unbroken bond): 4 states again. Finally, fully unfolded configuration with all couplings are broken.

The last thing we have to define before we can come to the results is how we practically decide when a certain bond is broken or not. We will say that

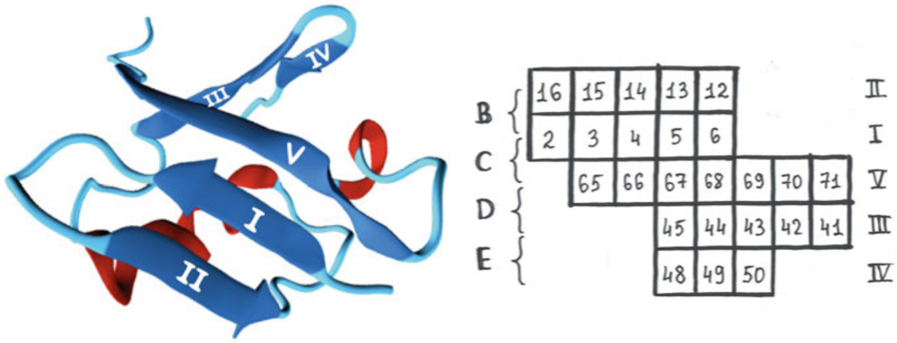


Figure 11.2. On the left: native conformation of ubiquitin with a schematic illustration of its secondary structure. On the right: the labeling of the β-strands and the connections between them.

the bond is broken if the distance between the center of masses of 2 β-strands is larger than 0.7 nm. This number was found empirically to make the graphs most pictorial. We can do this because we are interested in comparing different pathways, i.e., want to find a relative measure, not the absolute. In addition, we know that unfolding and breaking real chemical bonds happens with increasing distance between parts of the protein.

11.2.1 The graphs of unfolding

In Figure 11.4 we show the result for $E = 11$ V/nm. Let me explain it.

At every next moment of time, the 4-bit state can be the same or it can change. If it did not change, then we increase the size of the corresponding vertex on the graph. If the transition happens, we increase the thickness of the line for the corresponding transition. The line is green if the transition happens toward the bottom of the graph, i.e., some bond B, C, D, or E is broken. If some of those bonds are restored again, the transition goes toward the top of the graph. In this case, we add blue stripes for the corresponding transition line. The thickness of blue stripes are also proportional to the frequency of the "backward" transition.

In Figure 11.4, we present the results over all 100 simulations for all 50 ps. The time step of every simulation was 0.0005 ps. The state was written to the file every 100th step. Hence it means that for every run, we have $50/0.0005/100 = 1000$ transitions from time t to time t+dt. And over all 100 different simulations, we have 100 * 1000 = 10^5 transitions.

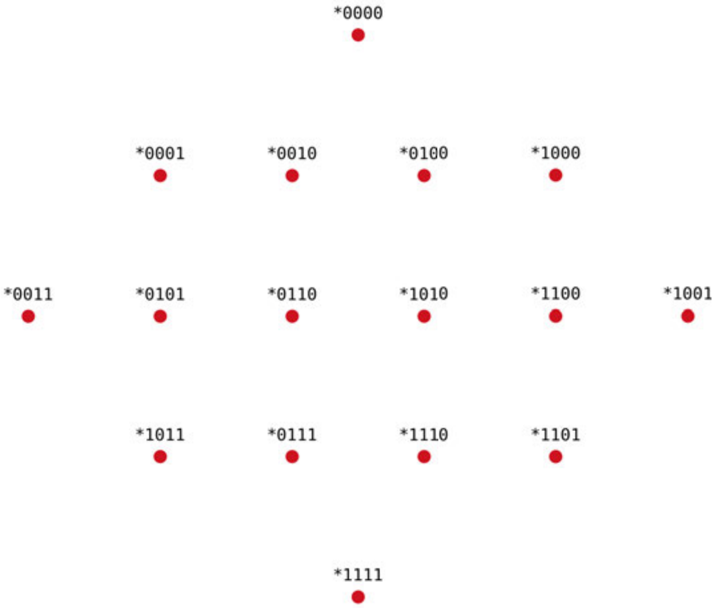


Figure 11.3. The structure of the graph of states. Every vertex represents the state with broken (1) or unbroken (0) bond between the corresponding β-strands *BCDE of ubiquitin as we label them in Figure 11.2.

The last thing to explain is the yellow vertices. We use them to denote that the corresponding state was the final state for at least one simulation. Again, the size of the point is proportional to the frequency of such cases or how many simulations ended there and how long they existed in this final state. The yellow dot can be smaller than the corresponding red dot or of the same size (then you will see the vertex as pure yellow, without red border around). The pure yellow vertex means that every time the simulations come to the corresponding state they stay there forever. Remember also that the number of transitions proportional to the area of the dot, not to the linear size (as for the case with the lines on the graphs). This means that even a thin red border on a big yellow dot can represent a fair amount of transitions.

Reading the first result

Now, let us show how we should read and understand the result in Figure 11.4. All the runs start with the native conformation of ubiquitin at 0000 state. We see that the initial state's size is larger than 1 000 transitions but smaller than 10 000 transitions. This means that either all configurations spend some time in 0000 before leaving the state or only one configuration stood there for a long

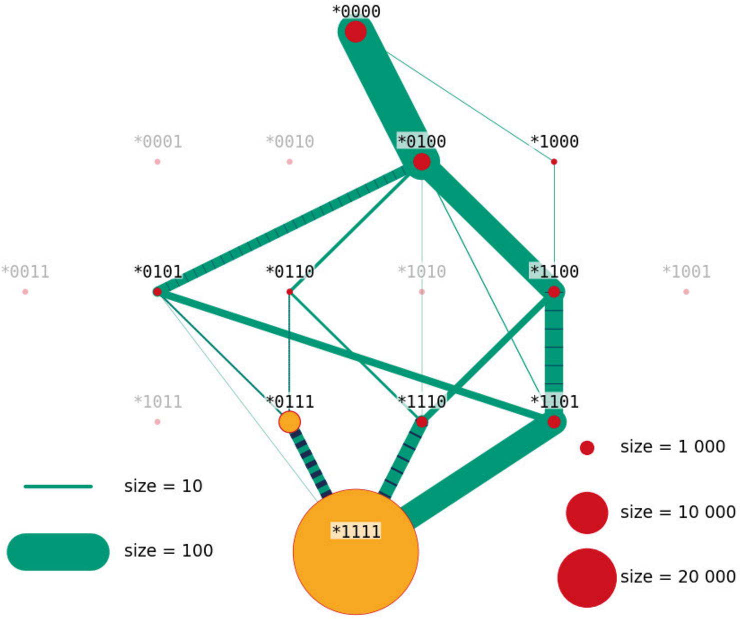


Figure 11.4. The graph of unfolding for $E = 11$ V/nm.

time, or, what is more likely, something in the between. We also see that the point is entirely red which tells us that independently of what scenario above was realized, all the simulations left the initial state eventually. There were 2 channels for that: to 0100 (breaking of C bond first) and 1000 (breaking B first). By the thickness of the lines, we see that the former path is much more preferable. Since there is no dark blue color in these lines, we understand that there were no backward transitions. Now we can conclude that almost all our 100 simulations undergo 0000 $\rightarrow$ 0100 as the first state transition.

The next transition happens quite fast after the first one, since the 0100 vertex is small. Here we have 5 different paths. First of all, we see 2 different transitions where 2 bonds were broken at the same time 0100 $\rightarrow$ 1110 (it did not go through 1010 since this point is faint in the Figure, which means the corresponding state was never visited) and 0100 $\rightarrow$ 1101. This happens because the field is very strong, and things can happen faster than we can resolve with our time grid. Now we should talk in terms of probabilities. The second step's preferable transition is 0100 $\rightarrow$ 1100 since the corresponding line is the thickest.

Following this path further we can see the leading channel in the graph: 0000 $\rightarrow$ 0100 $\rightarrow$ 1100 $\rightarrow$1101 $\rightarrow$ 1111. And by decoding the bit system back to the bonds between β-strands, we find the order of unfolding for the bonds: CBED.

Let us point out one more interesting observation you can get from the graph. The most likely final state is 1111: all bonds are broken. But you can see another state which is also a final state for some fraction of the simulations: 0111 where the B bond remains. The thickness of blue and green stripes on the corresponding link says that the forward and backward transitions are equally likely. But at the same time, 1111 is more likely than 0111. What does this mean? It means that most of the transitions 0111 → 1111 happens after the B bond was restored first! We can come to the same conclusion if we look at the links that come to 0111 from the top of the graph. There are only 2 transitions there: from 0101 and 0110. But the thickness of those 2 links together is smaller that 0111 → 1111 transition which can only be explained as we did above. Why is it an interesting observation? It tells us something about the B coupling. Even though the B bond is the second to break, it is the first to be restored after the protein is fully unfolded.

More results

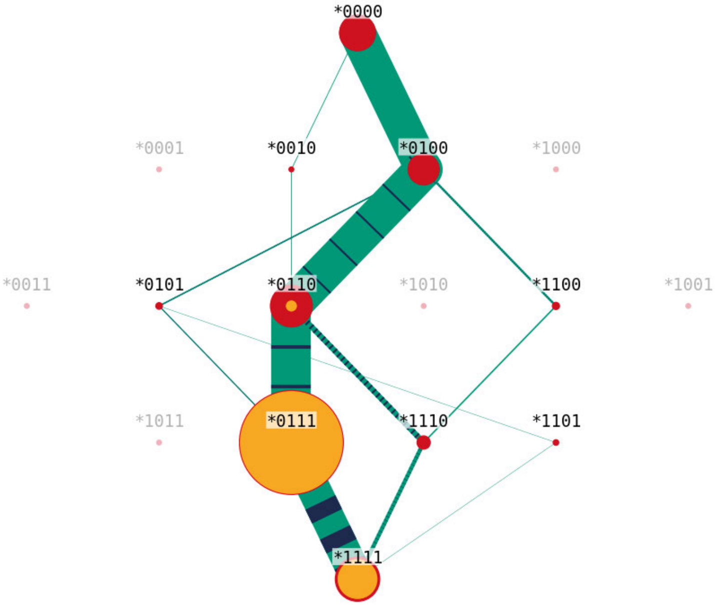


Figure 11.5. The graph of unfolding for $E = 5$ V/nm.

