NORTHWESTERN UNIVERSITY

Protein Folding under an Applied Force

A DISSERTATION

SUBMITTED TO THE GRADUATE SCHOOL
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

for the degree

DOCTOR OF PHILOSOPHY

Field of Mechanical Engineering

By

Pengfei Diao

EVANSTON, ILLINOIS

December 2008

ABSTRACT

Protein Folding under an Applied Force

Pengfei Diao

In this thesis we study protein folding with external force. In Part I we simulate the folding or unfolding procedure based on the two-state model. For constant-velocity experiments, the simulation results fit the experimental results. For constant-force experiments, we introduce cooperativity between domains to explain the different behavior between single domain folding and multi-domain folding. To make the simulation results fit the experimental results, in Part I we have to set the folding rates to unreasonable values. In Part II we solve this problem. In this part we present a model in which folding is comprised of smaller motions individually acted upon by the applied force. The model naturally explains how cooperativity arises when an applied force is present and why observed folding times become less sensitive to the external force as force increases, while the two-state model predicts the opposite trend.

Acknowledgements

My deepest gratitude goes to Professor Seth Lichter, my supervisor, for his constant encouragement and guidance. Not only as a student I have learned knowledge and techniques from a perfect teacher, but also as a person I learned what how noble one should be. Without his help, this thesis could not have reached its present form. Second, I would like to express my heartfelt gratitude to my parents, for their support over so many years.

Table of Contents

Part II

List of Tables

List of Figures

Part I

CHAPTER 1

Purpose of this work

The folding of proteins is one of the main problems of molecular biology. There are many possible configurations for any protein. But proteins can normally find the correct native state among all the possible choices. Misfolded proteins can cause diseases such as Alzheimer's disease, BSE (Mad Cow disease) and cancers [10, 35, 41]. Understanding protein folding can not only help in biomedical applications, but also in the design of protein-sized machines. Therefore, it is important to understand how proteins change from one configuration into another. There are many ways to make proteins change their state: force, temperature, chemical, etc. The purpose of this work is to formulate models to understand the detailed procedure of protein conformational change when subject to applied forces.

CHAPTER 2

Background about protein folding

2.1. Purpose of this section

Here we introduce some background knowledge about protein folding for the following reasons. First, we wish to formulate our model based on the facts already known. And, where assumptions need to be made, they can be based on observation. Second, we can be informed as to which aspects of the protein-folding procedure remain unclear; then we can attempt different guesses and test which one best agrees with experimental results. Sections 2.2 to 2.6 are based on Branden and Tooze [**7**].

2.2. General introduction

Proteins are composed of amino acids connected into a linear sequence by peptide bonds. The amino acid sequence of the protein's polypeptide chain is called its primary structure. Different regions of the sequence form local regular secondary structures, such as alpha (α) helices or beta (β) strands. The tertiary structure is formed by packing such structural elements into one or several compact globular units called domains. The final protein may contain several polypeptide chains arranged in a quaternary structure.

Different amino acids have different side chains, but they all have the same backbone *(-NH-CH-C=O-)*. If we ignore the side chains, a protein can be regard as a chain composed of amino acid backbones, *(-NH-CH-C=O-)*. The amino acid backbone is very rigid. This leads to the approximation in which each amino acid backbone is regarded as a rigid bar. Therefore, a protein can be regarded as a chain of bars. The three-dimensional unfolded structure of a protein is like a coiled chain.

Some proteins can spontaneously fold into the native state, while others need help, such as the help of enzymes.

"During the folding procedure, once most of the secondary structures are formed, the protein has looser tertiary structure, called the molten globular state, than the native state. The protein can spontaneously compact from the molten globular state into the native state." [**7**]

A protein, even in its native state, is not static. There are still fluctuations in its configuration. A native-state protein can also change its state. Some environment changes, such as pH or temperature, can make proteins in solution change from an active native state into an inactive denatured (unfolded) state. The energy difference between these two states is generally about 5-15 kcal/mol. As a comparison, the energy of one hydrogen bond is 2-5 kcal/mol.

To describe events in the protein world, it is useful to introduce its typical scales. The energy unit we used in the previous section is kcal/mol, which is about 7 nm·pN, or 1.7 k_BT. k_BT is the other unit frequently used, where k_B is Boltzmann constant, and T is temperature in Kelvin. At room temperature, 1 $k_BT \simeq 4.1$ nm·pN. The average mass (weighted by frequency of occurrence) of one amino acid is approximately 10^{-22} gram. The range of amino acid mass is from a low of 57 Da (glycine) to a high of 186 Da (tryptophan), a variation by a factor of over three [**15**]. The length of an amino acid can be characterized by the maximum distance between from an atom in one residue to the corresponding atom in the adjacent residue. This length is 0.380 nm for residues in the *trans* configuration, which is the most prevalent. Another characterization is the maximal distance projected along the end-to-end length when the protein is fully extended. This length is 0.363 nm [**15**]. (In the calculations of Part I, we use 0.375 nm as the length of one amino acid. In Part II, we use 0.38 nm. In Part I we use the 0.375 nm because it is an average of the amino acids' lengths 0.37-0.38 nm. In Part II we change it into the more frequently used value 0.38 nm. The difference is about 2%.)

2.3. Energy difference between two states

The energy difference between native and denatured states comes from two parts. One is enthalpy, which is the energy stored in the noncovalent bonds: hydrophobic interactions, hydrogen bonds and ionic bonds. The energy stored in noncovalent bonds is on the order of 5 kcal/mole (1.2 k_BT). These bonds are weaker than the covalent bonds, but what we are interested in is changes in energy. Therefore in the protein folding procedure, these bonds, which change during folding, contribute a greater net effect than the covalent bonds which, mostly, remain unchanged.

Some molecules, such as oil molecules, have the tendency to avoid water. This type of interaction is called a hydrophobic interaction. Some of the amino acid side chains, those that are nonpolar, behave similarly: their interaction with polar molecules like water are poor. Therefore, most of the nonpolar residues in globular proteins tend to stay inside of the protein, while the polar molecules, such as aspartic acid and lysine are on the surface of the protein, make contact with the solvent. The nonpolar residues tend to pack closely and displace water molecules.

Some proteins have charged groups. Oppositely-charged groups can form ionic bonds. Ionic interactions are highly sensitive to changes in pH and salt concentration.

Polar molecules, such as water molecules, have two different partially-charged regions. One is negative partially charged (the oxygen atom in water), the other is positively partially charged (the hydrogen atoms in water). Thus when water molecules are close together, their positive and negative regions can interact with nearby molecules. This interaction is called a hydrogen bond.

The covalent bonds are the same in both unfolded and folded states except for disulfide bonds formed between cysteine residues for which the native state has lower energy. The energy difference stored in all the bonds between the two states can reach several hundred kcal/mol.

The other part of the energy difference comes from entropic effects. The native state is highly ordered, and the denatured state is disordered. Therefore, the denatured state has lower energy from this part. The energy difference from entropy between these two states can also reach several hundred kcal/mol.

The total energy difference between the two states is called the free energy. The free energy difference is about 5-15 kcal/mol: much smaller than the energy difference from bonds or entropy. The low free energy difference is a severe complicating factor for predictions of possible native states. But the marginal stability is biologically important in order to degrade and synthesize proteins easily, and for some proteins to easily undergo allosteric changes.

2.4. Dynamics of folding

A protein can have many possible configurations. A simple guess might say that a protein searches through its configurations, until it finds the lowest energy state, *i.e.*, native state, and then that it will remain there. In 1968, Cyrus Levinthal showed by a simple calculation that this folding procedure is impossible. He assumed that every amino acid has three possible configurations:α helix configuration, β sheet configuration and loop configuration. Also, assume that an amino acid can change from one configuration into another in a very short time, one picosecond. A chain with 150 residues would take 10^{48} years to go through all the possible configurations. The actual folding time is in the range of 0.1–1000 seconds. This implies that the folding process can not be a totally random search. There must exist means which eliminate some possible configurations from the folding procedure. Say, once a native bond is formed, it can not be broken. These formed bonds will greatly decrease the possible configurations for the next step.

It is difficult to investigate the folding procedure experimentally or theoretically. It is hard to examine the intermediate states experimentally because they have short lifetimes. Theory and simulation also meet some trouble, even if we can calculate the free energies of all the configurations. Because we will find that the one of lowest energy, might not be the native state. The protein may stay in some deep local minimum and not fold further because it is stopped by a large energy barrier.

The molten globule state is an intermediate state. For some proteins, the whole folding procedure can be considered in a few steps. In some proteins the first step is from the unfolded state to the molten globule state. This change occurs quickly, normally in a few milliseconds. The next step, which can last up to one second, is from the molten globule state to the final state. The following schematic, Fig. 2.1, shows the free energy diagram for this type of folding procedure.

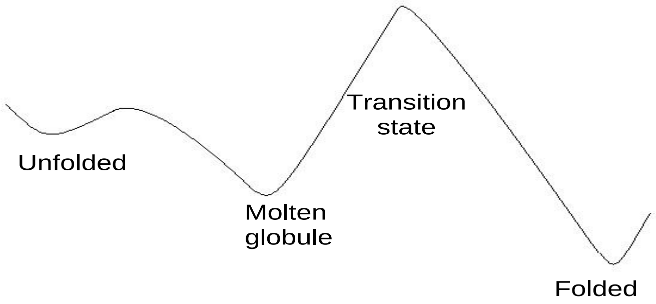


Figure 2.1. Free energy diagram for one type of folding procedure. The vertical-axis is the free energy, and the horizontal-axis shows the different stages during the folding procedure.

How the unfolded state collapses to the molten globule state is the main mystery of protein folding. What is the driving force which makes a protein fold from a random chain into the neighborhood of its native structure?

There is very little change in free energy by forming the internal hydrogen bonds of α helices and β sheets, because in the folding procedure the bonds with water molecules which have similar energy are broken, and then form the new bonds.

On the other hand, there is a large free energy change by bring hydrophobic side chains out from water and move into the interior region. This hydrophobic effects might be an important reason for protein folding, since the energy changes associated with them are large.

In order to fully understand the folding procedure, it is helpful to know all the intermediate states structurally and energetically. Alan Fersht developed a protein engineering procedure for this study. He investigated the effects on the energetics of folding of single-site mutations in a protein of known structure. If the mutation destabilizes some state, say mutation of an α helix destabilizes the intermediate state, then it means the helix structure has already been formed in the intermediate state. By this method, he has found that the molten globule state already has not only most of the native secondary structures but also the native-like relative positions of the α helix and β sheet as well as the relative positions of the β strands within the sheet.

2.5. Discussion–1

Based on the findings above, we can imagine the following folding procedure. First, stretch the protein in a random coil configuration, and then release the external force. The protein should begin to fold freely. In a few milliseconds, it should reach the molten globular state. If we draw a extension-time curve, the extension should have an abrupt collapse soon after the release of the force. But since at this stage all the structures are near the native-like position, the folding from the molten globule state to the native state will not cause a sizeable extension change. Therefore the extension-time curve should look like that shown in Fig. 2.2. At first, the extension is an almost horizontal line (fixed external stretching force), followed by a collapse (on release of the force), and then almost another nearly horizontal line.

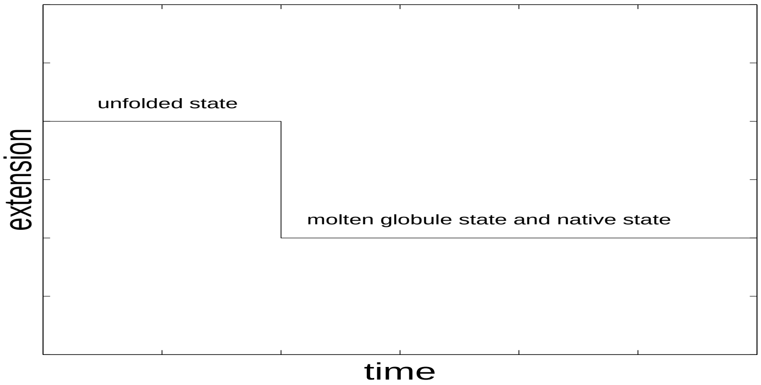


Figure 2.2. Simplified extension-time folding curve based on folding into a molten globule state. On release of the force, the protein rapidly collapses into the molten globule state. Thereafter, there would be only a small change in length, approximated here as the final horizontal line.

Fernandez and Li performed experiments to measure the the extension-time curve of ubiquitin [**21**]. The ubiquitin molecule was first stretched into an unfolded state. The force applied by the atomic force microscope (AFM) was then quickly reduced to a small fixed value at which folding occurred. Fig. 2.3 shows one of their results. The protein was stretched into a coil state under the application of a large force, approximately 100 pN. At a time of about 1.5 s, the force was decreased to approximately 15 pN. The protein begins to fold. There is a collapse just after this decrease in force. Some small vibrations follow. At a time of about 5 s, there is another collapse. This experimental result does not seem the same as the idealized molten globule folding shown in Fig. 2.2.

During folding, in the experiment shown, there is still a low applied force. Is this the reason for the difference between the experimental results and the idealized picture? Or, perhaps the molten globule state can not be applied to describe this protein's folding procedure? In this thesis I try to

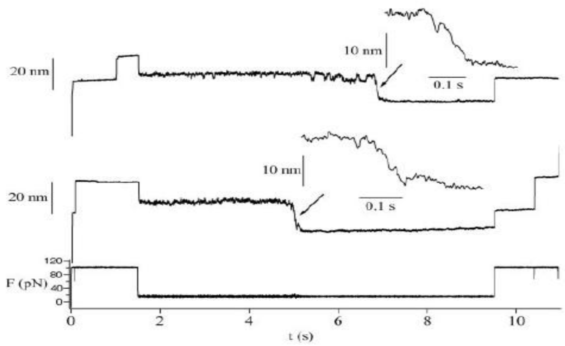


Figure 2.3. An example of the experimental results from Fernandez and Li [**21**].

build a model to explain the folding procedure which will help answer this question. We will find (in Sec. 11) that the answer to this question is that for a protein to change directly from unfolded state into the molten globule state, normally there should be some long-range interactions. This is because in the molten globule state the protein is already in a similar configuration as the native state. For this to occur, long-range interactions are normally necessary. We show that an applied external force can eliminate long-range interactions. Therefore under an external force, the molten globule state should not appear. In general, applied force changes the folding sequence.

2.6. Efforts to detect intermediate structures

The structure of intermediate states is useful to understand the folding procedure. One effort in this direction is by the group of Christopher Dobson. They used pulsed amide hydrogen-deuterium exchange to follow secondary structure formation. Amide hydrogen atoms are readily exchanged with the solvent in unfolded proteins, but this exchange is often strongly inhibited in a folded protein. They can detect the formation of structure during folding by measuring the rate of amide-hydrogen exchanges as a function of folding time. The protein they used in their initial experiments was lysozyme. In its native state, it has two lobes separated by a cleft. One lobe has five α helices and the other one is mainly three-stranded antiparallel β sheet. At 20 milliseconds, two major intermediate populations of lysozyme were detected. One in which the α helical lobe had achieved a high degree of secondary structure while the β sheet lobe contained no detectable structure. In the other population, no stable structure is detected in either of the lobes. A third less-populated state was also present. These observations suggest that intermediates are present along the folding pathway.

2.7. Discussion–2

The results of the experiment by Fernadnez and Li, described above, implies that the α helices formed faster than the β sheets. This is reasonable as the formation of α helices needs only the interaction of amino acids relatively close by, while the formation of β sheets needs hydrogen bonds to be formed between amino acids which are normally not nearby. How one pair of amino acids finds each other is still unknown. Do they move just by diffusion? Is there a driving force making them move toward each other? If so, what is the source of that force? In spite of all the unknowns, the initial distance of a pair of β sheet amino acids is normally longer than distance between α helix amino acids. It is reasonable that β sheets form slower.

2.8. Experimental results

Based on the results of H-D exchange studies of cytochrome c which lacks disulphide, it is found that on a time scale of approximate 10 milliseconds, a molten globule can form along with its secondary structure. On a time scale of approximate 100 milliseconds, hydrogen bonds form throughout the molecule. on a time scale of approximate 10 seconds, the complete hydrogen bonding pattern forms [**37**, **48**, **51**].

2.8.1. Numerical simulation

If the interaction forces between different atoms at difference distances can be measured, it seems that we could simulate the folding procedure numerically. Given an initial condition (configuration), we could calculate the motion due to the interaction forces. But there are many difficulties with this simple-sounding plan. First, how can the forces be determined exactly? We can use some approximate expressions, such as Van der Waals force. Then, how to include the Brownian force? We can ignore it, or use a random number to simulate the Brownian force. People set different models to simulate this procedure. But the time scale they can reach is only about a few milliseconds. Quoting Peter Kollman, [**18**]

> The limitation holding back this critical work has been the tremendous computational demand of the simulations, which must account for interactions between each atom in a protein and all the other atoms and surrounding water molecules. To capture protein movement at a useful level of detail, the full set of these interactions must be recalculated every femtosecond of protein time.

The Kollman group's simulation results show that a burst of folding in the first 20 nanoseconds quickly collapses the unfolded structure, suggesting that initiation of folding for a small protein can occur within the first 100 nanoseconds. Over the first 200 nanoseconds, the protein moves back and forth between compact states and more unfolded forms. The folded structures (molten state) have three-dimensional features, such as partially formed helices loosely packed together, that bear resemblance to the final folded form. They are only marginally stable, and unfold again before settling into other folded structures.

The next 800 nanoseconds reveal an intriguing "quiet period" in the folding. From about 250 nanoseconds until 400 nanoseconds the fluctuating movement back and forth between globules and unfolding virtually ceases. "For this period in the later part of the trajectory," says Kollman, "everything becomes quiet. And that's where the structure gets closest to the native state. It's quite happy there for awhile, then it eventually drifts off again for the rest of the period out to a microsecond."

2.8.2. Summary of the events at different time scales

Table 2.1 summarizes folding events and their times scales. The numerical simulation results are different from the experimental results. When needed, we will use the experimental results as the typical folding time scales.

2.8.3. Discussion

Here we can see that the forming of the molten globule state should happen on a time scale of less than 1 millisecond. As a reminder, the molten globule state has similar structure to the native state. That is, it should have almost the same end-to-end length as the native state. If we measure the end-to-end length during the folding, it should collapse in less than one millisecond, and fluctuate for a while, and then keep still. Of course this is only true for free refolding, *i.e.*, with no external force. What should the folding procedure be if there is an external force?

An external force increases the energy barrier for the folding procedure. We use the Arrhenius equation to estimate this effect,

$$k = \nu \cdot \exp(-E_b/k_bT) \tag{2.1}$$

The prefactor ν depends on the energy barrier E_b even in the absence of an external force. In a harmonic potential field ν is proportional to the square root of E_b. But, the most important effect of energy change is in the exponential term $\exp(-E_b/k_bT)$. For example, if we change E_b from 10 k_BT to 100 k_BT, ν will become three times larger, while the term $\exp(-E_b/k_bT)$ will become 10^{-39} times smaller. Compared to the change due the exponential term, the change from ν is negligible. Therefore, here we can assume ν remains constant while E_b changes.

Time scale	What's happening	Method	Protein & [Reference]
10 ms	molten globule forms	H-D exchange	cytochrome c [**37**, **48**, **51**]
100 ms	hydrogen bonds form	H-D exchange	cytochrome c [**37**, **48**, **51**]
10 s	hydrogen bonding complete	H-D exchange	cytochrome c [**37**, **48**, **51**]
20 ns	quick collapse	simulation	villin headpiece subdomain [**18**]
200 ns	change between molten globule and unfolded	simulation	villin headpiece subdomain [**18**]
next 800 ns	at the later part of this period, stays "quiet" in molten globule state	simulation	villin headpiece subdomain [**18**]
20 ms	α helical lobe achieves high degree of secondary structure, β sheet lobe does not	H-D exchange	lysozyme [**7**]
20 ms	neither of lobes achieve high degree of secondary structure	H-D exchange	lysozyme [**7**]

Table 2.1. Some time scales for events during folding as found from experiment and numerical simulation.

CHAPTER 3

Models describing the folding procedure

There are three main models [**36**]. The first is the framework model. Protein folding is thought to start with the formation of elements of secondary structure independently of tertiary structure, or at least before tertiary structure is locked in place. These elements then assemble into the tightly-packed native tertiary structure either by diffusion and collision or by propagation of structure in a stepwise manner. The second one is the hydrophobic collapse model. The initial event of the reaction is thought to be a relatively uniform collapse of the protein molecule, mainly driven by the hydrophobic effect. Stable secondary structure starts to grow only in the collapsed state. The third one is nucleation-condensation mechanism. Early formation of a diffuse protein-folding nucleus catalyzes further folding. The nucleus primarily consists of a few adjacent residues which have some correct secondary structure interactions, but is stable only in the presence of further approximately correct tertiary structure interactions. Fig. 3.1 illuminates the three models.

In this figure we can see that there are three stages in the folding procedure. Initially proteins are in the unfolded state. Then they are in the middle state, and at last go to the folded state. If we measure the end-to-end length of the proteins, these three models will give different results. For the hydrophobic collapse model, the protein's length collapses from the unfolded state to the middle state (molten globule state). Then the correct bonds need to be formed to reach the final state. Since all the amino acids have already congregated together in the first step, from the middle state to the folded state the protein's length will not change much. Therefore, if we only consider end-to-end length, there are only two choices for the protein: folded length and

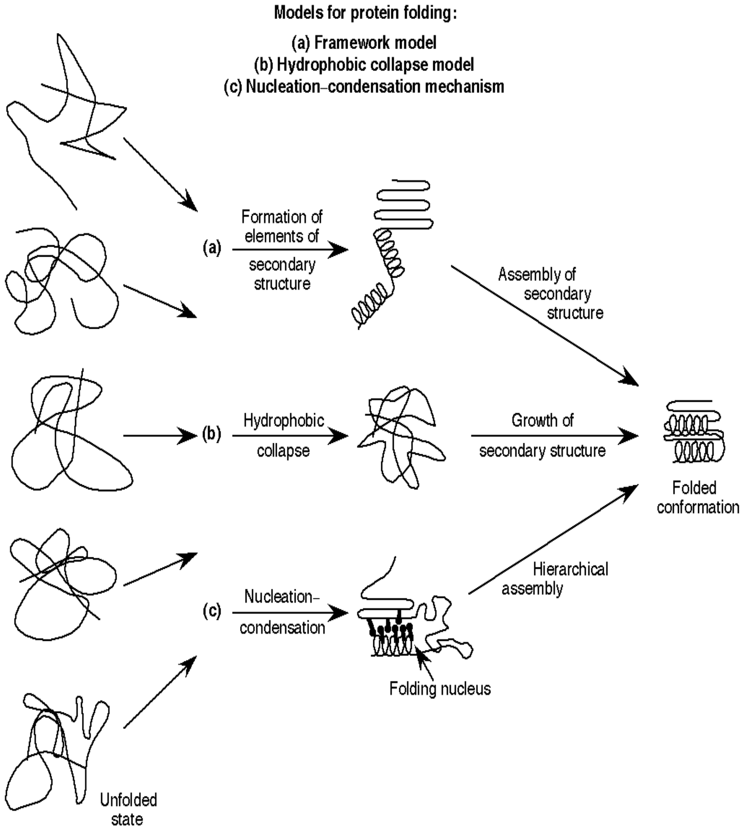

Models for protein folding:
(a) Framework model
(b) Hydrophobic collapse model
(c) Nucleation-condensation mechanism
(a)
Formation of elements of secondary structure
Assembly of secondary structure
(b)
Hydrophobic collapse
Growth of secondary structure
Folded conformation
(c)
Nucleation-condensation
Hierarchical assembly
Folding nucleus
Unfolded state

Figure 3.1. Three models for folding procedure [**36**].

unfolded length. We call this a two-state model. The third model is also a two-state model if end-to-end length is the only thing to consider. The first, the framework model, is different. From unfolded state to the middle state, its length can change. From the middle state to the folded state, its length can also change. We call this the multi-state model.

Knowledge about the protein folding procedure is increasing as more experimental data is collected and as better techniques are being developed. Increasingly, there is the data to check the accuracy of folding models. While it remains nearly impossible to observe directly the configurations of proteins as they fold, end-to-end length can be measured. There two types of such experiments. In one, the protein is first stretched and then released with constant velocity, measuring the forces at different lengths. This is called a constant-velocity experiment. The other type of experiment stretches proteins with constant force, measuring the end-to-end lengths at different times. This is called a constant-force experiment. In this thesis, I will construct a model to simulate the folding procedure and compare them with the experimental results, as shown in Sec. 4 and 6.

CHAPTER 4

Constant-velocity experiments

4.1. Devices and experimental results

The atomic force microscope (AFM) has been used in protein stretching experiments [**6**, **12**, **44**–**46**] Fig. 4.1 demonstrates the principle of operation of the AFM [**43**].

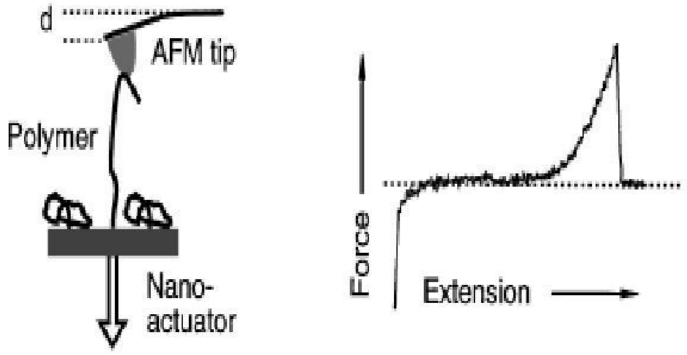


Figure 4.1. A typical experiment in which a single molecule is stretched by an AFM tip. The tip is brought into contact with the sample, which is covered by a layer of polymer molecules. If a molecule has bound to the tip, it can be stretched and the force measured via the deflection d of the cantilever spring as a function of the extension. When the maximum binding force is exceeded, the molecule ruptures from the tip and the tip is free again. On the right is a representative example of the force-extension curve. From [**43**, Fig. 1].

Examples of constant velocity experimental results are shown in Fig. 4.2 (for titin) and Fig. 4.3 (for dextran). (Dextran is not protein, but it is a polymer. We will include it in this discussion.)

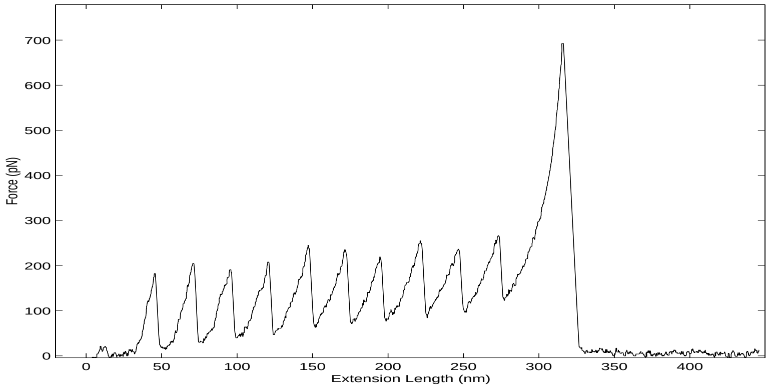


Figure 4.2. Experimental results of stretching titin with the AFM from [**34**].

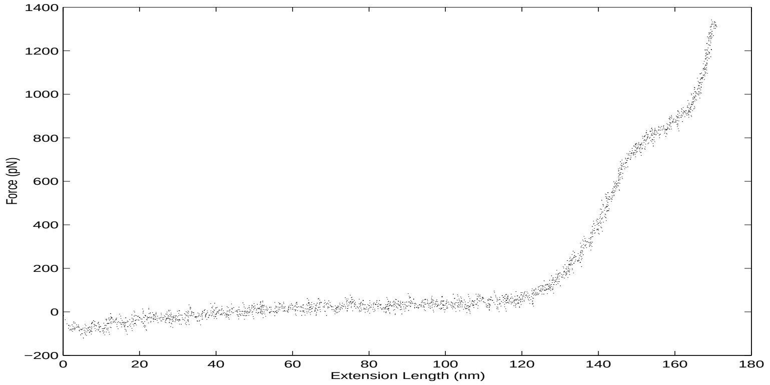


Figure 4.3. Experimental results of stretching dextran with the AFM [**34**].

4.2. Brief description of the model

Although the experimental results for titin and dextran looks different, it can be shown that they can be described by the same model. Here is a brief description of this model. Details will be shown in the following sections.

The polymers in the experiments have several identical domains (for titin) or a few hundred identical domains (for dextran). Each domain can be regard as a rigid bar. Each domain is considered to be in one of two possible states: a folded state or an unfolded state. Under the influence of an external force and thermal fluctuations, the projected length of each bar in the force direction can be calculated. All the domains are freely jointed. The total length of the protein is the sum of all the domains' lengths.

Therefore, to set up this model, we need to solve three problems. The first is to calculate the projected length in the force direction. The second is to determine how one domain changes from one state to the other. The last is to combine the freely-jointed-bars model and the two-state model together. Details are shown in the following sections.

4.3. Freely-jointed chain

A segment in a freely-orienting chain subject to no external force will usually have no preferred direction or orientation. When the chain is subjected to a tension f, however, a segment's potential energy will depend on its alignment relative to the direction of the applied force.

If the force is assumed to act in the z-direction, then the potential energy can be given by

$$V = -fI\cos\theta$$

where θ is the angle between the z-axis and $\mathbf{I}$. $\mathbf{I}$ is a vector whose direction is the same as the segment, and whose amplitude is the length I of the segment.

According to the Boltzmann distribution law, the probability that the segment makes an angle θ with the z-axis is proportional to

$$\exp\left(-\frac{V}{k_BT}\right)$$

Hence, the average value of the z-component of $\mathbf{I}$ as it undergoes thermal motion will be

$$\langle I_z\rangle = \frac{\int_0^\pi (I\ \cos\theta)\,(2\pi\sin\theta)\exp\left[fI\ \cos\theta/k_BT\right]\,d\theta}{\int_0^\pi (2\pi\sin\theta)\exp\left[fI\ \cos\theta/k_BT\right]\,d\theta} \tag{4.1}$$

or

$$\langle I_z \rangle = I \left[\coth \left(If/k_B T \right) - \left(k_B T/If \right) \right] \tag{4.2}$$

The total configuration length is

$$L = N\langle I_z \rangle = NI \left[\coth \left(If/k_B T \right) - \left(k_B T/If \right) \right]$$

If $N = 275$, $I = 0.5$ nm, and $T = 273K$, we get the force-length relationship shown in Fig. 4.4.

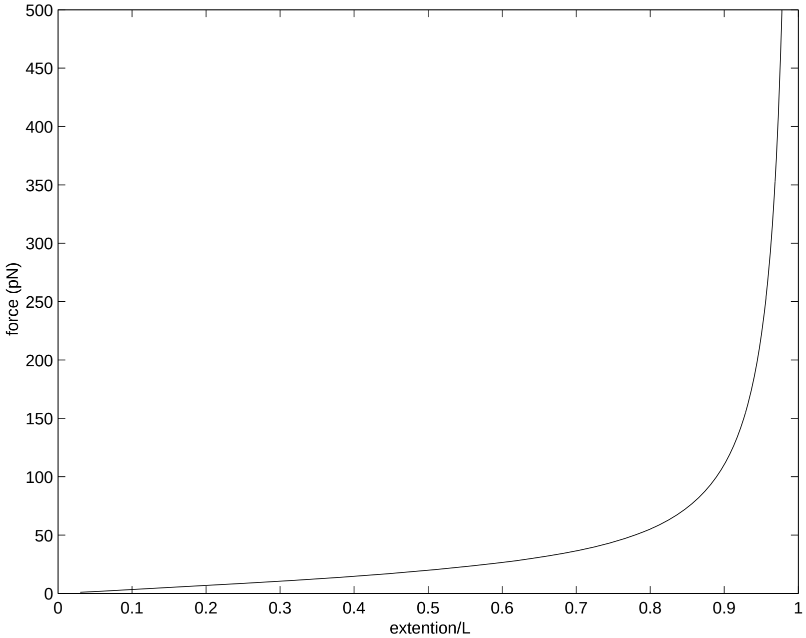


Figure 4.4. Force-length relationship for a freely-jointed chain with 275 bars each of length 0.5 nm. Temperature is 273 K. The force ranges from 0 pN to 500 pN, which spans the typical range used in experiments.

4.4. Two-state model

4.4.1. Two-state model can be regarded as a diffusion problem

The two-state model is introduced in Fig. 4.5. This model claims that each domain can be in only one of two states. There is an energy barrier between the two states. If the domain is in the folded state, its length is I_f. If it is in the unfolded state, its length is I_u. The energy difference between the folded state and the energy barrier is ΔG_u, and the distance between the folded state and the energy barrier is x_u. The energy difference between the folded state and the unfolded state is ΔG_0, and the distance from energy barrier to unfolded state is x_f. The change in length $I_u - I_f = x_u + x_f$. Because of thermal fluctuations, one domain can "jump" through the energy barrier and change state. This procedure is identical to particle diffusion as a random walk. One particle is initially in the position I_f. Because of thermal fluctuations, it can move as a random walker. If it can pass through the energy barrier, it will change state. Similarly it can walk from the unfolded position to the folded position. Therefore, the change-of-state problem can be regarded as a diffusion problem.

4.4.2. Probability equation for diffusion as a random walk

In this section we will develop the diffusion equation.

If a particle diffuses as a random walk, we hope to calculate the average time for it to pass some particular point. We start with conservation of mass,

$$\frac{\partial c(x,t)}{\partial t} = -\frac{\partial J(x,t)}{\partial x} \tag{4.3}$$

and the assumption that flux $J(x,t)$ is linearly proportional to the concentration gradient,

$$J(x,t) = -D\frac{\partial c(x,t)}{\partial x} \tag{4.4}$$

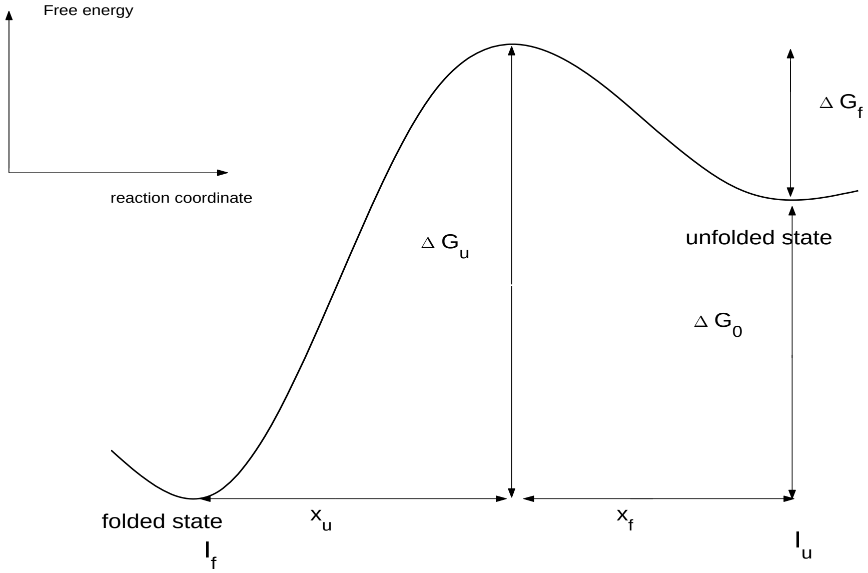


Figure 4.5. Free energy diagram for the two-state model in the reaction coordinate. If the domain is in the folded state, its length is I_f. If it is in the unfolded state, its length is I_u. The energy difference between the folded state and the energy barrier is ΔG_u, and the distance between the folded state and the energy barrier is is x_u. The energy difference between the folded state and the unfolded state is ΔG_0, and the distance from energy barrier to unfolded state is x_f.

where $c(x,t)$ is the concentration of mass and D is the diffusion coefficient. Combining (4.3) and (4.4) we get

$$\frac{\partial c(x,t)}{\partial t} = D\frac{\partial^2 c(x,t)}{\partial x^2} \tag{4.5}$$

We can replace $c(x,t)$ by $p(x,t)$, the probability of finding it at position x at time t. To do this we need one more equation

$$\int_{x_1}^{x_2} p(x,t)dx = 1 \tag{4.6}$$

where x_1 is the left end of the region and x_2 is the right end of the region.

Equations (4.3)-(4.5) are valid when there is no external force. An external force will cause an average velocity $v(x) = F(x)/\gamma$, where γ is the drag coefficient of the particle. Therefore, the probability flux, $J(x)$, becomes

$$J(x,t) = -D\frac{\partial p(x,t)}{\partial x} + \frac{F(x)}{\gamma}p(x,t) \tag{4.7}$$

Thus, in the presence of an external force, the probability satisfies

$$\frac{\partial p(x,t)}{\partial t} = D\frac{\partial^2 p(x,t)}{\partial x^2} - \frac{\partial}{\partial x}\left[\frac{F(x)}{\gamma}p(x,t)\right] \tag{4.8}$$

This is the basic equation for diffusion problems. One of the applications of this equation is to calculate the rate constant (shown in the next section). For example, in the two-state model there are two stable states separated by a energy barrier. Because of thermal fluctuations, a domain in one state has a certain chance to switch to the other domain. The rate of changing states can be calculated from the diffusion equation.

4.4.3. First-passage times and rate constants

We can now solve (4.8) under certain boundary and initial conditions given an external force. Then, based on the solution $p(x,t)$, we can calculate the other parameters we are interested in, such as rate constant, first passage time, and so on. Kramers [**29**] calculated the first passage time, *i.e.*, the average time for a particle to pass through a certain position for the first time. The external force he used was $F = -Kx$ in the region $0 \leq x \leq x_0$. That is, the energy barrier from $x = 0$ to $x = x_0$ is $U = \frac{1}{2}Kx^2$. At $x = x_0$, $U_0 = \frac{1}{2}Kx_0^2$. The result is

$$t_K = \tau\sqrt{\frac{\pi}{4}}\sqrt{\frac{k_BT}{U_0}}e^{\frac{U_0}{k_BT}} \tag{4.9}$$

where $\tau = \gamma/K$. The meaning of (4.9) is this. If you put a particle at $x = 0$, after an average time t_K it will arrive at x_0. The reciprocal of the first-passage time is call the rate constant,

$$k_K = 1/t_K \tag{4.10}$$

To understand (4.10), imagine you put a particle at $x = 0$ initially. Once it passes through $x = x_0$, put it back to $x = 0$ and let it move again. k_K is equal to the average number of times that the particle can pass through $x = x_0$ in unit time. Based on Kramers rate theory,

$$k_K = \frac{1}{\tau}\sqrt{\frac{4}{\pi}}\sqrt{\frac{U_0}{k_BT}}e^{\frac{-U_0}{k_BT}} \tag{4.11}$$

Note that the rate constant depends on the energy barrier in two parts. One is in the exponential term, which is the dominate term. The other part is the square root. The changes caused by this part can be ignored compared to the exponential term. Therefore for simplicity we can regard the second part as constant. In protein folding problems, the energy barrier is normally written as ΔG. (G is the Gibbs free energy. This is the measure of energy in a system at constant temperature and pressure.) From now we will use ΔG to replace U_0. Back to the two-state model. The rate constant for one domain to transfer from the folded state to the unfolded state is α_0, and from the unfolded state to the folded state, β_0. The expressions for the rate constants are:

$$\alpha_0 = \omega e^{-\Delta G_u/k_BT} \tag{4.12}$$

$$\beta_0 = \omega e^{-\Delta G_f / k_B T} \tag{4.13}$$

where ω is a constant given as

$$\omega = \frac{1}{\tau}\sqrt{\frac{4}{\pi}}\sqrt{\frac{\Delta G_f}{k_B T}} \tag{4.14}$$

When there is an external force [**43**]

$$\alpha(F) = \alpha_0 e^{F x_u / k_B T} \tag{4.15}$$

$$\beta(F) = \beta_0 e^{-F x_f / k_B T} \tag{4.16}$$

If there are N_f domains in the folded state, the average number of folded modules that become unfolded in a time Δt is

$$dP_u = \alpha(F) N_f \Delta t \tag{4.17}$$

Similarly, the average number of unfolded modules that become folded in a time Δt is

$$dP_f = \beta(F) N_u \Delta t \tag{4.18}$$

4.4.4. The relationship of probability and rate constant

In the previous section we calculated the average number of changes of state in a short time based on the rate constant. To describe the change-of-state procedure, we have two tools: one is the rate constant, the other is probability, normally of a particle being in a certain state. Since they are the two aspects of the same thing, we hope to know the relationship between them.

In Rief's two-state model [**43**], the two states are the folded state and unfolded state. Since the following analysis can be applied to any two-state change, we call them state 1 and state 2, rather than the folded state and unfolded state.

Initially one particle is in state 1 and it can transfer to state 2 with rate constant k. The probability of observing it in state 1 at time t is $p(t)$. Then

$$\frac{dp(t)}{dt} = -kp(t) \tag{4.19}$$

With $p(0) = 1$, we get

$$p(t) = e^{-kt} \tag{4.20}$$

If at t=0 there are N particles in state 1, during a short time interval Δt, what is the probability of transferring to state 2? The answer is,

$$P_0 = [p(\Delta t)]^N \tag{4.21}$$

$$P_1 = N[p(\Delta t)]^{N-1} * [1 - p(\Delta t)] \tag{4.22}$$

$$\vdots$$

$$P_i = C_N^i [p(\Delta t)]^{N-i}[1 - p(\Delta t)]^i \tag{4.23}$$

Where P_i is the probability of i particles transferring state during time interval Δt, and $C_N^i = \frac{N!}{i!(N-i)!}$ is the number of choices picking i particles from N different ones. The average number of transfers will be

$$\Delta N = \sum_{i=0}^{N} iP_i = N(1 - p(\Delta t)) \tag{4.24}$$

If Δt is small enough,

$$\Delta N \approx Nk\Delta t \tag{4.25}$$

In the next section we will describe how to do a computer simulation based on the average transfer number.

4.4.5. Simulation procedure and comparison with experimental results

Now we can show in detail how we simulate the folding procedure. As already noted, each domain has two possible states: the folded sate with backbone length I_f and the unfolded state with backbone length I_u. The total contour length (the longest possible length, *i.e.*, the length if all the domains are aligned into a straight line) of the protein is $L = N_f I_f + N_u I_u$, where N_f is the number of domains in the folded state, and N_u is the number of domains in the unfolded state. When there is no external force, the transition rate from folded state to unfolded state is

$$\alpha_0 = \omega e^{-\Delta G_u / k_B T} \tag{4.26}$$

Where ΔG_u is the activation barrier for folding, and ω is the reciprocal of a diffusive relaxation time. ω can be calculated using (4.14). The back reaction rate for the unfolding is

$$\beta_0 = \omega e^{-\Delta G_f / k_B T} \tag{4.27}$$

where ΔG_f is the activation barrier for folding.

If there is an external force, the rates become

$$\alpha(F) = \alpha_0 e^{F x_u / k_B T} \tag{4.28}$$

$$\beta(F) = \beta_0 e^{-F x_f / k_B T} \tag{4.29}$$

For example, for titin $\Delta G_0 = 13\ k_B T$, $\Delta G_u = 20\ k_B T$, $x_f = 27.7$ nm. If the external force is 15 pN, then at room temperature the rate constant for the refolding procedure will become much smaller compared than the rate constant when there is no external force, $\beta(F)/\beta_0 \approx e^{-100}$. There is clearly something suspect about the refolding rate. As it is observed that proteins fold on a time scale from microseconds to hours (and sometimes longer), this calculation would suggest that under the application of an external force of the order of pico-Newtons, proteins would never be observed to fold. In Sec. 11, we show that computing the refolding rate based on x_f is incorrect. In brief, x_f is a one-dimensional reaction coordinate formed by concatenating the reaction coordinates of the individual parts of the protein. That is, x_f sums spatially over all of the reaction coordinates accounting for the fact that they are indeed spatially sequential. We show that what is needed to determine folding times, is the fact that motion along the reaction coordinates of the individual components occurs simultaneously. Consequently, the spatial x that is needed is only the single most slowly folding component.

In a short time Δt, the average number of domains transferred from the folded state to the unfolded state is

$$dP_u = \alpha(F) N_f \Delta t \tag{4.30}$$

The number transferred back is

$$dP_f = \beta(F) N_u \Delta t \tag{4.31}$$

The simulation procedure proceeds as follows. We begin with $x = 0$, $F = 0$, and all the domains in folded states, *i.e.*, $N_f = N$ and $N_u = 0$, where N is the total number of domains. This is the initial condition. Here we show how to get the $(i+1)$th step from the ith step. We use $dP_u(F_i)$ to determine how many domains transfer from folded to unfolded. Here dP_u can be any value bigger than 0, but the number of domains that unfold dN_u must be an integer. If $dP_u < 1$, we pick a random number $0 < R < 1$: if $R < dP_u$, we set $dN_u = 1$, otherwise we set $dN_u = 0$. If $dP_u > 1$, we just use the integer part as dN_u. Similarly we can get dN_f. Then we get $N_f(i+1) = N_f(i) - dN_u + dN_f$, and $N_u(i+1) = N_u(i) - dN_f + dN_u$. We can now calculate the new contour length of the protein. The new extension length $x(i+1) = x(i) + V\Delta t$, where V is the stretching speed. Then based on the new x and L, we get the new force using

(4.2). Repeatedly applying these steps we can simulate the stretching procedure. The refolding procedure is almost the same; the only difference is the extension length is decreased at each step, $x(i+1) = x(i) - V\Delta t$.