Now let see what happens with E = 5 V/nm. The graph is presented in Figure 11.5 and has the same scale as Figure 11.4. In this case, we can see even more clearly the preferable unfolding pathway. It begins with breaking the C bond as for E = 11 V/nm case, but then the tendency changes. Instead of breaking the B bond as before, now the D bond breaks: 0100 → 0110. The next transition 0110 → 0111 – E breaks like for the previous case. And finally, the D coupling is unfolded. The pathway now is CDEB (compare to CBED for E = 11 V/nm).

We see that for the strong and weak fields, the protein undergoes 2 different pathways of unfolding. This phenomenon we will discuss in the next section. But here, in the end, let us again look at the final states (yellow dots). We see the same kind of oscillation between 1111 and 0111 like for E = 11 V/nm, but this time the preferable final state is 0111, where the B bond is still there.

Discussion

I will present 2 more results for different electric field strengths we have. For your convenience, I plot them together with 2 old results we just discuss to have the field's values in order: Figure 11.6- 11.9. The scale and the structure of all graphs are the same.

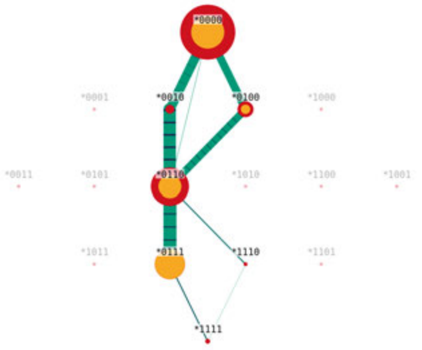


Figure 11.6. E = 3 V/nm.

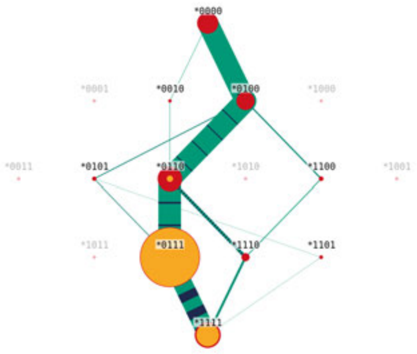


Figure 11.7. E = 5 V/nm.

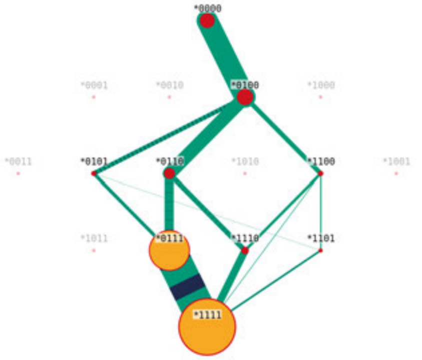


Figure 11.8. E = 7 V/nm.

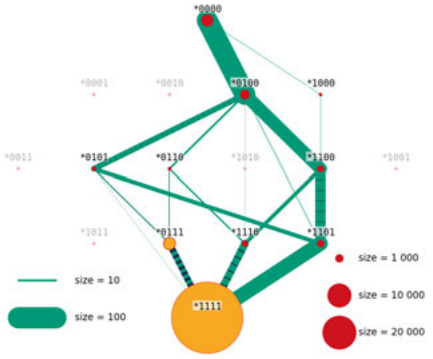


Figure 11.9. E = 11 V/nm.

One can see that the graph for E = 7 V/nm if something between the E = 5 V/nm and E = 11 V/nm cases. The leading channel still goes through 0100 → 0110 like for the weaker field, but the new pathway through 0100 → 1100 opens up and becomes dominant for the strongest field. Hence there is a smooth transition between the CDEB and CBED pathways.

The results for E = 3 V/nm shows that the structure is not unfolded completely. We have a lot of cases where none of the bonds were broken. This goes in line with the results for the radius of gyration in Figure 11.1. Notice that in the simulations where the unfolding began, the B bond was always preserved at the end.

So what is the summary of the study we just have discussed? The first bond to break is C. If we look again at the ubiquitin structure in Figure 11.2 we can see that C connects the beginning and the end of the chain. Thus this result should not surprise us. Other conclusions, however, are less trivial. The hardest coupling to break is B. If the protein is fully unfolded, then B would be the first to reconnect back. But if somehow you break the B bond before D and E (in our case with a very strong electric field), then the protein should undergo full unfolding before recreating it again.

Finally, we promised to compare our unfolding with the results for thermal and mechanical ones done by Irback *et al.* [77]. It turned out that our weak field E = 5 V/nm reproduces the thermal unfolding pathway: CDEB, while our strong electric field E = 11 V/nm reproduces the mechanical one: CBED. It makes perfect physical sense. The weak field behaves like “natural” unfolding by the temperature. And the strong electric field is similar to a “violent” mechanical unfolding.

11.3 Tertiary and secondary structures

The careful reader can ask a reasonable question: now we are talking about folding and unfolding but earlier in the thesis, we talked about collapsed and SARW phases. We always referred to the tertiary structure when we discussed the phase, and now we look at the β-sheet, which is a secondary structure. Is there any connection? This is an excellent question! And luckily, we already have developed the tool in Part III to answer it!

Let us start with plotting the RG observable VS scaling step in the same way as we did it in chapter 7 for Figure 7.2. The result is presented in Figure 11.10. Now we want to trace the evolution of the system with time, so having a 2D plot for classifying the phase is not convenient any more. What we need is one number instead.

From the discussion in Part III and Figure 7.2 we know that the critical thing to distinguish the phases is the sign of the RG flow curves. So let calculate the integral of the observable (or the area under the curve) in Figure 11.10 and compare this number to 0. If the integral is positive, then we are in the SARW phase, and a negative integral means the collapsed phase.

The dependence of the integral of the observable on time for different values of the electric field is presented in Figure 11.11. Note that the time axis goes

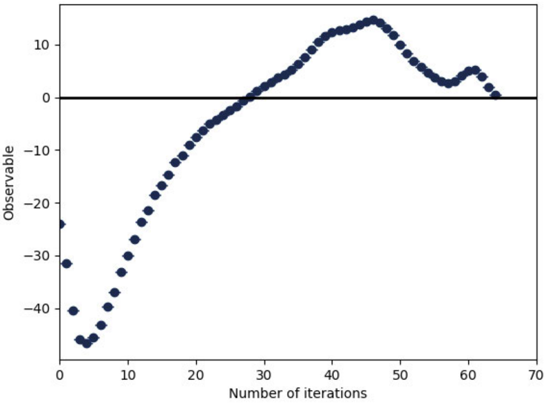


Figure 11.10. The RG observable from Eq. (6.12) for native ubiquitin conformation.

only up to 9 ps, not the full 50 ps as for Figure 11.1. What we immediately see is the different behavior of the green curve E = 11 V/nm. E = 3, 5 and 7 V/nm follows the same pattern, but E = 11 V/nm does not fit it because it has an initial hump. We also calculated electric field E = 9 V/nm to see if there is a smooth transition between those 2 different patterns. And it seems it is.

The result we got is fascinating! Two different pathways of unfolding which we cannot distinguish with the radius of gyration (Figure 11.1) show different behavior in our RG observable (Figure 11.11.)! To find 2 different unfolding pathways, we had to know everything about the secondary structure of the protein and implemented a special procedure for measuring the distances between the center of masses of single β-strands. The RG observable does not require more information than just atom positions. But it can "feel" secondary structures without any information about them, as our result shows.

Time graphs

There is one more interesting thing we can investigate. How does the secondary structure look at the transition point when the integral of the observable is equal to 0? At that point, there is a transition between the collapsed phase and the SARW phase.

We again will use our graphs from the previous section but in a bit different way. Before, we presented all the simulations at all time steps on the graph. We can do a similar thing but for only one time step, see Figures 11.12, 11.13. We took 100 simulations and analyzed the transitions for one fixed time step t where the integral of the observable in Figure 11.11 equals 0. It is done in a Markovian way: if the state at time $t - dt$ was different from the state at t, we link the states with a line. The convention for the colors is the same. The green line is the transition from the top of the graph toward the bottom, and the dark

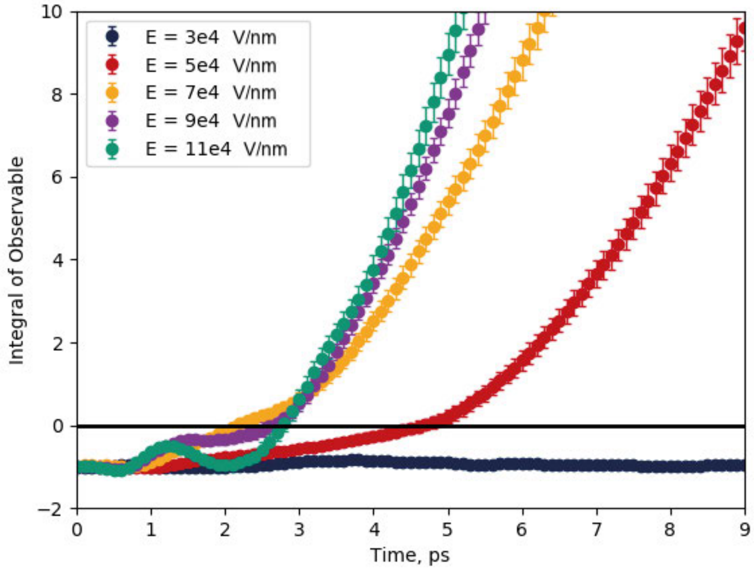


Figure 11.11. The evolution of the integral of the RG flow with time for various values of the electric field.

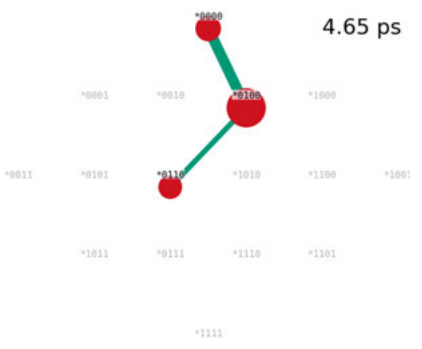


Figure 11.12. E = 5 V/nm.

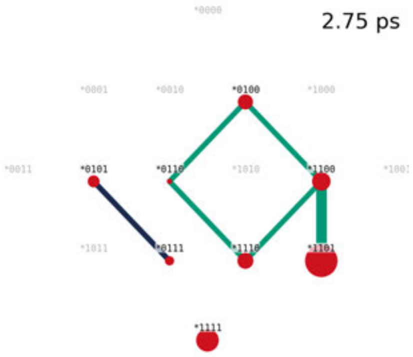


Figure 11.13. E = 11 V/nm.

blue line is vice versa. The size of the red points is proportional to the number of simulations in the corresponding state. The "sum" of all point sizes always corresponds to 100 simulations.

If we assume that the graphs represent the secondary structure and our RG analysis corresponds to the tertiary structure, we can make a nice conclusion from the results in Figures 11.12, 11.13. Both figures represent the time when the tertiary structure is broken, and how "far" the flow went toward the bottom of the graph reflects how much the secondary structure is disturbed. Thus we can say: *in a weak electric field, the tertiary structure breaks faster than the secondary structure. But in the strong electric field, the tertiary structure breaks slower than the secondary one.*

The conclusion we got shows that the RG method we developed in Part III can not only distinguish the phases but also distinguish the different pathways of the unfolding process. This works for ubiquitin. Does it work for other proteins? This is an interesting topic for further research.

12. Simulations: Orientation in a Time-Dependent Field

Let us look again at the already familiar paper of Marklund and others [65]. The authors consider a constant electric field in all their simulations. However, any real experimental implementation will give a ramping up of the field: when the field increases from 0 to a desired constant value. In this chapter, we will discuss our study of the orientation of ubiquitin in such ramping up fields.

12.1 Simulations

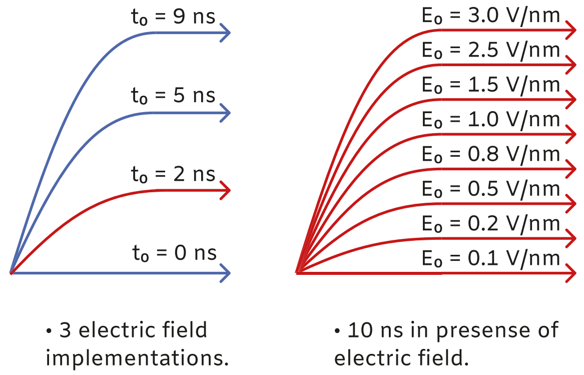


Figure 12.1. The schematic picture of the simulations. We increased the electric field from 0 to E_0 in the ramping time t_0. We did 4*8 = 32 simulations for different parameters E_0 and t_0 including 8 constant electric field (where t_0=0).

The GROMACS software which we use allows to add an electric field to the MD simulations in the form of a Gaussian envelop

$$E(t) = E_0 \exp\left(-\frac{(t-t_0)^2}{2\sigma^2}\right) \cos(\omega(t-t_0)) \;, \tag{12.1}$$

where t is time. Then the needed electric fields (as schematically pictured in Figure 12.1) can be compiled from two pieces

$$E(t) : \begin{cases} \omega = \pi/2t_0,\ \sigma = 1/\omega, & 0 < t \leq t_0 \\ \sigma \to \infty,\ \omega = 1/\sigma & t_0 < t \end{cases} . \tag{12.2}$$

Our aim is to test different final fields E_0 and ramping times t_0, as shown in Figure 12.1 to study the problem.

We will continue to work with ubiquitin in a vacuum because our target is SPI experiments. The starting configuration corresponds to the perpendicular orientation of the protein's dipole moment to the electric field. The length of all simulations is 10 ns. Every single run was repeated 10 times to collect the statistics.

12.2 Results

Since we are interested in the protein orientation in an electric field, we need some observable to measure the orientation. Let us introduce Θ the degree of orientation as

$$\Theta = 1 - \cos(\angle \mathbf{Ed}) \quad , \tag{12.3}$$

where $\angle \mathbf{Ed}$ is an angle between $\mathbf{E}$ – a vector of the applied electric field and $\mathbf{d}$ – the dipole moment of ubiquitin.

We start our simulations at $\Theta = 1$ and see if the observable vanishes as it should be when the protein is fully aligned with the electric field.

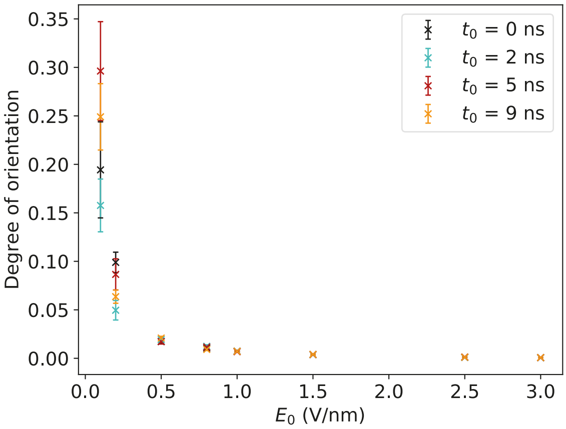


Figure 12.2. The degree of orientation Θ (Eq. (12.3)) as a function of the maximum value of the electric field E_0 and the ramping time t_0.

In Figure 12.2 we plot the orientation degree Θ after 10 ns of simulations averaged over 10 independent runs versus the maximum value of the electric field E_0 and as a function of the ramping time t_0 (including constant field $t_0 = 0$). Here we can see that all ramping times work almost the same. Only when the final field E_0 is very week smaller t_0 orients ubiquitin better. But starting with $E_0 = 0.5$ V/nm, the results seem not to depend on t_0 and all fields work quite well. Not an interesting result at all. But maybe we can measure the time of orientation somehow and see something interesting there?

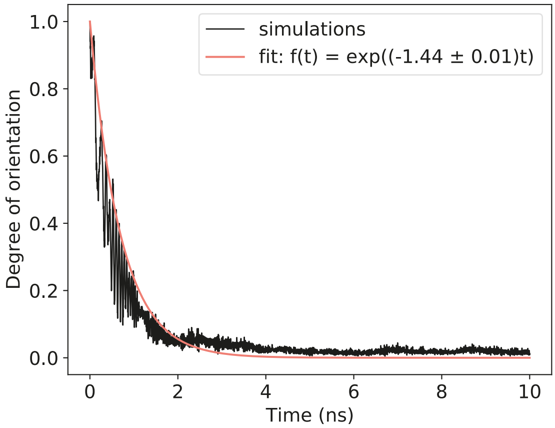


Figure 12.3. The time evolution of degree of orientation averaged over 10 independent runs for the parameters $E_0 = 0.5$ V/nm, $t_0 = 2$ ns (black line). The red line is the result of fitting it with the function $f(t) = \exp(-kt)$.

If we look at the evolution of Θ with time, we can see its exponential nature as shown with the black line in Figure 12.3. This means that we can fit it by the exponent $f(t) = \exp(-kt)$, as shown with the red line. Using the fitting parameters we can find the time τ when the protein looses 90% of its initial orientation, which means that it is 10% away from the desired orientation along the field.

$$\frac{f(\tau)}{f(0)} = \frac{\exp(-k\tau)}{1} = 0.1 \Rightarrow \tau = \frac{\ln 10}{k}. \tag{12.4}$$

Let us now plot time of orientation τ for all E_0 and t_0, Figure 12.4. There should be no surprise: the “slower” fields (with larger ramping time) require more time to orient the molecule in the field. But let us investigate the value of the electric field at the time of orientation τ. The plot is presented in Figure 12.5. We can see that for all E_0 and the ramping times t_0 (except the trivial case $t_0 = 0$) the orientation happens at the same value of the current electric field $E \approx 0.5$ V/nm. *Apparently, the orientation time for the protein depends only on the value of the electric field at the moment, but is independent of the maximum value E_0 and the ramping time t_0.*

The last question we should answer before celebrating the good result is the structure’s stability in our simulations. This time we only want to orient ubiquitin but not unfold it. So we should check that the fields we use do not change the original conformation significantly.

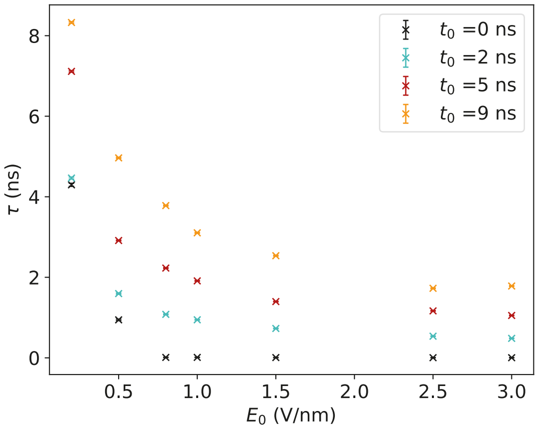


Figure 12.4. The time of orientation τ (Eq. (12.4)) as a function of the maximum value of the electric field E_0 and the ramping time t_0.

For investigating this we will calculate the *Root Mean Square Deviation*

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\mathbf{r}_i - \hat{\mathbf{r}}_i)^2}\ , \tag{12.5}$$

where $\mathbf{r}_i$ is a radius vector of i-th and $\hat{\mathbf{r}}_i$ is the radius vector of the same atom but in the reference configuration. N is the number of atoms. Unlike the radius of gyration, $RMSD$ is a relative measure: it shows the structure's deviation from the reference structure. In our case, we choose the reference structure to be the untouched ubiquitin or ubiquitin at $t = 0$.

In Figure 12.6 we plot the dependence of $RMSD(\tau)$ on E_0 and τ_0. When $RMSD$ is smaller than 0.1, the configuration can be considered the same as the reference. We see that almost everywhere (except the constant field for high E_0), the configuration does not change significantly.