The comparison of experimental results and simulation results are shown in Fig. 4.6 for titin and Fig. 4.7 for dextran. Numerical results and experimental results fit well.

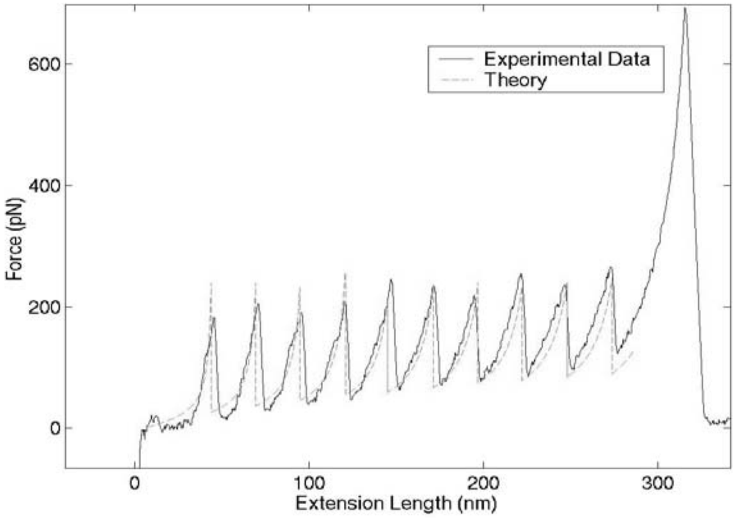


Figure 4.6. Comparison of experimental and simulation results for titin. The solid line is the experimental results. There are 10 domains. $I_f = 4$ nm, $I_u = 32$ nm, $\alpha_0 = 3 \times 10^{-5}/s$, $G_0 = 9.2\ k_BT$. Temperature is 300 K. The experimental data is from [**34**].

We can see that for titin, the stretching curve looks like a row of sawteeth: the force increases to a certain value, then decreases dramatically forming a peak. There are 10 peaks. There is a large length difference between the folded and unfolded state. In the folded state one domain has length 4 nm, while in the unfolded state one domain has length 32 nm. Therefore, as one domain unfolds, the protein's total contour length increases greatly. This causes the force to decrease markedly, yielding the sharp dropping edge of the sawtooth.

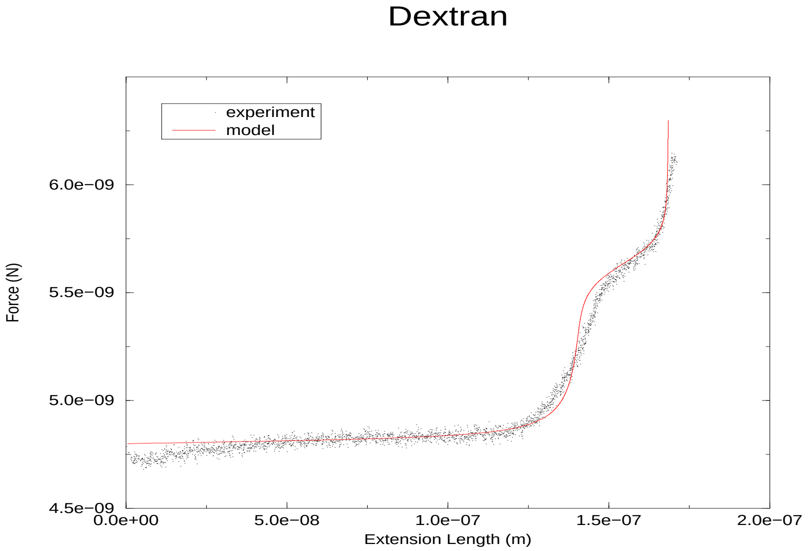


Figure 4.7. Comparison of experimental and simulation results for dextran. The dots are the experimental results. There are 310 dextran monomers. $I_f = 0.5$ nm, $I_u = 0.565$ nm, $\alpha_0 = 2 \times 10^{-4}/s$, $G_0 = 13.2\ k_BT$. Temperature is 300 K. The experimental data is from [**34**].

One the other hand, for dextran the length difference between the folded and unfolded state is quite small. In the folded state one domain has length 0.5 nm, while in the unfolded state one domain has length 0.565 nm. The number of domains for one dextran polymer is about 300. Therefore one domain that becomes unfolded will not cause a notable change of the total contour length. Hence, the extension curve for dextran is smooth. The slope of the model is steeper than the experimental results in the region of $5 \times 10^{-9} < F < 5.5 \times 10^{-9}$ N, and similar elsewhere. That is, simulation results seem stiffer than the experimental results. This problem comes from the FJC model. It regards each amino acid as a rigid bar: that's realistic. But, these bars are connected freely, this is not a good approximation at all forces. For example, for the FJC model in the high-force region, the simulated protein is already almost fully stretched and can hardly be stretched any more, that is why the force-extension curve is so stiff.

4.5. Further discussion about dP_u and dP_f

Note that although we use the notation dP_u and dP_f, they do not mean the probabilities of changing state. From (4.30), dP_u is the average number of domains transferred from the folded state to the unfolded state. And, from (4.31), dP_f is the average number of domains transferred from the unfolded state to the folded state. When used as the number of domains changing state, these values can be used for any Δt, large or small, and the expressions yield the correct value for the number of folded or unfolded domains. However, for the stochastic calculations which we and [**43**] carry out, we require the *probability* of at least one change in an interval of time, not the mean number of changes within an interval of time. (The difference between the "number" and the "probability" is that "number" could be larger than one but "probability" is always smaller than unity. Hence, we could interpret dP_u and dP_f as above, in terms of change in number, use an arbitrarily large Δt, and find the change. If larger than unity, the change implemented is the integer part plus one (zero) if a uniformly-distributed random number is less (greater) than the non-integer part.)

From (4.20), we can find $p(t) = e^{-kt}$, the probability of remaining in the original state after a time t. The probability of N identical domains remaining in their original state is then $p_N(t) = (e^{-kt})^N$. And, so the probability of at least one of the N domains changing state is $1-(e^{-kt})^N$. When the time interval is small $t = \Delta t$, then we find that $1-(e^{-kt})^N = Nk\Delta t + \ldots$. In [**43**] it is only this first term which is used to compute the probabilities of change, where the rate k is given the appropriate value α or β, and N is assigned as the number of folded domains. Is this always true during the simulation?

Figures 4.8-4.10 show the extension-force curves and the corresponding extension-dP_u curves. The three figures use the same parameters and the differences between them arises from the random numbers used to determine whether one folding or unfolding event could happen. We can see that dP_u is always much smaller than unity during the simulation. dP_f is even smaller than dP_u. Therefore, the approximation (4.25) used to calculate dP_u and dP_f is valid.

One thing to note is that during the simulation, we chose $\Delta t = 0.001$. In numerical simulation, we can choose Δt small enough to make the approximation valid. But, for protein folding does this Δt have a finite physical limit? Let's model the protein as a spring and calculate its vibration frequency. In Fig. 4.2, we can see that per peak, the extension length increases by approximately 25 nm, and the force increases by approximately 200 pN. So for one domain, its spring constant k_{spring} ≈200 pN/25 nm=0.008 N/m. The mass of one domain is about 2×10^{-23} kg. Therefore, the frequency $\omega \approx \sqrt{k_{spring}/m} = 2 \times 10^{10}$ /s. So, the physical time limit is $\sim 10^{-10}$ s. The $\Delta t = 0.001$ s used in the simulation is much longer than this limiting value, but is still small enough so that the approximations underlying dP_u and dP_f are valid.

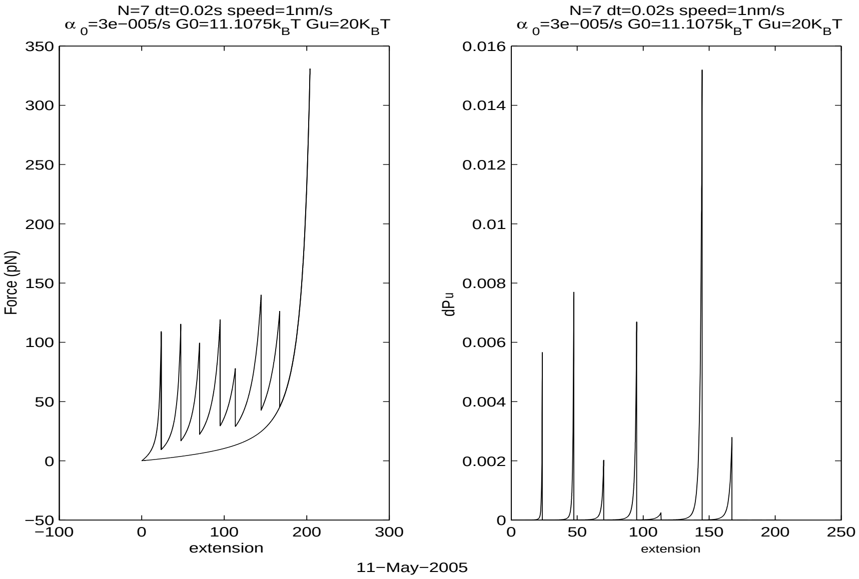


Figure 4.8. Extension-force curve and extension-dP_u curve.

Note that in the previous simulations, if there are N_f domains in the folded state, then we say $dP_u = \alpha(F) N_f \Delta t$. And we only generate a random number once and compares it with dP_u. (As shown in (4.20)-(4.25), This is valid as far as $k\Delta t \ll 1$. From Figs. 4.8-4.10 we can see

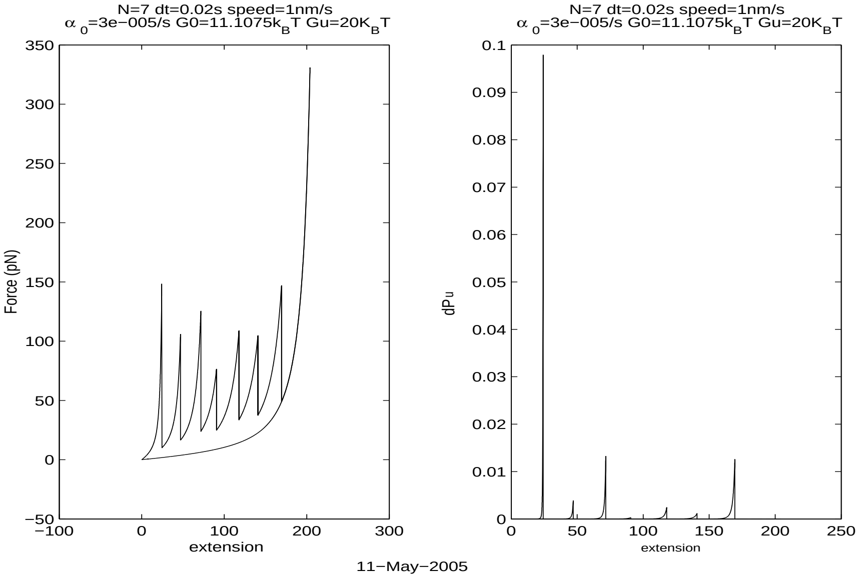


Figure 4.9. Extension-force curve and extension-dP_u curve. For the same parameters as in Fig. 4.8, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

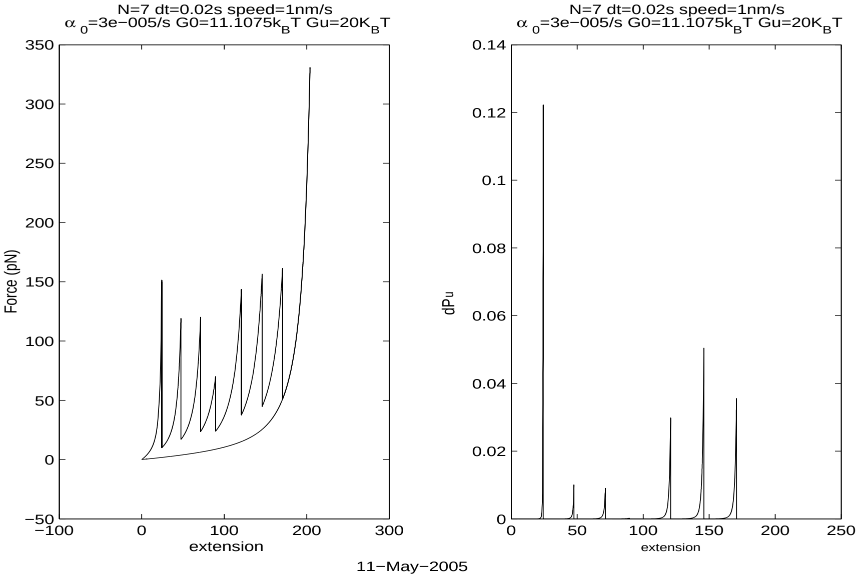


Figure 4.10. Extension-force curve and extension-dP_u curve. For the same parameters as in Fig. 4.8, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

that $dP_u \ll 1$. dP_u is greater than $k\Delta t$. So, as noted, when Δt is small enough, we can use the approximation $dP_u = k\Delta t$. We can also do the simulation in some other way. The average transfer number for one domain is $dP_u = \alpha(F)\Delta t$. We can generate one random number for one domain, compare dP_u with that random number to determine whether a transfer will occur. Since there are N_f domains in the unfolded states, we need to repeat this procedure N_f times and sum up all the transfer numbers as dN_u. We call this the single-domain method.

Figures 4.11-4.13 show the extension-force curves and the corresponding extension-dP_u curves. Again, they all have the same parameters and the difference comes from the random numbers. Once again, the values for the numbers transferred is small. A detailed comparison of the statistics of the single-domain method vs. the one-shot method in Figs. 4.8-4.10 would show the extent to which the distribution of changes is similar, and whether any noticeable differences arise from the two methods.

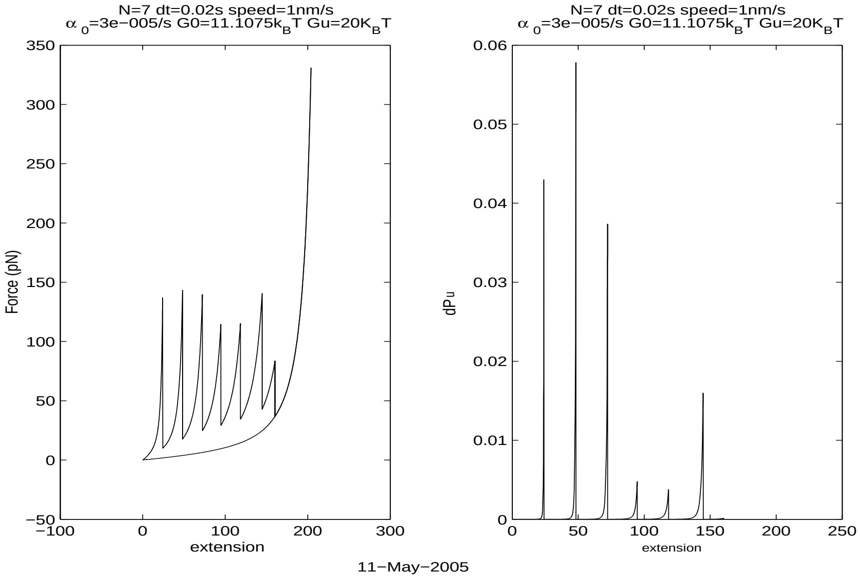


Figure 4.11. Extension-force curve and extension-dP_u curve for the single-domain method.

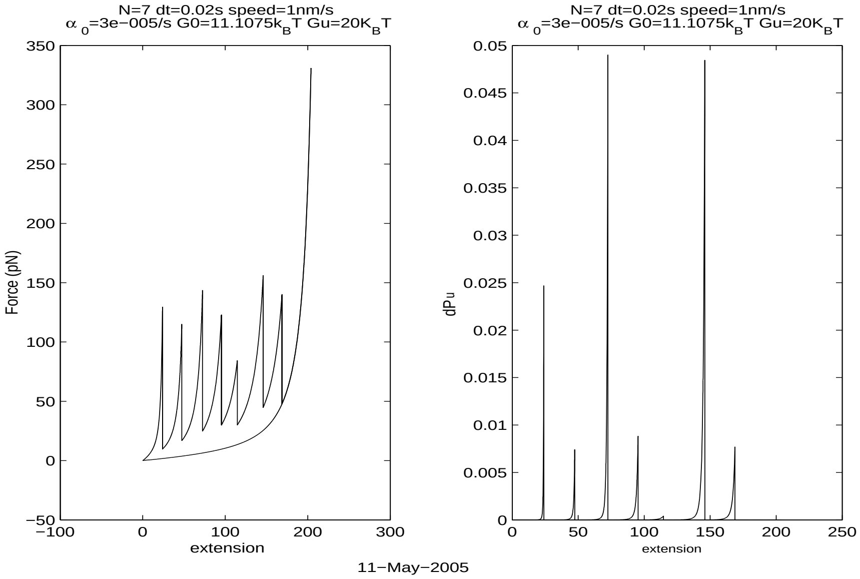


Figure 4.12. Extension-force curve and extension-dP_u curve for the single-domain method. For the same parameters as in Fig. 4.11, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

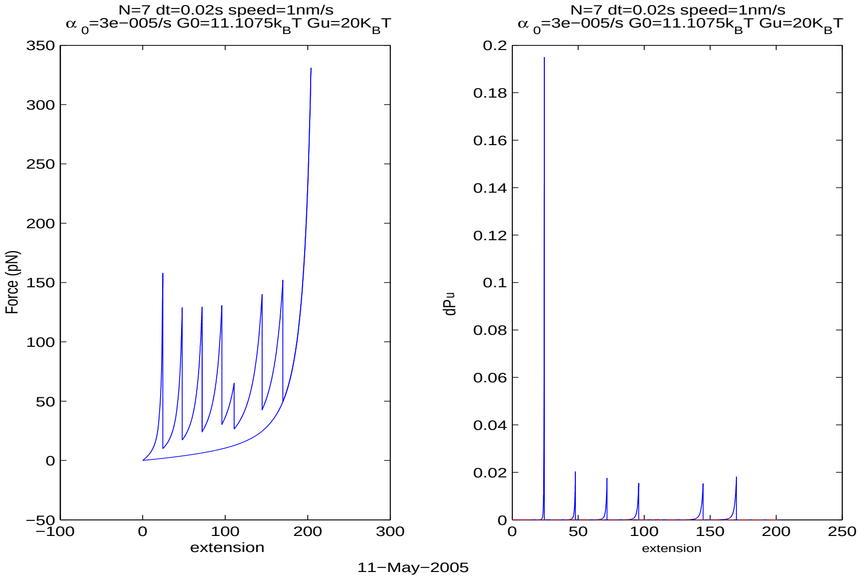


Figure 4.13. Extension-force curve and extension-dP_u curve for the single-domain method. For the same parameters as in Fig. 4.11, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

CHAPTER 5

Variance for constant velocity experiments

Since proteins are not static structures, it is necessary to describe variations about their mean configuration to gain information about their flexibility. This internal variation of flexibility is an integral part of protein function.

5.1. Fluctuation dissipation theorem

In this section we will calculate the variance of protein length due to the thermally-induced change in direction of the component monomers.

In Sec. 4.3 we have shown that the average length along the force direction for one freely rotating bar with a single state is

$$\langle I_z \rangle = \frac{\int_0^\pi (I\ \cos\theta)\,(2\pi\sin\theta)\exp\left[fI\cos\theta/k_BT\right]\,d\theta}{\int_0^\pi (2\pi\sin\theta)\exp\left[fI\cos\theta/k_BT\right]\,d\theta}$$

The $\langle\ldots\rangle$ and the top bar (for example, in (5.3)) both mean average.

If the monomer has two possible states with energy difference $\triangle G_o$, the average length along the force direction is

$$\begin{aligned}\langle I_z \rangle \quad = \quad & \frac{\int_0^\pi (I_f\cos\theta)\,(2\pi\sin\theta)\exp\left[a\cos\theta\right]d\theta}{\int_0^\pi (2\pi\sin\theta)\exp\left[a\cos\theta\right]d\theta + \int_0^\pi (2\pi\sin\theta)\exp\left[-\triangle G_o/kT + b\cos\theta\right]d\theta} \\ & + \frac{\int_0^\pi (I_u\cos\theta)(2\pi\sin\theta)\exp\left[-\triangle G_o/kT + b\cos\theta\right]d\theta}{\int_0^\pi (2\pi\sin\theta)\exp\left[a\cos\theta\right]d\theta + \int_0^\pi (2\pi\sin\theta)\exp\left[-\triangle G_o/kT + b\cos\theta\right]d\theta}\end{aligned}$$

(5.1)

where $a = \frac{fI_f}{kT}$ and $b = \frac{fI_u}{kT}$. Then we get

$$\begin{aligned} \langle L \rangle &= N\langle I_z \rangle \\ &= N\frac{I_f(\frac{1}{a})^2\left[a\cosh a - \sinh a\right] + I_u(\frac{1}{b})^2\left[b\cosh b - \sinh b\right]\exp\left[-G_o/kT\right]}{\frac{1}{a}\sinh a + \left[\frac{1}{b}\sinh b\right]\exp\left[-G_o/kT\right]} \end{aligned}$$

(5.2)

The variance in length is given by $\overline{(L-\bar{L})^2}$, which depends on the differences between the measured values of L and the mean value $\bar{L}$. However, it can be shown that the variance can be determined from measurement of the mean force-extension curve alone. In particular, we show that

$$\overline{(L-\bar{L})^2} = Nk_BT \cdot \frac{d\bar{L}}{dF} \tag{5.3}$$

That is

$$N^2 \cdot \overline{(I_z-\bar{I}_z)^2} = Nk_BT \cdot \frac{N \cdot \bar{I}_z}{dF} \tag{5.4}$$

$$\overline{(I_z-\bar{I}_z)^2} = k_BT \cdot \frac{d\bar{I}_z}{dF} \tag{5.5}$$

$$\langle I_z^2 \rangle - \langle I_z \rangle^2 = k_BT \cdot \frac{d\bar{I}_z}{dF} \tag{5.6}$$

First we calculate $\langle I_z^2 \rangle$

$$\begin{aligned}
\langle I_z^2 \rangle &= \frac{\int_0^\pi (I_f \cos\theta)^2 \, (2\pi \sin\theta) \exp\left[a\cos\theta\right] d\theta}{\int_0^\pi \, (2\pi \sin\theta) \exp\left[a\cos\theta\right] d\theta + \int_0^\pi (2\pi \sin\theta) \exp\left[-\triangle G_o/kT + b\cos\theta\right] d\theta} \\
&\quad + \frac{\int_0^\pi (I_u \cos\theta)^2 (2\pi \sin\theta) \exp\left[-\triangle G_o/kT + b\cos\theta\right] d\theta}{\int_0^\pi \, (2\pi \sin\theta) \exp\left[a\cos\theta\right] d\theta + \int_0^\pi (2\pi \sin\theta) \exp\left[-\triangle G_o/kT + b\cos\theta\right] d\theta} \\
&= \frac{I_f^2 (\frac{1}{a})^3 X + I_u^2 (\frac{1}{b})^3 Y \exp\left[-G_o/kT\right]}{\frac{1}{a}\sinh a + \left[\frac{1}{b}\sinh b\right] \exp\left[-G_o/kT\right]}
\end{aligned}$$

(5.7)

where $X = \left[a^2 \sinh a - 2a\cosh a + 2\sinh a\right]$, and $Y = \left[b^2 \sinh b - 2b\cosh b + 2\sinh b\right]$.

Therefore,

$$\begin{aligned}
\langle I_z^2 \rangle - \langle I_z \rangle^2 &= \frac{I_f^2 (\frac{1}{a})^3 X + I_u^2 (\frac{1}{b})^3 Y \exp\left[-G_o/kT\right]}{\frac{1}{a}\sinh a + \left[\frac{1}{b}\sinh b\right] \exp\left[-G_o/kT\right]} \\
&\quad - \left(\frac{I_f (\frac{1}{a})^2 \left[a\cosh a - \sinh a\right] + I_u (\frac{1}{b})^2 \left[b\cosh b - \sinh b\right] \exp\left[-G_o/kT\right]}{\frac{1}{a}\sinh a + \left[\frac{1}{b}\sinh b\right] \exp\left[-G_o/kT\right]} \right)^2
\end{aligned}$$

(5.8)

For $k_B T \cdot \frac{d\bar{I_z}}{dF}$, if we write the $\langle I_z \rangle$ in the form

$$\begin{aligned}
\langle I_z \rangle &= \frac{I_f (\frac{1}{a})^2 \left[a\cosh a - \sinh a\right] + I_u (\frac{1}{b})^2 \left[b\cosh b - \sinh b\right] \exp\left[-G_o/kT\right]}{\frac{1}{a}\sinh a + \left[\frac{1}{b}\sinh b\right] \exp\left[-G_o/kT\right]} \\
&= \frac{A+B}{C+D} \qquad (5.9)
\end{aligned}$$

where

$$A = I_f (\frac{1}{a})^2 \left[a\cosh a - \sinh a\right] \qquad (5.10)$$

(5.11) $$B = I_u(\frac{1}{b})^2 \left[b\cosh b - \sinh b\right] \exp\left[-G_o/kT\right]$$

(5.12) $$C = \frac{1}{a}\sinh a$$

(5.13) $$D = \left[\frac{1}{b}\sinh b\right] \exp\left[-G_o/kT\right]$$

then we have

$$\begin{aligned} k_B T \cdot \frac{d\bar{I_z}}{dF} &= k_B T \cdot \frac{\partial I_z}{\partial a}\frac{\partial a}{\partial f} + k_B T \cdot \frac{\partial I_z}{\partial b}\frac{\partial b}{\partial f} \\ &= I_f \frac{\partial I_z}{\partial a} + I_u \frac{\partial I_z}{\partial b} \end{aligned}$$

(5.14)

and we also have

(5.15) $$\frac{\partial}{\partial a}\left(\frac{A+B}{C+D}\right) = \frac{A'(C+D) - (A+B)C'}{(C+D)^2}$$

(5.16) $$\frac{\partial}{\partial b}\left(\frac{A+B}{C+D}\right) = \frac{B'(C+D) - (A+B)D'}{(C+D)^2}$$

and

(5.17) $$\frac{dA}{da} = I_f(\frac{1}{a})^3 \left[a^2\sinh a - 2a\cosh a + 2\sinh a\right]$$

$$\frac{dB}{db} = I_f(\frac{1}{b})^3 \left[b^2 \sinh b - 2b \cosh b + 2 \sinh b\right] \exp\left[-G_o/kT\right] \tag{5.18}$$

$$\frac{dC}{da} = -\frac{1}{a^2} \sinh a + \frac{1}{a} \cosh a \tag{5.19}$$

$$\frac{dD}{db} = -\frac{1}{b^2} \sinh b + \frac{1}{b} \cosh b \exp\left[-G_o/kT\right] \tag{5.20}$$

Plugging these into (5.14) we get

$$\langle I_z^2 \rangle - \langle I_z \rangle^2 = k_B T \cdot \frac{d\bar{I}_z}{dF} \tag{5.21}$$

Therefore, we get variance

$$v = N k_B T \cdot \frac{d\bar{L}}{dF} \tag{5.22}$$

Now, we can use the force-extension curves of data to find the right-hand-side of the above expression, and so determine the variance. We can also use (5.8) as the theoretical prediction.

Fig. 5.1 shows the comparison for the dextran experimental results with theoretical results. We can see that they have similar behavior. Initially the variance is very big. This is because the dextran is in the coil state, which is a loose structure. Any small perturbation can cause a large change in length. With increasing force, all the domains begin to align and become stiffer. Therefore the variance tends to decrease. There is a peak in the middle. In this region, dextran domains change from the folded into the unfolded state. This changing-state procedure introduces larger variance.

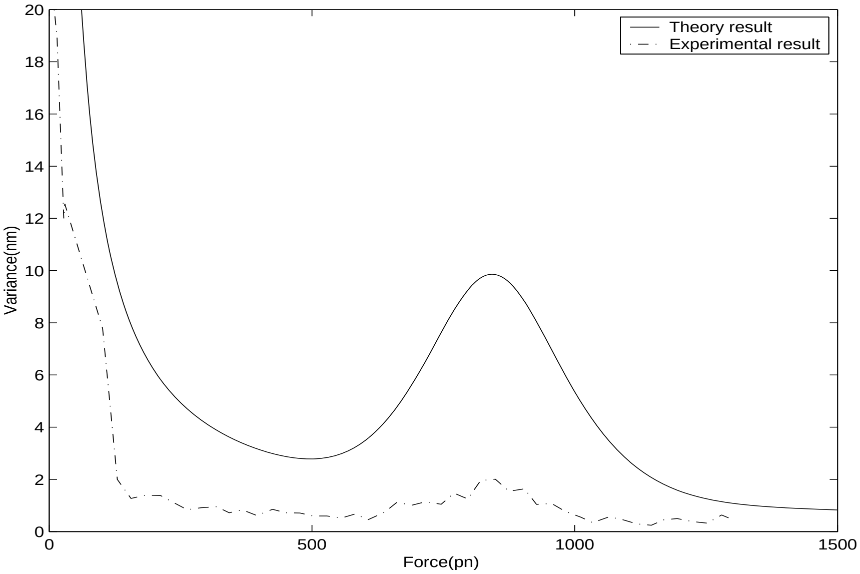


Figure 5.1. Comparison of experimental results and theoretical results for variance (dextran).

Equation (5.22) shows that variance is proportional to the inverse of the spring constant: $v = Nk_BT \cdot \frac{d\bar{L}}{dF} \propto \frac{1}{k_{spring}}$. Where $k_{spring} = \frac{dF}{dL}$ is the spring constant of dextran. Figure 5.2 shows the comparison of $C \cdot \frac{1}{k_{spring}}$ and the experimental variance. The conclusion that variance is proportional to the inverse of the spring constant seems correct. However, why is the coefficient constant $k_BT/5$ instead of k_BT. My guess is that since the experiments are done in solvent, viscous forces might reduce the variance. To check this hypothesis we can do the experiments in different solvents and compare the variance. Another guess is that the stiffness of AFM tip might also reduce the variance.

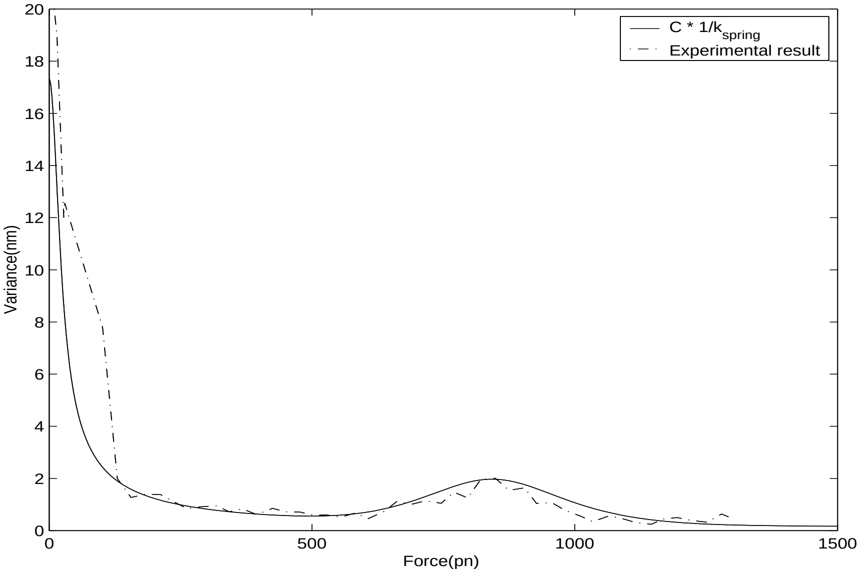


Figure 5.2. Comparison of experimental results and $C \cdot \frac{1}{k_{spring}}$, where $C = k_B T/5$.

Based on the expression for the variance (5.8), for some given force it is easy to calculate the corresponding variance. But for some given extension, it is difficult to get the variance. Therefore we show the force-variance curve. But for titin, the same force can corresponding to several different peaks. So we do not compare the experimental and theoretical results for titin.

CHAPTER 6

Constant-force experiments

In this section, we made an effort to reproduce the constant-force experimental results by numerical simulation. We try a set of models, from very simple test model to more complicated models. In general, parameters are chosen to fit the experimental results the best, not chosen based on measured values. How force effects the folding procedure is discussed in Sec. 11, where a more physically-meaningful model is given. Therefore, this section should be regard as an "intermediate state" of our study of protein folding.

In this section, the step-like behavior of single domain folding is reproduced (e.g. experimental results: Fig. 6.3; simulation results: Fig. 6.30). We also find that if we introduce some cooperativity between domains (*i.e.*, if one domain finishes folding it can accelerate the folding of other domains), then the folding procedure of a multi-domain protein will fold in stages (experimental results: Fig. 6.1; simulation results: Fig. 6.38). The reason for this cooperativity between domain is unknown. Sec. 11 discusses the reason of cooperativity inside one protein domain.

6.1. Results of constant-force experiments

Fernandez and Li used force-clamp AFM to record time-extension curves for ubiquitin [**21**]. Nine individual ubiquitin domains were linked together to form a single poly-ubiquitin molecule. The AFM tip would intercept this molecule at a random location and so pick up an indeterminate number of domains.

Figures 6.1-6.3 show some of their experimental results. The two-state model described in Sec. 4 predicts that for the constant-force experiments, the folding procedure should be step like. In the two-state model, one domain can only have two possible lengths, the folded length or the unfolded length. When a domain changes from the unfolded state to the folded state, there is a collapse of the total end-to-end length. If there are several domains in the protein, there should be several step-like collapses. However, in Fig. 6.1 and Fig. 6.2, we do not see this step-like behavior. It seems that we need to modify the two-state model.

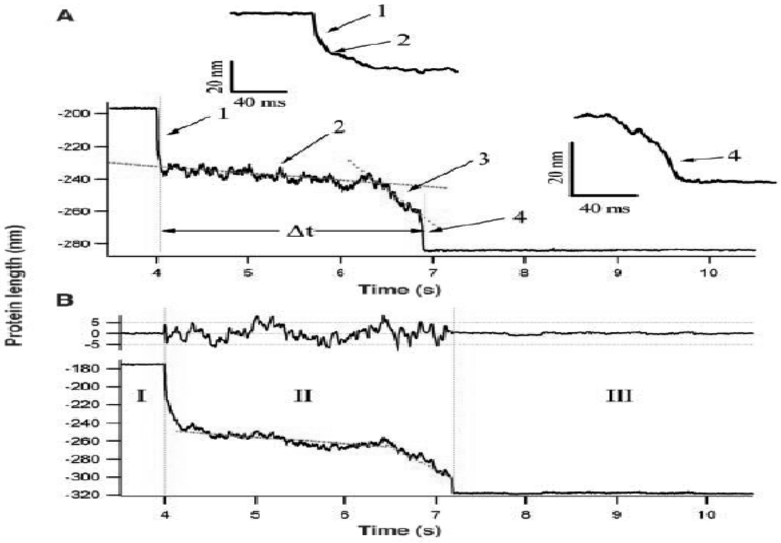


Figure 6.1. Examples of the experimental results for the constant-force folding experiments of [**21**]. Other of their results are shown in Figs. 6.2 and 6.3.

6.2. Modeling constant-force folding: Introduction

The model in this section divides the ubiquitin folding procedure into five stages, as shown in Fig. 6.4. The first stage is the nearly instantaneous transition from the coil-like structure under high external force to a shorter coil-like structure at the lower quenched force. In the second

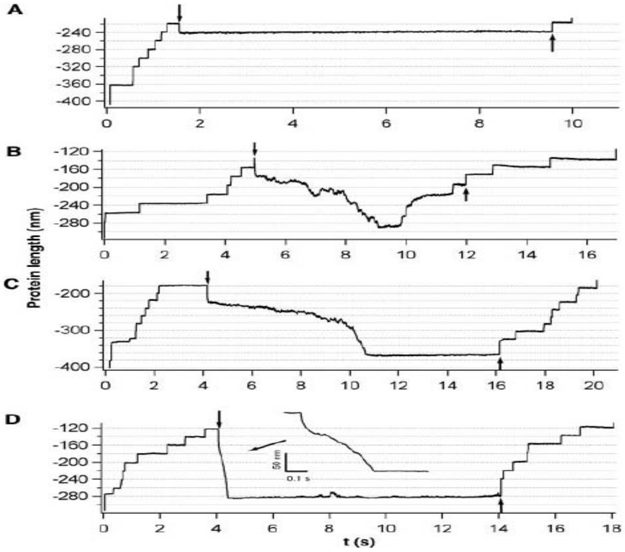


Figure 6.2. Additional results from [**21**].

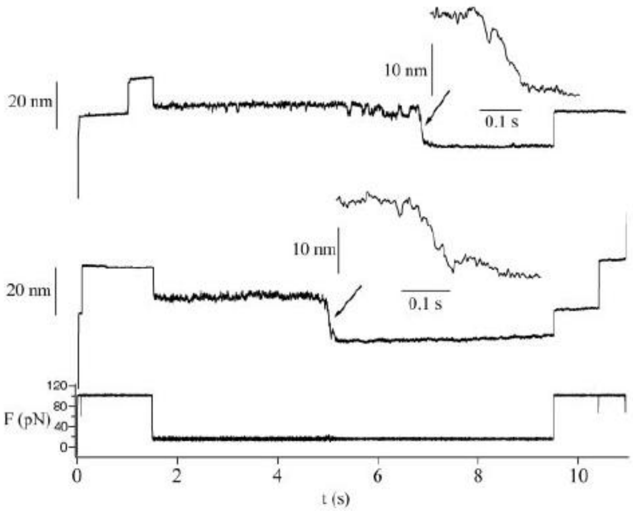


Figure 6.3. Additional results from [**21**], where it is assumed that there is only one ubiquitin domain being stretched by the AFM. Note, in comparison with Fig. 6.1 for poly-ubiquitin, that Stage 2 is nearly of constant extension, Stage 3 is absent, and while Stage 4 is an abrupt decrease, it is less sudden than in Fig. 6.1.

stage, the α helices form with folding rate k_{pf} and with A_h as the amplification factor: once one helix appears, the neighboring amino acids will fold faster. These two parameters determine the duration of this step. The details will be shown later in this section. At the next stage, β sheets form with parameters k_{pf} and A_b. In the last step, α helices and β sheets form the tertiary structure (native state) with parameters k_{sf} and A_s.

6.3. Modeling constant-force folding: Simplified model

In this section, we first introduce the domain structure in more detail, and hope to find a reasonable simulation based on the change of the structure during the unfolding and refolding procedure. Recall that amino acids are connected by peptide bonds to form the protein chain. This is called the primary structure. Different regions of the amino acid sequence form local regular structures, such as α helices and β sheets, held together by hydrogen bonds. The *domain* is the tertiary structure, which is formed by the interaction of several secondary structures. Though it is not certain what is happening during the stretching procedure, a reasonable assumption is that the entangled secondary structures are first separated, and then the secondary structures are broken, leaving the amino acid chain. We regard this chain as the unfolded state. The refolding procedure should be the reverse: first the amino acid sequence forms into secondary structures–we can call this the coil-helix procedure; then the secondary structures nucleate and fold to form the tertiary structure, *i.e.*, the folded domain. In the following we will concentrate on the refolding procedure.

This is a brief description of the folding and refolding procedure. In order to describe the experiments in [**21**], we first describe the ubiquitin domain. Since the unfolded length of the domain is 32 nm and the average length for one amino acid is 0.375 nm, we find $N_a = 32/0.375 = 85$. (The actual value for ubiquitin is 76.) This 32 nm is an estimate from the experimental single domain folding results shown in Fig. 6.3: the total length change during folding is about

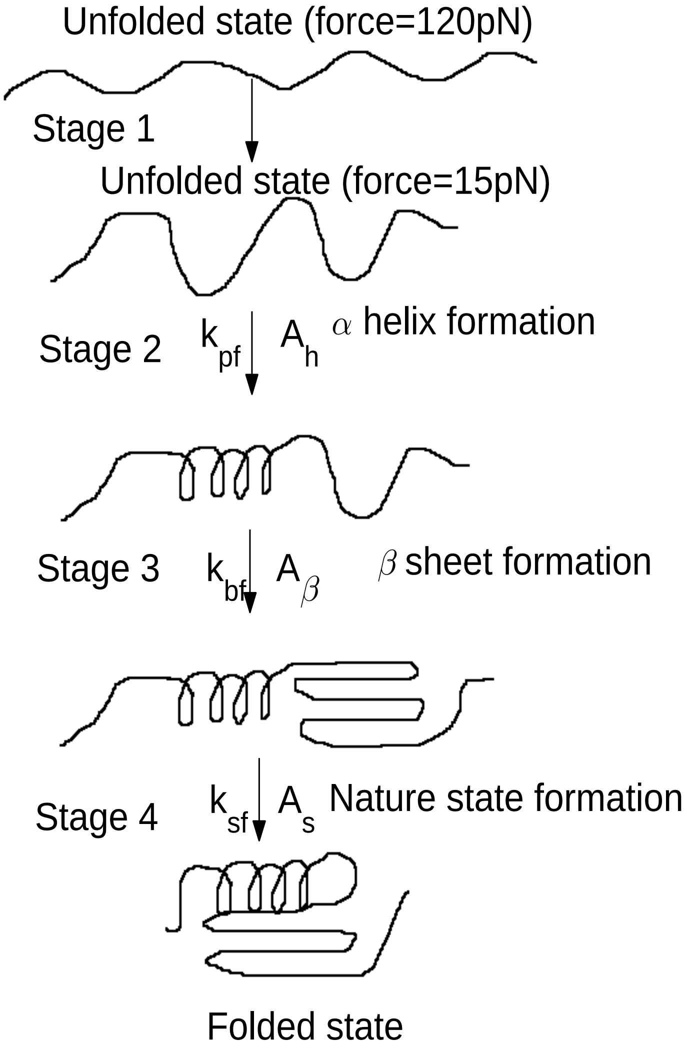


Figure 6.4. Simulation procedure and the parameters used in each step. The unfolding rates are set to be zero.

28 nm, and the folded unbiquitin has length 3.8 nm [**14**]. So, we assume that the total length of an unfolded ubiquitin is about 32 nm. The estimated value of 85 amino acids is larger than the actual value. But here we only give a simple model to see how to improve the model, so we use the value 85. The actual value will be used in the improved model. As for secondary structures,

for simplicity, here we assume that all of them are α helices (actually there are three α helices and four β sheets [**3**]). Two of the α helices are so short that the structure is sometimes classified as one α helix and four β sheets [**3**]. But, we consider seven secondary structures, each an α helix of 12 amino acids and hence $N_a = 84$.

Now we want to determine how the coil-helix procedure happens. If the protein is in the coil state, it has more possible configurations and therefore larger entropy. To form the more regular helix structure, the entropy decreases, so energy is required to make this happen. On the other hand, once the regular structure is formed, the hydrogen bonds between amino acids will decrease the energy. This is equivalent to a two-state model: first a energy barrier needs to be overcome and then the system will go to a lower energy state. This implies that we can still set some rate constant to determine the probabilities of changing state. And, as in the two-state model, this rate should depend on the external force.

In the two-state freely-jointed chain model, all the domains are refolding independently. But for the coil transfer to α helix, once one hydrogen bond is formed, it should make the neighboring bonds become easier to form. So, we introduce an amplification factor A: once a hydrogen bond is formed, the neighboring rate constants will become larger by a factor of A. We hope to get the cooperativity effect by this factor.

We now ask, "When do the secondary structures begin to nucleate?" and, "How do they nucleate?" We still hope to use the rate constants to describe this procedure. But, the secondary structures must be formed before they can nucleate to form the tertiary structure. A simple way to enforce this is not to simulate this nucleation until all the secondary structures are completed. Therefore, we introduce a parameter to describe the percentage of completion, C. The rate constant of the secondary structures' nucleation depends on C: if C is zero, the rate constant of nucleation is zero, when C increases the rate constant should also increase. A simple scheme to implement this is to assume that when $C = 1$, the rate constant for the secondary structure is k_s, otherwise it is Ck_s. Also, we have a amplification factor A for this nucleation.

Finally, we have the simple model, summarized as follows. Initially, there are N_p primary units, which are the amino acids with length $l_u = 0.375$ nm. We divide those amino acids into seven regions. In one region, they have some rate constant k_{pf} to fold to length $l_f = 0.15$ nm. If one amino acid is in the folded state, the neighboring amino acids will have the rate constant Ak_{pf}. At the same time, they also have some unfolding rate constant k_{pu}. This is the dynamics for coil-helix folding.

For secondary structure (helix) folding, the length of one secondary structure S_u is equal to the sum of the lengths of all the amino acids inside if the helix is in the unfolded length. Once folded, we assume its length becomes $S_f = 4/7$ nm, since it is observed in [**14**] that the length of one folded domain (tertiary structure) is about 4 nm and there are seven secondary structures in one domain. Of course, this value of S_f is based on the unreasonable view that all of the structures are aligned linearly in a row. The folding rate constant is $C \cdot k_{su}$, and the unfolding rate constant is k_{sf}. The amplification factor has the same effect as in the coil-helix folding procedure.