Our result is optimistic for experimentalists. It gives much freedom to the producers of the orientation device. They should make an apparatus that can provide the required value of the electric field ($E \approx 0.5$ V/nm), but how fast it will be reached does not matter. The second good result of the stability of the structure we shall call *orientation before destruction* by analogy with *diffraction before destruction*. The latter, as we discussed in Part IV, makes SPI a possible alternative for protein imaging since the structural information can be obtained before the high-intensity laser destroys the sample. Our result shows that the protein's orientation in the strong electric field can be done before this strong field unfolds the sample, which is also necessary for being used in an SPI pipeline.

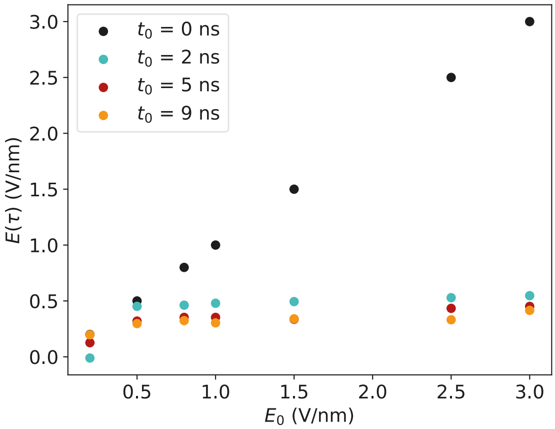


Figure 12.5. The electric field at the time of orientation $E(\tau)$ as a function of the maximum value of the electric field E_0 and the ramping time t_0.

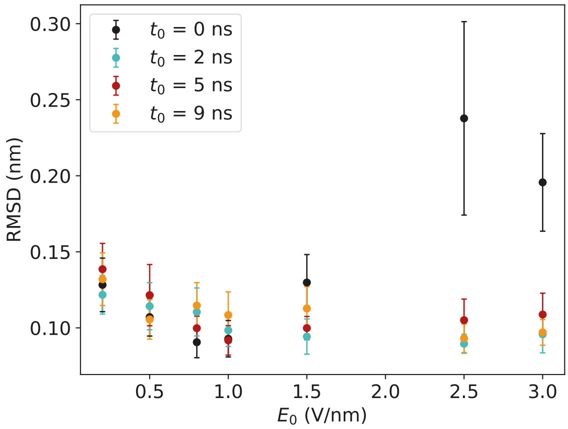


Figure 12.6. RMSD (Eq. (12.5)) as a function of the maximum value of the electric field E_0 and the ramping time t_0.

13. NMR data converter for GROMACS

In Part IV we mentioned the NMR technique to obtain the experimental data about protein geometry. NMR experimentalists have special databases and file formats to upload their results. MD simulations can gain from using NMR data. In Paper VI we present a Python package for converting NMR data to GROMACS format [78] and suggest the values for such simulations' parameters.

This work is not as interesting from a physical point of view as the things we discussed before. However, it contributes to the community as a new tool for improving MD simulations.

All the details can be found in Paper VI, no need to repeat them here.

Part VI:
Epilogue

14. What is next?

In the thesis, we have answered many questions about polymers and proteins simulation, behavior, and properties. Nevertheless, as with any good research, it opens up even more questions that wait for their answers. Let us state some of them clearly.

In Part II and corresponding **Paper I** we improved the effective Hamiltonian for chiral polymer chains presented in papers [37, 35] to obtain the real collapsed phase and several other remarkable phases and crossovers. The main question for future investigation is to find a way to describe real proteins with the model. Can we put in correspondence the data from PDB and 6 free parameters for each atom of our Hamiltonian? Then this description can be a new tool to study protein folding.

Part III and **Paper II** suggested a novel way for phase definition for a single polymer chain. The robust mathematical background, possible practical implementation, and clear usefulness of the method make it an interesting subject for further study. However, the new field it opens is so big that one could make another PhD thesis there. One can start with trying other scaling procedures and compare them with the existent results. In the best case, the result should be independent of the scaling procedure, but we do not know this unless we try. The role of the truncating parameter k in Eq. (7.1) is slightly discussed but not investigated enough. Can it indeed "probe" the polymer on different scales?

Can we use our RG method for the classification of proteins like it is done in Structural Classification of Proteins (SCOP) database or CATH Protein Structure Classification database, or Families of Structurally Similar Proteins (FSSP) database? Will we get similar classifications to those databases or different? Why?

We also used the RG method in part V for the unfolding of ubiquitin in a strong electric field. What about other proteins? Will it work for them as well? Can we find a universal way to define protein unfolding pathways with just the notion about all atoms' coordinates?

Paper III is also a first humble step in a statement of a new smoothing algorithm. An extensive study is needed there. However, that topic lies in the field of computer science rather than physics.

Papers IV and **V** applies molecular dynamics to simulate ubiquitin in a strong electric field. We have studied the unfolding of the protein and the orientation without significant structural damage. The next step there can be to study the orientational stability after the field is turned off. Then the question of the electric field boundary conditions in time will be entirely closed. The other obvious way to go is to try other structures and compare the results.

Finally, **Paper VI** is an example of improvement of the GROMACS framework. The power of any open software is in the collaboration of many different people around the world. Every even small improvement is an important step in the GROMACS development and can help many researchers. The more open and transparent GROMACS can be toward the potential collaborators, the more efficient the scientific and academic community can be. And then all together we can move toward a happy future.

15. Summary

Proteins are often called “molecules of Life”. This type of biomolecules build our tissues and define our health, mood and behavior. Half of the human body’s dry mass is proteins, which is approximately 10% of the total mass. Almost all the drugs we use target some protein. The proper functioning of any organism depends on the proper functions of its proteins. And the proper functioning of proteins depends not only on their chemical composition but also on their three-dimensional shape. The process of creating this shape is called protein folding. The same chemical structure allows several shapes. The correct one is called native conformation, while the others are misfolded conformations. In the best case, the latter works a bit worse than the protein in the native conformation. In the worst case misfolded proteins become toxic and lead to severe diseases like Alzheimer’s or cancer.

As we see, the protein study is vital for humanity. The better we understand the corresponding processes the faster and cheaper we can make new effective medicine when it is needed.

Any science can be divided into experimental and theoretical ones. I am working in the field of theoretical physics. The main instrument for my study is computer simulations. And there are different ways to simulate proteins. We started with the Monte Carlo method.

Monte Carlo is a statistical method which allows to find the ground state of a system. By varying the parameters of the effective model used for the system simulations, one can obtain its phase diagram. We did this for our homopolymer model of proteins in **Paper I** and build a three-dimensional phase portrait for it.

In the paper above, we used the classical definition of polymer phases, which work only for homopolymers. The real proteins in any representation are heteropolymers. In **Paper II** we developed a new method based on the renormalization group (RG) theory to distinguish phases of heteropolymers. This activity led to the quite interesting smoothing algorithm presented in **Paper III**.

In **Papers I**, **II** and **III**, I used my own software, which I specially developed for this research. It unites several programs in different programming languages and now is distributed openly under the Apache 2 license agreement.

Another way for protein simulation is the Molecular Dynamics (MD) approach. This is a “fair” way to simulate the dynamics of proteins, taking into account the real physics. We use one of the most popular software for MD –

GROMACS. But before talking about the next paper we have to look at experimental science.

We are interested in protein structure recognition. For most (80%) proteins ever experimentally resolved, scientists used the method of X-ray crystallography. The method's idea is to crystallize the sample and then expose it to X-ray radiation to get a diffraction pattern. Later, from this diffraction pattern, one can recreate the 3D structure of the protein. The obvious problem of X-ray crystallography with the application to proteins is the crystallization part. This process unavoidably changes the original conformation of the molecule. So the picture we get does not completely (or completely does not – we do not know) corresponds to the protein of interest.

The new era of lasers allows to do X-ray sampling of just one protein, with no need for crystallization. This conception got the name Single Particle Imaging (SPI). It was predicted that the very intense X-ray laser beam (which is needed for SPI) has to destroy the sample every time a new measurement is done. Then one has to prepare many samples and set up the sample delivery pipeline, for example, using an aerosol spray method. The downside of aerosol spray is that the delivered particles are orientated arbitrarily. This makes the experiment more complicated and expensive.

A smart way around the random particle orientation was suggested some time ago. Most of the biomolecules have a non-zero dipole moment, and can thus be orientated in an electric field. This orientation will fix 2 out of 3 degrees of freedom. If one increases the applied electric field, the protein starts to unfold. The question is: does the unfolding happen the same way every time? If it is so, then SPI can record the unfolding video for the first time in history. We can prepare many samples of the same state of this process and thus resolve one frame of unfolding. By repeating this procedure for every unfolding frame, we can compile the movie of the phenomenon. This question is the topic of **Paper IV**, where we investigate different pathways of the unfolding for the protein ubiquitin. For this purpose I suggest a special sort of graph representation. In this thesis, I also connect the unfolding with the phase definition (**Paper I**) using the RG method developed in **Paper II**.

In **Paper V** we investigate how a time-dependent electric field orients ubiquitin. Apparently, it becomes orientated when the electric field reaches a particular value, but how fast this value is reached is unimportant.

The last **Paper VI** is very technical. We developed a Python code to integrate the experimental data from the Nuclear Magnetic Resonance database with GROMACS. We also suggested the optimal parameters for further MD simulations. The code is also open source under the Apache 2 license agreement.

16. Sammanfattning på svenska

Proteiner kallas ofta för “livets molekyler”. Denna sorts molekyler bygger upp vår vävnad och avgör vår hälsa, humör och beteende. Hälften av människans torra massa är proteiner, vilket är ungefär 10% av vår totala massa. Nästan alla läkemedel riktar in sig på något protein och organismer är beroende av att deras proteiner fungerar som de ska för att själva fungera. Att proteinerna fungerar som de ska beror inte bara på deras kemiska uppsättning utan också på deras tre-dimensionella form. Den process genom vilken denna form skapas kallas för proteinveckning. Samma kemiska struktur tillåter flera olika former och den rätta formen kallas för dess naturliga tillstånd, medan övriga tillstånd är felveckade. I bästa fall fungerar de senare lite sämre än det naturliga tillståndet. I värsta fall är felveckade proteiner giftiga och kan leda till allvarliga sjukdomar såsom Alzheimers och cancer.

Som vi ser är studier av proteiner livsviktiga för mänskligheten. Ju bättre vi förstår proteiners natur, desto snabbare och billigare kan vi utveckla nya effektiva mediciner då de behövs.

Varje vetenskap kan delas in i experimentell och teoretisk vetenskap. Jag jobbar inom teoretisk fysik. Det främsta verktyget för mina studier är datorsimuleringar och proteiner kan simuleras på olika sätt. Vi börjar med Monte Carlo.

Monte Carlo är en statistisk metod som gör det möjligt att hitta ett systems grundtillstånd. Genom att variera parametrarna i den effektiva modellen som används för att simulera systemet så kan ett fasdiagram hittas. Vi gjorde detta för homopolymermodellen för proteiner i **Paper I** och skapade ett tredimensionellt fasdiagram.

I artikeln ovan använde vi den klassiska definitionen av polymerfaser, vilket endast fungerar för homopolymerer. Verkliga proteiner är heteropolymerer. I **Paper II** utvecklade vi en ny metod som baseras på renormeringsgruppsteori för att särskilja olika faser för heteropolymerer. Denna forskning ledde också till den intressanta utjämningsalgoritmen som presenteras i **Paper III**.

I **Paper I**, **II** och **III** använde jag min egenutvecklade mjukvara som var specifikt utvecklad för denna forskning. Den sammanlänkar flera program i flera olika programmeringsspråk och finns nu fritt tillgänglig online under en Apache 2 licens.

Ett annat sätt att simulera proteiner på är med hjälp av molekylärdynamik (MD). Detta är ett “rättvisande” sätt att simulera proteindynamik som reflekterar den verkliga fysikaliska utvecklingsprocessen. Vi använder en av de mest populära mjukvarorna för MD – GROMACS. Men innan vi går in på de följande artiklarna behöver vi nämna den experimentella vetenskapen.

Vi är intresserade av att känna igen proteinstrukturer. För de flesta (80%) proteiner där strukturen bestämts experimentellt så har forskare använt röntgenkristallografi. Metoden bygger på att provet kristalliseras och sedan utsätts för röntgenstrålning för att skapa ett diffraktionsmönster. Från detta diffraktionsmönster kan sedan proteinets 3D-struktur återskapas. Det uppenbara problemet med röntgenkristallografi är kristalliseringen. Denna process leder ofrånkomligen till en förändring av molekylens struktur. Så den bild vi får är inte helt (eller helt enkelt inte alls – detta vet vi inte) samma som den proteinstruktur som vi är intresserade utav.

Den nya lasereran tillåter röntgenbestrålning av enskilda proteiner utan krav på kristallisering. Denna idé har fått benämningen enpartikelsavbildning, eller single particle imaging (SPI) på engelska. Det har förutspåtts att väldigt stark röntgenstrålning förstör provet varje gång en mätning utförs. Därför måste många prov förberedas för att sätta upp en provleverans-pipeline. För att åstadkomma detta är det möjligt att använda en aerosolmetod. Nackdelen med denna metod är att de levererade partiklarna kan vara orienterade i godtycklig riktning, vilket gör experimentet mer komplicerat och dyrt.

Ett smart sätt att kringgå problemet med godtycklig orientering föreslogs för ett tag sedan. De flesta biomolekyler har ett nollskilt dipolmoment och kan därför orienteras i ett elektriskt fält. Denna orientering fixerar två av tre frihetsgrader. Höjs det elektriska fältet ytterligare börjar proteinet att vecka ut sig. Frågan är om denna utveckning sker på samma sätt varje gång. Skulle det vara så, då skulle SPI för första gången någonsin kunna spela in utveckningsprocessen på film. Vi kan förbereda många prov i samma tillstånd av denna process och på så vis framkalla en bild av ett enskilt steg i denna process. Genom att upprepa denna procedur för varje steg så kan vi skapa en film av detta fenomen. Denna fråga studeras i **Paper IV**, där vi undersöker de olika sätt som proteinet ubiqiuitin veckar ut sig på. För detta ändamål föreslår jag en speciel slags grafrepresentation. I denna avhandling kopplar jag också samman utveckningen med fasdefinitionen i **Paper I** med hjälp av RG metoden som utvecklats i **Paper II**.

I **Paper V** undersöker vi hur ett tidsberoende elektriskt fält orienterar ubiquitin. Tydligen orienteras det när det elektriska fältet når en viss styrka men hur fort detta värde uppnås är mindre viktigt.

Den sista artikeln **Paper VI** är mera teknisk. Vi utvecklade en Python-kod för att integrera experimentella data från NMR-databasen (kärnmagnetisk resonans) i GROMACS. Vi föreslog också de optimala parametrarna för MD-simuleringar. Koden är också open source under en Apache 2 licens.

17. Краткое содержание на русском языке

Белки нередко называют молекулами жизни. Они формируют наши ткани и определяют наше здоровье, настроение и поведение. Половина сухой массы тела человека составляют протеины, что приблизительно 10% от общей массы. Почти все лекарства, которые мы используем, нацелены на какой-то конкретный белок. Правильная работа любого живого организма зависит от правильной работы его белков. В свою очередь, правильная работа белков зависит не только от их химической структуры, но и от их трехмерной формы. Процесс формирования этой самой формы называется свертыванием белков. Одна и та же химическая структура может приводить к разным формам. Правильная свертка белка называется нативной структурой. Остальные конформации являются неправильными. В лучшем случае они работают немного хуже, чем нативная структура, а в худшем — становятся токсичными и вызывают ряд серьезных болезней, в том числе болезнь Альцгеймера и рак.

Как видно, исследование белков очень важно для человечества. Чем лучше мы понимаем процессы, с ними связанные, тем быстрее и дешевле мы сможем производить новые необходимые лекарства.

Изучение всех естественных наук можно разделить на экспериментальное и теоретическое. Я работаю в сфере последнего, и мой главный инструмент в этом — компьютерные симуляции. Существует несколько подходов для симуляции белков, но давайте начнем с метода Монте-Карло.

Монте-Карло это статистический метод, который позволяет довольно быстро (в сравнении с методом молекулярной динамики, который мы обсудим далее) найти основное состояние системы. Варьируя параметры эффективной модели, которой описывается данная система, можно найти ее фазовую диаграмму.

В статье I мы сделали это для нашей модели однородной цепи и получили ее трехмерный фазовый портрет. В этой статье мы использовали классическое определение фаз полимера, которое работает только в случае однородной цепи. Но реальные белки в любом представлении являются неоднородными. В статье II мы разработали новый способ определения фаз неоднородных цепей, используя идею ренормгрупп (РГ). Эта работа привела к довольно интересному алгоритму сглаживания, описанному в статье III.

В статьях I, II и III я использовала свое собственное программное обеспечение, специально написанное для этих целей. Оно состоит из нескольких отдельных программ, написанных на разных языках программирования, и находится в открытом доступе по лицензии Apache 2.

Другой способ симуляции белков — метод молекулярной динамики (МД). МД — это «честный» способ симуляции, так как он основан на настоящей физике процесса. Для такой работы мы используем один из самых популярный МД пакетов — GROMACS. Но перед тем, как начать рассматривать следующие статьи, давайте обратимся к эксперименту.

Нас интересует распознавание структуры белков, т.е. их формы.

Для большинства (80%) белков, когда-либо исследованных экспериментально, ученые использовали метод рентгеновской кристаллографии. Идея заключается в том, чтобы кристаллизованный образец облучить рентгеном для получения дифракционной картины. Эта картина нужна для воссоздания трехмерной структуры белка. Очевидная проблема этого метода — при кристаллизации первоначальная структура протеина неизбежно меняется. Так что картинка, которую мы получаем в итоге, не совсем (или совсем не — мы не знаем) соответствует искомому белку.

Новая эра лазеров дает возможность рентгеновского анализа для одного протеина без необходимости кристаллизации. Этот метод получил название одночастичная рентгенография (ОР) (Single Particle Imaging). Было предсказано, что лазерный луч, который в этом случае должен быть очень интенсивным, неизбежно приводит к разрушению образца в каждом новом измерении. Это значит, что для каждого эксперимента необходимо подготовить множество образцов и наладить их поставку к лучу лазера. Например, для этого можно использовать технику аэроспрея. Однако, у такого метода есть недостаток — исследуемая частица расположена в пространстве случайным образом, что усложняет эксперимент и делает его более дорогим.

Совсем недавно был предложен способ решения проблемы случайной ориентации.

Большинство биологических молекул имеют ненулевой дипольный момент. Это позволяет ориентировать молекулы в электрическом поле, сокращая число степеней свободы системы с трех до одной. Если начать увеличивать электрическое поле, то белок начнет разворачиваться. И возникает вопрос: один и тот же белок разворачивается всегда одинаково или по-разному? Если одинаково, то мы смогли бы записать видео разворачивания белка впервые в истории. Приготовив много образцов одной и той же стадии разворачивания, мы могли бы использовать ОР для воссоздания трехмерной модели белка. Повторяя этот процесс для каждого момента времени разворачивания, мы получили бы запись всего этого явления. Этот вопрос — тема исследования статьи IV, где мы изучаем разные пути разворачивания белка юбикутина. Для этого я предлагаю использовать метод графов.

В данной диссертации я показываю связь разворачивания юбикутина (статья IV) с определением фаз (статья I), пользуясь методом РГ, разработанным в статье II.

В статье V мы изучаем, как переменное электрическое поле ориентирует протеин юбикутин. Оказывается, полная ориентация происходит, когда электрическое поле достигает определенного значения, а как быстро это происходит — неважно.