6.4. Modeling constant-force folding: Including the externally-applied force in the coil-helix model

We now include the externally applied force in the model. Force can affect the model in two ways. First, the rate constant depends on force. Second, it tends to align the amino acids along the same direction. Because of thermally-induced fluctuations, normally the backbone is not a straight line. And, the actual end-to-end length, *i.e.*, extension, is normally shorter than its backbone length, *i.e.*, contour length. The difference between them will become smaller as the force is increased. There are two models to define extension as a function of force. One is the worm-like-chain model (WLC) [**33**]. In this model, at a certain extension length x and contour length L, the force F can be calculated as

$$F = \frac{k_B T}{p}\left(\frac{1}{4(1-x/L)^2} - \frac{1}{4} + \frac{x}{L}\right) \tag{6.1}$$

where p is the persistence length, which describes the chain stiffness.

The other model is the three-dimensional freely-joint-chain model (FJC). In this model, the system is composed of N bars with length l. These bars are connected one-by-one into a chain, and all the joints can rotate freely. If we stretch the system with a certain force f, then the average length projected in the force direction is

$$x = N \cdot l \cdot \frac{1}{a}(a \cdot \mathrm{ctanh(a)} - 1) \tag{6.2}$$

where $a = fl/(k_B T)$ and ctanh is the hyperbolic cotangent.

Figures 6.5 and 6.6 compare the force extension relationship for WLC and FJC, using a linear scale and a log-log scale.

Now, we need to consider how to include the WLC or FJC model into our simulation. In the previous section, we regarded the length of an unfolded secondary structure as the sum of the lengths of all the amino acids. This is the same as the contour length of one secondary structure. If we use the WLC model, then for a certain force, (6.1) can be applied to calculate the extension length. If we use the FJC model, then the average length can be determined by (6.2). All the other parts of the simulation remain unchanged. Figures 6.5 and 6.6 are the numerical results for WLC and FJC. In these two plots, the persistence length is $p = I$. Taylor expansion shows that if $p = I/2$, the WLC and the FJC chain models give the same values in the low-force region.

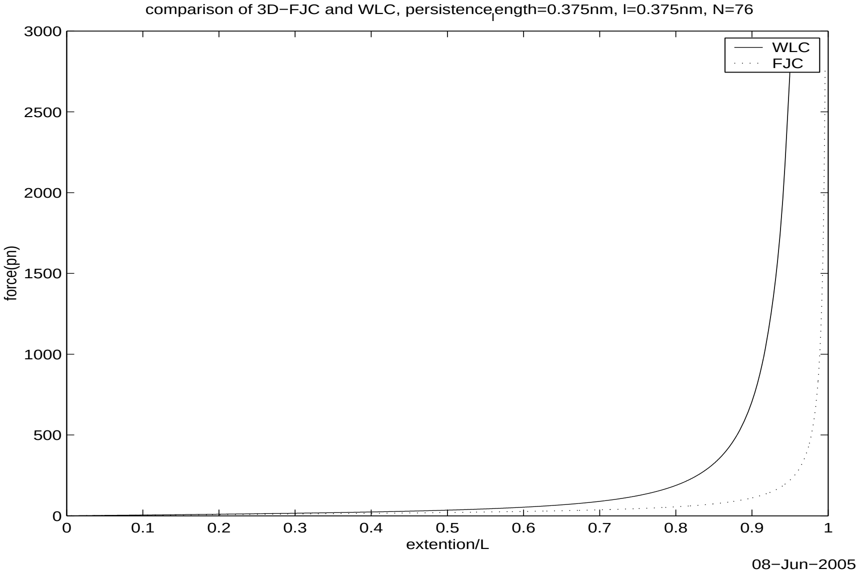


Figure 6.5. Force-extension relationship for WLC and FJC models using linear scales. The persistence length is set equal to the length of an amino acid, $p = I$.

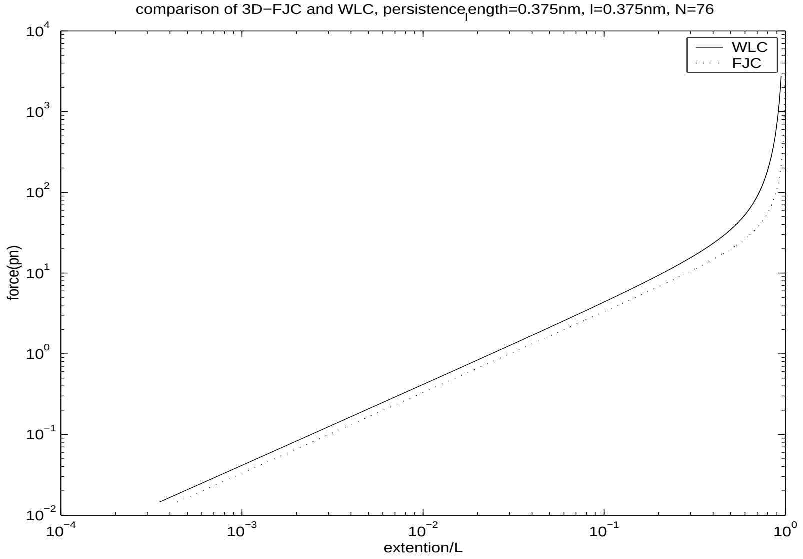


Figure 6.6. As in Fig. 6.5, but plotted on a log-log scale.

Fig. 6.7 shows the results for $I = 0.375$ nm and $p = I/2$. We can see that in this case, the WLC and FJC give the same simulation results as the force goes to zero.

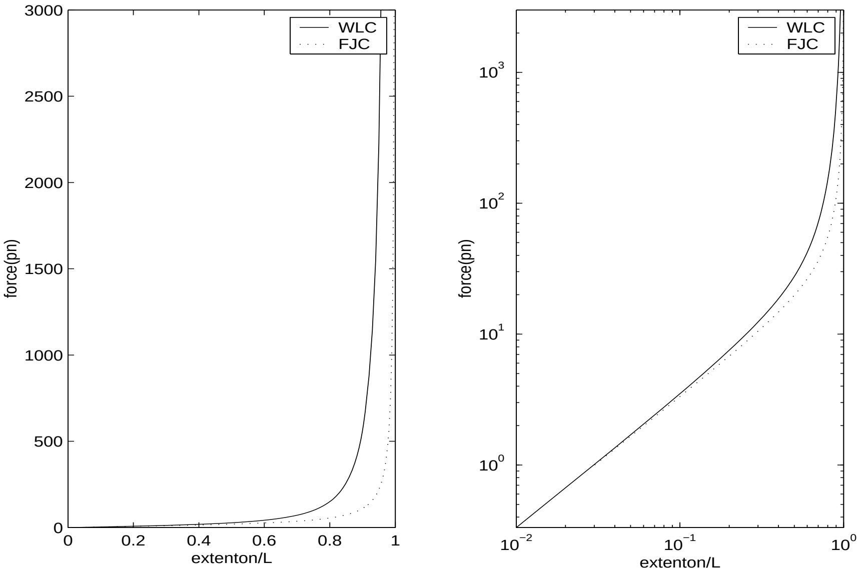


Figure 6.7. On the left is the force-extension relationship for WLC and FJC models using linear scales, p=I/2. The right plot uses a log-log scale. As in Figs. 6.5 and 6.6, as force goes to zero, the two lines approach one another.

Now, we can combine the coil-helix model and externally-applied force model to get some simulation results. The effects from each of the parameters are easy to parse out. In Sec. 6.6, parameter effects are discussed in details. Then, in Sec. 6.8, other modifications can be considered. Figure 6.8 shows the simulation results using the model described in Sec. 6.3 and 6.4.

6.5. Some analysis based on the experimental data

While most of the results shown in [**21**] are for poly-ubiquitin, two examples of the experimental results appear to be for the folding of a single domain, see Fig. 6.3. These two results show step-like behavior. The folding of one domain should be easier to understand than multiple domains. We now do some analysis for the one-domain experimental results.

The lowest trace in Fig. 6.3 shows the applied force. The upper two traces show the results from two experiments. After the applied force decreases from 100 pN to about 15 pN, there is an

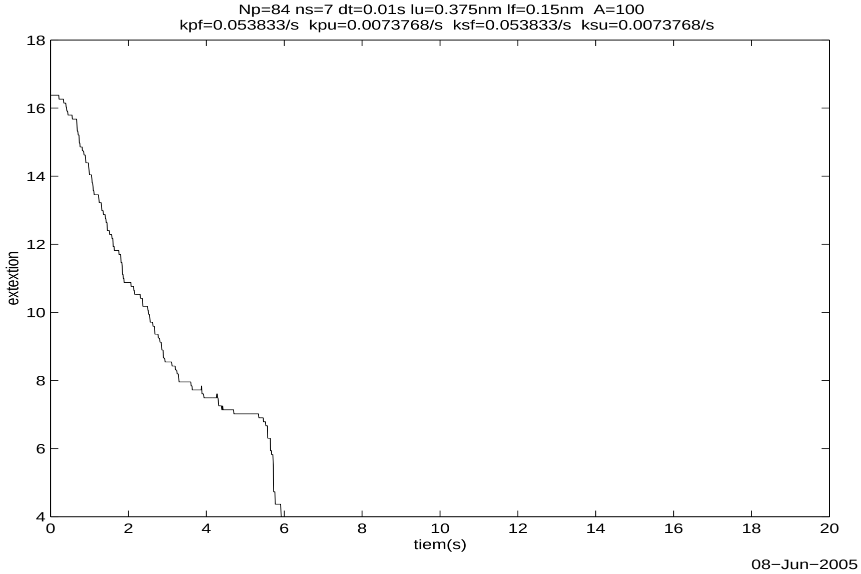


Figure 6.8. Refolding under constant external force 15 pN. The WLC model is applied to calculate the length of the secondary structures as described in Sec. 6.4.

immediate collapse in length of about 14 nm. Then, the length fluctuates for about 4-6 seconds during which time the average length does not change too much. Then there is a second quick collapse of about 20 nm.

The initial total length of the domain should be more than $14 + 20 = 34$ nm. We know that the end-to-end length of one amino acid is 0.375 nm, and there are 76 amino acids in one domain. Even if all of them are aligned in a straight line, the total length is only $76 \times 0.375 = 28.5$ nm, still shorter than 34 nm. One of the possible reasons for this is in the experiments they might pick up not one domain but one plus part of the next domain. And, between two domains there is a linker, which also has a certain length.

We consider the first collapse to be due to the coil relaxation (coil relaxation is the length change only due to reorientation, there is no state change) on the release of the high force. There might also be some coil-helix transition during this short time. The initial total length of the domain is more than 34 nm, therefore the first collapse is less than $14/34 = 40\%$ of the contour length. If we chose the FJC model to describe this procedure with unit length $l = 0.375$ nm, using (6.2) we can get at $f = 100$ pN, the extension divided by the contour length is $x/L = 89\%$, and at $f = 15$ pN, $x/L = 41\%$. Then, the length change would be $89\% - 41\% = 48\%$. This is bigger than 40%, and if there is coil-helix transfer, then the change should be even bigger. If we chose the WLC model with persistence length 0.375/2 nm, the prediction is $78\% - 34\% = 45\%$, still bigger than the experimental results. This error comes from the WLC model or FJC model.

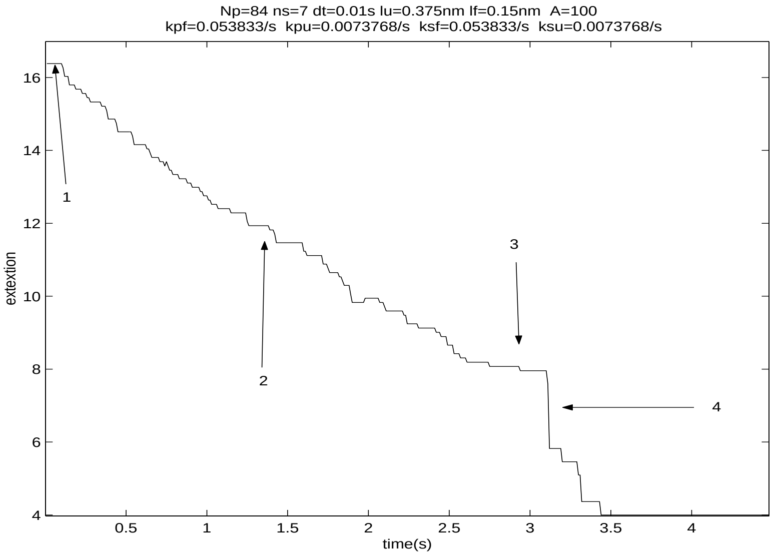


Figure 6.9. Typical results of the combined model described in Sec. 6.3 and 6.4. The Stages are labeled here, because it does not looks like the step-like single domain experimental results; it looks like the stage-like multi-domain folding. Further modification (by accounting for the actual ubiquitin structure, as described in Secs. 6.7 onward) makes these stages disappear: single domain folding shows step-like behavior.

6.6. The effect of the parameters in the model

We have many parameters in the model: k_{pf}, k_{pu}, k_{sf}, k_{su}, A_s (amplification factor for secondary structures to form tertiary structure) and A_p (amplification factor for amino acids to form helices). To show each one's effect on folding, we vary them one-by-one while keeping the others constant.

In Fig. 6.10 we show the results for different k_{pf} and k_{sf}, the folding rate constants for coil-helix transfer (rates that amino acids change from unfolded state into helix structure.) and secondary structure folding. All the plots in the same row (column) have the same k_{pf} (k_{sf}) value. Figure 6.11 is the percentage of amino acids in helix structure. We include this figure because we have two state changes: coil-helix transfer and secondary structure folding. This figure shows how much of the extension change comes from coil-helix transfer and therefore we can also know how much change comes from the other transfer. Here we can see that k_{pf} and k_{sf} determine the length of Stages 1 and 3,shown in the previous section. The length of the stage 1 is the waiting time before the first α helix structure appears, therefore it is determined by the folding rate of the α helix structure k_{pf}. Similarly for Stage 3, which is determined by the β sheet structure folding rate. We use the first row in Fig. 6.10 as an example to show the effect of k_{sf}. In the left plot, the extension is always above 10 nm, which means the protein never reach the folded state (4 nm). In the middle plot, the protein folds at time $t \simeq 3$ s. In the right plot, the protein folds at time $t \simeq 2$ s. It is clear that larger k_{sf} leads to faster folding. All the three rows show the same tendency. Similarly, larger k_{pf} can also make the folding faster.

Figures 6.12 and 6.13 are for different A_s and A_p. A_s is the amplification factor for secondary structure folding. A_p is the amplification factor for primary structure folding. Here we can see they determine the slope of Stages 2 and 4. The duration of Stage 2 is the time to finish α helix formation, which is determined by both k_{pf} and A_p. So A_p can effect the slope of Stage 2. Similarly for Stage 4 and β sheet formation. Larger values correspond to steeper slopes.

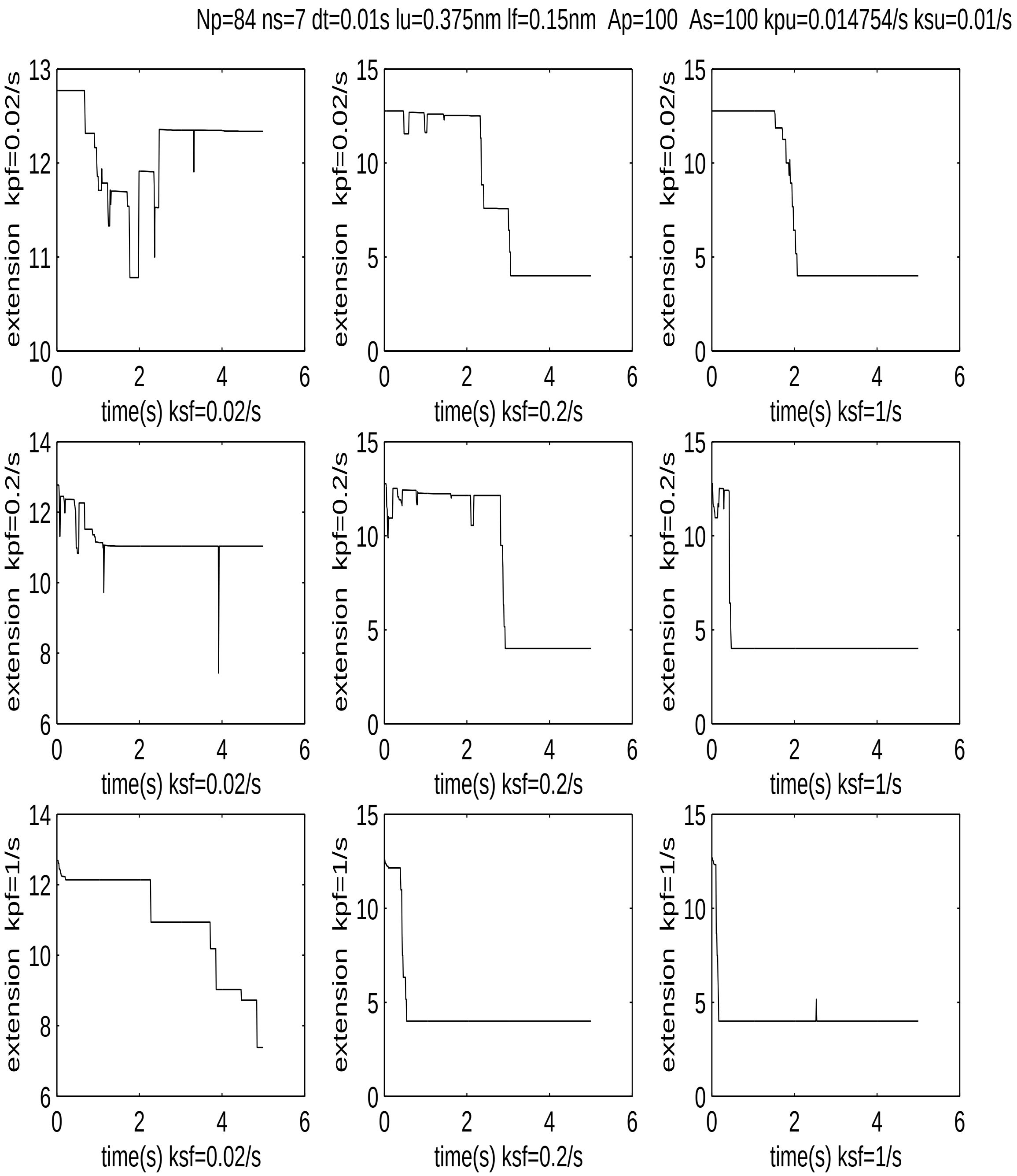


Figure 6.10. The effect of k_{pf} and k_{sf} on refolding. From the top to the bottom, k_{pf} increases. From the left to the right, k_{sf} increases.

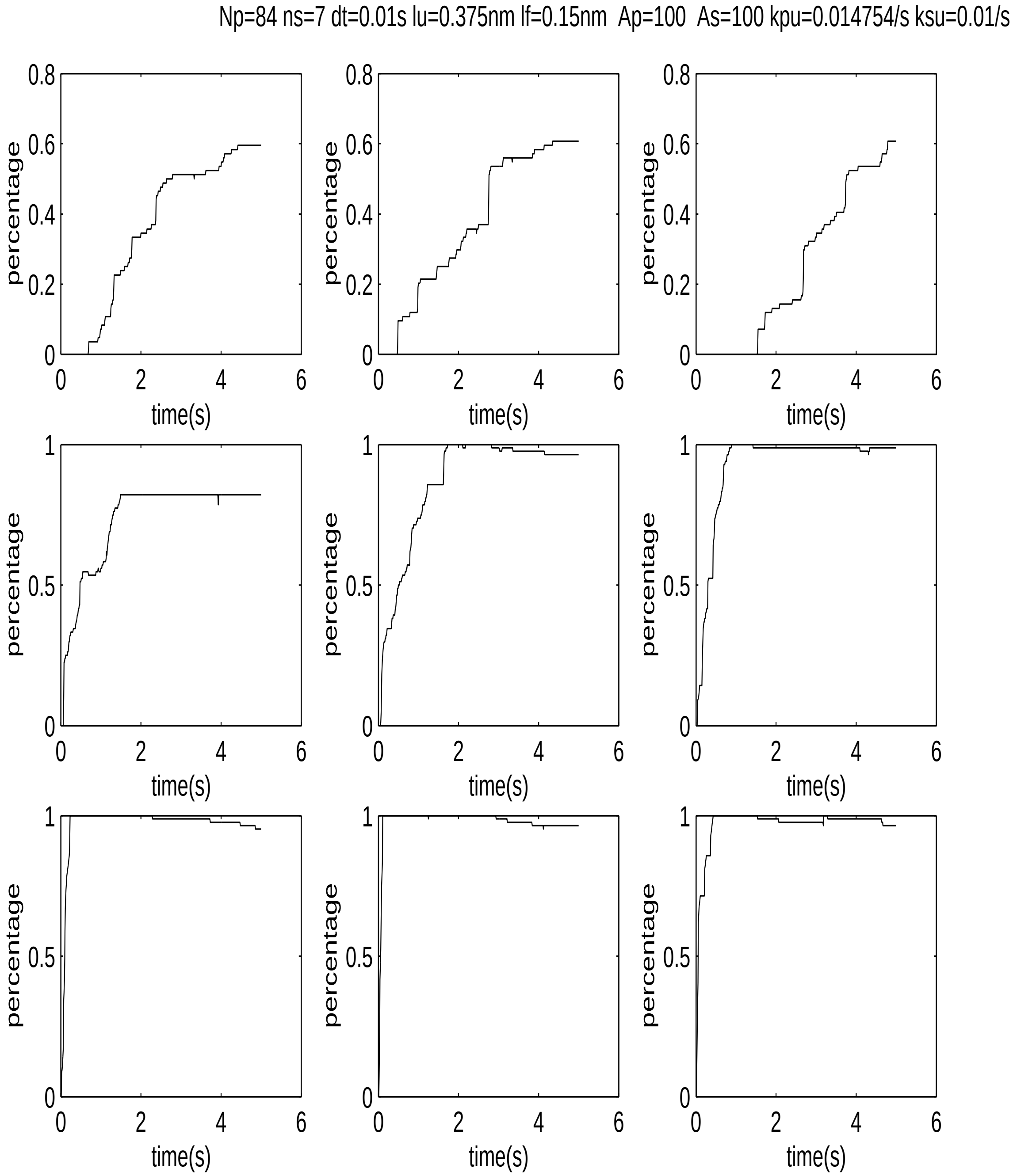


Figure 6.11. The effect of k_{pf} and k_{sf} on refolding. Here "percentage" means the percentage of amino acids in helix state. The values of k_{pf} and k_{sf} are as shown in Fig. 6.10.

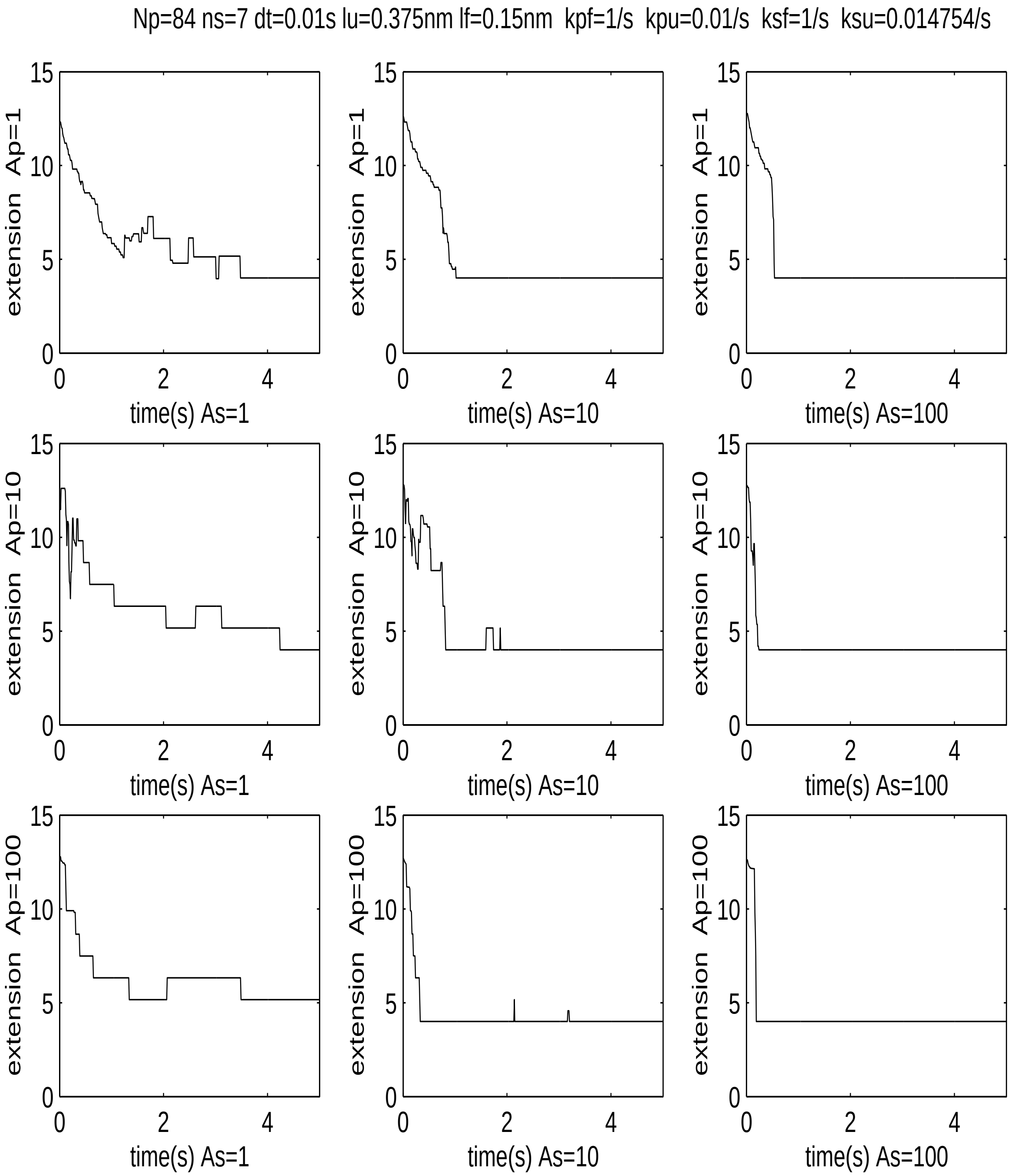


Figure 6.12. The effect of A_s and A_p on refolding. From the top to the bottom, A_p increases. From the left to the right, A_s increases.

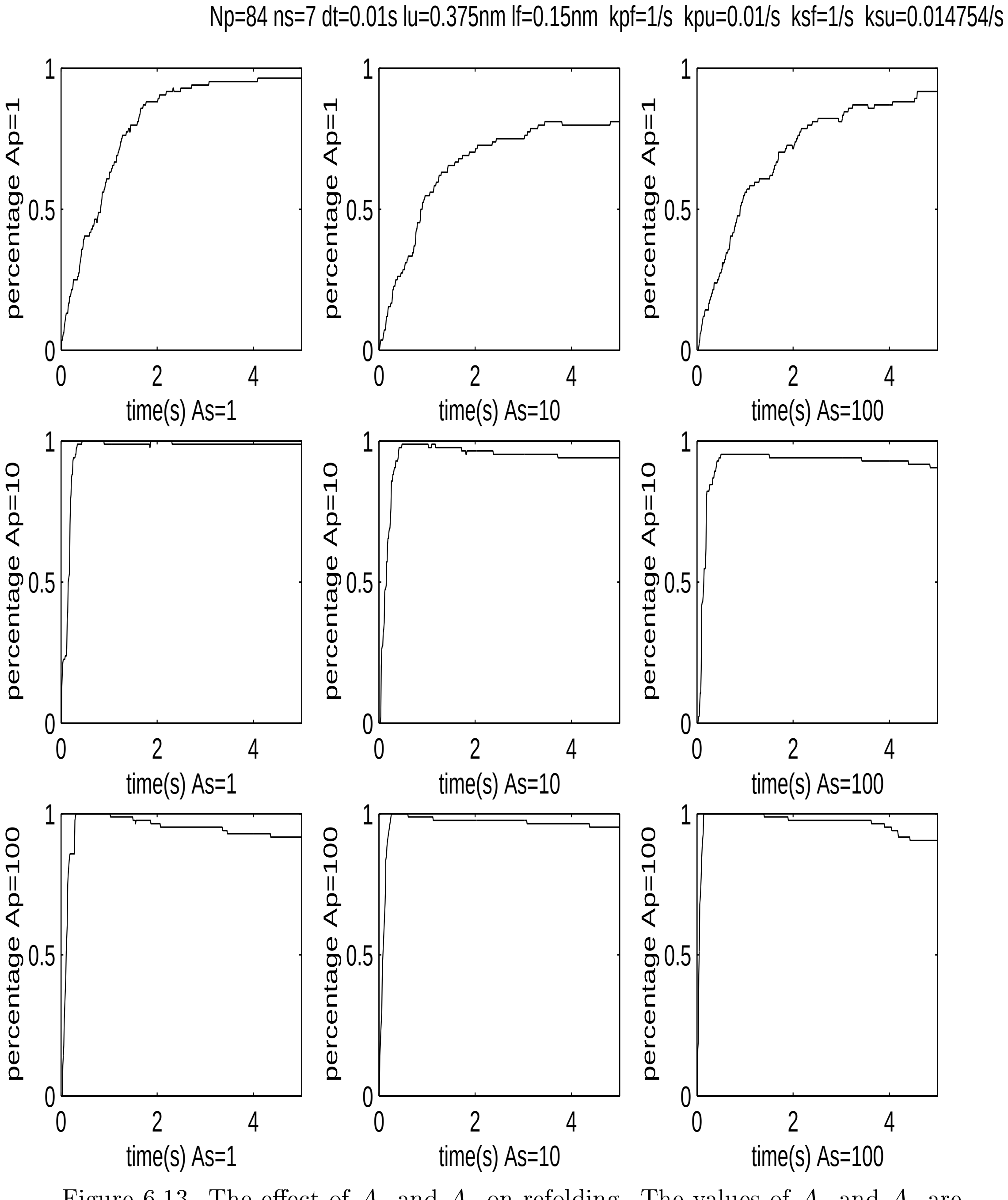


Figure 6.13. The effect of A_s and A_p on refolding. The values of A_p and A_s are as shown in Fig. 6.12

Figures 6.14 and 6.15 are for different k_{pu} and k_{su}. They determine how frequently the structures can become unfolded. We use the first row as an example to show the effect of k_{su}. In the left plot, the protein always has the same length once it reaches the folded state at time t=0.3 s. In the middle plot, there are two peaks, which means there are two unfolding events. In the right plot, there are many peaks, which means there many unfolding events. So a larger value implies easier unfolding, and therefore larger oscillations about the mean. All the three rows show the same tendency. Similarly, larger k_{pu} can also make the unfolding easier.

6.7. More information about ubiquitin structure

In the current model we regard the ubiquitin as a amino acid chain with 84 units and every twelve of them make an α helix. This is not the actual structure of ubiquitin. In this section we hope to see the actual ubiquitin structure and then in the next subsection develop a model to better simulate its folding procedure.

One ubiquitin domain has 76 amino acids. Its sequence is

1 MQIFVKTLTG KTITLEVEPS DTIENVKAKI QDKEGIPPDQ QRLIFAGKQL

51 EDGRTLSDYN IQKESTLHLV LRLRGG

This set of words won't help us too much to tell the ubiquitin structure. Figure 6.16 of the secondary structures is more useful. The ubiquitin domain is composed of α helices, β sheets and loops.

Fragmentation studies, multidimensional NMR experiments, and molecular dynamics simulation on ubiquitin have indicated that strands b3, b4, and b5 are less conformationally stable (easier to break) than strands b1 and b2 and the helix [**14**].

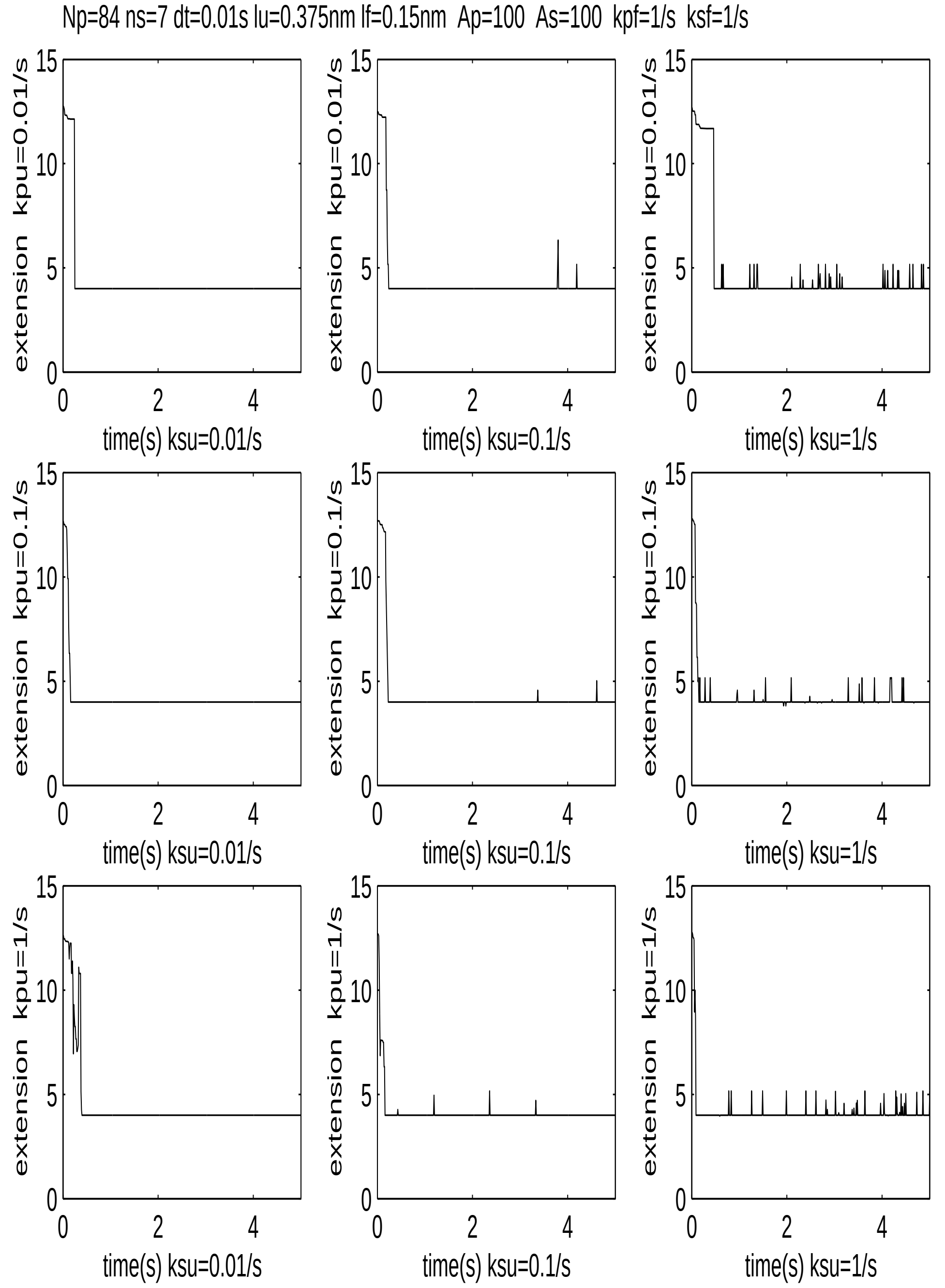


Figure 6.14. The effect of k_{pu} and k_{su} on refolding. From the top to the bottom, k_{pu} increases. From the left to the right, k_{su} increases.

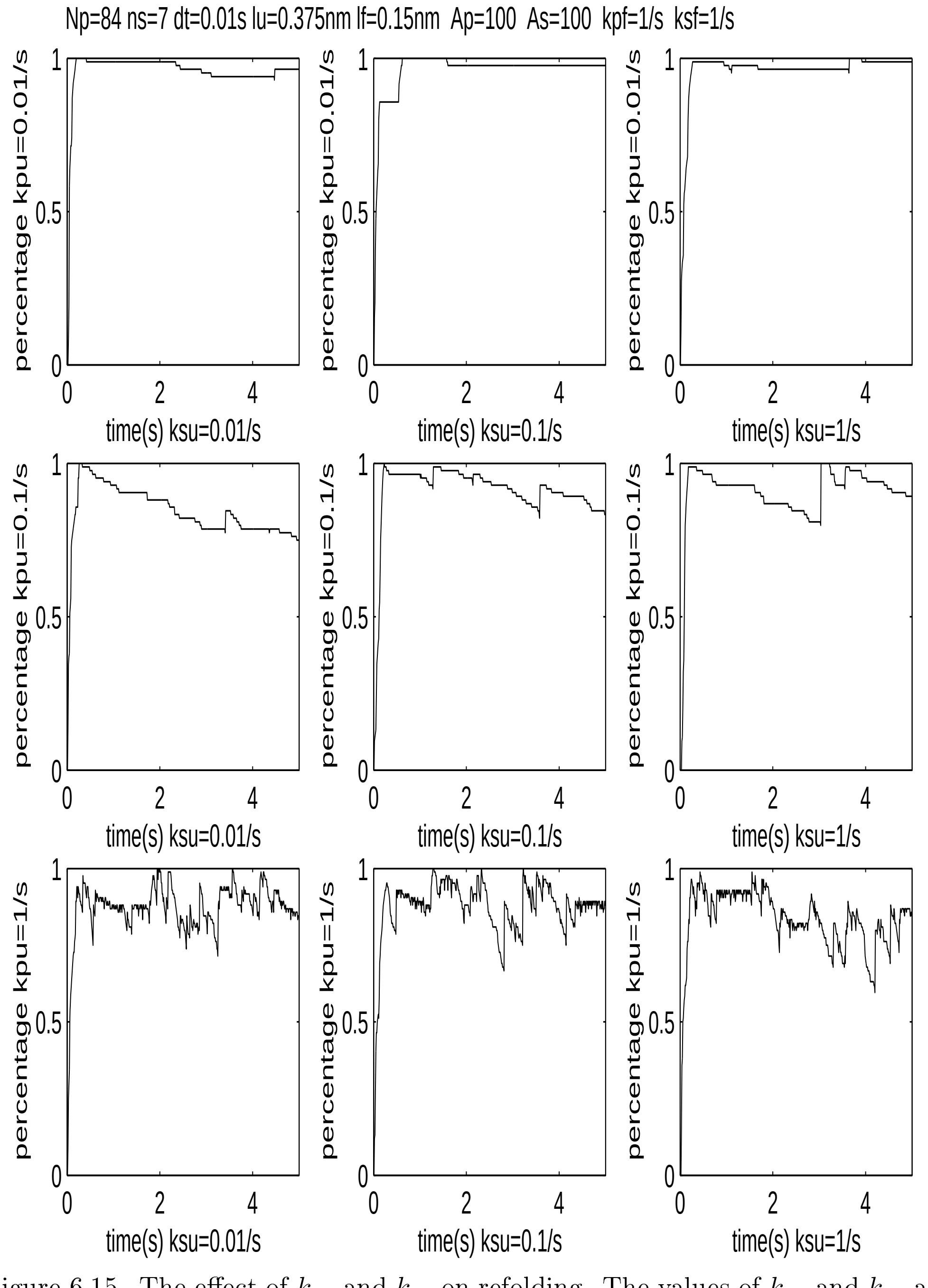


Figure 6.15. The effect of k_{pu} and k_{su} on refolding. The values of k_{pu} and k_{su} are as shown in Fig. 6.14.

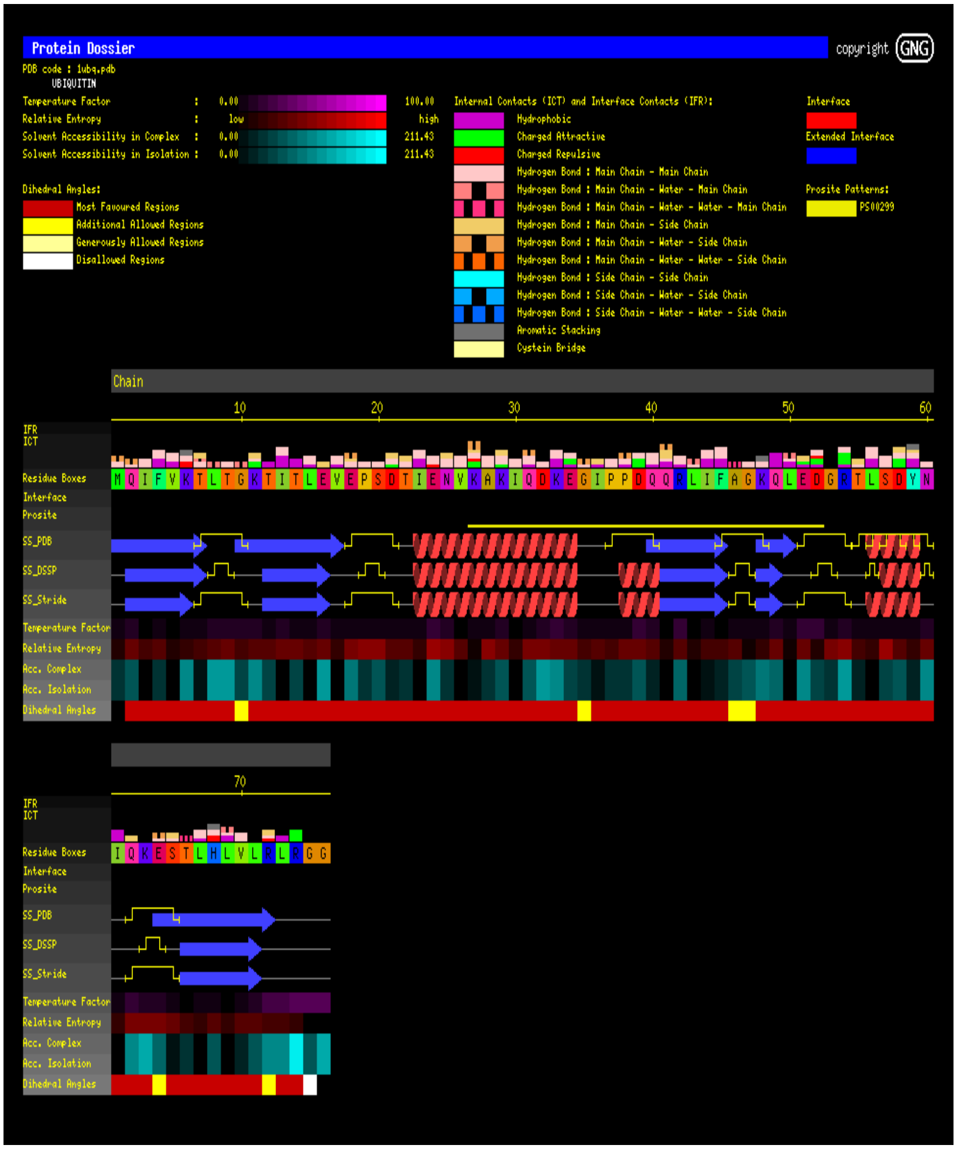

Protein Dossier
copyright GNG
PDB code : 1ubq.pdb
UBIQUITIN
Temperature Factor : 0.00 100.00
Relative Entropy : low high
Solvent Accessibility in Complex : 0.00 211.43
Solvent Accessibility in Isolation : 0.00 211.43
Dihedral Angles:
Most Favoured Regions
Additional Allowed Regions
Generously Allowed Regions
Disallowed Regions
Internal Contacts (ICT) and Interface Contacts (IFR):
Hydrophobic
Charged Attractive
Charged Repulsive
Hydrogen Bond : Main Chain - Main Chain
Hydrogen Bond : Main Chain - Water - Main Chain
Hydrogen Bond : Main Chain - Water - Water - Main Chain
Hydrogen Bond : Main Chain - Side Chain
Hydrogen Bond : Main Chain - Water - Side Chain
Hydrogen Bond : Main Chain - Water - Water - Side Chain
Hydrogen Bond : Side Chain - Side Chain
Hydrogen Bond : Side Chain - Water - Side Chain
Hydrogen Bond : Side Chain - Water - Water - Side Chain
Aromatic Stacking
Cystein Bridge
Interface
Extended Interface
Prosite Patterns:
PS00299
Chain
10
20
30
40
50
60
IFR
ICT
Residue Boxes
Interface
Prosite
SS_PDB
SS_DSSP
SS_Stride
Temperature Factor
Relative Entropy
Acc. Complex
Acc. Isolation
Dihedral Angles
M Q I F V K T L T G K T I T L E V E P S D T I E N V K A K I Q D K E G I P P D Q Q R L I F A G K Q L E D G R T L S D Y N
70
I Q K E S T L H L V L R L R G G

Figure 6.16. The details of ubiquitin structure. The three different types of structures are from PDB (Protein Data Bank), DSSP (Database of Secondary Structure in Proteins) and Stride (a software tool for secondary-structure assignment from atomic resolution protein structures). In the following, we use the structure from the PDB. The five β strands from the beginning to the end are named from b1 to b5. Also see Fig. 6.17.

The two ends of proteins are designated as the N-terminal and the C-terminal. The amino acid at the N-terminal of ubiquitin is Met1, where “Met” is the abbreviation of the name of that amino acid and "1" designates the position. It is 1 since this is the conventional beginning position. The other end, C-terminal (Gly76), is regarded as the end. All amino acids in a protein can be represented by a name and a number, showing its type and position counted from the N-terminal.

A polyubiquitin chain can be connected in two ways. One way is the C-terminal of one ubiquitin connected to the N-terminal of the next ubiquitin. This is called the N-C-linked polyprotein. The other way is the C-terminal of one ubiquitin connected to Lys48 of the next ubiquitin. This is called Lys48-C-linked polyprotein.

For different types of connections, constant velocity stretching experiments give out different results [**14**]. For N-C-linked polyprotein, the measured change of contour length in one unfolding is $\Delta L_c = 24 \pm 5$ nm. The unfolding force is $\Delta F = 203 \pm 35$ pN at an average velocity of 30 nm/s(range 25-41 nm/s). For the Lys48-C-linked polyprotein, $\Delta L_c = 7.8 \pm 2.8$ nm. The unfolding force is $\Delta F = 85 \pm 20$ pN at an average velocity of 30 nm/s(range 28-31 nm/s). Based on the change of length, the polyubiquitin used in Fernandez's experiments should be N-C linked [21].

Figure 6.17 shows the secondary structures of one ubiquitin domain. All the beta strands are named as b1, b2, b3, b4 and b5 from the N-terminal to the C-terminal. For N-C-linked polyprotein, simulations show that the main unfolding barrier comes from the 5 hydrogen bonds between the two parallel beta strands: b1 and b5. For the Lys48-C-linked polyprotein, the barrier comes from the 5 hydrogen bonds between the two antiparallel beta strands: b3 and b5. Lys48 is the first amino acid of b4, the smallest beta strand [**14**].

6.8. More modifications for the model

In the previous section we gave the detailed structure of ubiquitin. There are three different types of secondary structures: α helix, β-sheet and loops. In this section, we formulate a model

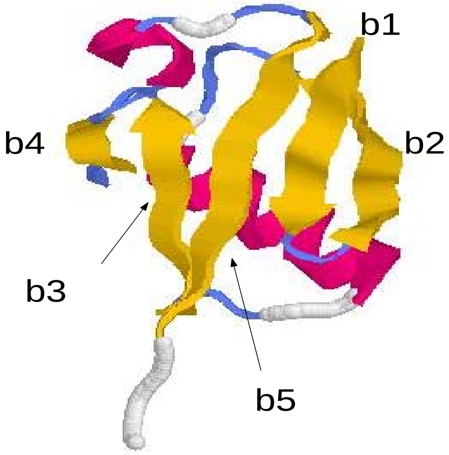


Figure 6.17. The β strands of ubiquitin. Also see Fig. 6.16. b1 is near the N-terminal beginning, and b5 is near the C-terminal end. Both of them point toward the reader, forming a parallel β sheet.

of how the secondary structures are formed and how they interact to form tertiary structure. We will look at all of the three structures one-by-one. We assume that initially (fully stretched), the ubiquitin is in the coil-like structure.

6.8.1. α helix

When the stretching force decreases, all the amino acids can change their length (as projected along the force direction) according to the FJC force-extension relationship (6.2). For example, the end-to-end length for one amino acid is 0.375 nm. When the external force is 120 pN, the projected length in force direction is 0.34 nm. When the force decreases to 15 pN, the projected length becomes 0.15 nm. This length change due to the external force is spontaneous. At the same time, neighboring amino acids can form helix structures with rate constant k_{pf}. Once one helix turn appears, all the neighboring amino acids will become A_h times easier to extend this helix. During the folding procedure there are two possible structures in one helix region: coil-like amino acids and helix-like amino acids. The length of the coil-like part is calculated according to the FJC force-extension relationship with end-to-end length 0.375 nm. The helix-like part (sub-helix structure, the helix turns have appeared, but the complete helix structure has not formed yet) is also calculated with the same relationship, but with a different length. Since the completed helix-like structure is quite rigid, we should regard all the amino acids in one structure as one bar. Its length is $n \times 0.15$ nm, where n is the number of amino acids in the helix state.

6.8.2. β sheet

Before one β strand meets some other one, we calculate its length from the FJC force-extension relationship with unit length 0.375 nm. Once one meets another, they form a β sheet, with combined length $n \times 0.375$ nm, where n is the number of amino acids in the larger β strand. In the following we describe how to calculate the end-to-end length of a β sheet. First, we consider a β sheet composed of two β strands. We use an arrow to represent a β strand pointing from the beginning amino acid to the ending amino acid. Then the directions of the first β strand and the last β strand have two possibilities: they are pointing to the same direction (parallel) or opposite directions (antiparallel).

First consider antiparallel connections. In this case we have three different possibilities. The two strands may have the same length, and therefore we can guess that when they combine together, they exactly match. The length from the beginning part of the first strand to the ending part of the second one is zero, as shown in Fig. 6.18a. The second possibility is that the two strands have different lengths, but when they hydrogen bond, the ending amino acid of the first strand just meets the beginning amino acid of the second one. In this case the net length after combination should be the length difference, as shown in Fig. 6.18b. The last possibility is that the two β strands do not have the same length, and are not connected as one's ending to the other's beginning. In this case the net length depends on how they are connected, as shown in Fig. 6.18c.

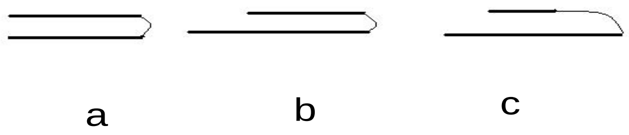


Figure 6.18. Three different types of antiparallel connection of β strands.

If two β strands are connected in parallel, there are two cases. The first one is that one's beginning exactly meets the other's beginning. Then the net length is determined by the second strand's length, as shown in Fig. 6.19a. The second case is that the two β strands do not meet beginning to beginning. Then the net length depends on how they are connected, as shown in Fig. 6.19b.

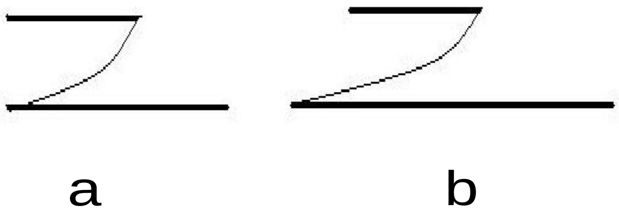


Figure 6.19. The two different types of parallel connection of β strands.

Once two strands hydrogen bond together, they behave like one new strand. Therefore, if we have a third one with which to combine, we can follow the same analysis procedure.

Figure 6.17 is the structure of the ubiquitin. We can see that first and the last strands are connected as type parallel b. Therefore the **final** total end-to-end length of protein **in the**

native state is determined by the length of this pair, in particular the last strand, which is the longest strand.

6.8.3. Loop region

The loop region is always regarded as a coil-like structure. Its length is calculated according to the FJC force-extension relationship with monomer length 0.375 nm.

6.9. Tertiary structure

In the previous section, we discussed how the secondary structures are formed. Here we hope to give a reasonable procedure for forming the tertiary structure.

First, we summarize the folding procedure. Initially, all the amino acids are in the coil state. The length projected in the force direction is calculated using the FJC model. Then, the secondary structures are formed with certain rate constants. The details of this procedure are described in the previous section. Then, once the three main parts are formed (two β sheets and one α helix, see Fig. 6.20), they begin to form a tertiary structure with rate constant k_{sf}. This folding is like the forming procedure of a β sheet: two secondary structures “merge” into one whose length is given as the longer one of the two structures. Note that the geometry used for merging is an expedient simple choice and it neglects three-dimensional effects.

6.10. Simulation procedure

The folding procedure was described before. Here it is reviewed to make clear the details of the simulation procedure. The Matlab code used to simulate is given in Appendix 13.3.

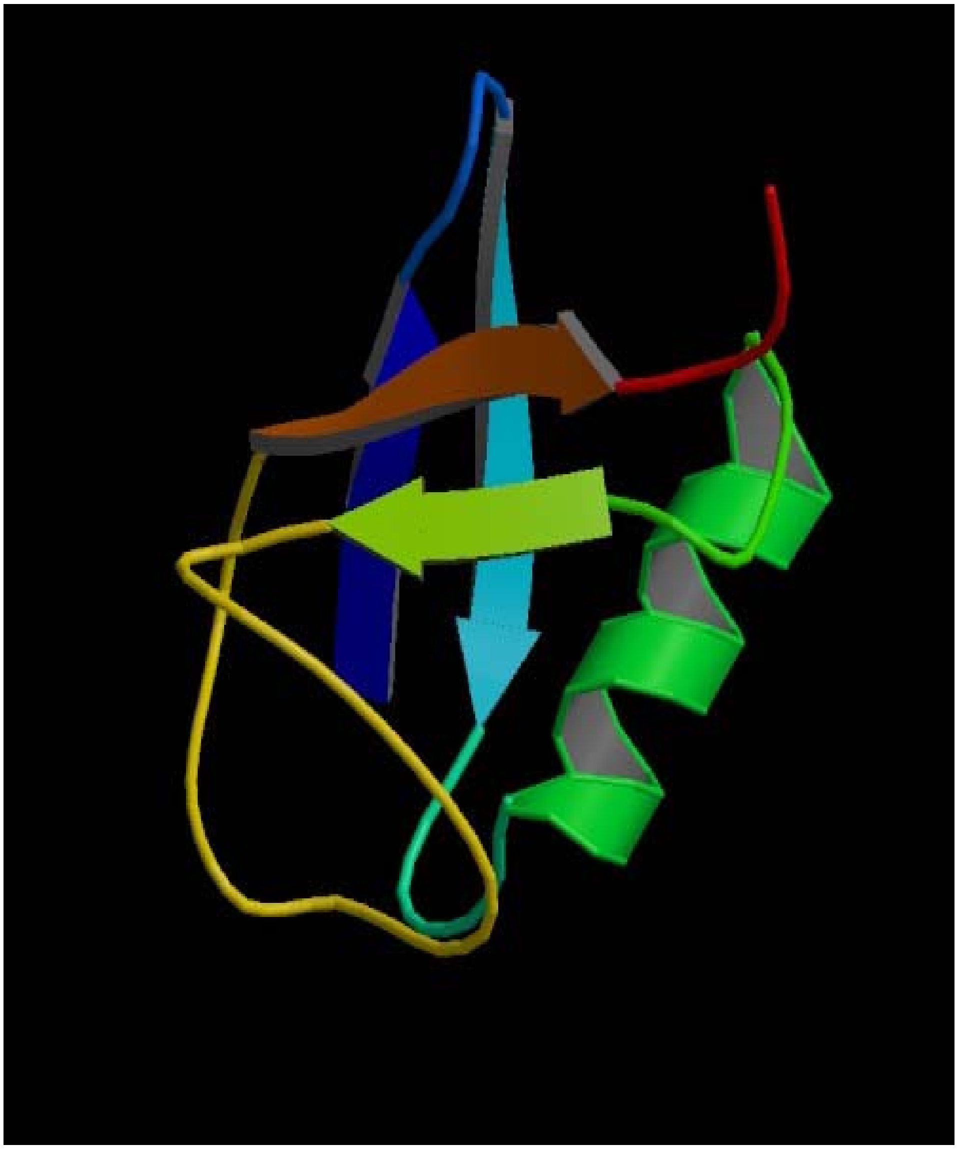


Figure 6.20. The folded ubiqintin structure. Here only the β strands b1, b2, b3 and b5 are shown. The smallest β strand b4 is not shown. The smaller α helix is not shown neither. We can divide the whole ubiquitin into three main parts: the first part is composed of b1 and b2 (the two horizontal β strands), the second part is the larger α helix, and the last part is composed of b3 and b5 (the two vertical strands).

6.10.1. Secondary folding

From Fig. 6.20 we can see that the folded ubiquitin has three main parts: two β sheets and one α helix. If we account for the two loops that connect them, there are five parts. We name these parts as 1 to 5 from the blue end to the red end. Part 1 is the first β sheet region, it includes a β strand with 7 amino acids (aa), a loop with 2 aa and a β strand with 8 aa. Part 2 is a loop with 5 aa. Part 3 is a α helix with 12 aa. Part 4 is a loop with 5 aa. Part 5 includes 6 secondary structures: 3 β strands and 3 loops. (The smaller α helix is sometimes regard as loop, here we regard it as a part of one loop.) The loops, part 2 and 4, are the easy parts, they are regarded as a freely jointet chain. Their length is calculated as the sum of freely jointed bars.

Part 3, for the α helix, is calculated based on the coil-helix transfer procedure described in Sec. 6.7.1: coil state amino acids with length 0.375 nm transfer into 0.15 nm long helix state amino acids with rate constant k_{pf}. Once any amino acid changes into the helix state, all the neighboring ones will become A_p times easier to transfer into the helix state. At the same time, amino acids in the helix state still can transfer into the coil state with rate constant k_{pu}. Unfolding is allowed only during the phase in which primary structure is folding into secondary structure.

Parts 1 and 5 are similar. In each region there are two main β strands. They have a certain rate constant k_{bf} to combine. Before one β sheet strand meets the other one, it behaves like a coil structure. Its length is calculated according to FJC force-extension relationship with length 0.375 nm. Once one meets another, they form a β sheet, where the combined length is $n \times 0.375$ nm, where n is the number of amino acids in the longer β strand. Once two β strands combine together, the end-to-end length is calculated based as in Sec. 6.8.2 on the combination of β strands.

6.10.2. Tertiary folding

After we calculate the length of each part, we need to calculate the total length of ubiquitin. For parts 1, 3 and 5, they have certain rate constants to combine. We call them $k13$ for part 1 and 3 to combine together, $k35$ for parts 3 and 5, $k15$ for parts 1 and 5. If none of them combine, the total length is the sum of the 5 parts. If any two combine together, the length will become the longest one among all the parts between these two plus all the length left. For example, if 1 and 3 are combined, the total length will become the longest one among part 1, 2 and 3 plus the length of part 4 plus the length of part 5.

6.11. Simulation results

First we take a look at a typical folding extension-time cure. The folding procedure can be separated into several stages. In Stage 1 we see a collapse. This is due to the external force change from 100 pN to 15 pN. When the external force is 100 pN, for one 0.375 nm long amino acid, the average length projected in force direction is 0.3336 nm. The total length for 76 amino acids is about 25.4 nm. When the force decreases to 15 pN, the projected length for one amino acid becomes 0.152 nm. Then the total length becomes 11.6 nm.

Stage 2 is a mostly flat region with some small wiggles and steps. These wiggles and steps come from the folding of primary structure into the secondary structures. For example, in the helix region, amino acids fold from coil-like structure into helix structure; in beta strands regions, two β strands meet and form a β sheet.

The fast collapse in stage 3 comes from the secondary structures folding to form the tertiary structure.

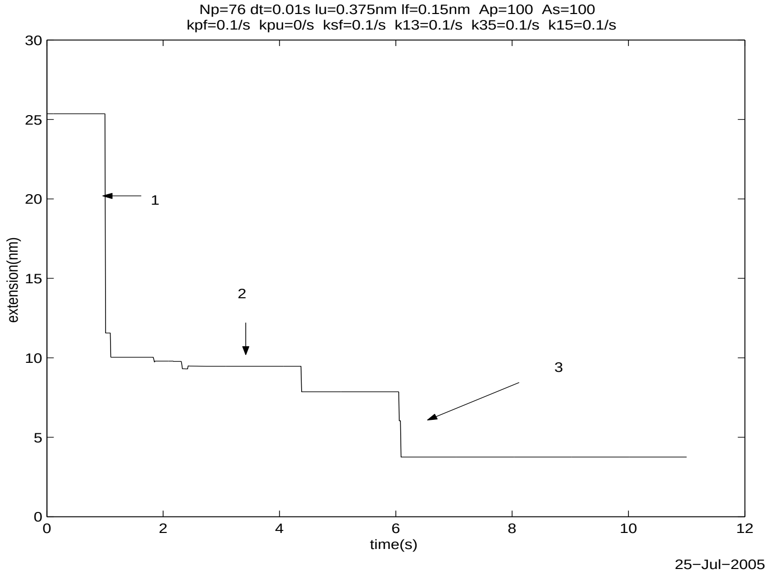


Figure 6.21. Simulation result. Described in Sec. 6.11.

6.12. Parameters used and results for different parameters

The initial high force used is 100 pN, and the low force is 15 pN. These come from Fig. (5) in [**21**].

The length of one amino acid is not the same for different types of amino acids. Normally used values can be 0.38 nm [**7**] or 0.37 nm [**22**]. The values used in the code is 0.375 nm. The length of one amino acid in an α helix is 0.15 nm [**15**]. The structure of ubiquitin comes from the PDB (Protein Data Bank). All the parameters listed above are kept unchanged during the simulation.

Here we look at the parameters we can change. The folding rate k_{bf} determines how long it takes for the β strands to fold. In Fig. 6.22, collapse b (the extension changes from about 10 nm to 8 nm, which occurs at t=2.) is due to the folding of β strands. The left plot is for k_{bf}=0.1 /s, it takes some time t_b for the collapse to happen. In the right plot k_{bf}=10 /s, and the collapse occurs in a much shorter time.

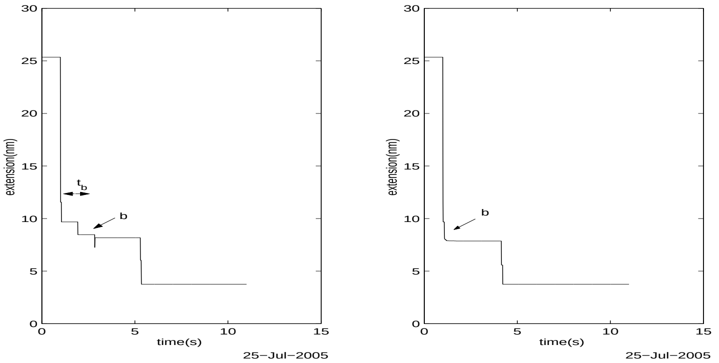


Figure 6.22. Plots for different k_{bf}: the left plot k_{bf}=0.1 /s, the right plot k_{bf}=10 /s. The arrow "b" points to the the several small collapses in the left plot. In the right plot, the small collapses shrink into one quick collapse.

Similarly, the folding rate k_{pf} determines how long is needed to wait for the forming of α helix. $k13$, $k15$ and $k35$ determine how fast the three main parts described in Fig. 6.20 fold to form tertiary structure.

A_h is the parameter used to describe, once one helix structure appears, how many times larger is the folding rate for subsequent helix turns. In Fig. 6.23 the left plot is for A_h=1, we can see many small steps for helix folding. For the right plot, A_h=100, the helix folding looks like a single step.

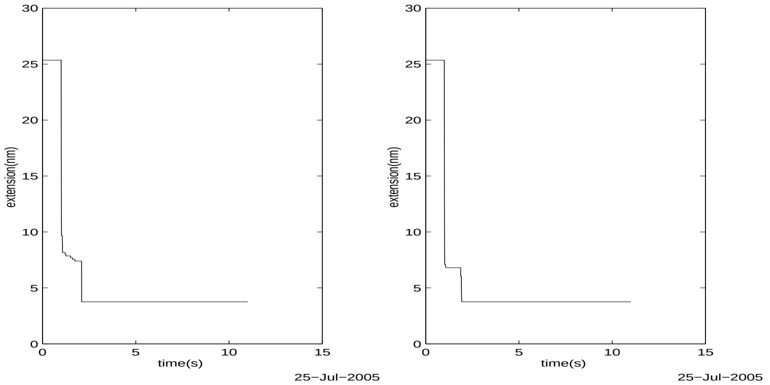


Figure 6.23. Plots for different A_h: on the left A_h=1, on the right $A_h = 100$.

The time step size used in the code is $dt = 0.01$ s.

6.12.1. A table of the parameters used in the simulation

Table 6.2 lists out the parameters used in the code. A discussion of these parameters will be given at the end of this section.

6.13. Experimental results

Fernandez and Li [**21**] used single-molecule atomic force microscopy techniques in the force-clamp mode to apply a constant force to a single polyprotein composed of nine repeats of the small protein ubiquitin.

In the experiments, a constant high force is first applied to the polyprotein. The force is then decreased to a low value. A spontaneous collapse is observed. This is called Stage 1, lasting ~ 10 ms. The second stage is characterized by a noticeable increase in fluctuations and a long lasting plateau with a slow collapse rate. Then Stage 3 appears with a abrupt increase in the slope of the collapse. The last stage is a fast and step-like collapse.

For the poly-ubiquitin experiments, they obtained a random sample of single molecules containing anywhere between one and nine repeats. In most cases, they picked up 3-5 ubiqintins.

In Table 6.3 and 6.4 we list the results gathered from the figures in [**21**]. The stages are shown in Fig. 6.1: Stage 1 is a quick collapse, Stage 2 is a flat region, Stage 3 is a steeper decrease and Stage 4 is a quick collapse. For some figures, stages are missing: these are listed here as "N.A.". For some others, the stages are not distinguishable: for these the tables show the total change of the combined stages. For [**21**]'s Fig. 3B, the protein first folds and then unfolds at the constant force. The value shown is for the folding procedure. (Fig. 2 in [**21**] is reproduced here as Fig. 6.1, Fig.3 in in [**21**] is our Fig. 6.2.)

The next table shows the time for the different stages.

[**21**] also measured the constant force folding for a single ubiquitin domain (Fig. 6.24). The two results for single domain folding show different behavior compared with the poly-ubiquitin folding. The first stage looks similar, but in the second stage, the single ubiquitin length remains almost constant except some small vibrations. And then a fast collapse occurs. Based on the time the collapse lasts, it should correspond to Stage 4 in the poly-ubiquitin experiments. Therefore, the flat plateau in the single ubiquitin experiments should correspond to Stages 2 and 3 in the poly-ubiquitin experiments.

The force they claimed in the paper is 100 pN for high force and 26 pN for low force. But based on the force-time plot they give, the low force appears to be around 15 pN. They claimed that under their experimental conditions, the unfolded ubiquitin chain can be considered as a polymer coil that is placed in a poor solvent. In this condition, the coil-globule phase transition [**1**, **16**] satisfies a well known theory to describe the folding procedure.

6.14. Comparison with experimental results

One result from [**21**] is that during the folding under the high force of 100∼120 pN, the unfolding extension of one step is about 20 nm. For a fully unfolded ubiquitin chain subject to these high forces, most or all of the secondary structure of a protein will be unraveled [**7**]. Therefore, the freely-jointed-chain is a good approximation. For one amino acid with length 0.375 nm, the projected length in the force direction is about 0.334 nm. That is, $0.334/0.375 \approx 89\%$, the same as claimed in [**21**] (extension by 85-90% of its contour length). The total length of 76 amino acids should be $0.334 \times 76 = 25.4$ nm. A folded ubiquitin measures 3.8 nm its termini [**14**]). Then unfolding one ubiquitin domain makes the length change $25.4 - 3.8 = 21.6$ nm, close to the experimental results.

Now we look at the folding procedure. We begin with the single ubiquitin domain folding experiments, as shown in Fig. 6.24.

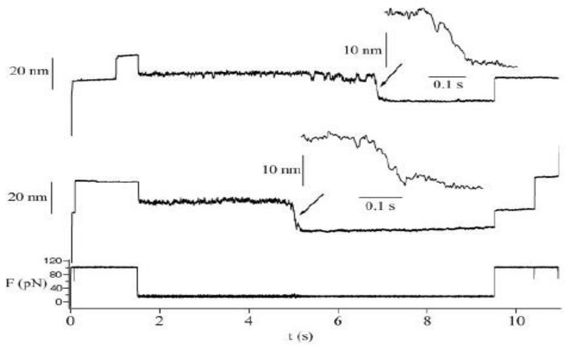


Figure 6.24. Single domain results from [**21**].

In Stage 1, the collapse of experimental results are about 12∼14 nm. In the simulation, shown in Fig. 6.25, a low force of 15 pN is used, based on the corresponding force-time plot shown as the lower plot in Fig. 6.24. If we still use the FJC model to calculate this length, the length of one amino acid becomes 0.152 nm. Then the total length should be $0.152 \times 75 = 11.6$nm. The change of length is $25.4 - 11.6 = 13.8$ nm, which agrees quite well with the experimental results. Figure 6.25 is the simulation result. In the simulation, after the force decreases, there is a collapse similar as experimental results. In the experimental results, Stage 1 is followed by an almost flat

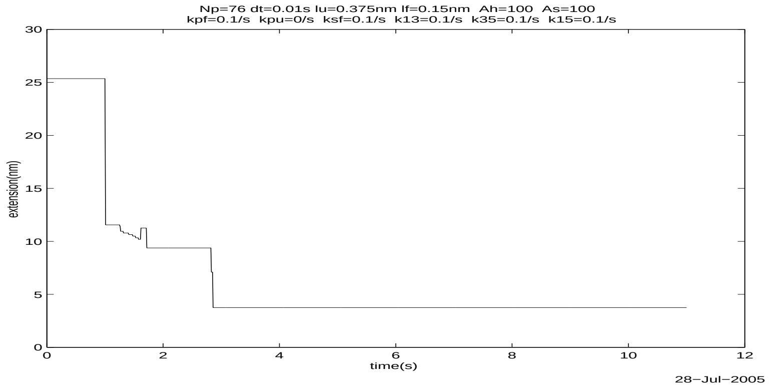


Figure 6.25. Simulation result. At t=1s, force drops from 100 pN to 15 pN.

region with some small fluctuations(Stages 2 and 3). The fluctuations become larger toward the end of the plateau. In the simulation there are also some small wiggles around t=1.5 s. These come from the helix structure folding. The only length increase at t=1.6 s also comes from the helix forming procedure, as explained in Sec. 6.7.1. Helix formation can be the explanation for the large fluctuations shown in the experiment. The experimental fluctuations seem bigger than simulation, but this is reasonable since experimental error can enlarge the vibrations. The time period of the flat plateau in experiments is around 5 s, which is longer than in the simulation. This difference can be overcame by a change in the parameters k_{pf}, k_{pu} and A_h.

There are three fast collapses. One immediately after the force decrease at t=1 s, and the last two are almost at the same time, t=2.8 s. The last collapse is due to tertiary folding and the first two come from the forming of two beta sheets. The last three collapses (two final collapses, plus one at $t = 1$ s) in simulation should become one big collapse, if we want to get the same results as experiments. This implies that once two β strands meet and form a β sheet, all the other β strands will join this β sheet quickly, and the tertiary folding will happen spontaneously. We can simulate this by making some changes of the simulation procedure. For example, once two β strands meet and form a β sheet, all the folding rates for β strands folding and tertiary fold will become A_h times larger.

The ending length of 3.75 nm, agrees well with the measured folded length.

In the experimental results, the last collapse is about 16∼18 nm. But in the simulation results, this collapse is approximately $11.8 - 3.8 = 8$ nm. The experimental results seem wrong. In unfolding procedure, the change in length is about 20 nm per ubiquitin, while in folding it becomes around 30 nm. This seems unreasonable. For a chain with 76 amino acids, the longest end-to-end length is about $76 \times 0.375 = 28$ nm. Even if changed from a straight line into a point, the total length change is only 28 nm.

6.15. Modification of the model

Based on the discussion in the previous sections, the model is modified as follows. Helix structure is the first secondary structure to form. This is because in the flat plateau (Stages 2 and 3) of single ubiquitin folding, only fluctuations are observed, which is the character of helix folding. If beta sheet forming or tertiary structure folding is allowed, one or several collapses should be observed. To make the time of the flat plateau similar to the experimental results, values for k_{pf} and A_h are changed. The unfolding rate k_{fu} is sill not introduced here.

After the single α helix is 100% finished, the beta sheets are allowed to form. Once any two beta strands meet together and form a beta sheet structure, the folding rate constant k_{bf} will become A_b=1000 times larger than the original value. Because the final collapse in the experimental results only includes one step, once one collapse happens, all the following ones should happen in a short time.

After the beta sheets are fully formed, the tertiary structure folding is allowed. The rate constants for tertiary folding are set to large values to make the collapse happen in a short time.

Figure 6.26 is the simulation result. Compared with the experimental results, the fluctuations still seem too small.

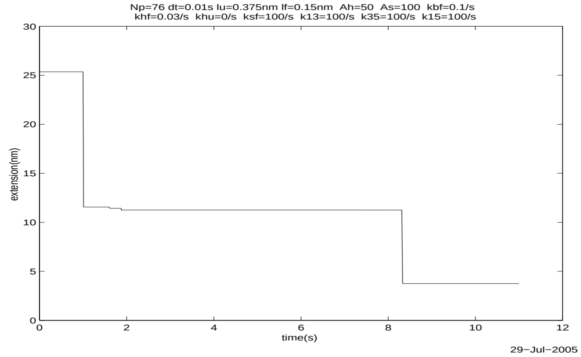


Figure 6.26. Simulation result. At t=1 s, force drops from 100 pN to 15 pN.

6.16. Random number determination of the projected length of a freely rotating bar in external force

In the model we regard the loop region as a freely-jointed-chain, which is composed of many rigid bars(amino acids) with length 0.375 nm. Each bar can rotate independently. Under a given external force, we can calculate the average length projected in the force direction. In the helix region and β sheet region, we still apply the FJC model to calculate the projected length in the force direction, but the length of one bar might be different. So far we only used the averaged length. Since the system is in a thermal environment, the length of all the bars should be changing all the time. Therefore, it is necessary to randomize the length of the bars at any time. This will introduce the variance that we threw out by using the average length.

6.16.1. Distribution function

Let θ be the angle between one bar's direction and the force direction. We need to determine the distribution function $P(\theta)$. Because of thermal fluctuation, all the bars can jump from one direction into any other direction. If there is no external force, all the directions are equally distributed. The probability of observing one bar in the direction between θ and $\theta + \Delta\theta$ is proportional to the area of the region on the unit sphere: $P(\theta) \cdot \Delta\theta \propto 2\pi \sin\theta \cdot \Delta\theta$.

Under the effect of an external force, all the bars tend to align in the force direction. The weighting factor due to the external force is $e^{Flcos\theta/k_BT}$. Now the probability of observing one bar in the region $\theta \sim \theta + \Delta\theta$ becomes: $P(\theta) \cdot \Delta\theta \propto 2\pi \sin\theta e^{Fl\cos\theta/k_BT} \cdot \Delta\theta$, where F is the external force, l is the length of one bar, k_B is the Boltzmann constant and T is temperature.

We still need the integration of $P(\theta)$ to be equal to unity, $\int_0^\pi P(\theta) = 1$. We can now get the distribution function,

$$P(\theta) = \frac{Fl}{k_BT} \cdot \frac{\sin\theta e^{Fl\cos\theta/k_BT}}{e^{Fl/k_BT} - e^{-Fl/k_BT}} \tag{6.3}$$

6.16.2. Using a random number to determine θ

Note that the random number produced by Matlab is an equally distributed random number in the region $[0, 1]$. How do we use this equally distributed number R to generate the unequally distributed number θ? We use Fig. 6.27 to explain the method.

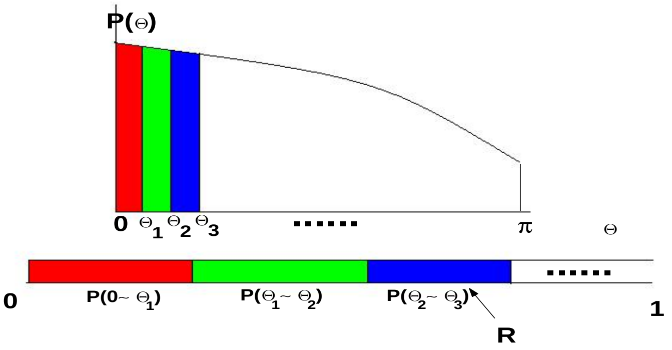


Figure 6.27. Using the equally distributed number R to generate the unequally distributed number θ.

We have a distribution $P(\theta)$ in the region $[0, \pi]$, as shown in the upper plot. We divide $[0, \pi]$ into many small regions: $[0, \theta_1]$, $[\theta_1, \theta_2]$, $[\theta_2, \theta_3]$ $\cdots$. Based on the distribution function, we can calculate the probabilities that the system is in each region:$P(0 \sim \theta_2)$, $P(\theta_1 \sim \theta_2)$, $P(\theta_2 \sim \theta_3)$ $\cdots$. Then we connect all the regions as in the lower plot, the length of each region being equal to the probability of that region. The total length of all the regions is equal to 1. Then we generate a random number R in the region of [0,1] and see which region the random number R falls into. The corresponding θ will be the angle we hope to find. For example, in the plot R falls into the third region, so we choose θ_3 as the angle we want.

Mathematically, if we generate a random number R, then the corresponding angle θ_R should satisfy the distribution function,

$$\begin{aligned} R &= \int_0^{\theta_R} P(\theta) d\theta \\ &= \frac{\int_0^{\theta_R} \sin\theta e^{Fl\cos\theta/k_BT} d\theta}{e^{Fl/k_BT} - e^{-Fl/k_BT}} \\ &= \frac{e^{Fl/k_BT} - e^{-Fl\cos\theta_R/k_BT}}{e^{Fl/k_BT} - e^{-Fl/k_BT}} \end{aligned} \tag{6.4}$$

i.e.

$$\cos(\theta_R) = \frac{k_BT}{Fl} \log\left[e^{Fl/k_BT} - R\cdot(e^{Fl/k_BT} - e^{-Fl/k_BT})\right] \tag{6.5}$$

Once we know the value of $\cos\theta$, we can calculate the length projected in the force direction as $l\cdot\cos\theta$. For a system composed of N units, we just repeat this procedure N times and sum all the lengths to get the total length.

6.16.3. Numerical simulation

Figure 6.28 is a comparison of the analytical expression and the numerical simulation result. The solid line is for $P(\theta)\cdot d\theta$, the bars are the percentage of events in every $\Delta\theta$ region. Here $\Delta\theta$ of each small region is $\pi/30$, force is 15 pN, and length is 0.375 nm. The numerical simulation repeats $100,000$ times. The numerical results fit well to the analytical expression.

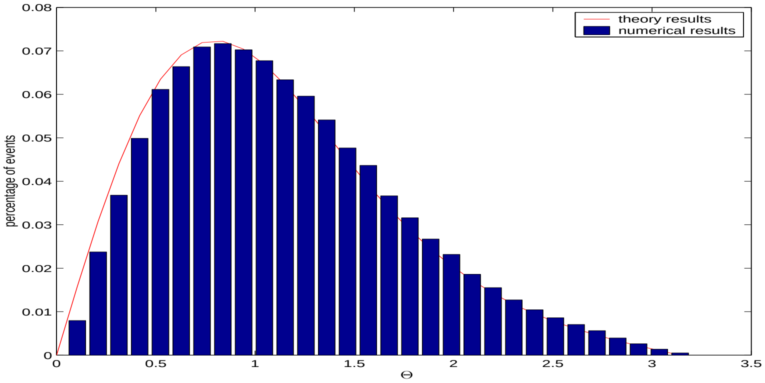


Figure 6.28. Comparison of the theory result and the numerical simulation result.

6.16.4. Variance

Knowing the distribution function, we can also calculate both the average length and the variance of one bar. For the average length,

$$\begin{aligned}
\langle l\rangle &= \int_0^\pi l\cdot\cos\theta\cdot P(\theta)d\theta \\
&= \frac{Fl}{k_BT}\frac{\int_0^\pi l\cdot\cos\theta\cdot\sin\theta e^{Fl\cos\theta/k_BT}d\theta}{e^{Fl/k_BT}-e^{-Fl/k_BT}} \\
&= b\cdot l\int_0^\pi \cos\theta\cdot\sin\theta\cdot e^{a\cos\theta}\,d\theta \\
&= l\left(\frac{1}{\tanh a}-\frac{1}{a}\right) \qquad (6.6)
\end{aligned}$$

Where $a = \frac{Fl}{k_BT}$, $b = a/(e^{Fl/k_BT} - e^{-Fl/k_BT})$. The variance is

$$\begin{aligned} V &= \langle (l - \langle l \rangle)^2 \rangle \\ &= \langle l^2 \rangle - \langle l \rangle^2 \\ &= \frac{Fl}{k_BT} \frac{\int_0^\pi l^2 \cdot \cos^2\theta \cdot \sin\theta e^{Fl\cos\theta/k_BT} d\theta}{e^{Fl/k_BT} - e^{-Fl/k_BT}} \\ &= b \cdot l^2 \int_0^\pi \cos^2\theta \cdot sin\theta \cdot e^{a\cos\theta}\, d\theta \\ &= \frac{l^2}{a^2}(a^2 + 1 - a^2 \frac{1}{\tanh^2 a}) \end{aligned}$$

For $l = 0.375$ nm, $T = 300$ K, and $f = 15$ pN, the calculated variance is 0.0336 nm^2. The numerical simulation results is 0.0339 nm^2. The numerical result is based on a data set of 10,000 elements. The theory and numerical average length are 0.1520 nm and 0.1524 nm, correspondingly.

Figure 6.29 is the numerical simulated results for a freely rotating bar with length 0.375 nm in external force 15 pN. Repeating the simulation 100 times, each time we can get a different projected length. Here the numerical variance is 0.038 nm^2. Figure 6.30 is the simulation result

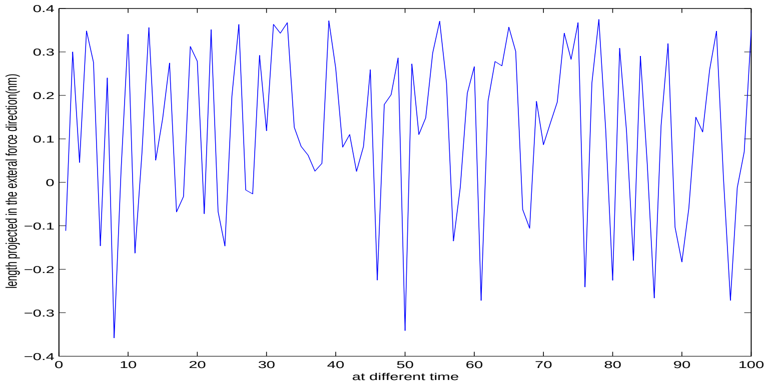


Figure 6.29. Projected length in force direction at different time.

of the folding procedure using the random numbers to determine the projected length. We can see that the variance seems too large this time.

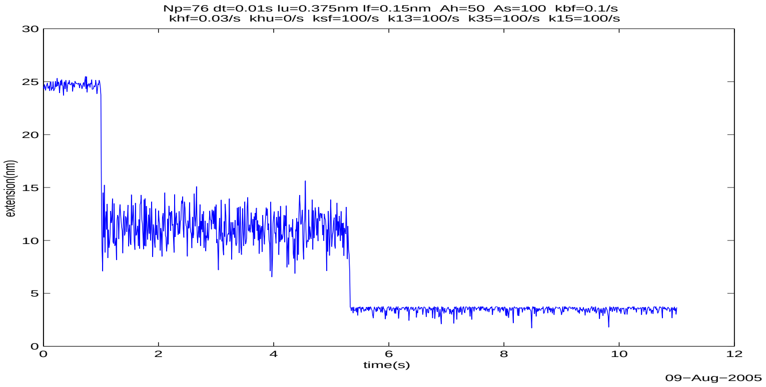


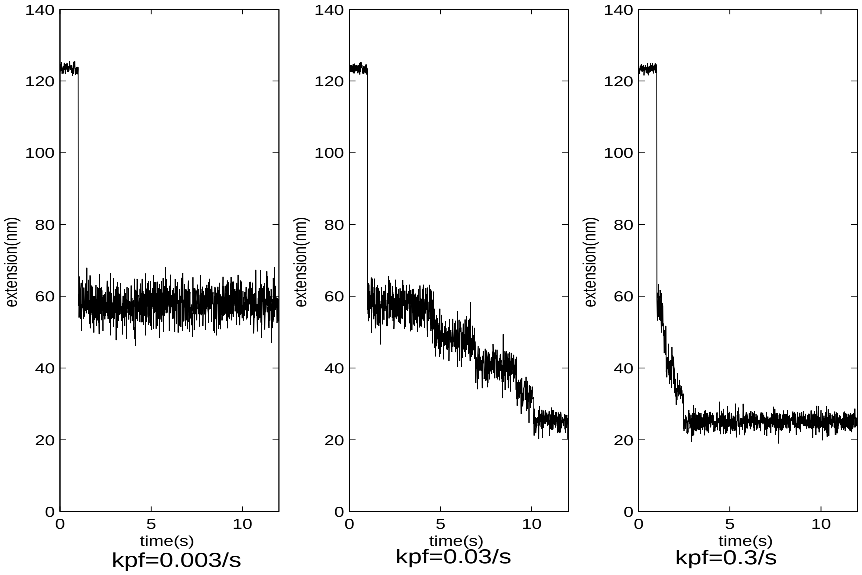
Figure 6.30. Numerical simulation of the folding procedure.

6.16.5. How sensitive are the simulation results to the parameters

The parameters k_{pf}, k_{bf}, k_{sf}, A_h, A_b and A_s affect the folding time. Larger values can make the folding procedure faster. Figures 6.31-6.36 show the effects of these parameters. Note that even for the exact same parameters, the folding times are not the same if we repeat the simulations several times. Therefore, the folding time shown in the plots are just the value from that particular realization. We use them to show the trends on varying the parameters.

The folding process have several steps. The total folding time is mainly determined by the slowest step. For the parameters used, the slowest step is α helices formation with the slowest rate 0.03/ s. Increasing this rate can decrease the total folding times dramatically, as shown in Fig. 6.31.

The folding time is not sensitive to A_h and A_s, because in the simulation these two parameters are related to the fast steps. Accelerating the steps that are already fast can not much decrease the time for the whole folding process.

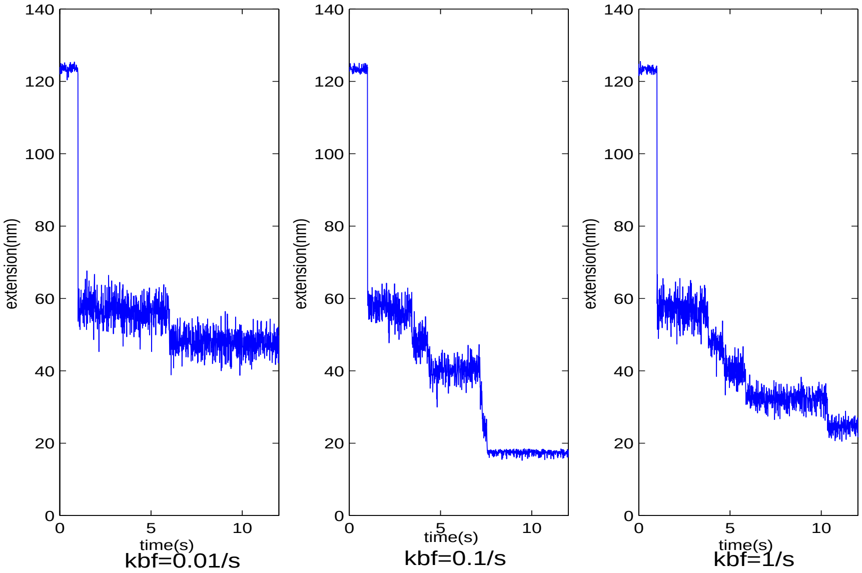


kpf=0.003/s

kpf=0.03/s

kpf=0.3/s

Figure 6.31. Effects of different k_{pf} values. Larger value can make the folding faster.



kbf=0.01/s

kbf=0.1/s

kbf=1/s

Figure 6.32. Effects of different k_{bf} values. Larger value can make the folding faster.

6.17. Multi-domain simulation results and comparison with experimental results

Figure 6.37 shows the simulation results for multi-domain folding procedure. More simulation results are shown in Appendix 13.5. For multi-domain simulation, we just simulate several single

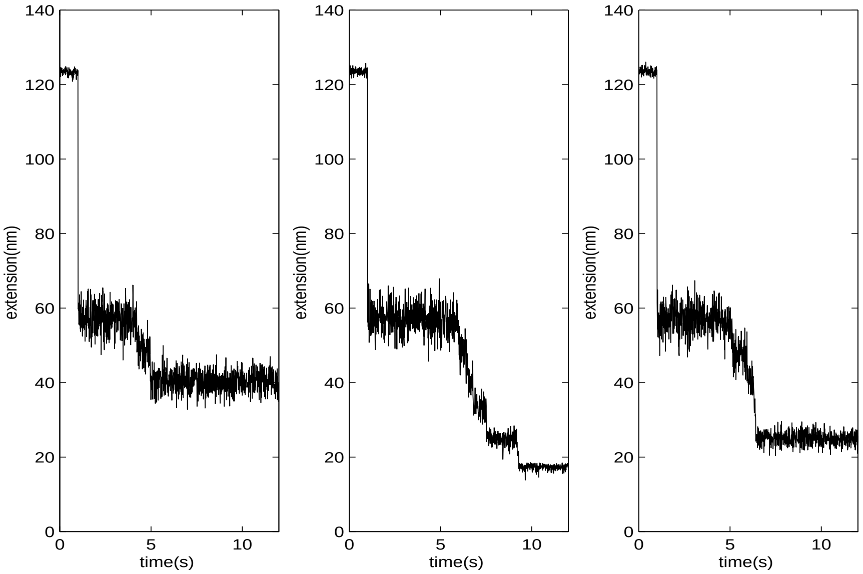


Figure 6.33. Effects of different k_{sf} values. The plots from left to right are corresponding to k_{sf}=100 /s, 1000 /s, and 10000 /s.