Последняя статья VI в большей мере техническая — мы разработали код на Питоне для интеграции экспериментальных данных, полученных методом ядерного магнитного резонанса, с пакетом GROMACS. Также мы предложили оптимальные параметры для дальнейших МД симуляций. Этот код тоже находится в открытом доступе под лицензией Apache 2.

Acknowledgements

First of all, I would like to express my deepest appreciation to the people who made this journey possible. Thanks to professor **Antti Niemi** who invited me to this magic world of proteins and to my new home — Sweden. Your excellent physical intuition and broad knowledge let to my most exciting projects. I am deeply indebted to my main supervisor **Johan Nilsson** who gave me the most valuable things: money and freedom. You allowed me to find my way, and I am very grateful for that. Special thanks to **Maksim Ulybyshev**, the great physicist and the great person. Your contribution cannot be overstated.

My success would not have been possible without the support and nurturing of my new second supervisor **Carl Caleman**, who gave me a chance and invited me to his wonderful scientific group. Our work together was not long but very productive. I would like to extend my sincere thanks to my collaborators **Erik Marklund**, professor **David van der Spoel** and **Emiliano De Santis**. It was my pleasure to work with you. Thanks should also go to the whole international **MS SPIDOC group**, an inspiring collaboration of bright and purposeful people.

Many thanks to professor **Jonas Fransson**, whose doors were always open for me. It was helpful to talk to you.

I must also thank **Swedish National Infrastructure for Computing** and especially **Uppmax** for all the core hours they provided. Besides, their support team is amiable and was always ready to help.

Thanks to the whole **Materials Theory division**, it was five happy years together. I also had the great pleasure of working with people from **Molecular and Condensed Matter Physics division** and from **BMC**.

Most of my sweet memories are connected to a little red villa, which became my second home for this time. This house no longer belongs to the University, but is still in our hearts. I am very grateful to all the villa's inhabitants! We had so much fun together. Exploring the attic and finding hundreds of fly wings, our BBQ outside altogether, music evenings, walks in the forest, watching the World Cup in the meeting room, eating Surströming, and many more. And special thanks for our exciting lunches. Thanks to **Henning** — I learned economics and became an investor only because you were so engaging to discuss this. Thanks to **Dushko**, who knows a lot of exciting stuff and is not scared to think big. We had several great ideas about future projects, like the anti-matter factory with magnetic field stabilization or the factory for cleaning the air from CO2 and compressing the side product to diamonds. The world is just not ready for our progressive ideas yet! Thanks to **Juan David**, who is always ready to help. I am glad we found a way to be friends. Thanks to **Francesco**, who brought a piece of sunny Italy to the villa. The guitar music you played in our office was fantastic. Special thanks to room 120: **Johann** and **Tomas**. The knowledge that you are my officemates helped me come to the office even when things went not well. With Tomas, I could discuss whatever, while Johann reminded us that not all our ideas are socially appropriate. I very much appreciate **Annica**, **Lucia**, **Mahdi**, **Ola**, **Fariborz**, **Manuel**, **Mahroo**, **Andreas**, **Yaroslav Oscar**, **Anders**, **Attila**, **Erik**, **Jorge**, **Paramita** — you made the villa so bright and diverse.

Many thanks to **Charlotta** and **Katharina**. You supported me in a tough period in my life. I would like to extend my sincere thanks to **Nina** and **Olga**. My lovely Russian friends who I was destined to meet only in Sweden.

I was raised and educated in Moscow and want to send my sincere gratitude to the people there. Thanks to my master supervisor **Oleg Pavlovsky**, you were the first who showed me what it means to be a scientist. Thanks to professor **Konstantin Sveshnikov** who was always kind to me despite my not perfect grades. Thanks to the **Faculty of Physics at Lomonosov Moscow State University** and all my teachers, they are still preparing world-class specialists. I would like to acknowledge my fellow students. You made the years at the university in Moscow much better. Especially, I would like to mention **Sasha Kazantsev**. You helped me to understand everything I had not got in the lectures.

Thanks to my **High School 1173** and especially **Anselma Nikolaevna Dubinina**, who is unfortunately not with us anymore. My path as a physicist started there with one phone call from her when she invited me to the school. Anselma Nikolaevna gathered the most talented and inspiring teachers in physics and mathematics to educate children and prepare them for the best Moscow universities. This great woman changed the lives of hundreds of people allowing a better future for them. Your contribution cannot be overstated. Thanks should also go to my classmates. I am glad that with many of you, we are still in contact. Special thanks to **Alina Petrushina**. Your unparalleled support with learning physics and mathematics led me to where I am now.

I would also like to extend my gratitude to **Marita and Thomas Björnssons** for their warm hospitality (and for their elder son). Most of this thesis was written in their wonderful house. It is always a holiday for me to visit you and the whole **Björnssons family**.

Thanks to my dear **Kristofer**. You made me a better programmer, a better artist, a better physicist, and a better person. I am happy to move forward together toward our bright future. You are my best team member, and I love you.

And finally, thanks to **my family**, for those who have gone and for those who have arrived. I am deeply indebted to my mother, **Olga Sinelnikova**, and my late father, **Boris Sinelnikov**, for everything I have achieved. You are my permanent and unwavering support; I love you. Special thanks to my brother

Ilya. You are cool; I am very proud of you. Thank you **Luba**, **little Sasha**, and **little Sonya**. I am glad our family grows.

I am happy and proud to know all the people above and to have them as a part of my life.

References

[1] World Health Organisation, “Archived: Who timeline - covid-19.” `https://www.who.int/news/item/27-04-2020-who-timeline---covid-19`, 2020.

[2] E. Callaway, H. Ledford, G. Viglione, T. Watson, and A. Witze, “COVID and 2020: An extraordinary year for science,” *Nature*, vol. 588, no. 7839, pp. 550–552, 2020.

[3] C. B. Anfinsen, “Principles that govern the folding of protein chains,” *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[4] L. Pauling, R. B. Corey, and H. R. Branson, “The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain,” *Proceedings of the National Academy of Sciences*, vol. 37, no. 4, pp. 205–211, 1951.

[5] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, “The shape and structure of proteins,” in *Molecular Biology of the Cell. 4th edition*, Garland Science, 2002.

[6] C. Hyeon and D. Thirumalai, “Capturing the essence of folding and functions of biomolecules using coarse-grained models,” *Nature Communications*, vol. 2, no. 1, pp. 1–11, 2011.

[7] D. J. Selkoe, “Folding proteins in fatal ways,” *Nature*, vol. 426, no. 6968, p. 900, 2003.

[8] L. M. Luheshi, D. C. Crowther, and C. M. Dobson, “Protein misfolding and disease: from the test tube to the organism,” *Current Opinion in Chemical Biology*, vol. 12, no. 1, pp. 25–31, 2008.

[9] P. J. Thomas, B.-H. Qu, and P. L. Pedersen, “Defective protein folding as a basis of human disease,” *Trends in Biochemical Sciences*, vol. 20, no. 11, pp. 456–459, 1995.

[10] C. M. Dobson, “The structural basis of protein folding and its links with human disease,” *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 356, no. 1406, pp. 133–145, 2001.

[11] L. C. Walker and H. LeVine, “The cerebral proteopathies,” *Molecular Neurobiology*, vol. 21, no. 1-2, pp. 83–95, 2000.

[12] A. Mukherjee and C. Soto, “Prion-like protein aggregates and type 2 diabetes,” *Cold Spring Harbor perspectives in medicine*, vol. 7, no. 5, p. a024315, 2017.

[13] H. Ecroyd and J. A. Carver, “Crystallin proteins and amyloid fibrils,” *Cellular and Molecular Life Sciences*, vol. 66, no. 1, p. 62, 2009.

[14] A. N. Bullock and A. R. Fersht, “Rescuing the function of mutant p53,” *Nature Reviews Cancer*, vol. 1, no. 1, pp. 68–76, 2001.

[15] J. C. Price, S. Guan, A. Burlingame, S. B. Prusiner, and S. Ghaemmaghami, “Analysis of proteome dynamics in the mouse brain,” *Proceedings of the National Academy of Sciences*, vol. 107, no. 32, pp. 14508–14513, 2010.

[16] B. H. Toyama and M. W. Hetzer, “Protein homeostasis: live long, won’t prosper,” *Nature Reviews Molecular Cell Biology*, vol. 14, no. 1, pp. 55–61, 2013.

[17] A. Sinelnikova, “Polymer chain Monte Carlo.” `https://github.com/Anny-Moon/PCMC`, 2017.

[18] A. Sinelnikova, “Polymer chain analyzer.” `https://github.com/Anny-Moon/PCA`, 2017.

[19] A. Sinelnikova, “Plotter for PCA.” `https://github.com/Anny-Moon/PlotterPyPCA`, 2018.

[20] A. Sinelnikova, “pdb2xyz.” `https://github.com/Anny-Moon/pdb2xyz`, 2017.

[21] B. Hess, C. Kutzner, D. van der Spoel, and E. Lindahl, “GROMACS 4: algorithms for highly efficient, load-balanced, and scalable molecular simulation,” *Journal of Chemical Theory and Computation*, vol. 4, no. 3, pp. 435–447, 2008.

[22] N. Metropolis and S. Ulam, “The Monte Carlo method,” *Journal of the American Statistical Association*, vol. 44, no. 247, pp. 335–341, 1949.

[23] S. M. Ulam, *Adventures of a Mathematician*. Univ of California Press, 1991.

[24] W. Krauth, *Statistical mechanics: algorithms and computations*, vol. 13. OUP Oxford, 2006.

[25] E. Schuster, “Buffon’s needle experiment,” *The American Mathematical Monthly*, vol. 81, no. 1, pp. 26–29, 1974.

[26] B. Gnedenko and A. Kolmogorov, *Limit distributions for sums of independent random variables*. Cambridge, Massachusetts: Addison-Wesley, 1954.

[27] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, “Equation of state calculations by fast computing machines,” *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[28] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[29] W. Kühnel, *Differential Geometry: Curves-Surfaces-Manifolds*, vol. 16. American Mathematical Soc., 2006.