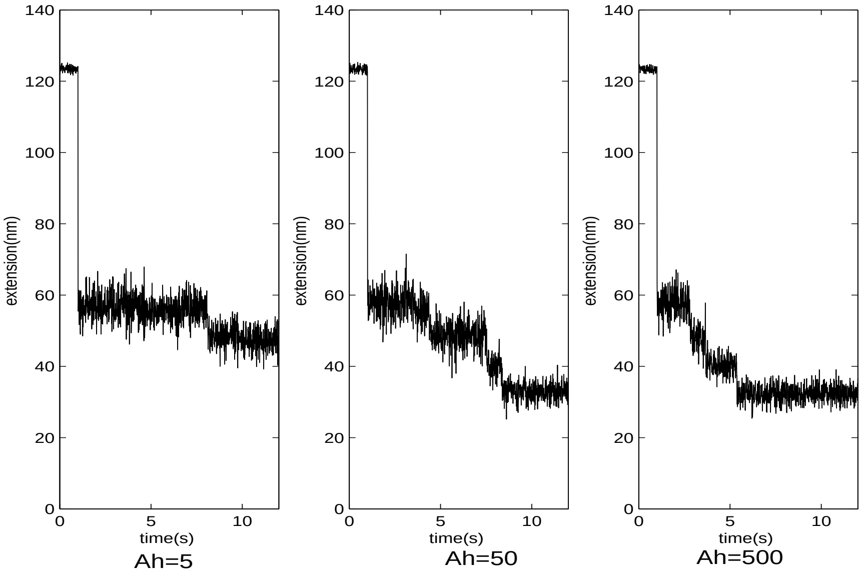


Figure 6.34. Effects of different A_h values. Larger value can make the folding faster.

domains simultaneously and use the sum of lengths of all the domains as the total length. Figures 6.39 and 6.40 are the experimental results. Figure 6.39 divides the folding history into four stages. Stage 1 is a quick collapse just after the decrease of the external force. Stage 2 is a slow decrease

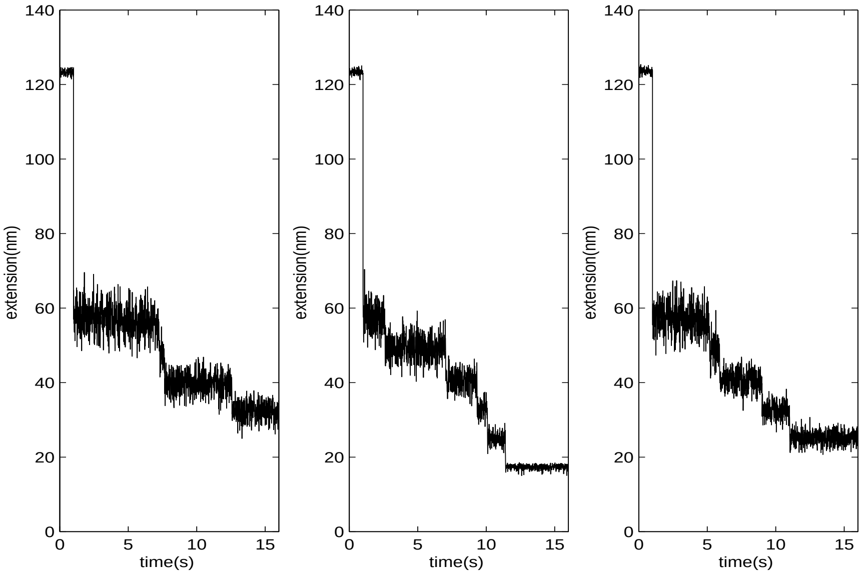


Figure 6.35. Effects of different A_b values. The folding time is not sensitive to the A_b values.

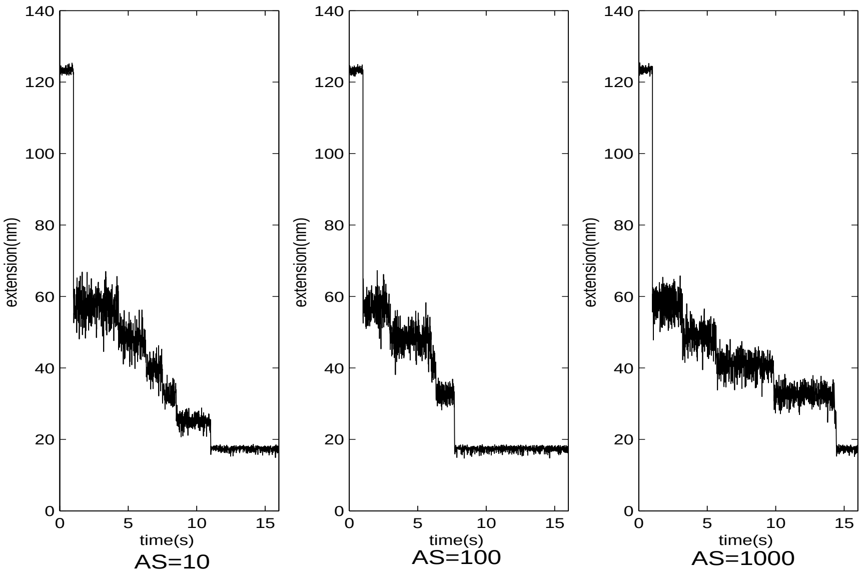


Figure 6.36. Effects of different A_s values. The folding time is not sensitive to the A_s values.

of length at a nearly constant rate. Stage 3 has a steeper decreasing slope. Stage 4 is another quick collapse. The simulation results reproduce the quick collapse of Stage 1. Then we can see a slow decrease, with no differentiation into stages. Figure 6.39 is only one of the many experimental results. Figure 6.40 shows more results. These results show quite different folding procedures. These procedures are not regular enough to divide into stages as in Fig. 6.39. It is hard to give a quantitative comparison. These experimental results have two characteristics in common. First, after decreasing the external force, they all have a quick collapse. This shows that our assumption, that the stretched protein can be described as a coil-like structure, seems reasonable. The second characteristic is that all the folding happens on a time scale of several seconds. These two characteristics are the same as our numerical results.

As noted, repeated simulations will yield different results. Figure 6.38 is the simulation repeated with the same parameters as in Fig. 6.37. More results can be seen in Appendix B. The results from this simulation have an extension history which is divided into the same four stages as found in the experimental results reproduced in Fig. 6.39. Following the decrease in force, there is an abrupt decrease in length as in Stage 1. The total decrease during this Stage is about 70 nm compared with 40-70 nm in the experiment. A relatively flat portion follows in the simulation similar to Stage 2, but lacking the slight decrease in length. In the experiment, this stage last 2.5-8 s, in the simulation it lasts about 4 s. The total decrease amounts to 20 nm, while in the simulation, there is 0 nm decrease. There is then a decrease like Stage 3 in which the rate of decrease is about 10 nm/sec compared with a rate of 43 nm/sec in the experimental results. ("Stage 2 and 3 can vary greatly in their rates of collapse and cannot always be distinguished." [**21**]) This stage lasts 1.5 s in the simulation and 0.5-1 s in the experimental results. Finally, the simulation shows Stage 4, an abrupt decrease in length of 20-30 nm compared with 16 nm in the simulation.

In the simulation there is no length change in Stage 2. This is because in this stage, the length change only comes from α helix formation, which gives almost zero change in extension length. β sheet formation is regarded as a one-step procedure, and immediately followed by the tertiary formation. These two formations are in the collapse region. There is no slow length change in the model. The detailed procedure of β sheets formation might help to improve this.

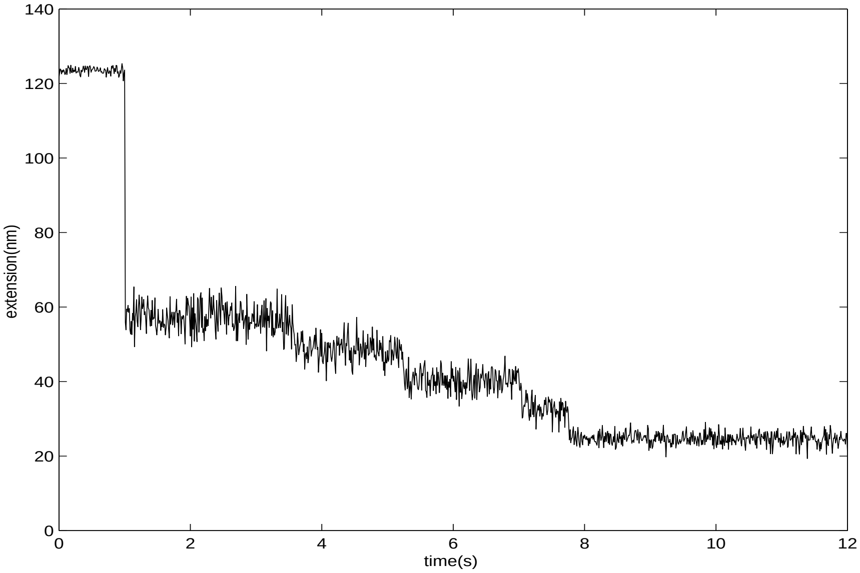


Figure 6.37. Simulation results for multi-domain folding procedure. k_{pf}=0.03/s, k_{bf}=0.1, k_{pu}=0/s, k_{bu}=0 /s, k_{su}=0 /s, A_h=50,A_b=1000, A_s=100.

Fig. 6.41 shows additional details of the folding procedure for Fig. 6.38. The top plot reproduces the folding curve for the five domains. The second to the sixth plot show the folding procedure for the individual domains, from one to five respectively. The slope of stage 2 comes from domains three and four. Domain three folds at a time of approximately 6 s, and domain four folds approximately 5 s, also see Fig. 6.42. We find that adjacent steps plus the noise induced by small-scale fluctuations takes on the appearance of Stage 3. The sum of them yield the decrease in length observed during the time from 5 s to 6 s in Fig. 6.38. Domain one and five fold at almost the same time, at approximately 6.5 s. The nearly simultaneous collapse of these two domains brings about the collapse.

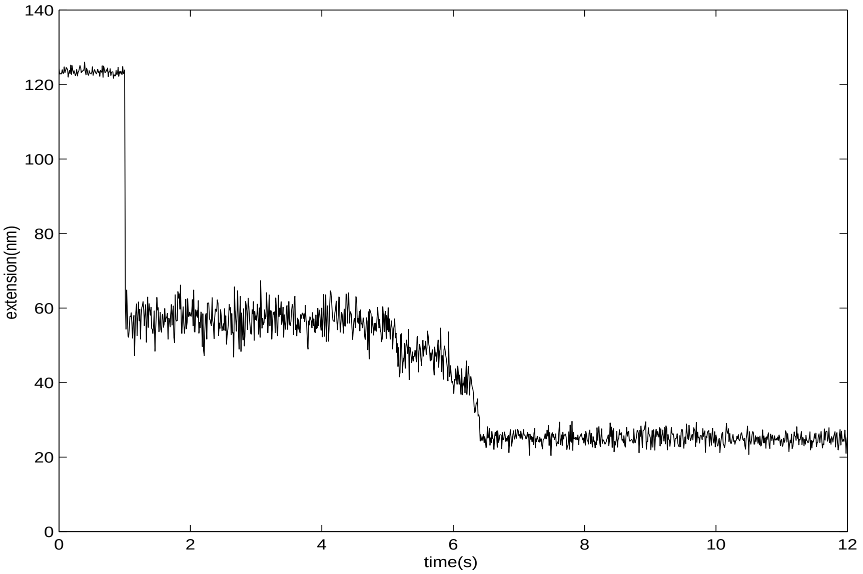


Figure 6.38. Simulation results for multi-domain folding procedure. This one looks like the experimental results of [**21**]. k_{pf}=0.03/s, k_{bf}=0.1, k_{pu}=0/s, k_{bu}=0 /s, k_{su}=0 /s, A_h=50,A_b=1000, A_s=100. This is the only one that happens to agree with experimental results among over 20 simulations. In Sec. 6.18 we show that once we introduce interactions between domains, this pathway will become typical.

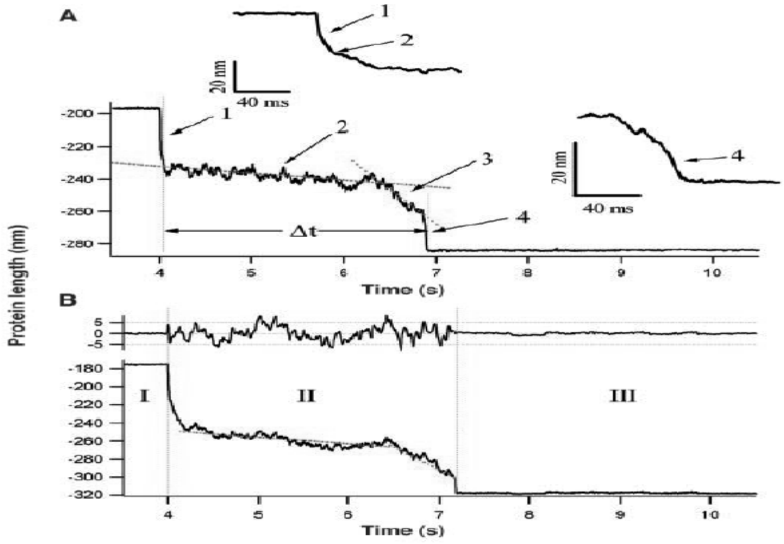


Figure 6.39. Experimental results for constant force folding procedure. From [**21**].

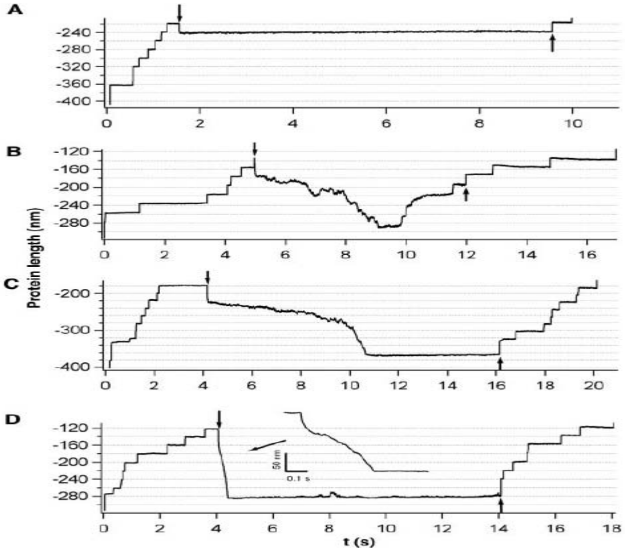


Figure 6.40. Experimental results for constant force folding procedure. From [**21**].

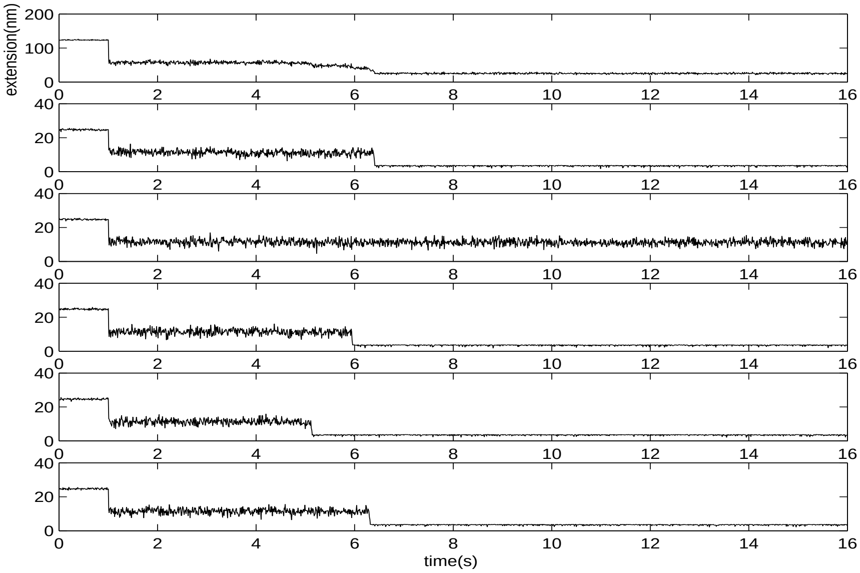


Figure 6.41. Details of the folding procedure for Fig. 6.38. The top plot is the folding curve for the five domains. The other five plots show the folding procedure for each of the five domains individually, from 1 (2nd plot) to 5 (6th plot).

Fernandez and Li repeated their multi-domain folding experiments over eighty times. They report that “most of the folding trajectories are qualitatively similar, following a continuous

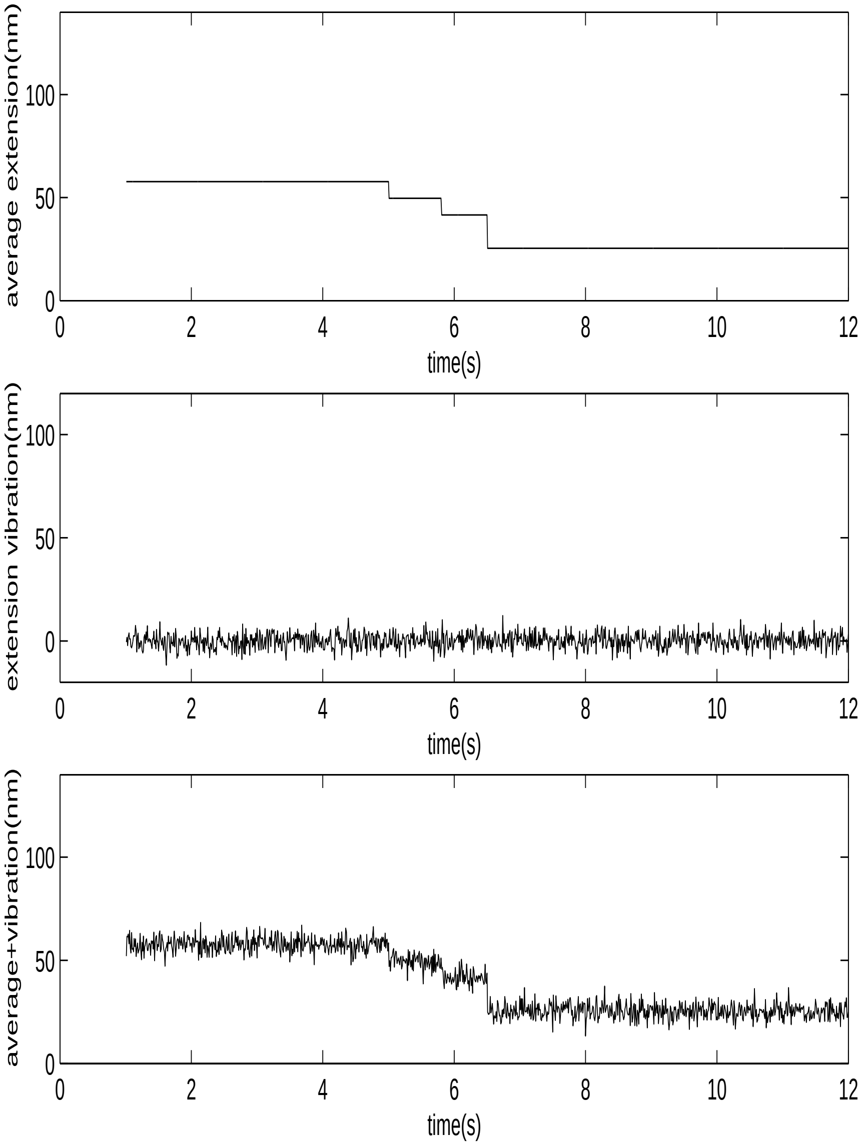


Figure 6.42. These three plots show how the simulation results in a folding process which shows an identifiable Stage 3 and Stage 4. The top plot is the average extension, showing step-like behavior: There is one small collapse at t≈5 s, another small collapse at t≈5.8 s, and one large collapse at t≈6.5 s. The middle plot shows the thermally-induced variation in length. The bottom plot is the sum of the first two plots. When the noise is superimposed on the step-like decreases, the extension is smoothed, as in Stage 3. The final collapse is so abrupt that there is little time for the fluctuations to do much smoothing , and the decrease remains quite sharp.

convex time course marked by abrupt changes in slope" [**21**]. But this experimental-like result appeared only once out of 20 times in my simulations. How can we get the experimental-like results more frequently? Note that for one single domain, the folding is always step-like. Increasing the number of domains introduces a continuous decrease. What will happen if we further increase the number of domains? Fig. 6.43 shows the simulation result for twenty domains. There is a continuous decrease, but the final collapse disappears. This is because the twenty domains are independent, they can fold at any time (still in the time scale of several seconds). Increasing the number of domains makes them folding at different time, therefore the folding curve looks continuous. But it becomes less likely that many of them fold at the same time, so the final collapse disappears. For example, if there are four domains, two domains folding at the same time is enough to appear as a collapse-like folding. If there are twenty domains, two domains folding at the same time can not be regard as a collapse any more. To observe the same size of collapse (relative to the total length change of all the domains), now ten domains need to fold at the same time, which is much less likely.

What kind of change can make simulation results more likely to reproduce the experimental results? If we look at Fig. 6.42 again, we can compare the folding of the domains with the four stages shown in Fig. 6.39. Stage 1 only comes from coil relaxation. Stage 2 is the waiting time for any domain to finish the folding procedure. One domain folded at time 5 s, after waiting for about 4 s. Then the next folding finished after 0.8 s. Then after 0.7 s, two folding finished at the same time, which brings the final collapse. This implies what kind of folding procedure can produce the experimental-like behavior. Waiting time for the first folding to finish, gave us Stage 2. Once one domain finished folding, the next few ones should happen in a short time, to give the steep slope. Then, the domains remaining should fold at almost the same time, to give the final collapse. This whole procedure is like a avalanche!

The multi-domain folding procedure is not the sum of five independent single domains, there are cooperations between the domains. This seems to give the cause of the multi-domain folding behavior and why it is different from the step-like single domain folding procedure.

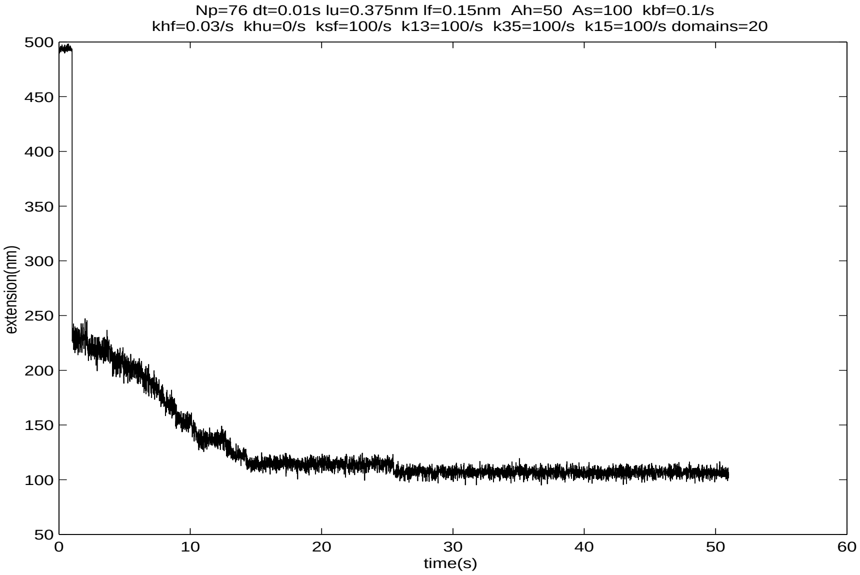


Figure 6.43. Typical simulation folding procedure with twenty domains. We can see that after the collapse due to coil relaxation, there is one continuous decrease. The final collapse has disappeared.

6.18. Introducing interactions between domains

In Fernandez and Li's experimental results, Stages 2-4 show a tendency of folding faster and faster: Stage 2 is a slow decrease, Stage 3 is a steeper slope, and Stage 4 is a collapse. This implies the existence of cooperation: once some domains finish folding, the rest of the domains will fold with a higher rate. To simulate this effect, we introduce a parameter A_t: for every completely folded domain, all of the unfinished folding procedures will speed up A_t times. The effect of A_t is, once any domain finishes its folding, all the folding rates of the other domains will become A_t times faster.

Figs. 6.44-6.49 show the simulation results. The parameters used are: k_{pf}=0.03/s, k_{bf}=0.1, k_{pu}=0/s, k_{bu}=0 /s, k_{su}=0 /s, A_h=50,A_s=1000, and A_s=100. The number of domains and A_t are given in the captions.

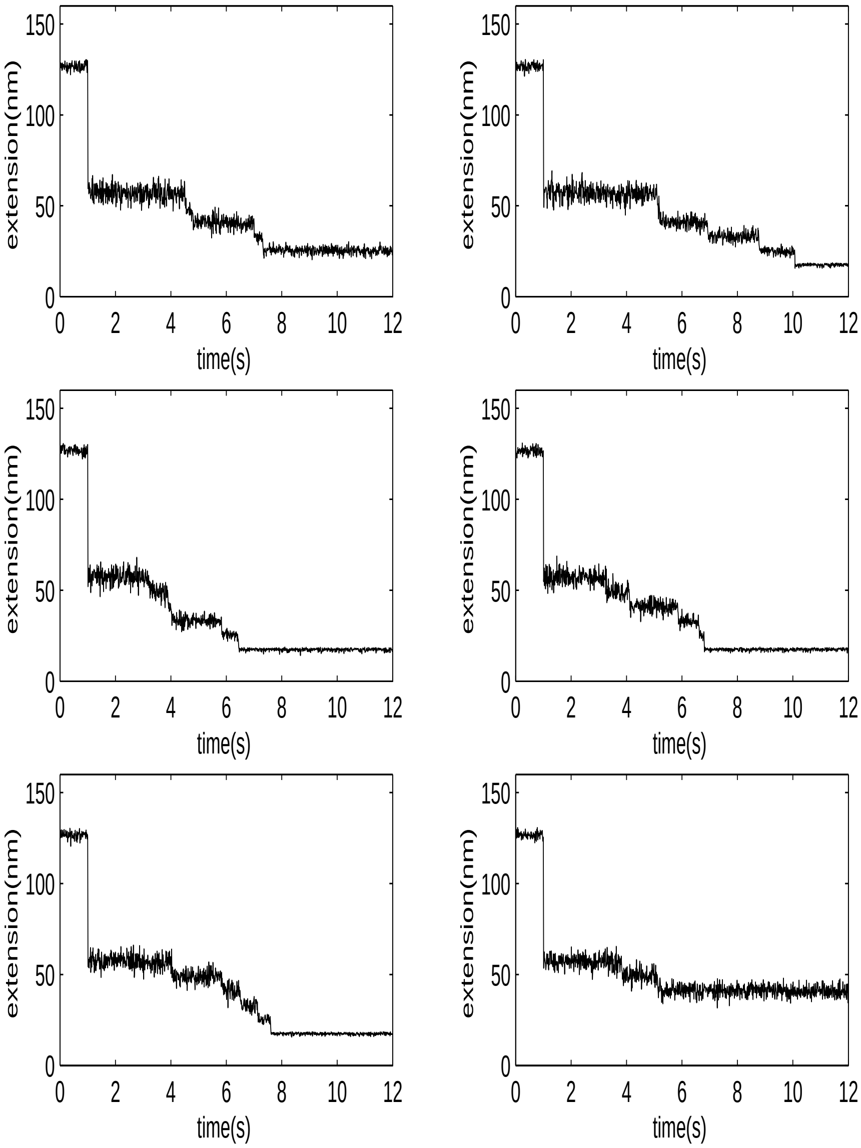


Figure 6.44. There are 5 domains. $A_t = 1.2$.

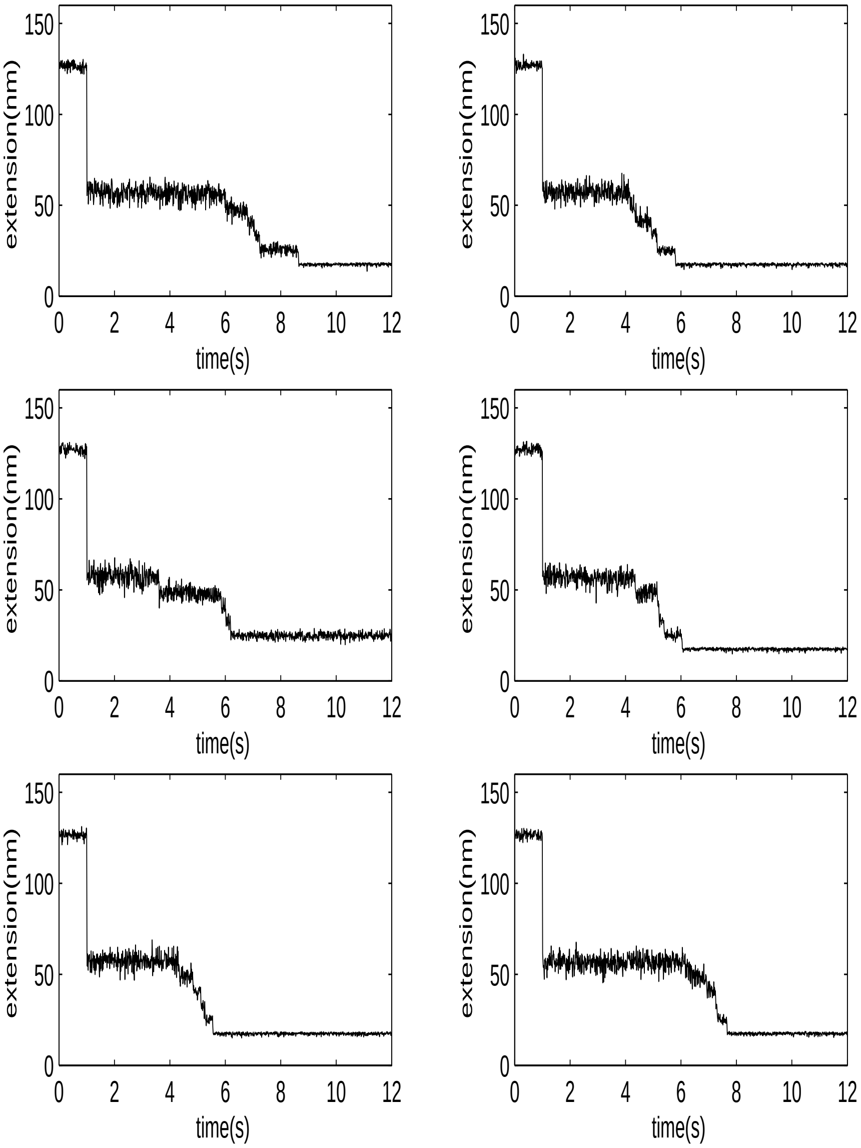


Figure 6.45. There are 5 domains. $A_t = 1.5$.

We first look at the folding curves of 5 domains. When $A_t = 1.2$, the effects of the interactions are not clear, the domains still appear as if they are folding independently. When $A_t = 1.5$, we can see the tendency to accelerate folding, but there is still no clear final collapse. When $A_t = 2$, the final collapse appears more frequently. When $A_t = 3$, the final collapse always shows up.

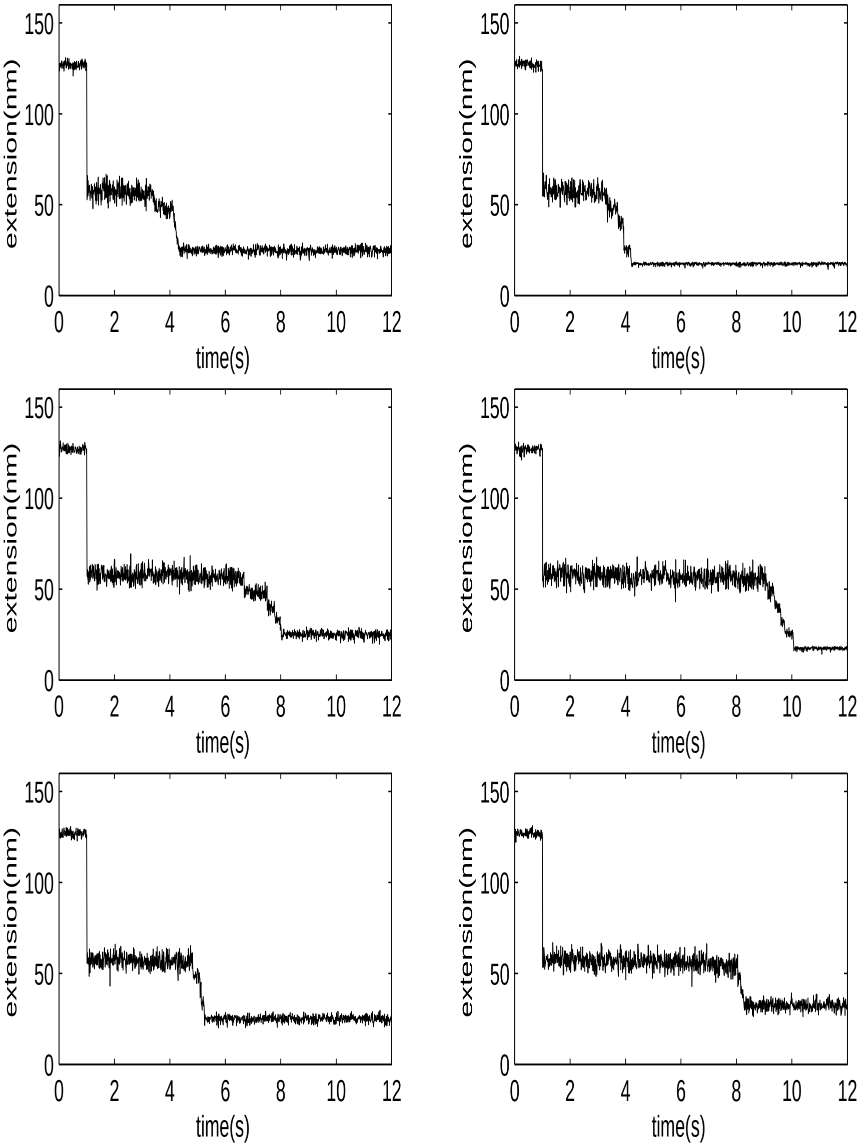


Figure 6.46. There are 5 domains. $A_t = 2$.

Now we look at the folding curves of 9 domains. When $A_t = 1.2$, the tendency for accelerated folding is already apparent. When $A_t = 1.5$, the final collapse is clear. The interaction factor A_t seems have more effect on 9-domain folding procedure than 5-domain folding procedure. This is because for the 5-domain folding procedure, the last domain folds with a rate A_t^4 times higher. For the 9-domain folding procedure, the last domain folds with a rate A_t^8 times higher. A value

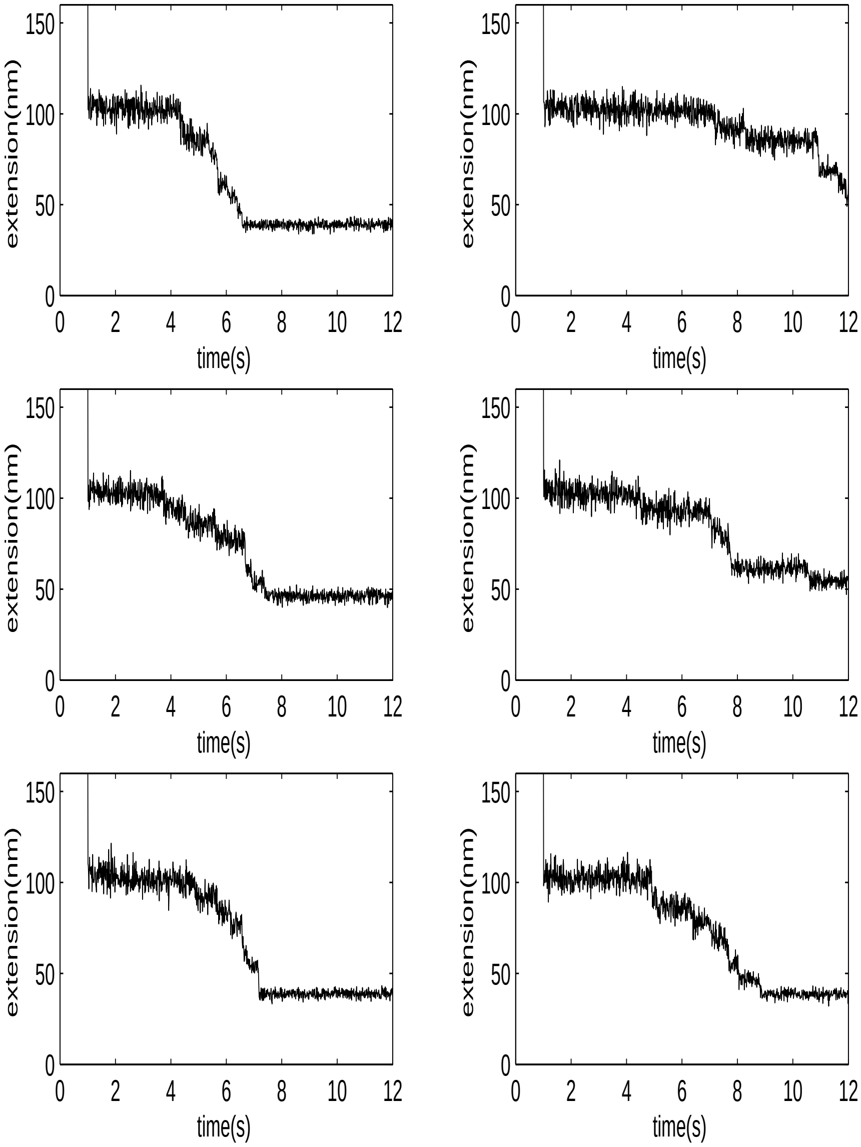


Figure 6.47. There are 9 domains. $A_t = 1.2$.

of A_t only slightly greater than unity will more greatly accelerate folding when acting on a large number of domains.

The simulations in this section show that the introduction of an interaction factor A_t can improve the simulation results by making Stages 3 and 4 appear more frequently, as in the experimental results. But we still do not know the dynamics of the interactions. Why might there

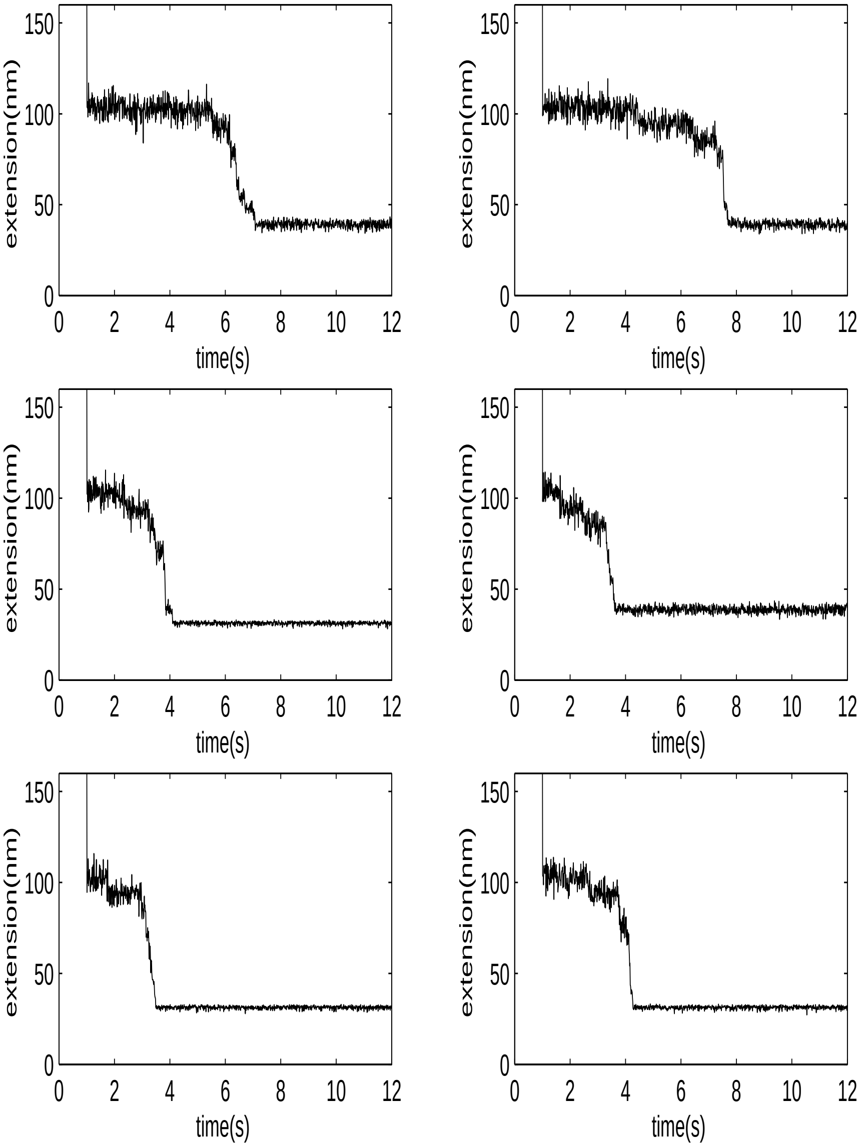


Figure 6.48. There are 9 domains. $A_t = 1.5$.

be interactions between domains? How do they effect the folding procedure? In the simulation, I assume the interaction appears when some domains completely fold: Is this necessary? Does the value of A_t depend on the number of total domains? Or does it depend on the number of folded domains? Or does it depends on the percentage of folded domains? There are still many puzzles in this interaction.

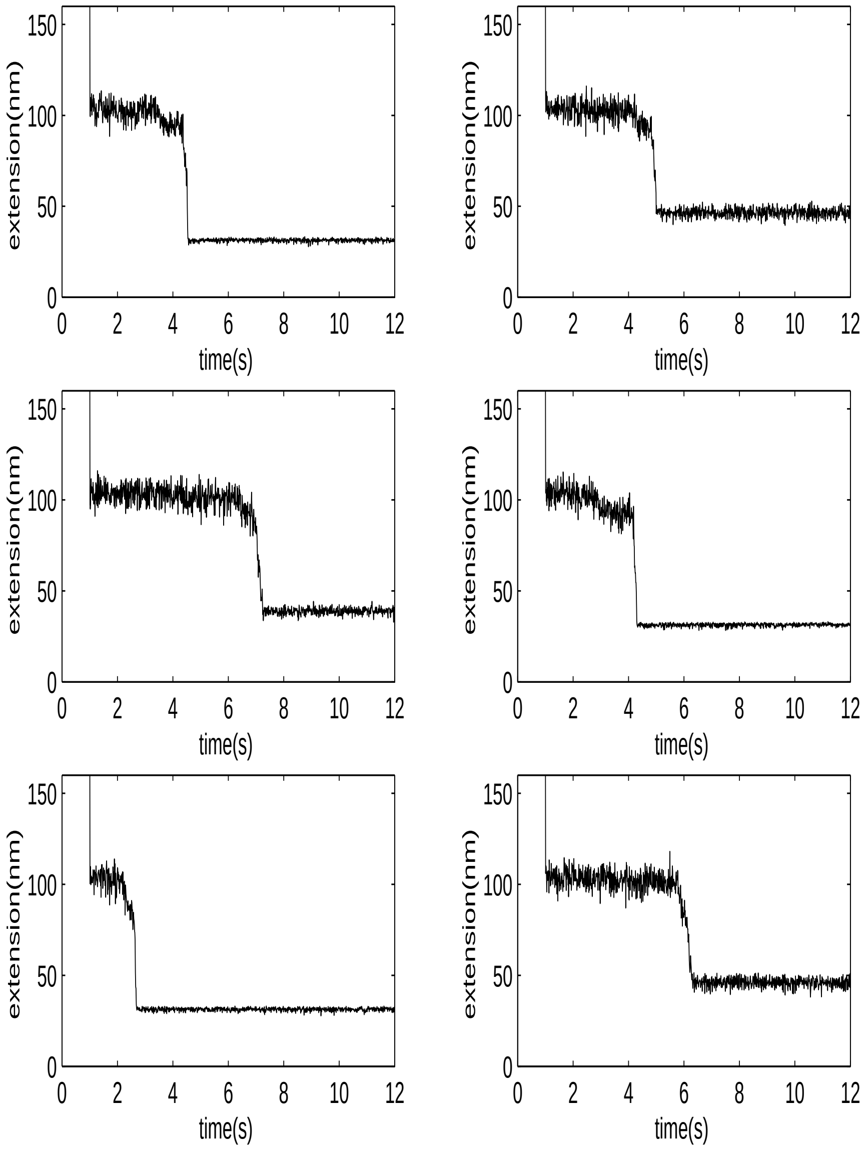


Figure 6.49. There are 9 domains. $A_t = 2$.

6.19. Discussion about the folding rates.

Here we want to see how the folding rate should change when there is an external force.

In Fernandez and Li's paper, the folding time Δt is fit as a function of force using $\Delta t \approx 0.01 * \exp(F \times 0.8/k_BT)$ with $\Delta x = 0.8$ nm. We want to look at this length scale for different forces. The folding rate for ubiquitin without any force is about $365/s$ [**27**], which means the folding time is about 0.003s. For Fernandez and Li's results, the reported folding time in the range 10-20 pN is about 0.2 s. Using the average of 15 pN, we see that the rate is slowed down about 70 times under 15pN. This can be fit to the expression above by using $\Delta x \approx 1.1$ nm. Increasing the range of forces about 10 pN further (increasing force from 10-20 pN to 20-30 pN) causes the folding time to become 10 times slower. This corresponds to $\Delta x \approx 0.9$ nm. Further increasing the force from 20-30 pN to 30-40 pN causes the folding time to become 2 times slower. This corresponds to $\Delta x \approx 0.28$ nm. It seems that when force increases the characteristic length Δx will decrease. Similar behavior is observed in the persistence length for the worm-like-chain model [**43**]: in the low-force region the persistence length is 0.8 nm while in the high-force region the persistence length decreases to 0.4 nm.

What is the meaning of the Δx? As will be shown in the next section, it is not the end-to-end length of the whole protein. If we consider the folding procedure as many small simultaneous diffusions, this Δx should be the length of one small diffusion procedure. Initially in the folding process, for small external forces, long range interactions are allowed. Amino acids could diffuse a long distant to find the correct bonds to form. Therefore Δx is large. When force increases, since long distance interactions are very sensitive to external force, they are eliminated very quickly. Only short range interactions are left, so Δx decreases. One interesting observation is that for 35 pN, the average projected length on the force direction of one amino acid is about 0.27 nm, which is close to the $\Delta x \approx 0.28$ nm for this force.

sequence	structure
1-7	β strand
8-9	loop
10-17	β strand
18-22	loop
23-34	α helix
35-39	loop
40-45	β strand
46-47	loop
48-50	β strand
51-55	loop
56-59	α helix
60-63	loop
64-73	β strand
74-76	loop

Table 6.1. The secondary structures of ubiquitin.

Parameters	Meaning	Values
dt	time step	0.01 s
monomer length	length of one amino acid	0.375 nm
monomer length in helix state	monomer length in helix state	0.15 nm
k_{pf}	rate of primary structure form α helices	0.03/s
k_{pu}	rate of formed α helices unfold again	0/s
k_{bf}	rate of primary structure form β sheet	0.1/s
k_{bu}	rate of formed α helices unfold again	0/s
k_{sf}	rate of secondary structures to form tertiary structure	100/s
A_h	once one helix turn appear, the times its neighbours's folding rate can accelerate	50
A_b	once one β sheet pair forms, the times its neighbours's folding rate can accelerate	1000
A_s	once one tertiary structure forms, the times of folding rate of the rest can accelerate	1000

Table 6.2. Parameters used in the simulation.

Figure	High force (unfold)	Low force (fold)	Length change (stage 1)	Length change (stage 2)	Length change (stage 3)	Length change (stage 4)
Fig. 1A	122 pN	15 pN	30 nm	10 nm	15 nm	20 nm
Fig. 2A	120 pN	15 pN	36 nm	12 nm	18 nm	24 nm
Fig. 2B	120 pN	15 pN	75 nm	19 nm	38 nm	19 nm
Fig. 3A	100 ~ 120 pN	50 pN	20 nm	0		
Fig. 3B	100 ~ 120 pN	35 pN	20 nm	115 nm		
Fig. 3C	100 ~ 120 pN	35 pN	40 nm	75 nm		70 nm
Fig. 3D	100 ~ 120 pN	23 pN	60 nm	105 nm		

Table 6.3. The results from [**21**].This table gives out the estimated length changes for the stages.

Figure	Time (stage 1)	Time (stage 2)	Time (stage 3)	Time (stage 4)
Fig. 1A	0.01s	7s	1.5s	N.A.
Fig. 2A	0.02s	2.5s	0.4s	0.04s
Fig. 2B	N.A.	2.8s	0.8s	N.A.
Fig. 3A	N.A.	8s		
Fig. 3B	N.A.	4s		
Fig. 3C	N.A.	6s		0.7s
Fig. 3D	0.1s	0.2s		

Table 6.4. The results from [**21**]. Continuation of Table 6.3, this table gives out the estimated time for the stages.

Force (pN)	$\Delta x(nm)$	Conditions	Source
0 - 10	3.1	simulation	[**6**]
0 - 20	1.1	folding	[**21**]
20 - 30	0.9	folding	[**21**]
30 - 40	0.28	folding	[**21**]
0 - 40	0.8	folding	[**21**]
<50	0.8	unfolding	[**43**]
>50	0.4	unfolding	[**43**]

Table 6.5. Δx as inferred from folding and unfolding experiments, where the folding or unfolding time is fit to $\exp(F \times \Delta x/k_B T)$. The values in lines 2, 3 and 4 are found by accounting for the changes that occur in Δx as the force is varied. Note that as force increases, Δx decreases. The value in the fifth line is that given in [**21**] by assuming that Δx is insensitive to the force. Even under unfolding conditions, it is found that as force increases Δx decreases and that the value of the Δx under folding and unfolding conditions are comparable.

CHAPTER 7

Conclusions to Part I

In the first part we simulated the folding procedure for constant velocity and constant force folding procedure. These simulations are based on the two-state model.

The simulation results of force-extension curve for constant velocity fit the experimental data. The variance based on two-state model can show the same tendency (shape) as the experimental results, though with a bigger amplitude. The reason for this discrepancy is still unknown.

For the constant force experiments we solve one puzzle: why the single domain folding is step-like while the multi-domain folding show stages-like behavior. The answer is the cooperativity between domains. One domain finishing its folding can help the folding of the others, leading to the stages-like behavior. But we do not know the mechanism of this cooperativity. In Part II, we give an explanation of the cooperativity inside of one domain.

In this part, when we set the value of parameters, we choose them such that we can get the best fits to the experimental results. Following the two-state model [**43**], when there is no external force, the folding rate is α_0. When there is an external force F, the force will have an effect of $\exp(F\Delta x/k_BT)$ on the folding time. Fernendez and Li [**21**] stretch ubiquitin and then allow it to refold under a force of approximately 15 pN. Using $F = 15$ pN and $x_f = 20$ nm, the folding time will be $\exp(15 \cdot 20/4.1) = 6 \cdot 10^{31}$ times slower. This means, if the folding time is 10^{-6} second with no applied force, then under 15 pN the folding time will be 10^{24} years. Simply stated, under 15 pN ubiquitin can never fold. This result is at odds with the experimental finding of ubiquitin refolding and suggest that there may be something wrong with the two-state model. Solving this puzzle is the task of the next part.

7.1. Two-state model can simulate the unfolding procedure for constant velocity stretching

The two-state model is widely used to describe the folding procedure. For this model, a protein can stay in one of the two possible states, the native state or the unfolded state. When there is no external force, the folding (unfolding) rate is α_0 (β_0). When there is an external force, the force will have an effect $\exp(F\Delta x/k_BT)$ on the folding (unfolding) time, where Δx is the distance from unfolded (native) state to the transition state. Use this model, we can numerically simulate the force versus length curve for constant velocity folding for dextran. The simulation results fit the experimental results quite well, as shown in Fig. 7.1.

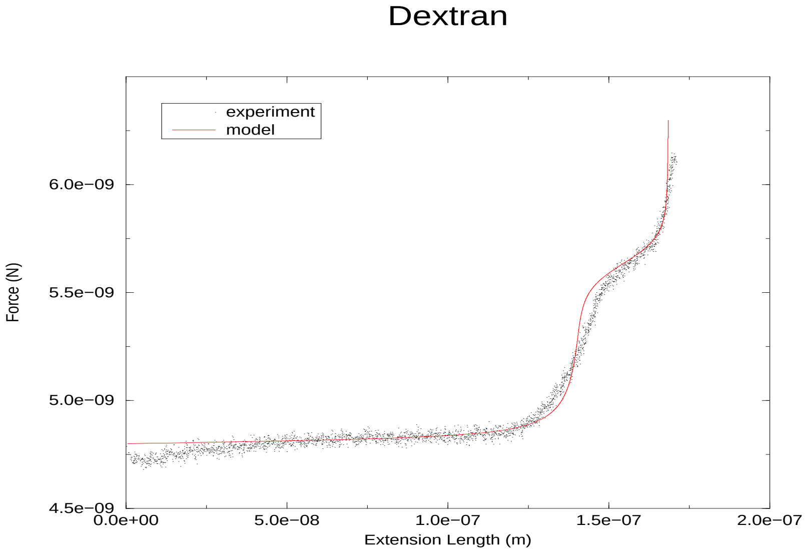


Figure 7.1. Comparison of experimental and simulation results for dextran. The dots are the experimental results. There are 310 dextran monomers. $I_f = 0.5$ nm, $I_u = 0.565$ nm, $\alpha_0 = 2 \times 10^{-4}/s$, $G_0 = 13.2\ k_BT$, and temperature is 300 K. The experimental data is from [**34**].

We can also calculate the variance based on the two-state model. The calculated results give the same tendency as the experimental results: large variance for low force; when force increases, the variance decreases; for moderate forces, there is a peak for the variance as the states transition from short to long length, see Fig. 7.2.

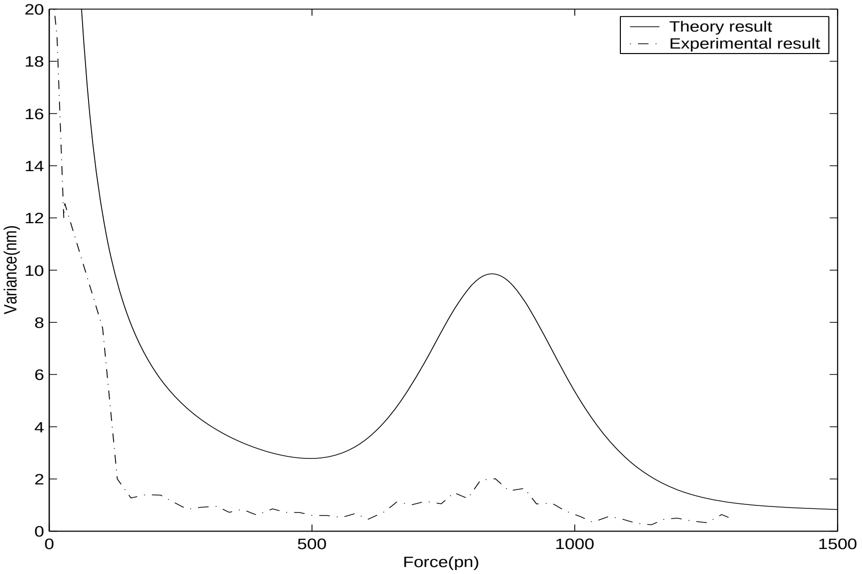


Figure 7.2. Comparison of experimental results and theoretical results for the variance of extension of dextran.

7.2. The disappearance of step-like behavior might come from inter-domain interaction

We can reproduce step-like behavior for single domain for constant force folding procedure. Fig. 7.3 and 7.4 show the experimental and simulation results.

For multi-domain folding procedure (constant force), we find that if we allow an interaction coefficient between domains, which means once one domain finished the folding it will help the other domain fold, we can also reproduce the stages of folding shown in Fernandez and Li's paper. Shown in Figs. 7.5 and 7.6.

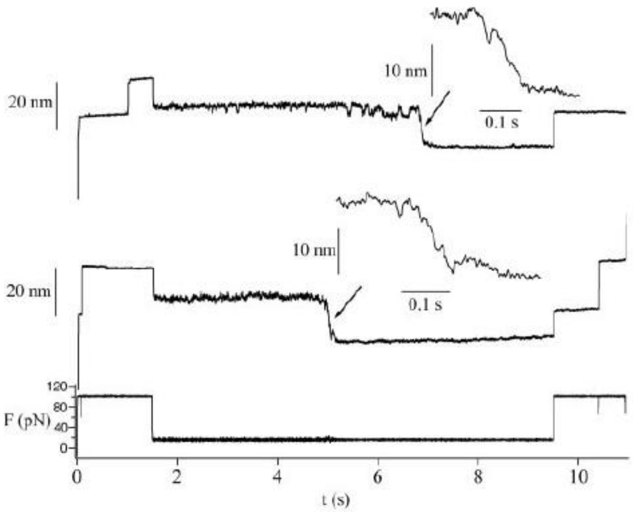


Figure 7.3. Results from [**21**], where it is assumed that there is only one ubiquitin domain being stretched by the AFM.

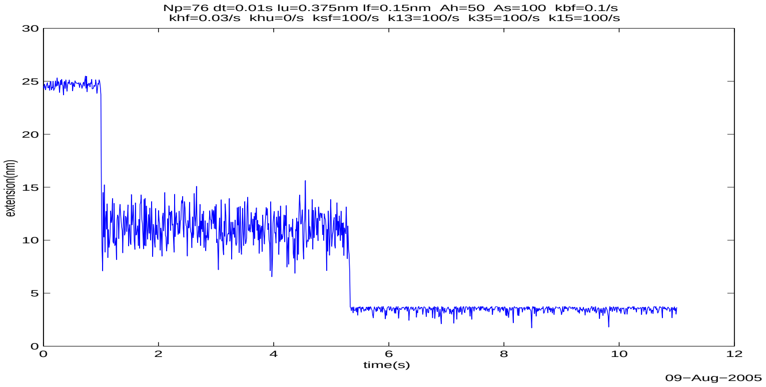


Figure 7.4. Numerical simulation of the folding procedure for single domain.

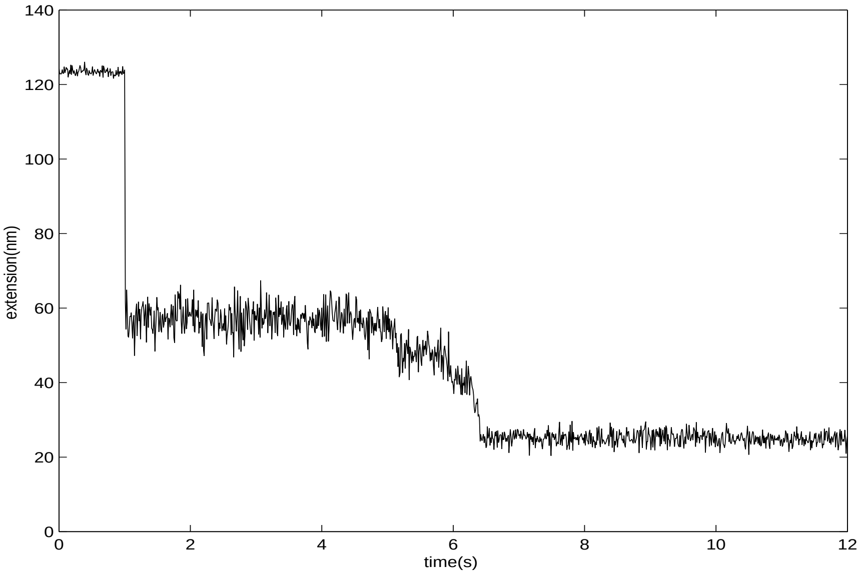


Figure 7.5. Simulation results for multi-domain folding procedure. This one looks like experimental results.

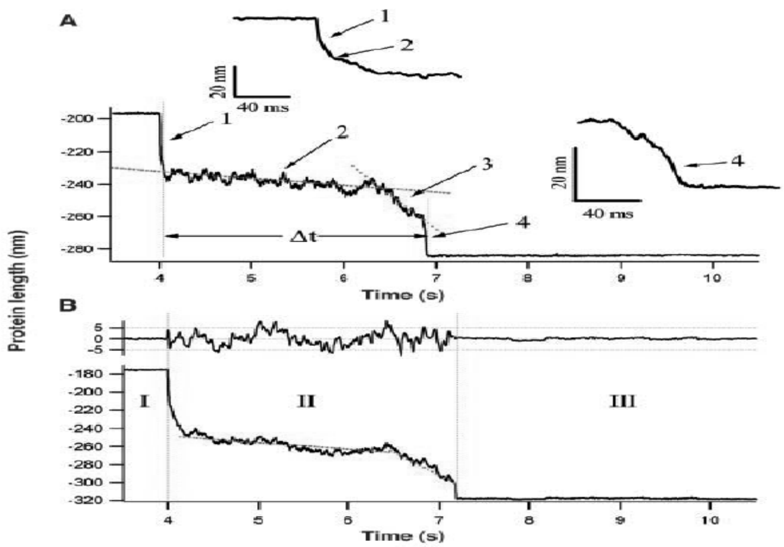


Figure 7.6. Experimental results for constant force folding procedure. From [**21**].

Part II

In Part I we mentioned that there is a problem for the two-state model: a small force might slow down the folding procedure so much that some proteins can never fold under force, which is not corresponding to experimental observation. In Part II, we try to solve this problem.

In Sec. 8 we present a first attempt to improve the two-state model. Instead of assuming that the protein can only be in one of two possible states, we assume that the protein can be in many different configurations. These configurations are simplified as a one-dimension reaction coordinate. For the stretching experiments, this reaction coordinate is the end-to-end length of the protein. We assume that for any end-to-end length, there is a corresponding free energy. Therefore we can get one curve: free energy versus reaction coordinate. We assume that the folding procedure can be simulated as one-particle diffusion through this energy field. We simulate with different kinds of potential fields to get a better understanding of the folding procedure.

In Sec. 9 we show that one particle-diffusion simulation can be replaced by a more realistic simulation: multi-particle diffusion. In this new simulation all the particles can diffuse simultaneously. Then, the slowest procedure determines the overall folding time. Though many particles diffusion is assumed here, one-particle diffusion is not useless: for any particle, its diffusion is still a one particle diffusion. The change is that the free energy used for any particle should not be the free energy of the whole protein, but only part of the whole protein's free energy. This is the key of making the whole protein's folding time becomes less sensitive to the external force.

In Sec. 10 we apply the idea of multi-particle diffusion to an imaginary protein. We make some reasonable assumptions about how the force affects the formation of bonds. We have some interesting results, shown in details in the section.

In Sec. 11, we apply the ideas in Sec. 10 to a real protein, RNase H.

CHAPTER 8

The change in MFPT with different potential fields

For two-state model, rate constants are the parameters to describe the folding time. In diffusion problems, we use mean first passage time (MFPT) instead. This is the average time for the particle to diffuse from its initial position to its ending position, which represents the protein folding from the unfolded to the folded state.

8.1. Numerical simulation for one intermediate state

We begin with the case where there is only one intermediate state. For simplicity, we choose a cos function to generate the potential

$$\begin{aligned} |x| &\leq 0.5: \quad G = -\text{depth}/2k_BT\cos(2\pi x) + \text{depth}/2k_BT \\ 0.5 &\leq |x| \leq 1: \quad G = -5k_BT\cos(2\pi x) + 5k_BT \end{aligned}$$

as shown in Fig. 8.1. The potential fields are set so that there are at least two local minimum, corresponding to the initial unfolded state and the final native state. The energy difference between the initial energy and the highest energy (transition state) is called the "height". If there are some other local minima, we call them intermediate states. The energy difference between the transition state and the intermediate states is called the "depth". The spatial distance between neighboring local minima is call the "width". This potential, then, can be specified in terms of its height(amplitude) and width. We first keep the width unchanged, and vary only the depth of the intermediate states. The MFPT's are shown in Tab. 8.1 and Fig. 8.2.

depth	MFPT
no intermediate	1483
0 k_BT	2392
2 k_BT	1313
4 k_BT	999
6 k_BT	882
8 k_BT	908
10 k_BT	1481
12 k_BT	5447

Table 8.1. MFPT for different depths of the intermediate state, as shown in Fig. 8.1.

The simulation procedure is as follows. We divide the potential field spatially equally into N points. There is one energy corresponding to each point. We assume the particle can jump from one point to its neighbors with certain rates. For example, the ith point has energy G_i, the $(i+1)$th point has energy G_{i+1}, and the energy difference is defined as $\Delta G_{i+1/2} = G_{i+1} - G_i$. The spatial distance between the two points is Δx, the diffusion coefficient is D. Then the rate that the particle jumps forward from the ith point to the the $(i+1)$th point is $F_{i+1/2} = \frac{D}{\Delta x^2} \cdot \frac{\Delta G_{i+1/2}/k_BT}{\exp(\Delta G_{i+1/2}/k_BT)-1}$. The backward rate from $(i+1)$th point to the the ith point is $B_{i+1/2} = \frac{D}{\Delta x^2} \cdot \frac{\Delta G_{i+1/2}/k_BT}{1-\exp(-\Delta G_{i+1/2}/k_BT)}$. Then we can use a random number to decide the particle's behavior. The first time it reaches the ending point (the native state), the folding procedure is complete. The code is shown in Appendix 13.1.

The presence of a intermediate state has two effects. First, it can provide a "rest area" for diffusion. As a local energy minimum, a particle can stay at the intermediate state, and use it as a new starting point to finish its travel. In this manner, the intermediate state can help the diffusion procedure. This is why moderate depths of the intermediate can accelerate the diffusion procedure. On the other hand, it takes extra time to get out of the intermediate state. In this manner, the intermediate state slows the diffusion procedure. This is the reason that for larger depths, the presence of the intermediate state increases the MFPT.

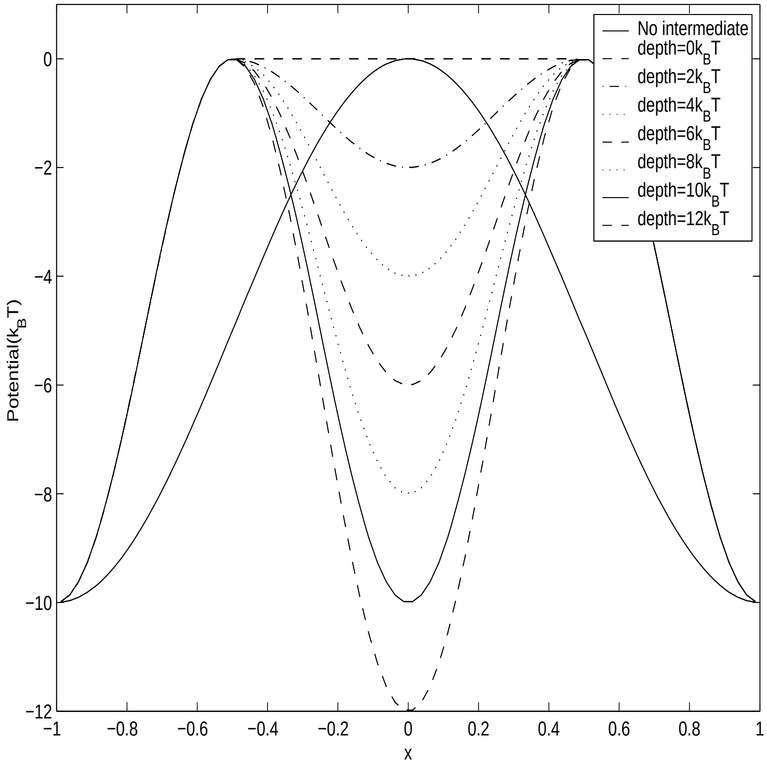


Figure 8.1. Here, the intermediate state is of constant unit width and varying depth. The solid line is for no intermediate state and is given by a single cos peak, $|x| < 1, G = 5k_BT\cos(\pi x) - 5k_BT$. The dashed line (designated $0k_BT$) is for an intermediate state of zero depth, i.e. a plateau, $|x| \leq 0.5 : G = 0$; $0.5 \leq |x| \leq 1 : G = -5k_BT\cos(2\pi x) + 5k_BT$.

8.1.1. Analytical calculation for one intermediate state

There is one stable intermediate state between two peaks as shown in Fig. 8.3. Denote the MFPT from the initial position A to the transition state B as t_1 . The MFPT for the backward transition from intermediate state C to B is t_2. The MFPT from C to the ending position D is t_3. A particle at position C has probability $p = \frac{t_2}{t_2+t_3}$ to go forward to D, and probability $q = \frac{t_3}{t_2+t_3}$ to go backward to A. Note that, $k = 1/t$, where k is the rate constant. At any position, the probability of going forward is proportional to the rate of going forward. Therefore we can get the probability of going forward or backward based on the MFPT. For simplicity, we will use notation $t_{23} \equiv t_2 + t_3$. For a particle initially in position A, after time t_1 it arrives at position

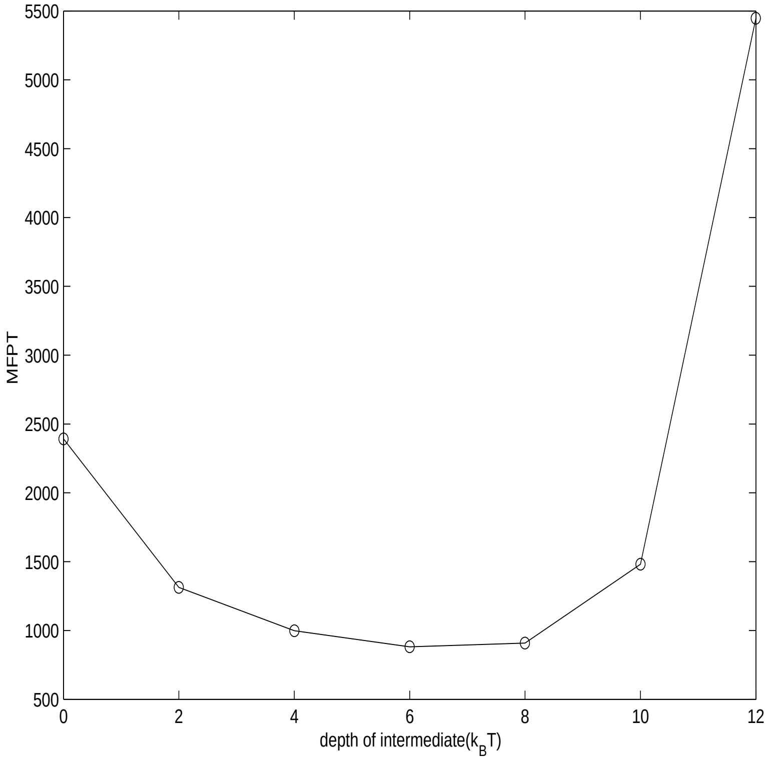


Figure 8.2. The MFPT results from Table 8.1 plotted versus the depth of the intermediate states shown in Fig. 8.1.

C. Now it has two choices: go to D after time t_3 with probability p, or go back to A after time t_2 with probability q. If it goes back to A, it needs to repeat the procedure again. The MFPT with this intermediate state is

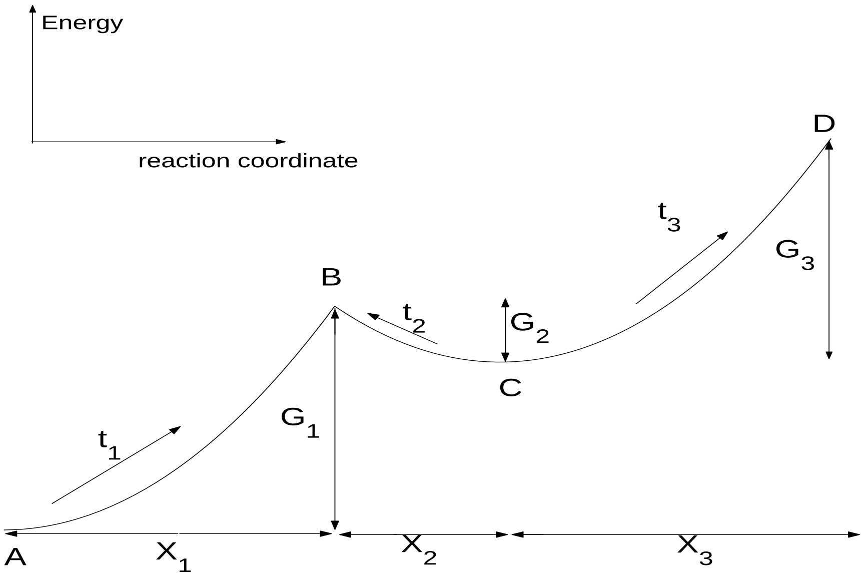


Figure 8.3. Energy barrier with intermediate state in the middle.

$$
\begin{aligned}
MFPT &= t_{13}p + (t_{13}+t_{12})pq + (t_{13}+2t_{12})pq^2 \\
&\quad +(t_{13}+3t_{12})pq^3 + \cdots \\
&= t_{13}p(1+q+q^2+\cdots) + t_{12}p(q+2q^2+3q^3+\cdots) \\
&= t_{13}p\frac{1}{1-q} + t_{12}pq(1+2q+3q^2+\cdots) \\
&= t_{13}p\frac{1}{1-q} + t_{12}pq\frac{d(1+q+q^2+q^3+\cdots)}{dq} \\
&= t_{13} + t_{12}pq\frac{1}{(1-q)^2} \\
&= t_{13} + t_{12}\frac{q}{p} \\
&= t_1 + t_3 + (t_1+t_2)\frac{q}{p} \\
&= (t_1+t_3) + (t_1+t_2)\frac{t_3}{t_2}
\end{aligned}
$$

(8.1)

The expression for the MFPT can be considered in two parts. The first term, t_1+t_3, accounts for the fact that no matter what the pathway is, to reach the right-hand end the particle must go through the entire potential field once, which takes time t_1+t_3. The second term, $(t_1+t_2)\frac{t_3}{t_2}$, accounts for the additional contribution from those particles which traverse part of the path repeatedly. For, we note that the particle does not always go forward; sometimes it will go backward. If it goes backward from the intermediate state, it takes time t_2 to reach the initial position, and t_1 to return to the intermediate state. Therefore, to go backwards out of the intermediate state to the initial state one time takes an extra time (t_1+t_2). This time is multiplied by $\frac{t_3}{t_2}$ which is the average number of times that the particle will go backwards out of the intermediate state. When t_3 is small and t_2 is large, the particle tends to proceed directly from the initial state to the final state. When t_2 is small and t_3 is large, the particle's trip to the final state will tend to be composed of repeat passes over the transition state at B.

A simple case to consider is $t_1=t_2=t_3$, in which the two peaks are identical, and each peak is symmetric about its maximum. It this case $\langle FPT\rangle = 4t_1$. The intermediate state makes the MFPT four times longer. Note that we will use MFPT or $\langle FPT\rangle$ interchangeably.

If only $t_1=t_3$, then the MFPT becomes $MFPT = 3t_1+\frac{t_1^2}{t_2}$.

If the two peaks are symmetry about the intermediate state, as in the numerical simulations shown above, then we have $t_2=t_3$. Then the MFPT becomes

$$MFPT = 2(t_1+t_2) \tag{8.2}$$

In [**52**], the MFPT is calculated using

$$MFPT_{approx} = \frac{2}{k_1}+\frac{1}{k_2}$$

Potential	Numerical results	Wagner [**52**]	$MFPT_{approx}$
Fig. 8.4	924.5	920.7	925.5
Fig. 8.5	924.6	467.5	925.5

Table 8.2. Comparison of the results from [**52**] and our results.

where the k's are rate constants. In our notation, this equation can be written as

$$MFPT_{approx} = 2t_1 + t_2 \tag{8.3}$$

We use numerical simulation to compare Wagner's results and our results, as shown in Figs. 8.4 and 8.5. These two potentials are both symmetric about the intermediate state. We can split the potential, at the intermediate state, into two potentials, and simulate the MFPT for each part separately, see Figs. 8.6 and 8.7.

For Fig. 8.4, $t_1 = 458.0$ and $t_2 = 4.775$. The numerical simulation result is $MFPT = 924.5$, Wagner's method gives the result 920.7 and our method gives 925.55. Both are close to the numerical results. This is because $t_1 \gg t_2$, and the difference between the two methods is not significant. For Fig. 8.5, $t_1 = 4.775$ and $t_2 = 458$. For this case, $t_1 \ll t_2$, and we can see that our method is accurate, Table 8.2.

8.1.2. One intermediate with flat potential regions

For symmetric step-like regions with one intermediate, Fig. 8.8), we can calculate the MFPT analytically [**23**]. Take the width of the one region to be W, the diffusion coefficient as D, the height of the first peak as U_H, and the depth of the intermediate as U. Then the MFPT is

$$\langle FPT \rangle = \frac{W^2}{D}\left[\frac{5}{2} + 4\cosh\left(\frac{U_H}{k_BT}\right) + 2\cosh\left(\frac{U}{k_BT}\right) + 2\cosh\left(\frac{U_H - U}{k_BT}\right)\right] \tag{8.4}$$

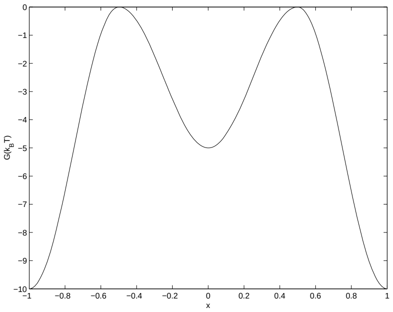


Figure 8.4. $MFPT = 924.5$. The potential is $|x| \leq 0.5$: $G = -2.5k_BT\cos(2\pi x) + 2.5k_BT$;$0.5 \leq |x| \leq 1$: $G = -5k_BT\cos(2\pi x) + 5k_BT$.

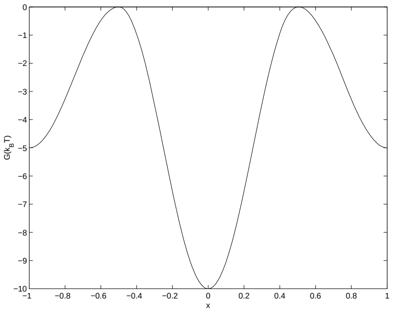


Figure 8.5. $MFPT = 924.6$. The potential is $|x| \leq 0.5$: $G = -5k_BT\cos(2\pi x) + 5k_BT$;$0.5 \leq |x| \leq 1$: $G = -2.5k_BT\cos(2\pi x) + 2.5k_BT$.

Fig. 8.9 shows a comparison between the numerical simulation results using (8.1) and the analytical prediction from (8.4). The two agree quite well. The small difference is due to simulation error which can be reduced by further decreasing the simulation time step.

To study the effects of the position and the width of the intermediate, we use a spike-like potential, as shown in Fig. 8.10. Table 8.3 and Fig. 8.11 give the results for different positions of the intermediate state.

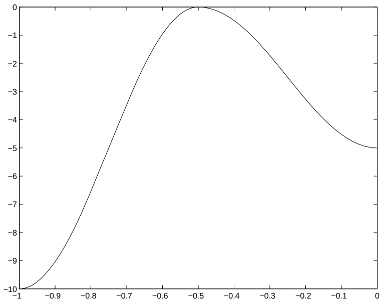


Figure 8.6. The $MFPT = 458.0$ for the left half $-1 \le x \le 0$ of the potential in Fig. 8.4. This $MFPT$ is used as t_1 in (8.2).

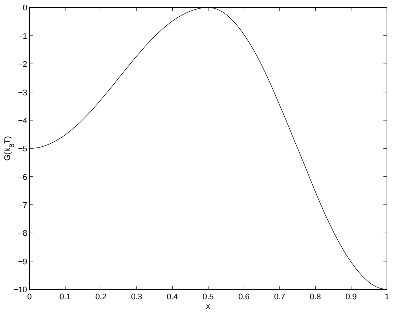


Figure 8.7. The $MFPT = 4.775$ for the right half $0 \le x \le 1$ of the potential in Fig. 8.4. This $MFPT$ is used as t_2 in (8.2).

Fig. 8.12 shows the simulation results for different widths of the intermediate state, while the position of the intermediate state is fixed at the middle of the interval.

8.1.3. Multiple identical peaks

In this section, we investigate the presence of multiple intermediate states on an otherwise smooth linearly changing potential. We use the potential $G = k_BT\,[kx + 1 - \cos(2\pi x)]$ as a numerical simulation example. If there are N identical peaks, the region of x is $0 \le x \le N$. When N increases, the length of the domain also increases. There are two parts in the potential: the linearly changing part $k_BT \cdot kx$ and the periodic part $k_BT\,[1 - \cos(2\pi x)]$. If $k > 0$, the underlying

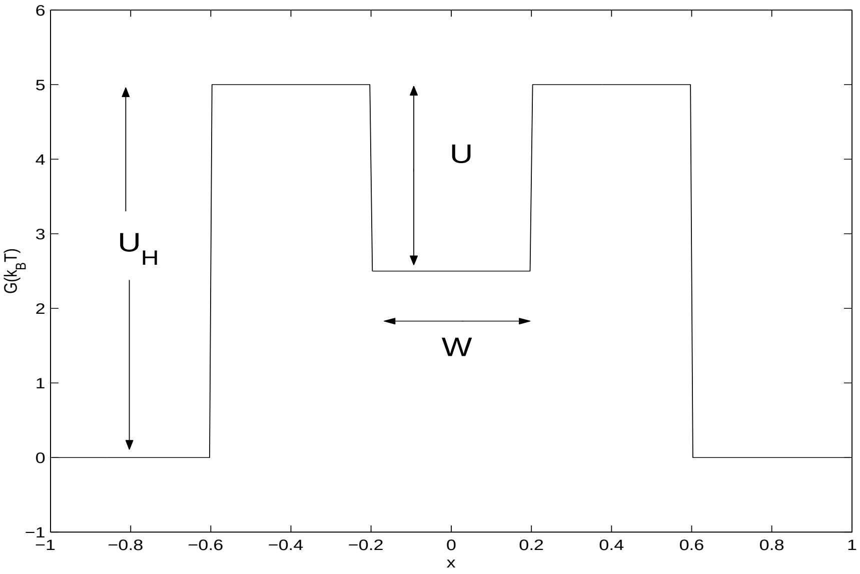


Figure 8.8. One intermediate with flat potential regions.

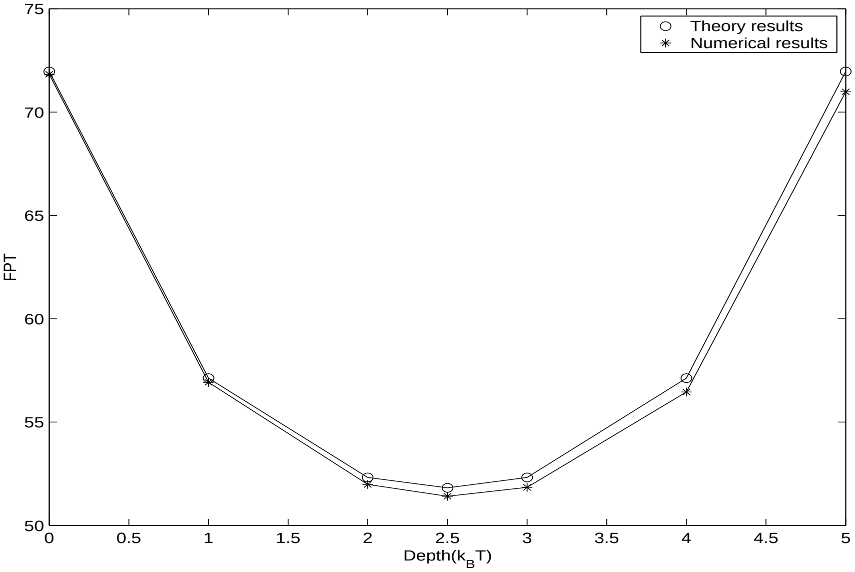


Figure 8.9. Comparison of numerical simulation results and analytical prediction from (8.4) for step-like potentials.

potential increases linearly while if $k < 0$, the underlying potential is decreasing. Fig. 8.13 shows the potential field for $N = 5$.

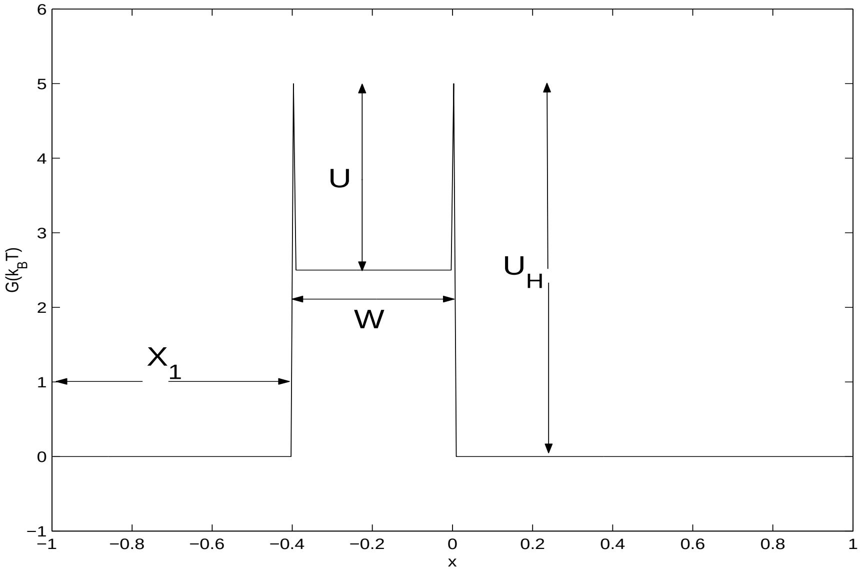


Figure 8.10. Spike-like potentials.

x_1	W	MFPT
0.1	0.2	2.5915
0.2	0.2	3.7378
0.3	0.2	4.8842
0.4	0.2	6.0304
0.5	0.2	7.1763
0.7	0.2	8.3220
0.7	0.2	9.4676

Table 8.3. Numerical simulation results for the MFPT for the potential shown in Fig. 8.10 for different positions of the intermediate state and $U_H = 5k_BT$, $U = 2.5k_BT$, $W = 0.2$. x_1 and W are normalized by the the total length of the whole region. The simulation spatial step was $\Delta x = 1/160$.

Numerical simulations show that if k is positive, then the MFPT time increases exponentially as the number of peaks increases. *I.e.*, $\langle FPT \rangle \propto \exp(Nc)$ where c is some constant, Fig. 8.14.

When $k = 0$, the MFPT time increases proportional to N^2, *i.e.*, $\langle FPT \rangle \propto N^2$, Fig. 8.15.

When $k < 0$, the MFPT time increases proportional to N, *i.e.*, $\langle FPT \rangle \propto N$, Fig. 8.16.

Fig. 8.17 shows the results for *fixed* length of the region length and fixed peak amplitude, but varying number of peaks in the region. For $k = 0$, the MFPT is constant, as expected. Based

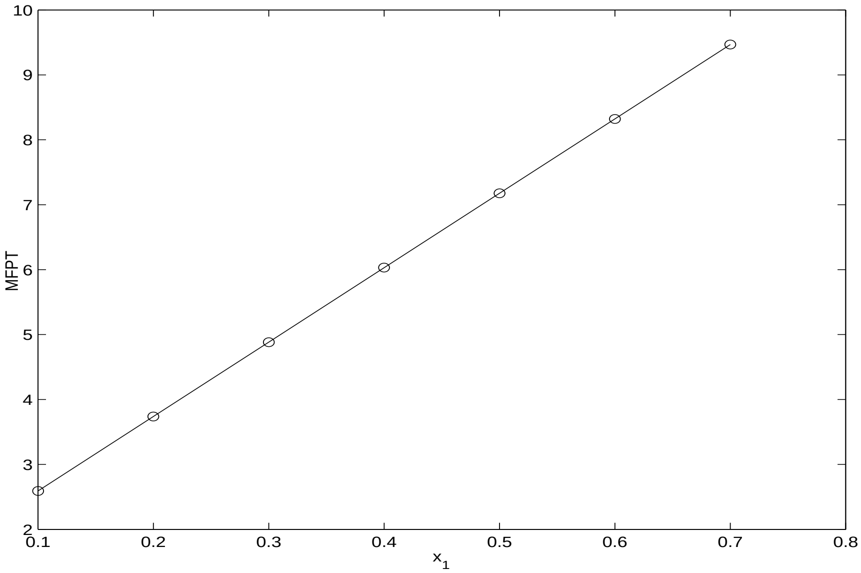


Figure 8.11. The results from Table 8.3 show that the MFPT increases linearly with the position x_1 of the intermediate state.

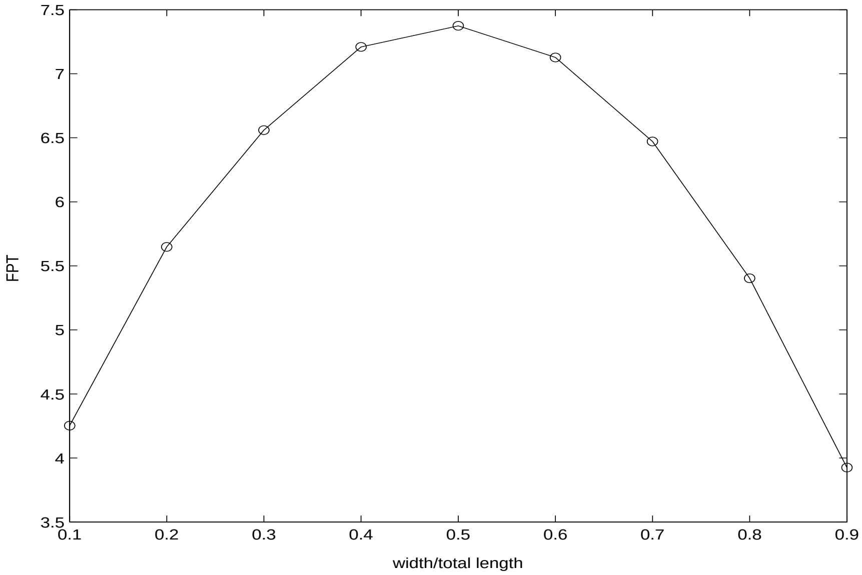


Figure 8.12. Numerical simulation results for the spike-like potential with different widths, $U_H = 5k_BT$, $U = 2.5k_BT$, and x_1 chosen such that the intermediate state is in the middle of the interval. The simulation spatial step was $\Delta x = 1/320$.

on dimension analysis, we know that the folding time for one peak is proportional to X^2, where X is the spatial size of the peak. That is, if we keep the shape of the potential unchanged and

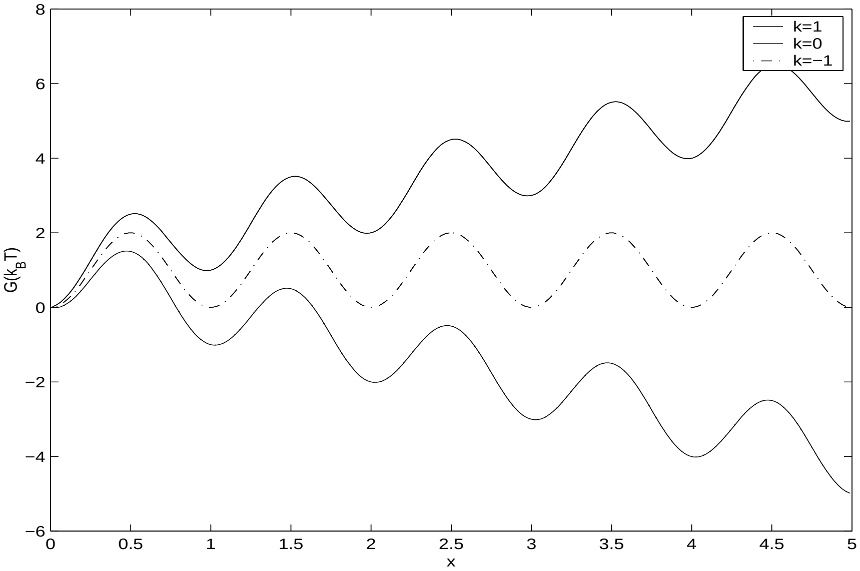


Figure 8.13. Potential $G = k_BT\,[kx + 1 - \cos(2\pi x)]$ for different k and the case of five peaks.