[30] S. Hu, M. Lundgren, and A. J. Niemi, “Discrete Frenet frame, inflection point solitons, and curve visualization with applications to folded proteins,” *Physical Review E*, vol. 83, no. 6, p. 061908, 2011.

[31] A. Y. Grosberg and A. R. Khokhlov, *Statistical Physics of Macromoleculas*. American Institute of Physics, 1994.

[32] H. K. Onnes, “Expression of the equation of state of gases and liquids by means of series.,” in *Through Measurement to Knowledge*, pp. 146–163, Springer, 1991.

[33] P. J. Flory, *Principles of polymer chemistry*. Cornell University Press, 1953.

[34] A. J. Niemi, “Phases of bosonic strings and two dimensional gauge theories,” *Physical Review D*, vol. 67, no. 10, p. 106004, 2003.

[35] U. H. Danielsson, M. Lundgren, and A. J. Niemi, “Gauge field theory of chirally folded homopolymers with applications to folded proteins,” *Physical Review E*, vol. 82, no. 2, p. 021910, 2010.

[36] M. Chernodub, S. Hu, and A. J. Niemi, “Topological solitons and folded proteins,” *Physical Review E*, vol. 82, no. 1, p. 011916, 2010.

[37] M. Chernodub, M. Lundgren, and A. J. Niemi, “Elastic energy and phase structure in a continuous spin Ising chain with applications to chiral homopolymers,” *Physical Review E*, vol. 83, no. 1, p. 011126, 2011.

[38] A. Krokhotin, S. Nicolis, and A. J. Niemi, “Long range correlations and folding angle with applications to α-helical proteins,” *The Journal of Chemical Physics*, vol. 140, no. 9, p. 03B605_1, 2014.

[39] “Research collaboratory for structural bioinformatics protein data bank (RSCB PDB).” https://www.rcsb.org.

[40] P.-G. De Gennes and P.-G. Gennes, *Scaling concepts in polymer physics*. Cornell university press, 1979.

[41] L. P. Kadanoff, “Scaling laws for ising models near T_c,” *Physics Physique Fizika*, vol. 2, no. 6, p. 263, 1966.

[42] K. P. Sullivan, P. Brennan-Tonetta, and L. J. Marxen, *Economic Impacts of the Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank*, 2017.

[43] W. H. Bragg and W. L. Bragg, “The reflection of x-rays by crystals,” *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 88, no. 605, pp. 428–438, 1913.

[44] J. C. Kendrew, G. Bodo, H. M. Dintzis, R. Parrish, H. Wyckoff, and D. C. Phillips, “A three-dimensional model of the myoglobin molecule obtained by x-ray analysis,” *Nature*, vol. 181, no. 4610, pp. 662–666, 1958.

[45] A. Klug, “From macromolecules to biological assemblies (Nobel Lecture),” *Angewandte Chemie International Edition in English*, vol. 22, no. 8, pp. 565–582, 1983.

[46] J. Drenth, *Principles of protein X-ray crystallography*. Springer Science & Business Media, 2007.

[47] I. I. Rabi, J. R. Zacharias, S. Millman, and P. Kusch, “A new method of measuring nuclear magnetic moment,” *Physical Review*, vol. 53, no. 4, p. 318, 1938.

[48] F. Bloch and E. M. Purcell, “The Nobel Prize in physics 1952,” *Nature*, vol. 170, pp. 911–912, 1952.

[49] E. Ruska, “The development of the electron microscope and of electron microscopy (Nobel Lecture),” *Angewandte Chemie International Edition in English*, vol. 26, no. 7, pp. 595–605, 1987.

[50] M. Adrian, J. Dubochet, J. Lepault, and A. W. McDowall, “Cryo-electron microscopy of viruses,” *Nature*, vol. 308, no. 5954, pp. 32–36, 1984.

[51] P. Emma, R. Akre, J. Arthur, R. Bionta, C. Bostedt, J. Bozek, A. Brachmann, P. Bucksbaum, R. Coffee, F.-J. Decker, *et al.*, “First lasing and operation of an ångstrom-wavelength free-electron laser,” *Nature Photonics*, vol. 4, no. 9, p. 641, 2010.

[52] E. A. Schneidmiller and M. V. Yurkov, “Photon beam properties at the European XFEL (December 2010 revision),” tech. rep., Deutsches Elektronen-Synchrotron (DESY), 2011.

[53] D. A. Deacon, L. Elias, J. M. Madey, G. Ramian, H. Schwettman, and T. I. Smith, “First operation of a free-electron laser,” *Physical Review Letters*, vol. 38, no. 16, p. 892, 1977.

[54] H. Motz, “Applications of the radiation from fast electron beams,” *Journal of*

Applied Physics, vol. 22, no. 5, pp. 527–535, 1951.

[55] M. J. Bogan, W. H. Benner, S. Boutet, U. Rohner, M. Frank, A. Barty, M. M. Seibert, F. Maia, S. Marchesini, S. Bajt, *et al.*, "Single particle X-ray diffractive imaging," *Nano Letters*, vol. 8, no. 1, pp. 310–316, 2008.

[56] R. Neutze, R. Wouts, D. van der Spoel, E. Weckert, and J. Hajdu, "Potential for biomolecular imaging with femtosecond X-ray pulses," *Nature*, vol. 406, no. 6797, pp. 752–757, 2000.

[57] H. N. Chapman, C. Caleman, and N. Timneanu, "Diffraction before destruction," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1647, p. 20130313, 2014.

[58] J. Bielecki, F. R. Maia, and A. P. Mancuso, "Perspectives on single particle imaging with X-rays at the advent of high repetition rate X-ray free electron laser sources," *Structural Dynamics*, vol. 7, no. 4, p. 040901, 2020.

[59] M. M. Seibert, T. Ekeberg, F. R. Maia, M. Svenda, J. Andreasson, O. Jönsson, D. Odić, B. Iwan, A. Rocker, D. Westphal, *et al.*, "Single mimivirus particles intercepted and imaged with an X-ray laser," *Nature*, vol. 470, no. 7332, pp. 78–81, 2011.

[60] T. Ekeberg, M. Svenda, C. Abergel, F. R. Maia, V. Seltzer, J.-M. Claverie, M. Hantke, O. Jönsson, C. Nettelblad, G. van der Schot, *et al.*, "Three-dimensional reconstruction of the giant mimivirus particle with an X-ray free-electron laser," *Physical Review Letters*, vol. 114, no. 9, p. 098102, 2015.

[61] A. Munke, J. Andreasson, A. Aquila, S. Awel, K. Ayyer, A. Barty, R. J. Bean, P. Berntsen, J. Bielecki, S. Boutet, *et al.*, "Coherent diffraction of single Rice Dwarf virus particles using hard X-rays at the Linac Coherent Light Source," *Scientific Data*, vol. 3, no. 1, pp. 1–12, 2016.

[62] H. K. N. Reddy, C. H. Yoon, A. Aquila, S. Awel, K. Ayyer, A. Barty, P. Berntsen, J. Bielecki, S. Bobkov, M. Bucher, *et al.*, "Coherent soft X-ray diffraction imaging of coliphage PR772 at the Linac Coherent Light Source," *Scientific Data*, vol. 4, p. 170079, 2017.

[63] G. van der Schot, M. Svenda, F. R. Maia, M. Hantke, D. P. DePonte, M. M. Seibert, A. Aquila, J. Schulz, R. Kirian, M. Liang, *et al.*, "Imaging single cells in a beam of live cyanobacteria with an X-ray laser," *Nature Communications*, vol. 6, no. 1, pp. 1–9, 2015.

[64] M. F. Hantke, D. Hasse, F. R. Maia, T. Ekeberg, K. John, M. Svenda, N. D. Loh, A. V. Martin, N. Timneanu, D. S. Larsson, *et al.*, "High-throughput imaging of heterogeneous cell organelles with an X-ray laser," *Nature Photonics*, vol. 8, no. 12, pp. 943–949, 2014.

[65] E. G. Marklund, T. Ekeberg, M. Moog, J. L. P. Benesch, and C. Caleman, "Controlling protein orientation in vacuum using electric fields," *The Journal of Physical Chemistry Letters*, vol. 8, pp. 4540–4544, 2017.

[66] "MS-SPIDOC Horizon2020." www.ms-spidoc.eu.

[67] B. J. Alder and T. E. Wainwright, "Phase transition for a hard sphere system," *The Journal of Chemical Physics*, vol. 27, no. 5, pp. 1208–1209, 1957.

[68] M. N. Rosenbluth and A. W. Rosenbluth, "Further results on Monte Carlo equations of state," *The Journal of Chemical Physics*, vol. 22, no. 5, pp. 881–884, 1954.

[69] E. Lindahl, M. Abraham, B. Hess, and D. van der Spoel, "GROMACS 2020.3

manual," July 2020.

[70] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. A. Swaminathan, and M. Karplus, "CHARMM: a program for macromolecular energy, minimization, and dynamics calculations," *Journal of Computational Chemistry*, vol. 4, no. 2, pp. 187–217, 1983.

[71] W. L. Jorgensen and J. Tirado-Rives, "The OPLS [optimized potentials for liquid simulations] potential functions for proteins, energy minimizations for crystals of cyclic peptides and crambin," *Journal of the American Chemical Society*, vol. 110, no. 6, pp. 1657–1666, 1988.

[72] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives, "Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids," *Journal of the American Chemical Society*, vol. 118, no. 45, pp. 11225–11236, 1996.

[73] W. D. Cornell, P. Cieplak, C. I. Bayly, I. R. Gould, K. M. Merz, D. M. Ferguson, D. C. Spellmeyer, T. Fox, J. W. Caldwell, and P. A. Kollman, "A second generation force field for the simulation of proteins, nucleic acids, and organic molecules," *Journal of the American Chemical Society*, vol. 117, no. 19, pp. 5179–5197, 1995.

[74] H. J. Berendsen, J. v. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," *The Journal of Chemical Physics*, vol. 81, no. 8, pp. 3684–3690, 1984.

[75] S. Vijay-Kumar, C. E. Bugg, and W. J. Cook, "Structure of ubiquitin refined at 1.8 Å resolution," *Journal of Molecular Biology*, vol. 194, no. 3, pp. 531–544, 1987.

[76] A. Irbäck, S. Mitternacht, and S. Mohanty, "Dissecting the mechanical unfolding of ubiquitin," *Proceedings of the National Academy of Sciences*, vol. 102, no. 38, pp. 13427–13432, 2005.

[77] A. Irbäck and S. Mitternacht, "Thermal versus mechanical unfolding of ubiquitin," *PROTEINS: Structure, Function, and Bioinformatics*, vol. 65, no. 3, pp. 759–766, 2006.

[78] A. Sinelnikova, S. Patel, and D. van der Spoel, "Read NMR data files for proteins and generate GROMACS input files."

[79] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Annals of Mathematical Statistics*, vol. 29, pp. 610–611, 1958.

[80] G. Casella, C. P. Robert, M. T. Wells, *et al.*, "Generalized accept-reject sampling schemes," in *A festschrift for herman rubin*, pp. 342–347, Institute of Mathematical Statistics, 2004.

Index

Appendix A. Generation of τ

The torsion angles are distributed normally as one can see from the Hamiltonian in Eq. (4.32):

$$\begin{aligned} \mu &= \frac{a(b\kappa_i + 1)}{c(d\kappa_i + 1)}, \\ \sigma^2 &= \frac{T}{c(d\kappa_i + 1)}. \end{aligned} \tag{17.1}$$

The algorithm for generating random numbers according a normal distribution $\mathcal{N}(\mu, \sigma^2)$ was proposed in the original paper [79] by Box and Muller and is named after the authors. Here it is:

1. Generate independently ξ_1 and ξ_2 according to the uniform distribution $\mathcal{U}(0, 1)$;
2. Define

$$\begin{aligned} Z_1 = R\cos\Theta = \sqrt{-2\ln\xi_1}\cos(2\pi\xi_2), \\ Z_2 = R\sin\Theta = \sqrt{-2\ln\xi_1}\sin(2\pi\xi_2). \end{aligned} \tag{17.2}$$

3. Finally

$$\begin{aligned} \widetilde{Z_1} &= Z_1\sigma^2 + \mu, \\ \widetilde{Z_2} &= Z_2\sigma^2 + \mu. \end{aligned} \tag{17.3}$$

Then $(\widetilde{Z_1}, \widetilde{Z_2})$ is a pair of independent random variables from the desired distribution.

Appendix B. Generation of κ

The generation of κ is a more complicated question than generation of τ. We have a double well potential for the curvature angles in the Hamiltonian in Eq. (4.32), which means that κ_i should be generated according corresponding double-peak distribution (we drop the index i from κ to increase readability):

$$P(\kappa) \sim e^{f(\kappa)} \equiv e^{-A\kappa^4+B\kappa^2+C\kappa} \, , \tag{17.4}$$

where

$$\begin{cases} A = q > 0 \\ B = ab\tau_i + 2qm^2 - \frac{c}{2}d\tau_i^2 - 2 \\ C = 2(\kappa_{i+1} - \kappa_{i-1}) \end{cases} . \tag{17.5}$$

The algorithm we will use to sample according to a random distribution is called *rejection sampling* [80].

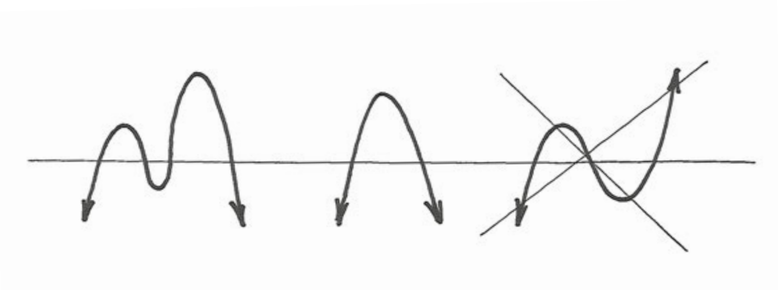


Figure 17.1. The asymptotic behavior of the function $f(\kappa)$.

Assuming we want to generate $x \in (x_1, x_2)$ from the distribution $P(x)$. The algorithm is as follows.

1. Generate independently x and ξ according to the uniform distribution $\mathcal{U}(0,1)$;
2.

$$\begin{aligned} &\text{if} \quad \xi < \frac{P(x)}{\max\limits_{(x_1,x_2)} P(x)} \qquad \Rightarrow \qquad \text{accept } x, \\ &\text{otherwise} \qquad \Rightarrow \qquad \text{reject } x. \end{aligned} \tag{17.6}$$

In fact, it is the same idea as the accept-reject algorithm in Metropolis (see sec. 3.6.2).

Let us investigate the asymptotic behavior of the exponential function $f(\kappa)$. Since the parameter A is always positive we can find

$$\begin{cases} f(+\infty) = -\infty \\ f(-\infty) = -\infty \end{cases} . \tag{17.7}$$

The function is continuous so the number of roots has to be even. Hence there can be only 2 or 4 roots which corresponds to 2 or 1 maxima like one can see in Figure 17.1. It is a good point, otherwise one of the extremum would be minimum and we cannot associate the function with a probability.

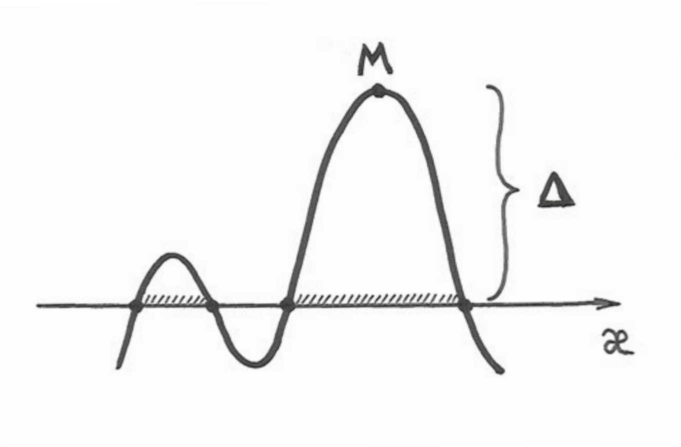


Figure 17.2. Roots of the function $f(\kappa)$. M is a global maximum and Δ is an offset variable.

The first step is to find the maximum

$$\begin{cases} f'(\kappa) = -4A\kappa^3 + 2B\kappa + C = 0 \\ f''(-\infty) = -12A\kappa^2 + 2B < 0 \end{cases} \Rightarrow f_{\text{max}} \equiv M. \tag{17.8}$$

Then we should manually set an offset Δ to limit the integral (see Figure 17.2):

$$|f(\kappa) - M| < \Delta. \tag{17.9}$$

Thus we will get 2 or 4 roots.

• If 2 roots, 1 maximum

This case is presented in Figure 17.3 in the left plot. The probability in this case:

$$\begin{aligned} \widetilde{P}(\kappa) &= \frac{1}{N} e^{+f(\kappa)}, \qquad \kappa \in (\kappa_1, \kappa_2) \\ N &= \int_{\kappa_1}^{\kappa_2} e^{f(\kappa)}. \end{aligned} \tag{17.10}$$

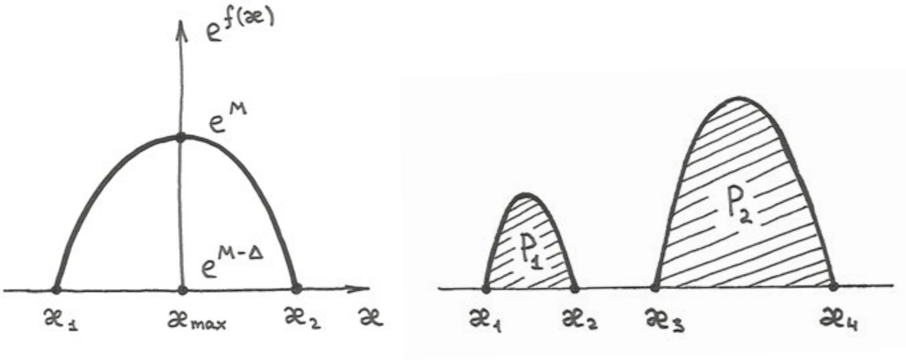


Figure 17.3. On the left: the case of one maximum, two roots. On the right: the case of two maxima, four roots.

And maximum of the distribution is

$$\widetilde{M} = \widetilde{P}(\kappa_{\max}) = \frac{1}{N} e^M. \tag{17.11}$$

And now we can use reject sampling: generate ξ_1 and ξ_2 from the uniform distribution $\mathcal{U}(0,1)$ and set

$$x = \kappa_1 + \xi_1(\kappa_2 - \kappa_1), \tag{17.12}$$

and accept it only

$$\text{if} \quad \xi_2 < \frac{\widetilde{P}(x)}{\widetilde{M}} \quad \Rightarrow \quad \text{accept } x. \tag{17.13}$$

• If 4 roots, 2 maxima

The probability to be in one interval or in the other one is proportional to the areas how it is shown in Figure 17.3 at the right

$$\begin{cases} N_1 = \int_{\kappa_1}^{\kappa_2} e^{f(\kappa)} \\ N_2 = \int_{\kappa_3}^{\kappa_4} e^{f(\kappa)} \end{cases} \Rightarrow \begin{cases} P_1 = \frac{N_1}{N_1+N_2} \\ P_2 = \frac{N_2}{N_1+N_2} \end{cases} \Rightarrow P_1 + P_2 = 1. \tag{17.14}$$

Here we generate ξ from the uniform distribution $\mathcal{U}(0,1)$ and pick the first interval (κ_1, κ_2) if $\xi < P_1$ or pick the second interval (κ_3, κ_4) otherwise.

Since we got only one interval we can follow the instructions in the previous case.

Acta Universitatis Upsaliensis

Digital Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 2015

Editor: The Dean of the Faculty of Science and Technology

Distribution: publications.uu.se
urn:nbn:se:uu:diva-434275

ACTA
UNIVERSITATIS
UPSALIENSIS
UPPSALA
2021