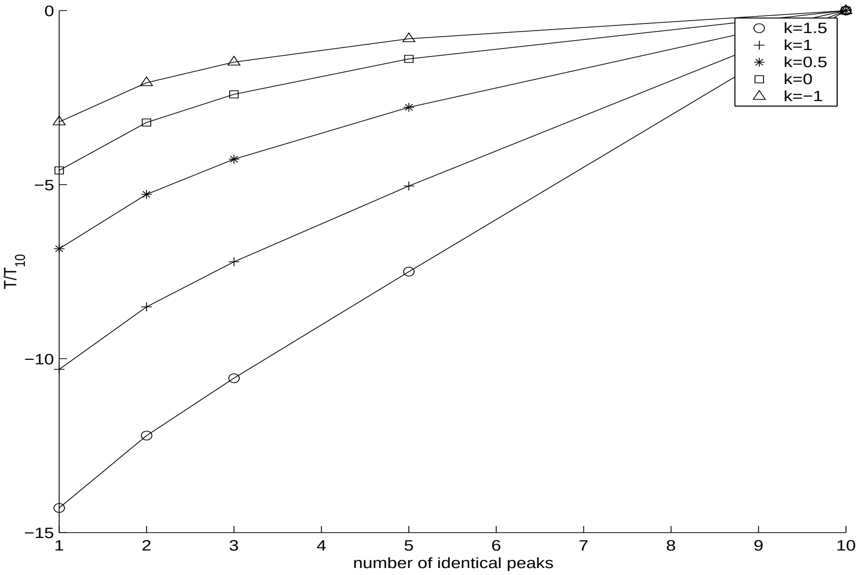


Figure 8.14. MFPT for different k on a log scale. For large positive k, we have a straight line fit. $G = k_BT\,[kx + 1 - \cos(2\pi x)]$. T_{10} is the mean FPT for $N = 10$.

only shrink the spatial distance by half, the folding time will become 4 times shorter. Now we need two such smaller peaks to keep the total length unchanged. For the identical two symmetry peaks, the total folding time is 4 time longer than one. These two effects are canceled, the total

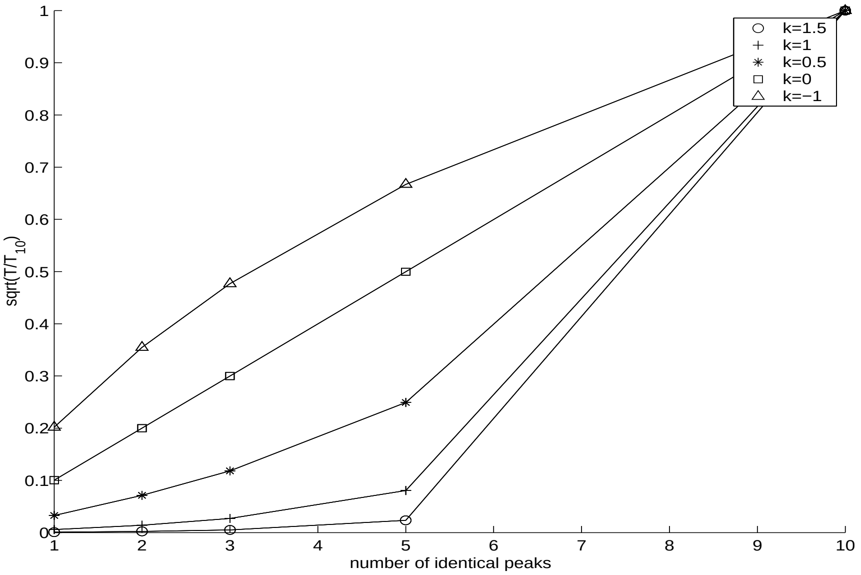


Figure 8.15. The square root of the Mean FPT for different k. For $k = 0$, we have a straight line fit. $G = k_B T\,[kx + 1 - \cos(2\pi x)]$. T_{10} is the mean FPT for $N = 10$.

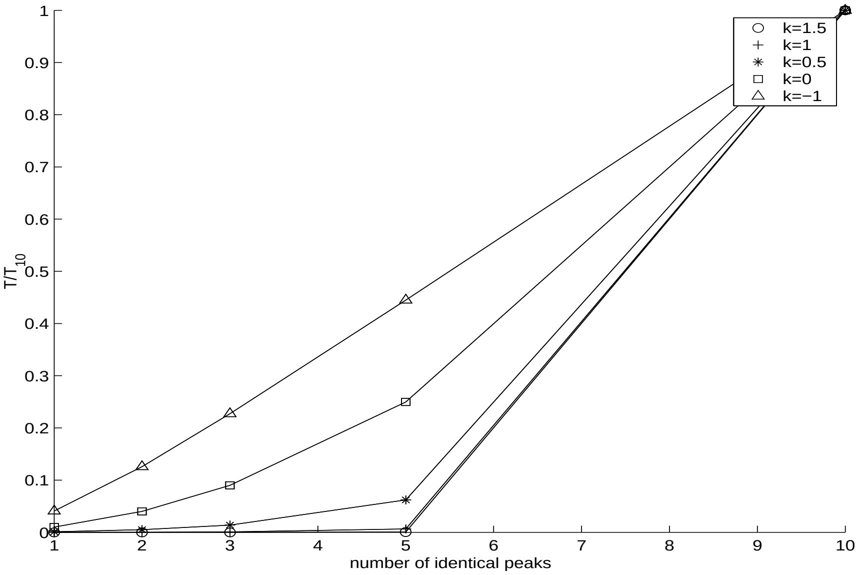


Figure 8.16. Mean FPT for different k in a linear scale. For negative k, we have a straight line fit. $G = k_B T\,[kx + 1 - \cos(2\pi x)]$. T_{10} is the mean FPT for $N = 10$.

time remains unchanged. This is the reason that in increasing the number of peaks from one to two, the folding time remains constant. Similarly for increasing the number of peaks to any

Δx	MFPT for $N = 10$
1/40	4125.10
1/80	4160.9
1/160	4169.9

Table 8.4. Numerical simulation results for the MFPT with different precision. Decrease in Δx will cause the results more precise. The difference is about 1% when changed from $\Delta x = 1/40$ to $\Delta x = 1/160$.

value. For both positive k and negative k, the increase of the number of peaks first causes some increase in time, but then the MFPT becomes almost constant while the number of peaks varies. This means if the length of the region is kept constant, then the MFPT is not sensitive the number of peaks.

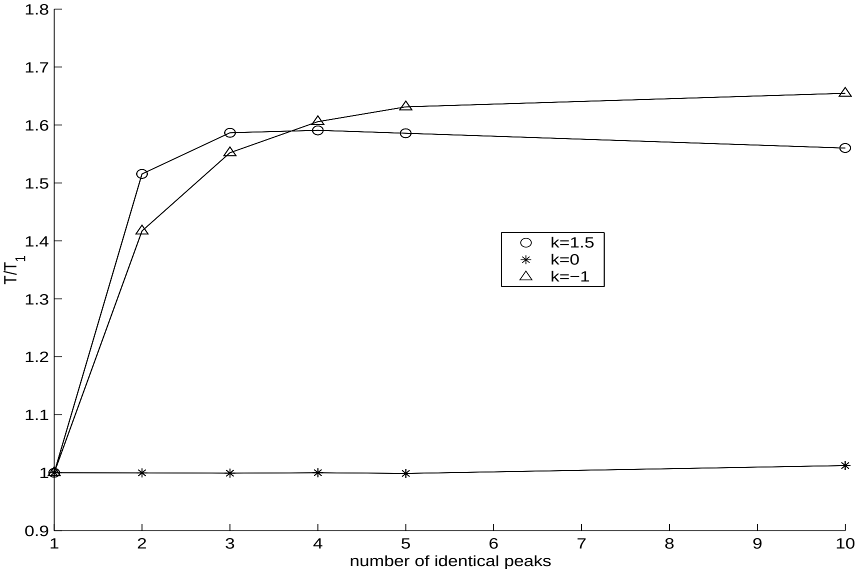


Figure 8.17. Mean FPT for different k. $G = k_BTkx - a\cos(2\pi x)$, where $a = 2k_BT$. T_1 is the mean FPT for $N = 1$. As shown in Table 8.4, the decrease in time with number of peaks for $k = 1.5$ seems not to be due to numerical error. Increasing the the simulation precision by a factor of 4 only makes the results change about 1%.

Fig. 8.18 shows the effects of the amplitude of the peaks. Increasing the amplitude makes the MFPT longer.

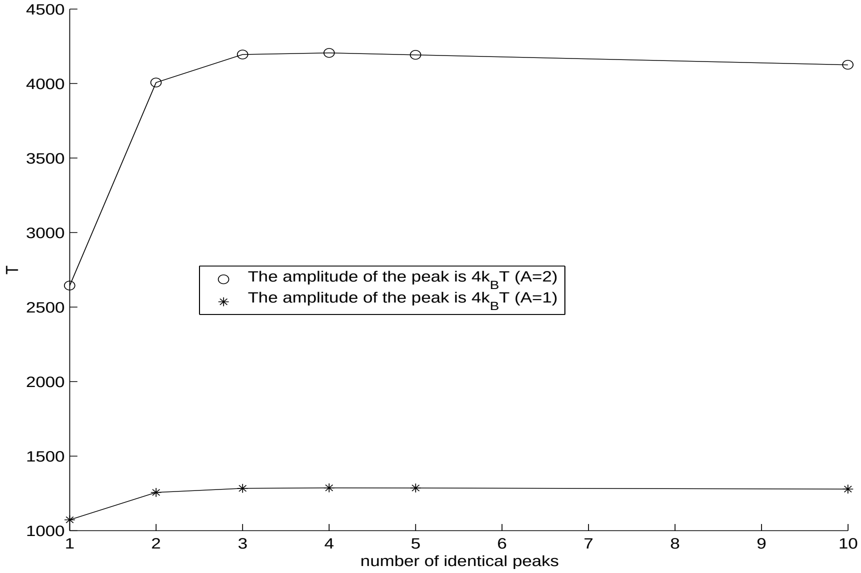


Figure 8.18. Mean FPT for different amplitude of peaks. $G = k_BTkx - a\cos(2\pi x)$, where $a = 2k_BT$ and $a = 1k_BT$.

CHAPTER 9

Can the folding procedure be simulated by the diffusion of a single particle?

9.1. Simultaneous diffusion simulation of protein folding

The two-state model is widely used in protein folding simulations. For proteins whose length change in folding is large, the two-state model predicts that the folding time is very sensitive to the external force. Experimental results, however, find to the contrary, see Sec. 11. However, if we use an alternative model, which we call the simultaneous diffusion model, this problem does not arise. As shown in the section on the constant force refolding simulation, Sec. 6, if we choose the folding rate (the inverse of the mean first passage time for diffusion problem) properly, a simultaneous diffusion model can also yield step-like behavior for single domain folding, which is regarded as a characteristic of the two-state model. Even though the simultaneous diffusion model is closer to the actual folding procedure, it can still be represented as one-particle diffusion, i.e. as diffusion over $G(x)$, where x is one-dimensional.

9.2. Experimental discrepancy for the two-state model

Fig. 9.1 shows the free energy diagram, G versus the reaction coordinate. The two possible states are separated by the energy barrier of the transition state. A protein moves from one state to the other with a certain rate. In force-stretching experiments, the end-to-end length is normally chosen as the reaction coordinate. One simple consequence of the two-state model is that $x_u + x_f = \Delta L$, where ΔL is the length change from one state to the other, and x_u and x_f are defined in the figure.

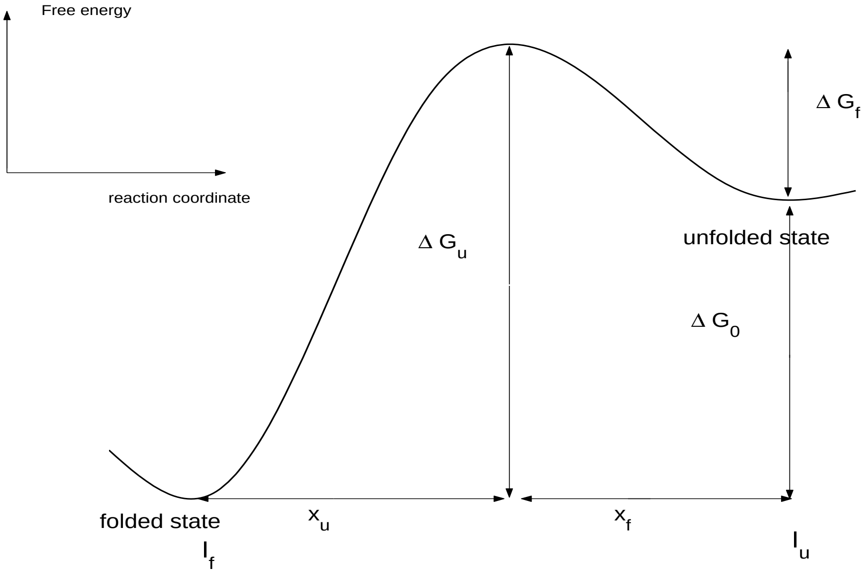


Figure 9.1. Free energy diagram in reaction coordinates. If the domain is in the folded state, its length is I_f. If it is in the unfolded state, its length is I_u. The energy difference between the folded state and the energy barrier is ΔG_u, and the distance between the folded state and the energy barrier is is x_u. The energy difference between the folded state and the unfolded state is ΔG_f, and the distance from energy barrier to unfolded state is x_f.

Schlierf [**50**] shows that $x_u = 0.17$ nm for ubiquitin. It is also shown that the total length change from the native state to the unfolded state is about 20 nm. Using the two-state model, we can estimate $x_f \approx 20$ nm. Fernendez and Li [**21**] stretch ubiquitin and then allow it to refold under a force of approximately 15 pN. Following the two-state model [**43**], when there is no external force, the folding rate is α_0. When there is an external force F, the force will have an effect of $\exp(F\Delta x/k_B T)$ on the folding time. Using $F = 15$ pN and $x_f = 20$ nm, the folding time will be $\exp(15 \cdot 20/4.1) = 6 \cdot 10^{31}$ times slower. This means, if the folding time is 10^{-6} second with no applied force, then under 15 pN the folding time will be 10^{24} years. Simply stated, under 15 pN ubiquitin can never fold. This result is at odds with the experimental finding of ubiquitin refolding and suggest that there may be something wrong with the two-state model.

Fernandez and Li [**21**] measure folding rate as a function force. Fitting this data by assuming that folding time $\sim \exp Fx_f$, they find that $x_f = 0.8$ nm. Therefore, $x_f + x_u = 2.5$ nm, instead of 20 nm. This discrepancy can not be explained by the two-state model.

9.3. Viscosity affects protein folding

Jacob [**28**] shows that viscosity of the solvent affects the folding rate. This means that diffusion plays a role in the protein folding procedure. But, what we need here is not a single particle diffusion simulation, because for one particle diffusion the folding time is still sensitive to the external force. Instead, we need a simultaneous diffusion model.

9.4. Reducing multi-dimensional folding in (x, t) to the diffusion of a single particle along one dimension

Protein folding is a complex process involving many degrees of freedom. The complexity of describing protein conformational changes is often reduced by considered that the amino acids are rigid units, so that only rotations about the peptide bond need to be considered. In this case, for a protein of N amino acids, there are $2(N-1)$ degrees of freedom, accounting for the two angles ϕ and ψ at each peptide bond.

Even with this simplification, we are left with a large problem involving many degrees of freedom.

Protein folding is sometimes considered as a problem of a single particle diffusion problem with the end-to-end length as the reaction coordinate. Then, there is a free energy potential $G(x)$ which corresponds to that length x. Therefore, we can get a length versus potential curve $G(x)$. We could then consider using that potential field for the one-particle diffusion problem, in order to simulate the protein folding procedure. That is, we would use only one degree of freedom, a one-dimensional reaction coordinate along which the protein folds. But, the folding procedure of a one-degree of freedom system is not the same as that for one of $2(N-1)$ degrees of freedoms. But the one-degree of freedom system can be used to approximate the higher-dimensional system under some conditions, as discussed below.

The change along the reaction coordinate comes from many contributions. For example, we can consider, as noted above, that the individual amino acids act as rigid components. During the folding procedure, we can track the configurational changes of each amino acid. Let us consider that the reaction coordinate is the end-to-end length. From the initial unfolded state to the final native state, the projected length of the first amino acid has some change Δx_1, also some energy change ΔG_1. Here the projected length means the length projected to the line between two ends of the protein. (With this definition, translations and rotations of the protein as a whole, while remaining static in configuration, do not alter the projected lengths of the component amino acids.) Similarly we define $\Delta x_2, \Delta x_3, \ldots, \Delta x_N$ and $\Delta G_2, \Delta G_3, \ldots, \Delta G_N$, where N is the number of amino acids. The N particles, each representing an amino acid are allowed to diffuse. When all of them finish folding, we can say the whole protein has then finished folding. That is, the slowest i of the N amino acids determines the MFPT of the protein.

For the alternative description in terms of the diffusion of a single-particle diffusion problem, the single particle is required to diffuse through the length $\Delta x = \Delta x_1 + \Delta x_2 + \cdots + \Delta x_N$ and through an energy $\Delta G = \Delta G_1 + \Delta G_2 + \cdots + \Delta G_N$. Now we ask, "Does the N-particle diffusion model give the same MFPT as the one-particle diffusion model?" If so, then the one-particle model would be a better model as it is much simpler.

Assume that the energy of each amino acid is only a function of end-to-end length. Then we have $\Delta x_1 = \Delta x_2 = \cdots = \Delta x_N = \Delta x/N$ and $\Delta G_1 = \Delta G_2 = \cdots = \Delta G_N = \Delta G/N$. For the first amino acid $i = 1$, we can do the diffusion experiment and get a FPT t_{11}. After repeating the diffusion many times, the MFPT for amino acid $i = 1$ can be computed, $\langle T_1 \rangle = \sum_{i=1}^{R\to\infty} t_{1i}/R$, where R is the number of times the experiment is repeated. Similarly, $\langle T_2\rangle = \sum_{i=1}^{R\rightarrow\infty} t_{2i}/R, \ldots$. If all the amino acids were identical, we would have $\langle T_1\rangle = \langle T_2\rangle = \cdots = \langle T_N\rangle$.

Since the FPT of the protein is determined by the slowest-folding amino acid, the MFPT for the protein is $\langle FPT_{N-particle}\rangle = max(\langle T_1\rangle, \langle T_2\rangle, \ldots, \langle T_N\rangle)$, where max chooses the maximum of the N single amino acid MFPT's.

9.4.1. Mean first-passage times

Now we want to compare the two folding procedures. In one-particle diffusion, the $\langle FPT_{One-particle}\rangle$ accounts for diffusion through length Δx with energy change ΔG. The second procedure is to divide the length into N smaller parts, and to also divide the total energy change into smaller parts. For simplicity, we assume that the smaller length regions and energies are equally divided. We use N-particle diffusion to simulate this procedure. Because of the shorter length, the MFPT for each small region should be shorter.

On the other hand, the completion of the N-particle diffusion requires all the particles to finish folding. The more regions, the more time we should wait. Can these two effects balance each other so that the one-particle diffusion has the same MFPT as the N-particle diffusion?

If the potential is constant or changes very little compared to k_BT, *i.e.*, $\Delta G << k_BT$, then the effects of the energy change are trivial. In this case, the problem can be regarded as a free diffusion problem. If the distance x for the diffusion becomes N times smaller, then it occurs N^2 times faster. As shown in Sec. 9.4.2, the conclusion that the MFPT of diffusion scales with N^2 is general and holds for any potential. Even for arbitrarily large energy changes, $\Delta G >> k_BT$, if only the x scale is compressed N times, the MFPT becomes N^2 times faster.

Moreover, a reasonable assumption is that the energy barriers encountered by a single particle would be less than the energy barriers of the whole system. In this case, the MFPT of

N	MFPT
1	0.9992
4	2.0783
9	2.8327
16	3.3813
25	3.8148
100	5.1896

Table 9.1. $\langle FPT_{N-particle}\rangle$ for different N.

each of the N particles should be even shorter. Therefore, we can use the estimate $\langle T_1\rangle \leq \langle FPT_{One-particle}\rangle/N^2$.

We now need to specify the FPT distribution for one small region. As an approximation, we choose the exponential decreasing distribution, *i.e.*, the probability that $t < T_1 < t + \Delta t$ is given by $P(t < T_1 < t + \Delta t) = exp(-t)\Delta t$. In this case, we find numerically that $\langle FPT_{N-particle}\rangle = \sqrt{N}\langle T_1\rangle$, as shown in Table 9.1. This square root scaling is only approximate.

But it is clear that $\langle FPT_{N-particle}\rangle \neq N^2\langle T_1\rangle$. Using the approximate square-root scaling, we have

$$\langle FPT_{One-particle}\rangle \geq N\sqrt{N}\langle FPT_{N-particle}\rangle \tag{9.1}$$

The two models give different results. That is to say, let us compare a model which uses a single-particle diffusing over the potential surface of the folding of the entire protein versus a model of N particles diffusing independently and simultaneously each over its local potential surface. The one-particle system must diffuse along the reaction coordinate, the entire distance from unfolded to native state. Each of the N particles needs to diffuse only its contribution to the whole-protein distance: in the case considered each particle contributes $1/N$ of the distance. We find that the one-particle diffusion is much slower than the N-particle model.

Fig. 9.2 shows the comparison of the simulation results with different fits. The square fit is needed to make the MFPT of N-particle diffusion the same as one-particle diffusion. Compared to the square fit, the square root fit is closer to the simulation results.

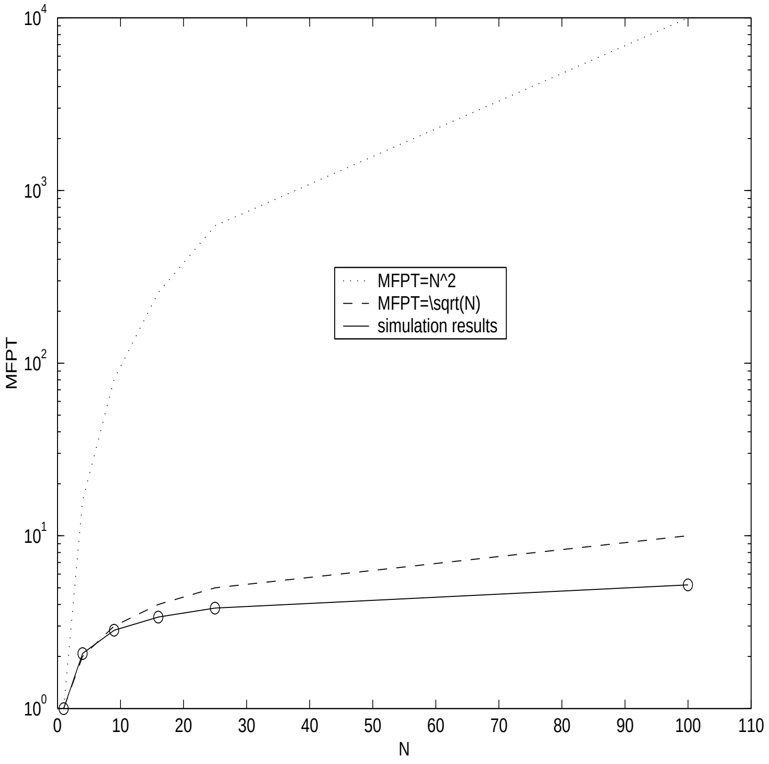


Figure 9.2. The comparison of the simulation results with different fits.

Though the square root is a better fit than the square fit, it still can not fit the simulation results well. A better fit is $\log(n)$, as shown Fig. 9.2. The larger n is, the better the log fit, as shown in Fig. 9.3. In this figure we choose $n = 100$ as the reference point: $MFPT_n = MFPT_{100}\log(n)/\log(100)$. Between $n = 25$ and $n = 100$, the error between the simulation results and log fit is less than 5%.

If the distribution of the FPT for the one particle diffusion problem is $p(t)$, then MFPT is $MFPT = \int_0^\infty tp(t)dt$. For any kind of problem, once we know $p(x)$ we can calculate the corresponding MFPT. We define $P(t) = \int_0^t p(x)dx$, where $P(t)$ is the probability that the MFPT

N	MFPT
1	1.000
4	2.0833
9	2.8290
16	3.3807
25	3.8160
100	5.1874

Table 9.2. $\langle FPT_{N-particle} \rangle$ for different N, calculated using the integral in (9.2.

is less than t. For the N-particle diffusion problem, the probability that the largest FPT is less than t is $P_N(t) = P(t)^N$. Then the probability distribution for $p_N(t)$ is $p_N(t) = \frac{dP_N(t)}{dt} = Np(t)P(t)^{N-1}$. If we plug $p(t) = \exp(-t)$ into the expression, we have:

$$MFPT_N = N \int_0^\infty t \exp(-t)(1 - \exp(-t))^{N-1} dt \tag{9.2}$$

Numerical calculation of this integral, shown in Table 9.2 gives the same result as the simulation results shown in Table 9.1. This is another proof to show that the simulation is correct.

It seems that the one particle diffusion formulation, which uses the free energy of the entire protein versus end-to-end length as the diffusion potential does not well simulate the actual protein folding procedure. Of course, this conclusion depends upon the assumptions. The first assumption is that the amino acid components fold independently. Therefore, all of them are folding simultaneously. At the opposite extreme is the situation in which the components fold sequentially one-by-one. Even if there are N components, at any time there would be only one component folding. This would be equivalent to the one-particle diffusion problem. In this case, the one particle diffusion would well approximate the folding procedure.

We would expect that protein folding falls somewhere between the totally-independent procedure and the sequential procedure. Perhaps, the whole procedure can be divided into several phases. In one phase, some of the components fold independently, while the others keep still. In

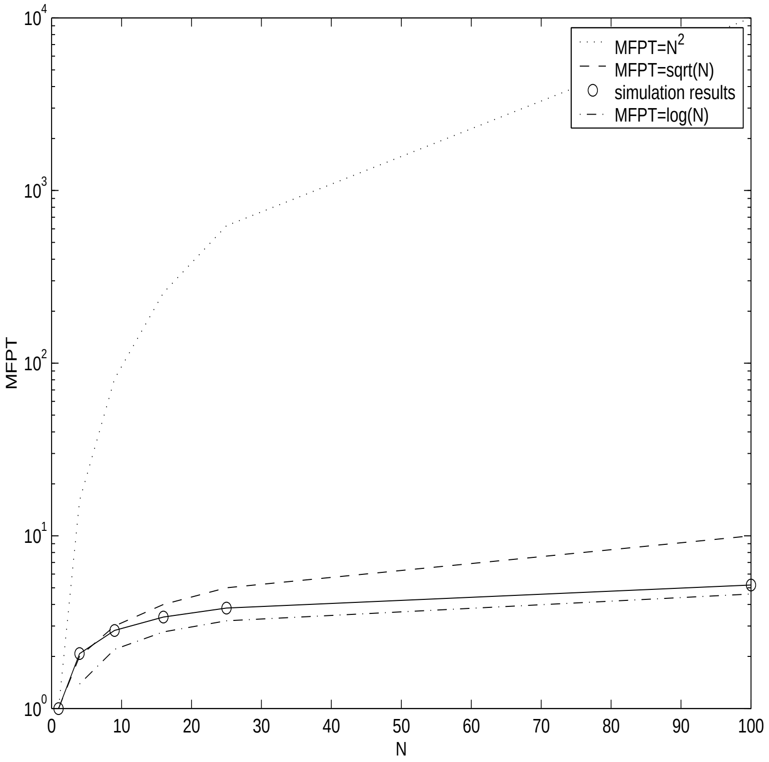


Figure 9.3. The log function is a better fit than the square root.

the next phase, some different components take part in the folding procedure. Then, the actual folding might lie between the one-particle and N-particle procedures.

The second assumption we made is that all the components are identical. If they are not, then the slowest independent component in each phase determines the FPT. We can use one particle diffusion to simulate this component. To simulate the whole folding procedure, the one-particle diffusion model can still be used, but now the potential field is not the free energy of the entire protein. Rather, it is made up by connecting all the energy changes of the slowest component in each phase. In this case, the one-particle diffusion problem is still useful to understand the protein folding problem.

9.4.2. MFPT scales as $1/L^2$

In this section we use dimension analysis to determine the MFPT when only the length of the potential field changed. There are five variables for this problem: T (time), L (length), G (energy), $k_BT\Theta$ (temperature) and D (diffusion coefficient). There are three basic variables: T, L and M. Based on dimension analysis theory we have two dimensionless functions:

$$F_1(\frac{G}{k_B\Theta}) = 0 \tag{9.3}$$

and

$$F_2(\frac{TD}{L^2}, \frac{G}{k_B\Theta}) = 0 \tag{9.4}$$

Eqn. 9.4 can be rewritten as

$$T = \frac{L^2}{D}F_3(\frac{G}{k_B\Theta}) \tag{9.5}$$

This means, if we keep energy and temperature unchanged and only compress length L into L/N, the corresponding MFPT will change from T into T/N^2.

9.5. Why protein folding is not sensitive to external force

As discussed above, since the folding should be a combination of many simultaneous foldings, if we want to use one-particle diffusion to simulate this procedure, we should use the energy of the slowest part instead of the energy of the whole system. If we apply an external force, for example, a stretching force, this force should make the folding procedure slower. But how large an effect will this external force have? We assume the MFPT is proportional to $\exp(G/k_BT)$ as an approximation, where G is the energy difference between the initial position and the ending position in the diffusion problem. The energy change due to external force for the whole protein is $F_{external}\Delta L$, where ΔL is the end-to-end length difference between unfolded and native state. If ΔL is large, the folding procedure should be very sensitive to the external force. In experiments, it is not sensitive [**21**]. The length change between stretched and native state is about 20 nm, while the folding takes place against an applied force of in the range $15-20$ pN. How does this force change the folding time change? It we use the total protein length change, it should be about $\exp(\Delta F\Delta L/k_BT) = \exp(5 \times 20/4.1) = 3 \times 10^{10}$ times the zero-force time. This means that if the folding time under a force of 15 pN is 1 second, then under 20 pN, the folding time should change into about 1000 years. This is not what is seen in the experimental results.

The experimental results are much less sensitive than this, whereby in the range $15-20$ pN in external force, the folding times vary only several times. All the experimental results show that the folding times are of the order of 10 seconds. This is because here we should not use the length of the whole protein, we should use the length of the slowest part. Therefore instead of ΔL, we should use Δl, the length change of the slowest part, which in general should be shorter than ΔL. For a change in folding time of only several times when force varies by 5 pN, we should have $\Delta F\Delta l/k_BT \approx 1$, which means the Δl should be the order of 1 nm. The length of one amino acid is 0.38 nm, so this length is about 2-3 amino acids length when aligned in a straight line. If the amino acids are not aligned, we use the FJC model to calculate the average length projected in the force direction. For a force of 15-20 pN, the projected length is about 0.15-0.19 nm, that is 5-6 amino acids in 1 nm. This is about the length between two interacting amino acids in an α helix.

CHAPTER 10

The effects of external force on the folding sequence

The simultaneous model can answer one question that the two-state model can not, “Why is the folding time not sensitive to the external force?” One might think that we need to change the two-state model only a little: wherever it comes to using Δx, we merely decrease Δx, and all the problems disappear. Is this true?

In section 6.19, it is shown that not only is Δx small compared to the total protein length change, but also it itself is a function of the value of the external force. What do these observations imply?

The prevalent point of view is that the external force changes the free energy diagram from $G(x)$ into $G(x) + Fx$, as shown in Fig. 10.1 [**4**, **25**, **43**]. That is, the force-free energy landscape is shifted by a linearly increasing amount. Is this always true?

In this section, an example is formulated to illustrate that the application of an external force can change not only the height of the energy barrier, but also the folding sequence. Therefore, when an external force is applied, the free energy diagram is not just slightly perturbed, as shown in Fig. 10.1. Rather, a totally new free energy diagram is required. This radical change also explains, as we show below, why Δx will vary under different applied forces.

10.1. A simple model to simulate the folding procedure

Here, a simple protein model is formulated. We assume that reaching the native state requires traversing three transition states and the formation of three bonds, as shown in Fig. 10.2. We might interpret this model as arising from free energy increases (forming the energy barriers)

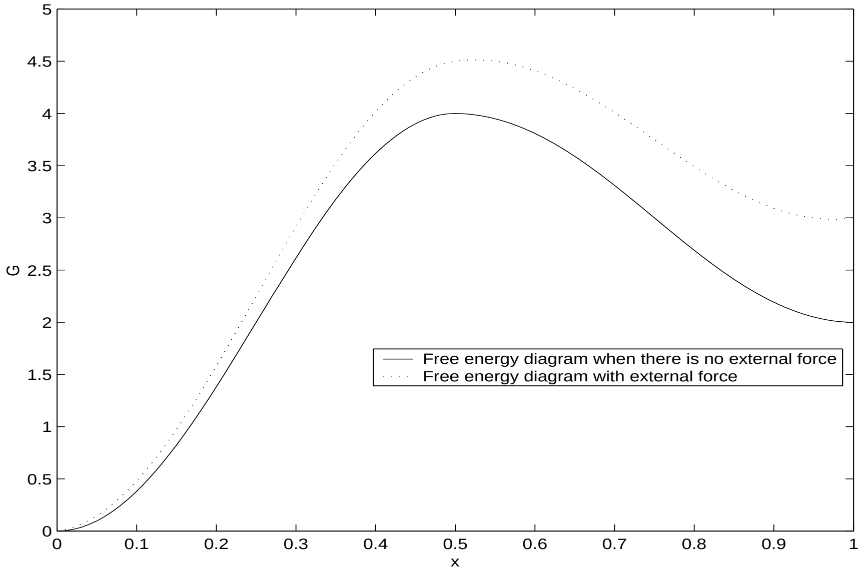


Figure 10.1. The free energy $G(x)$ in the absence of an applied force is different from that when there is an externally applied force. The usual view of how the force affects $G(x)$ is shown here. Depending on the sign of the force, the force-induced shift may be to either speed up or slow down the rate from one state to another. Here we show a force acting so as to slow the rate of the reaction which proceeds from left to right. In this section, we show that this view is not always correct.

due entirely to entropy (i.e. only from the contribution $-TdS$ to the free energy). The free energy decreases may be assumed to be due entirely to the formation of bonds (i.e. only from the contribution dU to the free energy). We also assume that once a bond is formed, it will not be broken again. The distance between **A** and **a** is 3 (nondimensionalized units of length). To bring **A** and **a** together, the entropy change will cause a free energy increase of $3k_BT$; **B** and **b** with distance 2 yields an entropy increase of $2k_BT$; **C** and **c** with distance 1 yields an entropy increase of $1k_BT$. Amino acid **A** forms a $3k_BT$ bond with amino acid **a**; **B** forms a $2k_BT$ bond with **b**; **C** forms a $1k_BT$ bond with **c** (Fig. 10.2).

We further assume that the fastest folding sequence will be chosen as the actual folding choice. The question becomes, when there is no external force, which folding sequence is fastest?

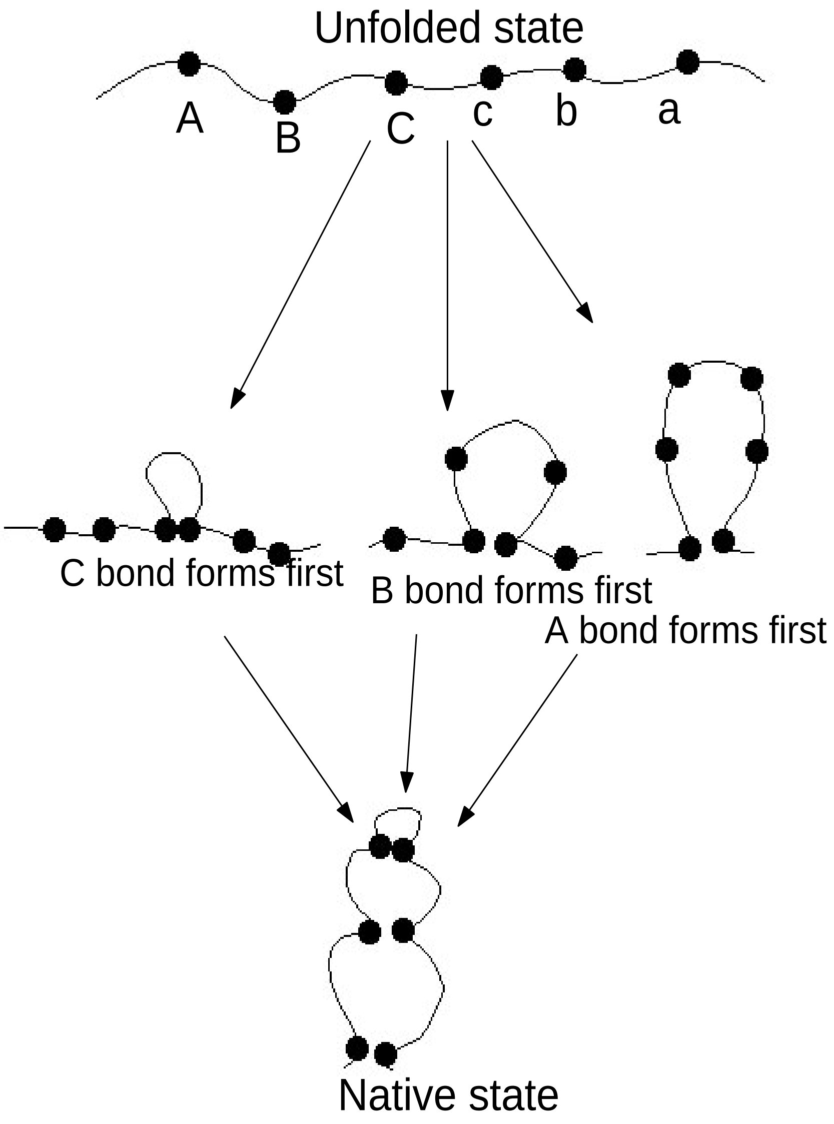


Figure 10.2. The toy protein used in the simple model. Three pairs of amino acids form bonds to achieve the native state. Here we show possible folding orders: **C-c** forms first, or **B-b** forms first, or **A-a** forms first.

First, we list out all the possible choices. **A** could finish folding first, then **B**, then **C**. The free energy diagram for this procedure will increase $3k_BT$ then decrease $3k_BT$, and increase $2k_BT$ then decrease $2k_BT$, and increase $1k_BT$ then decrease $1k_BT$. We depict this sequence as $A \rightarrow B \rightarrow C$, as shown in Fig. 10.3. We now choose the cos function with which to formulate the folding free energy profile, as was done in Sec. 8. Similarly, we can have the other sequences $C \rightarrow B \rightarrow A$ (shown in Fig. 10.4), $A \rightarrow C \rightarrow B$, $B \rightarrow A \rightarrow C$, $B \rightarrow C \rightarrow A$, and $C \rightarrow A \rightarrow B$.

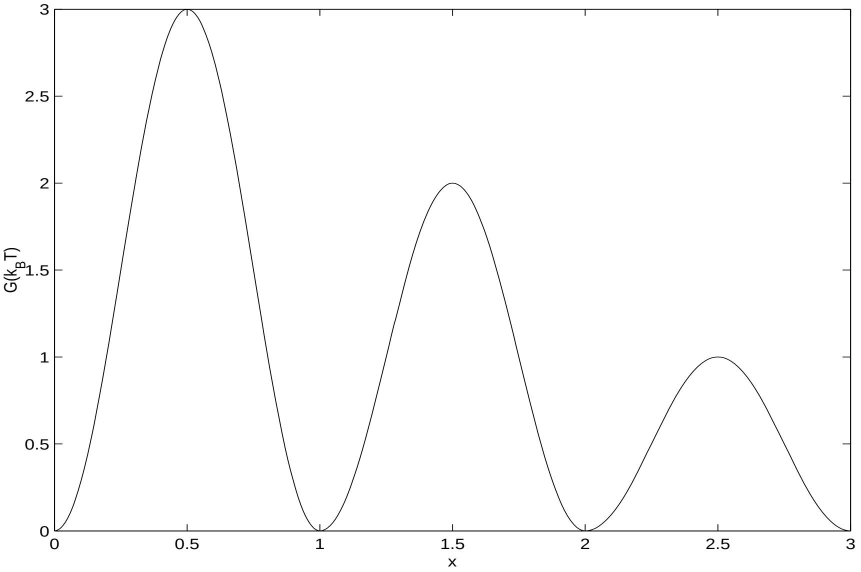


Figure 10.3. Free energy diagram $G(x)$ for the sequence $A \rightarrow B \rightarrow C$.

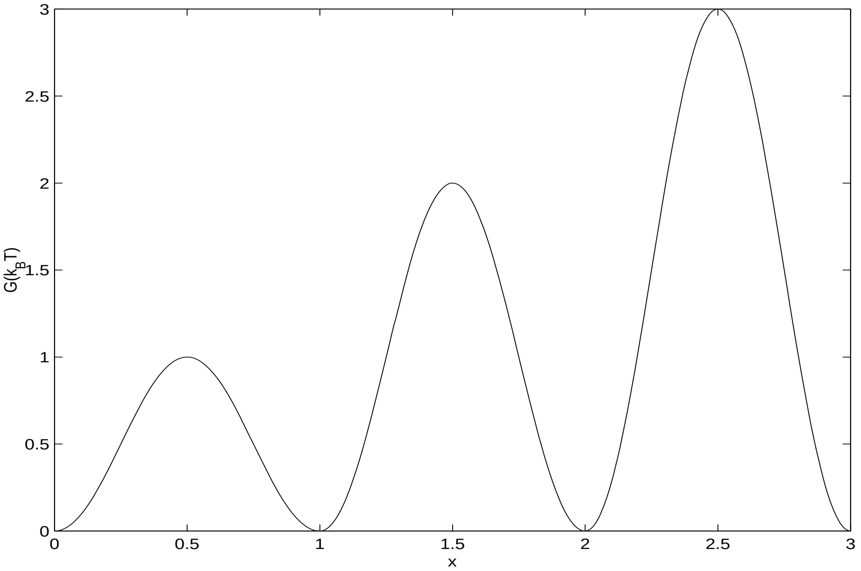


Figure 10.4. Free energy diagram $G(x)$ for the sequence $C \rightarrow B \rightarrow A$.

It is also possible that two bonds form at one time. Say, at the first step, **A** and **B** form together. Then, the free energy diagram will increase by $5k_BT$ and decrease $5k_BT$, then increase $1k_BT$ and decrease $1k_BT$. The free energy diagram is shown in Fig. 10.5. We call this sequence $AB \rightarrow C$. Similarly we also have sequences $C \rightarrow AB$, $AC \rightarrow B$, $B \rightarrow AC$, $BC \rightarrow A$, and $A \rightarrow BC$.

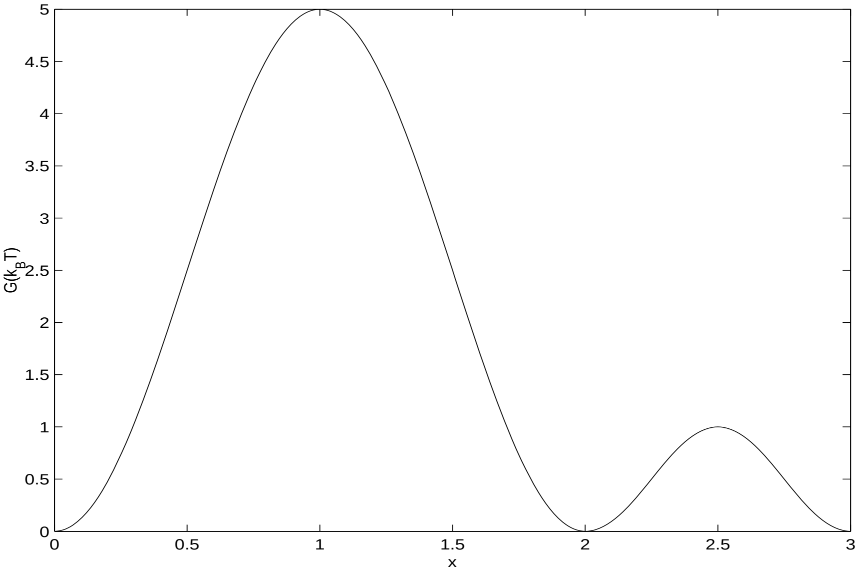


Figure 10.5. Free energy diagram $G(x)$ for the sequence $AB \rightarrow C$.

The final possibility is that all three pairs are waiting for each other to form bonds simultaneously. This is the molten globular model. The free energy diagram is shown in Fig. 10.6.

10.2. The simulation results without external force

We hope to know which sequence folds fastest. Table 10.1 summarizes the simulation results. Sequences like $B \rightarrow C \rightarrow A$ (three small steps one-by-one, in any order) yield the fastest folding.

No matter whether **A**, **B** or **C** forms first, all give the same folding time. Long range diffusion

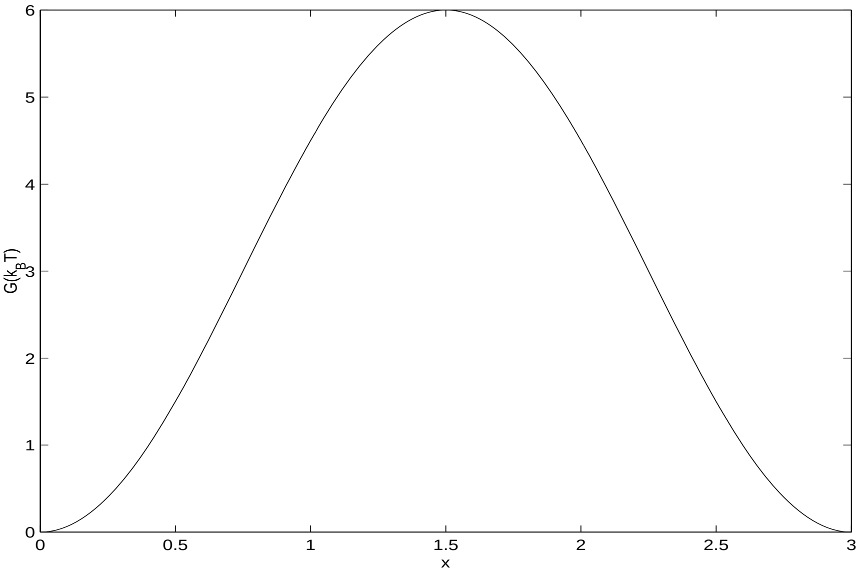


Figure 10.6. Free energy diagram $G(x)$ for the sequence ABC.

Sequence	MFPT
$A \to B \to C$	2.735
$A \to C \to B$	2.735
$B \to A \to C$	2.735
$B \to C \to A$	2.735
$C \to A \to B$	2.735
$C \to B \to A$	2.735
$AB \to C$	22.20
$C \to AB$	22.20
$AC \to B$	11.19
$B \to AC$	11.19
$BC \to A$	6.783
$A \to BC$	6.783
ABC	107.22

Table 10.1. MFPT for different folding sequences without external force.

(**A**) yields the same MFPT as the short range interaction **C**. In next section, we will see how the presence of an applied force changes this conclusion.

The molten globular state **ABC** gives the slowest folding pathway. The general rule for this simulation is: breaking one large peak into small peaks always accelerates the folding procedure.

For example, $A \rightarrow B \rightarrow C$ is faster than $AB \rightarrow C$. This conclusion is not limited to this particular protein model and the parameters used here. It is a general phenomena. For all the simulations tried, if one keeps the total diffusion distance unchanged and keeps the sum of the peaks' height unchanged, breaking one higher peak into smaller peaks always accelerates the folding procedure. This implies that the molten globule state should be the last choice as a folding pathway. The most efficient choice should be to form bonds whenever one has the chance.

To see how force changes the folding sequence, we now consider the folding procedure with external force.

10.3. Simulation results with external force

First we need to specify how the force affects the magnitude of the free energy changes during folding. We assume that the effect of force is greater for greater diffusion distances. So, for the diffusion of **A**, because it has the longest diffusion length, force causes the energy barrier to be $3k_BT$ higher. And, similarly, $2k_BT$ higher for **B** and $1k_BT$ higher for **C**. But, this energy barrier increase is sequence dependent. If **A** forms first, **B** and **C** will be protected from the external force and therefore their energy barrier will not change. So for the sequence $A \rightarrow B \rightarrow C$, the energy barrier will become $6k_BT$, $2k_BT$ and $1k_BT$ (shown in Fig. 10.6). If **B** forms first, the height of the **C** energy barrier will not change, but **A** can still feel the external force. With **B** already formed, **A** does not need to diffuse as far as when it diffuses first. The increase for **A** will become $1k_BT$. If the folding sequence is $B \rightarrow A \rightarrow C$, the energy barriers will become $4k_BT$, $4k_BT$ and $1k_BT$ (as shown in Fig. 10.8). Similarly we can construct the other sequences.

The MFPTs for folding with two or three bonds forming simultaneously, such as $AB \rightarrow C$ or ABC are not listed here. They are always longer than bond formation one-by-one.

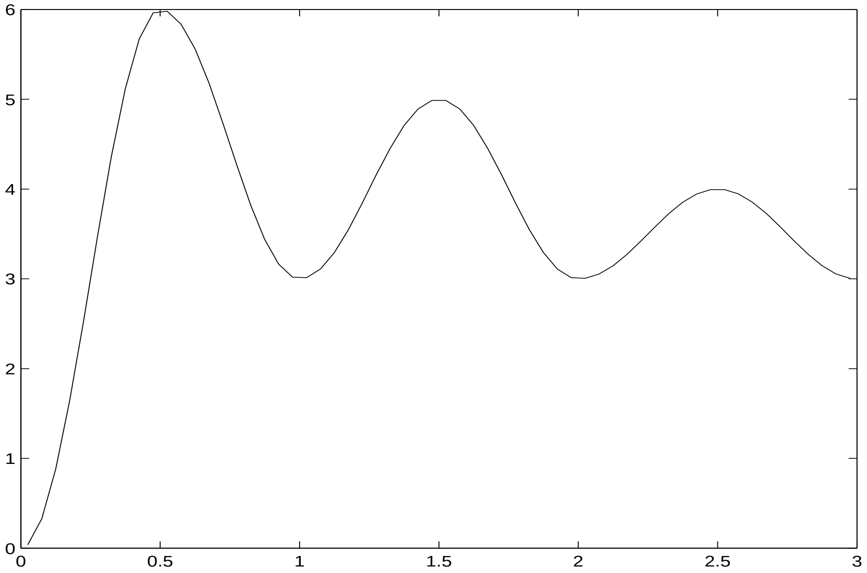


Figure 10.7. Free energy diagram for sequence $A \rightarrow B \rightarrow C$ with external force.

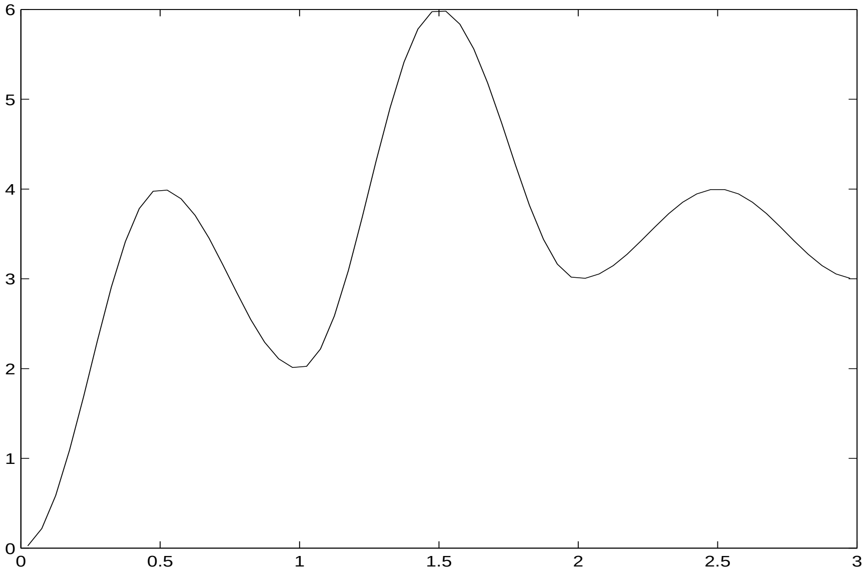


Figure 10.8. Free energy diagram for sequence $B \rightarrow A \rightarrow C$ with external force.

We can see from Table 10.2 that folding order matters this time. $C \rightarrow B \rightarrow A$ gives the fastest folding. This means, when there is external force, short-range diffusion should happen first, then the middle-range diffusion, then long-range diffusion.

Sequence	MFPT
$A \to B \to C$	16.32
$A \to C \to B$	16.32
$B \to A \to C$	6.7076
$B \to C \to A$	6.7076
$C \to A \to B$	8.1792
$C \to B \to A$	5.391

Table 10.2. MFPT for different folding sequence with external force F. (To gain an idea of approximately how large this force might be, if the distance between C and c is 1 nm, then the force is approximately 4 pN.)

Sequence	MFPT
$A \to B \to C$	221.5
$A \to C \to B$	221.5
$B \to A \to C$	24.34
$B \to C \to A$	24.34
$C \to A \to B$	38.68
$C \to B \to A$	11.58

Table 10.3. MFPT for different folding sequence with larger external force $2F$. (To give an idea of approximately how large this force might be, if the distance between C and c is 1 nm, then force is approximately 8 pN.)

When we double the force from F to $2F$, the MFPTs for different sequences are shown in Table 10.3. At the higher force, the advantage of sequence $C \to B \to A$ is made clearer. For external force F, the second fastest sequence takes time 24% longer than the fastest sequence. While for external force $2F$, the second fastest sequence takes 110% longer. The larger the external force is, the greater the advantage for short-range diffusion to occur first.

This gives us a picture of the protein folding procedure under external force. Large force eliminates long-range diffusion, makes the protein form neighboring bonds first, and then with the help of these bonds, initially long-range diffusion becomes easier to finish.

10.4. Discussion

“Why is the folding time not sensitive to the external force?” In section 6.19, it is shown that not only is Δx small compared to the total protein length change, but also it itself is a function of the value of the external force.

“Why is Δx a function of the applied force?” The answer to this question is that Δx decreases when the external force increases because a larger force eliminates long-range diffusion more efficiently. The MFPT increases exponentially as $\exp F\Delta x/k_BT$. So, folding through a length Δx is made much faster by dividing a single change in Δx into many smaller changes. Another point of view, to explain this result, is that diffusion over long distances without the aid of bonding, becomes exponentially less probable as force is increased. For larger external forces, shorter-range diffusion becomes more and more important. Therefore, if we use some kind of “average” Δx to describe the folding procedure, it will become shorter and shorter, until it reaches a physical limit. This limit might be the length of a single amino acid.

“Does the application of an external force result in only the linear shift of the energy landscape (as suggested by Fig. 10.1).” For this question, the answer is that force does not only change the height of the energy barrier. It can change the sequence of the folding procedure as is already apparent from the answer to the previous question.

How much of a change there is in the folding sequence depends on both the magnitude of the external force and the protein folding sequence when there is no external force. For example, if naturally (without an external force) in the protein folding sequence, the C-terminus comes to the N-terminus to form a bond at the first step, then a small force will eliminate this step. On the other hand, for short range interactions such as the formation of an α helix, the folding sequence might not be affected by the same force. It can be seen then, that a parameter which might describe the force-sequence effects quantitatively is $\exp(\bar{s}F/k_BT)$. Where $\bar{s}$ is the average distance between bonded amino acids (excluding peptide bonds) in the native state.

It is interesting to note that $\bar{s}$ is equal to the contact order times the contour length of the protein [**27**].

10.5. One misunderstanding about two-state model

One widely accepted diagram for the two-state mode is the solid plot shown in Fig. 10.1: two local minimum (unfolded state and native state) are separated by an intervening peak. With a certain rate, the protein can change from one state to the other. There is actually one hidden assumption for this description: the folding pathway is the reverse of the unfolding path way. Is this true? In general, no. We can use the three-bonds protein model introduced above as an example. To unfold the tight and compact native protein, the unfolding should break the three bonds simultaneously, and then all the amino acids will be free to move. This unfolding diagram is like that shown in Fig. 10.6: overcome one high energy barrier, then go to the unfolded state. For the folding procedure a long loose structure changes into a tight one. As discussed in the folding simulations, and shown in Fig. 10.4, folding via multiple peaks is a faster folding pathway than forming all three bonds simultaneously. Thus the folding and unfolding pathways are different.

Generally, then, the folding pathway is not the reverse of the unfolding pathway and can not be described on the same diagram simply by moving in the opposite direction.

Only one case is an exception: the molten globule model. For this model, the folding procedure is that the amino acids do not form bonds until all of them find the correct position. Once they are positioned near the native state, then bonds form at the same time. This is just the reverse of the unfolding in which all the bonds break simultaneously, and the amino acids then go to their unfolded positions.

CHAPTER 11

Simulation based on real protein: RNase H

The following section is based on a paper submitted for publication. Hence, its style is somewhat different from the previous sections. In this section we apply the ideas of the simple ABC model in Sec. 10.1 to develop a more sophisticated model and apply it to a real protein, RNase H. We show that by using this Individual Bond (IB) model, we can solve some puzzles: Why can proteins still fold fast with an applied external force? Why does the characteristic folding length decrease when force increases? Why are there small accumulated changes in extension before a large collapse happens?

11.1. Abstract

To become operational, proteins fold from their nascent extended conformation into a compact form. Folding is often described by the two-state model in which the protein is considered either folded or unfolded. When stretched by an applied force, the two-state model predicts that the free-energy landscape will be tilted and folding times will depend sensitively on applied force. We show that under an external force, the two-state model is inconsistent with measurements of folding times and folding pathway. Using RNase H as an example, we present a model in which folding is comprised of smaller motions individually acted upon by the applied force. The model naturally explains how cooperativity arises when an applied force is present and why observed folding times become less sensitive to the external force as force increases, while the two-state model predicts the opposite trend.

11.2. Background

Proteins assembled by the ribosome as linear chains of amino acids subsequently fold into complex three-dimensional functional forms. To study folding in the laboratory, single protein molecules are first unfolded by the application of a high force using atomic force microscopy or laser tweezers [**5**, **8**, **11**, **13**, **21**, **26**, **39**, **43**, **49**]. The force is then reduced in strength allowing folding to ensue. It has been suggested that folding under a finite applied force can be used to investigate states which otherwise are too transitory to be accurately observed and to investigate regions of the folding landscape which might otherwise be inaccessible [**24**].

It is important to determine if folding under an applied force illustrates the force-free folding process or if force alters the folding pathway. The two-state model postulates a high-energy barrier separating the folded and unfolded states in which the protein will be observed, Fig. 11.1A. According to two-state dynamics, the effect of an externally applied force F is to accelerate the force-free unfolding rate from α_0 to $\alpha_0 \exp(FX_u/k_BT)$, and to slow the zero-force folding rate from β_0 to $\beta_0 \exp(-FX_f/k_BT)$ [**43**].

Protein	ΔL (nm)	X_u (nm)	X_f (nm)	Source
ubiquitin	24	0.25	23.75	[**12**]
ankyrin	12.4	1.7	10.7	[**30**]
spectrin	31.7	1.5	30.2	[**47**]
FN-III	28.5	0.3	28.2	[**38**]
titin	25-28	0.3	24.7-27.7	[**44**]
projectin	27	0.17	26.8	[**9**]

Table 11.1. Lengths based on the two-state model. Using the length difference between folded and unfolded states ΔL and the distance between folded and transition states X_u, it is found that the folding distance $X_f = \Delta L - X_u$ between the transition and unfolded states is typically much larger than the unfolding distance X_u [**13**]. (If it is not reported, we use the maximum value $X_u = 1.7$ nm in computing X_f.)

The folding length scale X_f is typically much larger than X_u, Table 11.1. Consequently, according to the two-state model, an applied force tremendously slows folding. For ubiquitin,

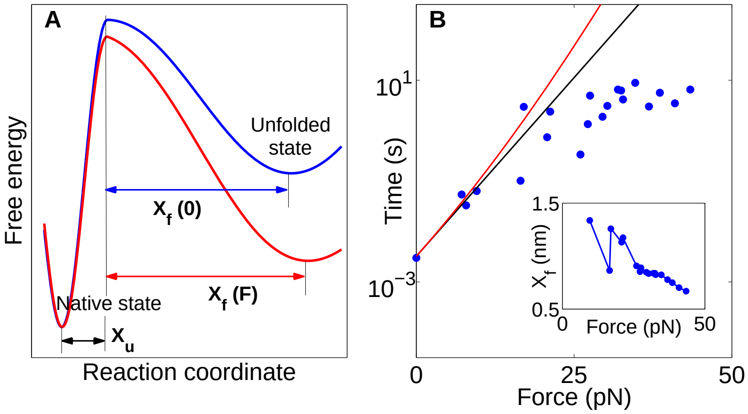


Figure 11.1. (**A**) Free energy versus end-to-end length for the two-state model. X_u (X_f) is the distance from the folded (unfolded) state to the transition state. The free energy for zero applied force, blue line, and for an applied force, red line. Force tilts the profile by a linear amount FX and lengthens X_f [**13**]. (**B**) Folding times as a function of applied force. Assuming $T(F) \propto \exp(FX_f/k_BT)$, a constant X_f yields a straight-line dependence, such as shown by the black line. The two-state model predicts that X_f increases with applied force as shown in (A), which yields a faster than linear increase, as shown by the red line. (**Inset**) However, fitting X_f through an increasing number of data points (blue dots) for the folding times of ubiquitin (from [**21**, **27**]) from the left-most four through to all points, shows that X_f decreases as a function of force.

the zero-force folding time is approximately 0.003 s [**27**]. Under an applied force of $F = 15$ pN, the folding time would become $0.003 \times \exp(15 \times 23.75/k_BT)$ or approximately 10^{22} years. Yet, that ubiquitin is observed to fold in seconds [**21**] suggests that folding does not proceed in an all-or-nothing two-state jump over a large length scale, but rather is broken down into smaller steps with $X_f << 20$ nm [**31**, **49**]. In Bullard *et al.* [**9**], although they measure $X_f \simeq 26.8$ nm considering a two-state point-of-view for projectin, they find that they must introduce the much shorter folding length scale of 1.1 nm in order for their Monte-Carlo simulation to correctly predict folding times. Moreover, though the two-state model predicts that the characteristic folding length increases with applied force (see Fig. 11.1A), measurements of folding time ([**21**] Fig. 4B) reveal the opposite trend, Fig. 11.1B.

In [**21**], the atomic force microscope is used to stretch a chain of linked ubiquitin domains. The observation ([**21**] Fig. S4A) that while remaining in the unfolded state the end-to-end length steadily decreases, does not fit with the two-state model which predicts an all-or-nothing transition between states. Rather, the steadily decreasing length suggests that there is an accumulation of small-scale transitions. Similarly, and as we will discuss in detail, antecedent to a large-scale transition, the refolding of RNase H also shows an accumulation of small-scale changes indicating a deviation from the two-state model [**13**].

11.3. Individual bond model

We introduce a minimalist Individual Bond (IB) model of folding under applied force, which is applied to the folding of RNase H, see Sec. 11.7. Using structural data, 100 hydrogen-bonded pairs of amino acids in the native state are identified. These pairs of amino acids undergo *interactions*, where at each time step, the probability to form a bond is proportional to $R_0 \exp(-F x_{rs}/(k_B T))$, where R_0 is the zero-force folding rate, and extension x_{rs} is the contour length between amino acids r and s projected onto the direction of the applied force by using the worm-like chain model. As each bond forms, the extension of other ongoing, not-yet bonded interactions may change, Fig. 11.2. All interactions occur simultaneously [**19**, **40**, **42**, **53**, **54**], and the applied force acts on the interactions individually [**20**]. The effect of force is equivalent

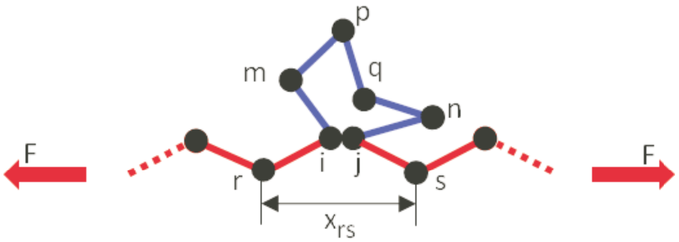


Figure 11.2. IB model geometry. When a bond forms between amino acids i and j: (1) The interacting pairs of amino acids internal to the bond (shown in blue) which have not yet bonded, such as m and n, are shielded from the applied force F and continue their folding subject to $F = 0$. (2) The contour length for all interactions external to the bond (shown in red), such as between r and s, are shortened by eliminating the extension length of the amino acids now internal to the bond, see Sec. 11.7.

to tilting the energy profile as shown in Fig. 11.1A, but with the crucial distinction that rather than being applied to the protein as a whole, as in the two-state model, the tilt is applied to each interaction individually. The same zero-force folding rate R_0 is assumed for all bonds. This equal-rate assumption may not be realistic, but we use it to emphasize that the IB model is not biased, by pre-selecting folding rates, toward any particular length scale. Rather, as we will see, changes in length scale and sequence are due to force-induced cooperativity.

11.4. Results

By running the IB model with an applied force which is decreasing in time, we can mimic the laser-tweezers experiment of Cecconi *et al.* [**13**] who measured changes in length of RNase H, Fig. 11.3A. Both the experimental and simulated results show similar folding behavior, Fig. 11.3. Note that the experiments measure extensional length, while our model yields contour length. Above approximately 15 pN, the length fluctuates about a constant value in the measurements, while in the simulation, the length also is nearly constant. Near 15 pN, both the experimental and simulated curves change slope as the length begins to gradually decrease. At a force of approximately 5.5 pN in the experiments and slightly lower in the simulation, there is

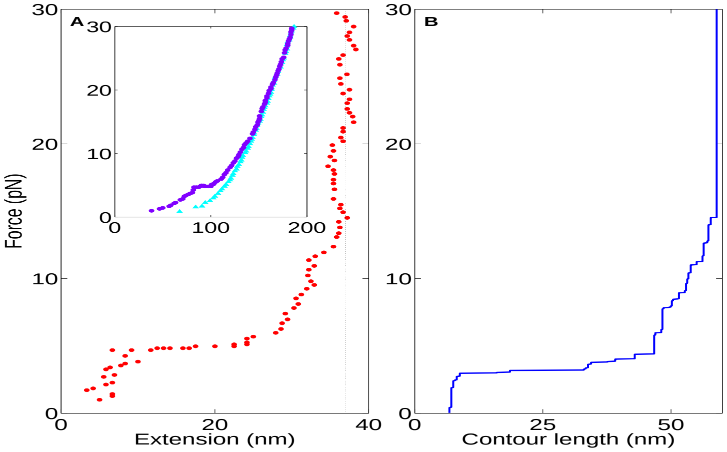


Figure 11.3. Comparison of experimental and simulation results. (**A(Inset)**) Experimental results digitized from [**13**]'s Fig. 1, the magenta circles mark the force versus extension curve for RNase H protein + DNA handles. The cyan triangles are the force-extension curve for the DNA handles alone, shifted 37 nm along the extension axis to show that above approximately 15 pN, as they are changing length at the same rate, the two curves have nearly identical slopes. Near 15 pN the curves deviate as the slope of the RNase H + DNA curve decreases due to RNase H folding. Near 5.5 pN, RNase H undergoes a large decrease in length. (**A**) The difference in extension between the unshifted curves in the Inset gives the force-extension curve for RNase H alone. (**B**) A typical simulation result of contour length as a function of force from the IB model.

a precipitous decrease in length. In the simulation, this large decrease in length can be seen to come not from the formation of one bond, but from many steps in rapid succession [**31**].

11.5. Cooperativity and folding sequence

In the IB model, the formation of a bond leads to two forms of cooperativity which tend to accelerate the formation of other bonds: amino acids internal to a bond are shielded from the applied force F, and amino acids external to a bond have their interaction distance shortened, Fig. 11.2. This force-induced cooperativity yields two distinctive types of pathways, sequential folding at high forces and simultaneous folding at lower forces. Under large constant applied forces, long-range interactions can not directly form bonds. Short-range interactions bond most easily, decreasing the length of moderate length interactions so that they can subsequently bond, all the while bringing closer together the longer-range interactions until they too can bond. This is a zipper-type mechanism [**53**]—not operating through the entropic or enthalpic contribution to the zero-force free energy, but—due to sequential small-scale changes to the work term Fx_{ij}. Folding under a large applied force can then appear as a steady accumulation of small changes in length.

Cooperativity can also lead to large abrupt changes in length. Consider, as a toy example, a protein which has five short-range interactions, each of length 1 nm, and three long-range interactions with lengths 20 nm, 30 nm, and 40 nm. The short interactions are assumed to lie internal to the long ones. If the protein is subject to a decreasing force, initially, the force is so high that even the short interactions have no chance to bond. As force decreases, the five shortest interactions will form bonds first. Consequently, (assuming for simplicity that the length between bonded amino acids is zero) the contour length for the three long-range bonds becomes 15 nm, 25 nm, and 35 nm, respectively. Even with the aid of the short-range bonds, the long-range interactions may still not be sufficiently shortened to bond. The force keeps decreasing until the shortest remaining interaction, 15 nm, forms a bond, reducing the contour lengths of other two interactions to 10 nm and 20 nm. Since under this force the 15 nm interaction can bond, the shorter 10 nm interaction will quickly finish bonding. This bond reduces the remaining unbonded length to 10 nm, which too, will rapidly bond. So, for the three long-range interactions, once the shortest one forms, all the others will form in rapid succession. To an observer, this can appear as one-step behavior.

Both types of cooperative behavior are seen in RNase H, Fig. 11.3A: sequential bonding beginning near 15 pN leads to a gradual reduction of length, preceding the nearly simultaneous folding at 5.5 pN. We can more closely follow the cooperative folding by manually pausing the

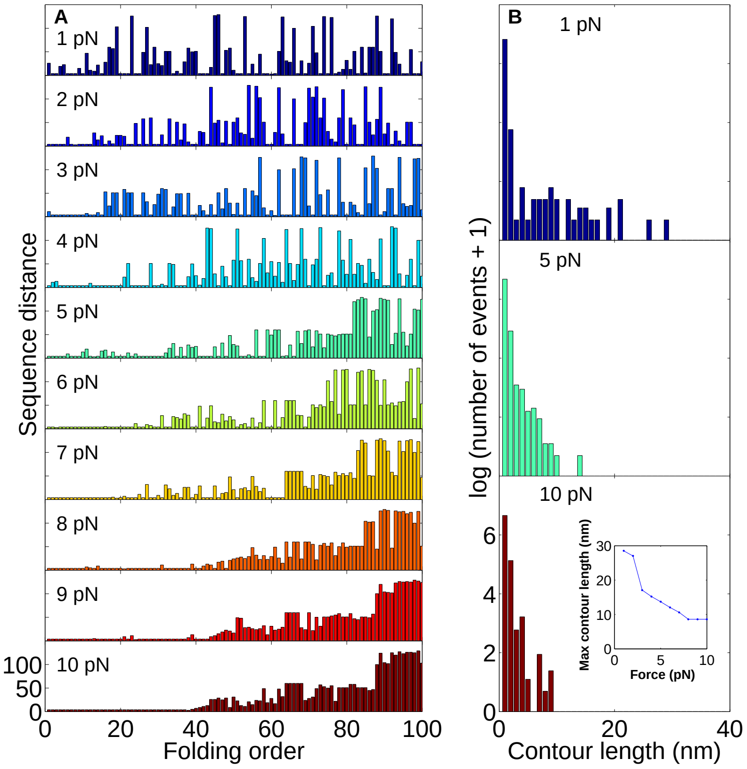


Figure 11.4. Constant force simulations for the IB model of RNase H. (**A**) For each applied force $F = 1, 2, \ldots, 10$ pN, as each bond forms, its initial separation along the protein in terms of its sequence distance is plotted in order of formation from 1 to 100. (**B**) At each of the three forces $F = 1, 5, 10$ pN, the number of occurrences, summed over ten simulations, in which the contour length changes by a given amount. (**Inset**) The value of the maximum change in contour length as a function of applied force.

simulation and querying the output (not shown). We find that by $F = 6$ pN, most of the short-range bonds within the α helices and antiparallel β sheets have folded leaving long-range interactions with lengths from 7 to 39 nm. The shortest of these, the bond between helix 1 and helix 2, forms first, decreasing the contour length by 7 nm, and setting off the swift decrease in length seen at about 5 pN in Fig. 11.3B.

Fig. 11.4 shows results from running the IB model as a constant-force simulation. Fig. 11.4A shows that there is a continual change in folding order under applied force. The order of bonding between amino acids which are close versus far in sequence separation becomes increasingly graded by size. (The shortest sequence distance, other than one β sheet interaction of length three, is equal to four, for interactions within α helices; longer distances (the longest is 129) are generally due to interactions between different secondary structures.) At the lowest force $F = 1$ pN, there is no apparent preferred folding order, and the longest sequence distances > 100 disperse throughout the folding order alongside the shortest. As the force is increased, $2 \leq F \leq 4$ pN, the longest distances are delayed till later in the folding order $\overset{\sim}{>}40$, while the shortest and intermediate distances remain interspersed throughout the folding order. For $F \geq 8$ pN, all of the shortest distances occupy a nearly continuous streak, folding first in order. They are followed by longer sequence distances graded in distance, from short to long.

Contour length between any particular pair of interacting amino acids continually changes as other pairs bond, Fig. 11.2. The final change in contour length between an amino acid pair comes when it itself forms a bond, Fig. 11.4B. At the lowest force, $F = 1$ pN, long interactions > 20 nm can occur. As force increases, interactions through long distances have little chance to form directly, but rely on shorter interactions to form first in order to incrementally decrease their contour length. At $F = 5$ pN, bonds form directly only between amino acids separated by less than 20 nm. At $F = 10$ pN, the distribution of contour lengths which directly bond is shifted to even shorter lengths, < 10 nm. The decrease, with increasing force, in the value of the largest contour length change which occurs during folding is accompanied by a shift of the entire distribution of contour lengths to shorter lengths. This suggests that the length scale which should be used to characterize folding will decrease as force increases. This is consistent with measurements from ubiquitin [**21**] which show that the length scale X_f determined from folding times decreases as force increases, Fig. 11.1B(Inset).

11.6. Conclusions

Similar to Gō models, once a bond is formed in the IB model, it can not break. Furthermore, the IB model is strictly one-dimensional, accounting for changes in length along the one-dimensional reaction coordinate. As a protein folds, its three-dimensional geometry is constructed, yielding an increasing contribution from bending and torsional stiffness. Bond breaking and three-dimensionality, ignored in the IB model, are undoubtedly needed to describe other aspects of folding, such as bistability [**13**, **32**].

The main insight of this paper is that applied force does not act directly on the unfolded state, but rather acts individually on the many interactions which collectively form the protein's instantaneous state. Several consequences ensue: cooperativity, incremental accumulation of length changes, and large step-like changes in length. Force differentiates states by length scale, making the folding pathway a function of applied force. Large forces favor shorter interactions and increasingly eliminate long-range interactions by sequentially pairing down their contour length.

For zero-force folding, the question of heterogeneity remains, i.e. how different portions of the protein fold at vastly different rates [**2**]. Folding under an applied force may be a window onto that question, in that it introduces heterogeneity along the reaction coordinate, and so can be directly assayed. Structures of different length scales can be tracked in time to see how and when they fold, and how they are aided cooperatively by the folding of other portions of the protein.

In vivo, proteins may be subject to external forces as they are expressed by the ribosome, in the crowded cellular environment, as they translocate through membranes, interact with substrates, and are degraded. Understanding proteins under applied force helps understand proteins' response to these stresses. The view supported here, that applied force alters the folding pathway, offers a new possibility of how misfolded proteins may arise within the cell and hence lead to diseases potentiated by misfolding [**17**].

11.7. Methods: Simulation of the Individual Bond model

We use structural data from the Protein Data Bank (PDB) for protein 1rch. Amino acids are numbered sequentially $i = 1, \ldots, 155$ from the C to N terminus. Each amino acid is ascribed to an α helix, β sheet or loop motif according to its native structure. All bonding pairs within a motif plus bonds ≤ 0.35 nm between motifs are identified. The amino acids forming these bonds are said to undergo an *interaction*. There are 100 such interacting pairs. In the unfolded state, the contour length between amino acid pair i and j is calculated by $(j - i) \cdot 0.38$ nm. As folding proceeds, the contour length changes. The contour length between amino acid i and j is simply the sum of the contour length contributions from each amino acid. The contour length contribution of one amino acid is calculated as follows: if it is inside of a formed β sheet bond, its contour length contribution is zero; if it is not inside of a formed β sheet bond and it is inside of a formed α helix bond, its contour length contribution is 0.15 nm; otherwise its contour length contribution is 0.38 nm. All the bonds that do not form within a β sheet or α helix are regarded as secondary bonds. They are treated as β sheet bonds whose contour length contribution is 0.38 nm before bonding and zero after forming a bond. The extension length x_{ij} (contour length projected along the force direction) is computed from the contour length using a worm-like chain with persistence length 0.7 nm. Once a bond is formed, it is not permitted to break.

The simulation can now be described according to the following three steps. (i) Independent rate rule: At each time step Δt, the probability of folding for any interaction between amino acids i and j is proportional to its rate $R_0 \exp(-F x_{ij}/k_B T)\Delta t$, where F is equal to the applied force unless i and j is an interaction interior to a previously formed bond, in which case $F = 0$ according to (iii). Whether or not a particular interaction form a bond is decided by a random number. All bonds are given the same zero-force folding rate $R_0 = 1.4/s$. For simulations in which force is varied, force is decreased at a constant rate over time τ, where $R_0\tau = 14$. (ii) Exterior length step: After a bond forms, amino acid pairs exterior to the bonding pair are brought closer together. For example, the bond which forms between amino acid $m = 28$ in β Strand 2 with amino acid $n = 31$ in β Strand 1 is interior to amino acids $i = 27 < 28$ and $j = 32 > 31$ whose extension x_{27-32} is then decreased by ((extension length of x_{mn}) - (interior bond length)). When the interior bond is, as in this example, part of a β sheet, its bond length is zero; bonds within an α helix are taken to be 0.15 nm. (iii) Interior force step: Amino acids inside a previously-formed bond are thereafter shielded from the applied force. That is, if amino acids i and $j > i$ form a bond, then amino acids $i < k < j$ subsequently will not be subject to the applied force as they continue to fold, Fig. 11.2.

CHAPTER 12

Conclusions to Part II

The conclusions from the simultaneous diffusion model are as follows.

(1) For long proteins such as ubiquitin, the two-state model is not valid. The length change for the folding of a single domain of ubiquitin is approximately 20 nm [**21**]. Hence, folding by the two-state model should be very sensitive to the external force. If we use this length change of 20 nm as the Δx, use 15 pN as the external force, the folding time will be $exp(15 \cdot 20/4.1) = 6 \cdot 10^{31}$ times slower. If we take the zero-force folding time as 10^{-6} seconds, then under 15 pN the folding time will become 10^{24} years. The experimental results show that this is not true.

If we use the simultaneous diffusion model to describe the folding procedure, then the sensitive force effects disappear. Instead of using the two-state model to describe the whole protein, we assume the folding procedure is composed of many small simultaneous foldings. The slowest one determines the folding time. So the force should be applied only to a small length. Using this smaller length scale, the folding time is no longer sensitive to the external force.

As a particular implementation, we further assume that the many small folding procedures comprise the formation of the secondary structures. Once the secondary structures are formed, then the tertiary structures begin to form by coalescence of the secondary structures. By assigning different folding rates for all the folding procedures, we can generate a similar length versus time cure for the single domain folding as observed by Fernandez and Li [**21**]. Fig. 12.1 and 12.2 show the simulation results and experimental results for single domain folding under constant force. The simultaneous diffusion model can reproduce the step-like behavior, which is regarded as the characteristic of the two-state model.

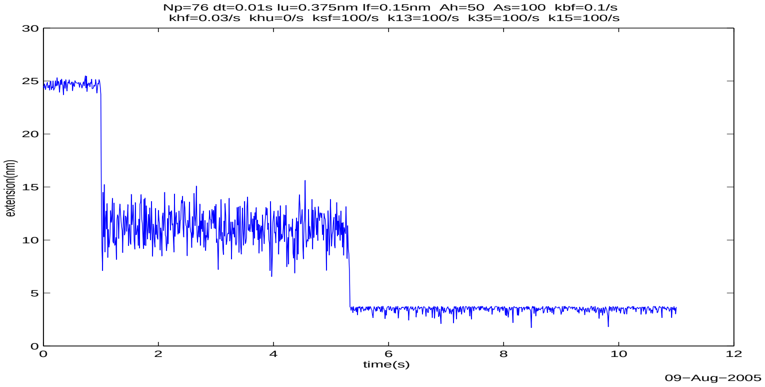


Figure 12.1. Numerical simulation for the folding procedure

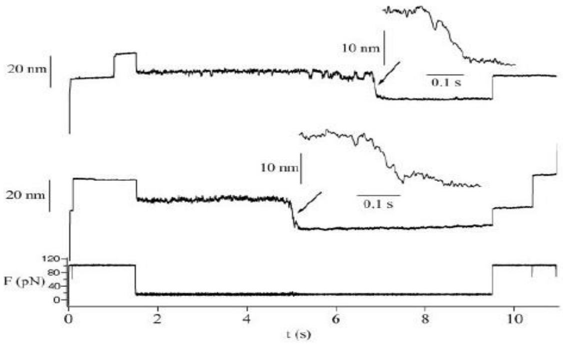


Figure 12.2. Single domain results from [**21**].

(2) The simultaneous diffusion model solves the problem of why folding time is not sensitive to external force. The folding time is due to the slowest folding component. Each such component (perhaps one of the secondary structures folding into its tertiary position) is characterized by a displacement which is only a fraction of the total change in extension. Thus, while the dependence of folding time on force is proportional to $\exp F\Delta x/k_BT$, where $\Delta x <<$ (total change in extension).

(3) Application of an external force may change the folding sequence. The application of a force discriminates against diffusion through longer distances. The energy change as a function of distance increases exponentially with distance, as noted in 2 Changes involving large distances, then, become much less probable.

(4) The length scale which governs the rate of folding under an external force is force dependent. As force increases, the probability of having a long-range interaction falls off exponentially. Consequently, as force increases, short-range interactions become increasingly prominent.

(5) The usual view is that application of an external force simply tilts the zero-force free energy profile. This would be the case if the potential experienced by the folding protein were independent of the folding process. The potential felt by any part of the protein is composed of two parts, one due to the applied force and the other due to the potential of the rest of the protein. Only the potential due to the applied force is independent of the folding process. The other part depends on the conformation of the entire protein which, as noted in 3, changes on the application of a force.

(6) The folding process is not the reverse of the unfolding process. This implies that unfolding is cooperative: the protein fails catastrophically. Folding, on the other hand, is rate limited by one of many independent processes. There are many of such processes, with smaller energies and distances. (These processes may be the diffusion of secondary structures.)

Unfolding can be seen as breaking parallel bonds [**12**]. (Observations of unfolding show that the forces needed are very large–an order of magnitude greater than the strength of a hydrogen bond.) That is, unfolding occurs by the simultaneous breaking of bonds. But the formation of those bonds, i.e. their folding, need not happen simultaneously. In fact, we have shown in sections 10.2 and 10.3, that simultaneity in folding is, in general, the slowest folding pathway. Fast folding relies on the simultaneous diffusional processes having different time scales. That is, diffusion begins simultaneously and ends sequentially.

CHAPTER 13

Appendices

13.1. Matlab code to compute MFPT

Code to compute MFPT. The code can be run to a short time T, at which time, the decrease in probability within the interval can be checked to see if the decrease is exponential. If so, the simulation can be continued analytically, as described in Sec. 8.1.1.

```
clear;

%dx=0.0125/2;       % size of one region in x
     % Check that it divides x into an integer number of sections.
dx = 0.05
aa=100;      %record the result every aa steps
T=100;        % total simulation time
dt=10^(-4); % time interval for one step. Must have dt<min(1./(F+B))
steps=ceil(T/dt);

x=-1+dx/2:dx:1-dx/2;
N=max(size(x)) ;%number of regions
kbt=1.38*3;

%%%% The potential field %%%%
% For different potentials only this part needs to change.
barrier1=5;   % the height of the first barrier (unit kbt)
depth=2.5;       % depth of the intermediate(unit kbt)

Nsections = 5       % Nsections = number of different potentials
tempn=N/Nsections;

vtest = 0*x; vtest(tempn+1:2*tempn)=barrier1*kbt;
vtest(2*tempn+1:3*tempn)=(barrier1-depth)*kbt;
vtest(3*tempn+1:4*tempn)=barrier1*kbt;
```

```
plot(x,vtest/kbt); xlabel('x'); ylabel('G(k_BT)') %break

%%%% The diffusion equation %%%%
% Set the absorbing boundary for the right end.
alphax=[x(N)-dx/2, x(N)+dx/2]; energyalpha=0*alphax;
alpha=(energyalpha(2)-energyalpha(1))/kbt;

%%%% Another example for how to set potential field %%%%%
% Since the potential field is the part varies the most, here
% I give another example for G=k_BT*cos(2*pi*x), 0<x<1. The
% following shows how to set the potential field and how to
% set the absorbing boundary
% x=-1+dx/2:dx:1-dx/2;
% vtest=kbt*cos(2*pi*x);
% alphax=[x(N)-dx/2, x(N)+dx/2];
% energyalpha=kbt*cos(2*pi*alphx);
% alpha=(energyalpha(2)-energyalpha(1))/kbt;
%%%%%% the end of this example %%%%

D=1;

% Set F (forward rates) and B (backward rates)
dv=vtest(2:N)- vtest(1:N-1); F=D./(dx)^2.*dv/kbt./( exp(dv/kbt)-1 );
B=D./(dx)^2.*dv/kbt./( -exp(-dv/kbt)+1 );
    % Check for those cases for which dv = 0 and reset to the correct
    % F and B.
F(find(dv==0))=D./(dx)^2; B(find(dv==0))=D./(dx)^2;
F(find(abs(F)==Inf))=D./(dx)^2; B(find(abs(B)==Inf))=D./(dx)^2;

jump=zeros(N,N);   %jump is the matrix of F's and B's

% Check if dt is small enough. If dt is too large, stop running
% and print time-step size recommendation.
if dt>min(1./(F+B))
    sprintf('\n \n dt is too large, we need dt<min(1./(F+B)). dt should less than')
    min(1./(F+B))
    break
end


% Put F's and B's into jump.
for i=2:N-1
    jump(i,i-1)=F(i-1);
    jump(i,i)=-(F(i)+B(i-1));
    jump(i,i+1)=B(i);
end
jump(1,1)=-(F(1));          %reflecting boundary at the left end
jump(1,2)=B(1); jump(2,1)=F(1); jump(N-1,N)=B(N-1);
jump(N,N-1)=F(N-1); if alpha~=0
        %absorbing boundary condition at the right end
    final=D./(dx)^2*alpha^2/(exp(alpha)-1-alpha);
else
    final=D./(dx)^2;
end


jump(N,N)=-final-B(N-1);

% the initial condition.
% p is the probability distribution
% sump is the sum of the probabilities inside the region.

p=zeros(N,1); p(1)=1/dx; sump=[sum(p*dx)]; savep=[];
jump=sparse(jump);




%% Estimate the running time
steps=ceil(T/dt); testp=p; tic for i=1:10000
    k1=jump*testp*dt;
    k2=jump*(testp+k1/2)*dt;
    k3=jump*(testp+k2/2)*dt;
    k4=jump*(testp+k3)*dt;
    testp=testp+k1/6+k2/3+k3/3+k4/6;
end time_test=toc; time_estimate=steps/10000*time_test;
text=num2str(time_estimate); time_text=['The estimeate running time
is: ' text ' seconds']; disp(time_text); disp('Press any key if you
want to continue running the code ...') pause


sump=[sum(p*dx)]; savep=[];
% Simulate from 0 to T with time step dt.
 for i=1:steps
    k1=jump*p*dt;
    k2=jump*(p+k1/2)*dt;
    k3=jump*(p+k2/2)*dt;
    k4=jump*(p+k3)*dt;
    p=p+k1/6+k2/3+k3/3+k4/6;
    if mod(i,aa)==0
     sump=[sump sum(p*dx)];
     i;
     sum(p*dx)
     if mod(i,aa*100)==0
         savep=[savep p;];
     end
 end
end


%%%% Plot the results.   %%%%
% You can change this part if you are interested in something else.
figure(1);
plot(x,p);
xlabel('x');
ylabel('probabilty distributionat
the end');
figure(2);
ss=max(size(sump));
dsump=-(sump(2:ss)-sump(1:ss-1))/dt/aa;
plot(x,vtest/kbt)
xlabel('x');
ylabel('G(k_BT)');
% average first passage time ues the old method
FPT=dt*aa*dsump*(1:ss-1)'*dt*aa


% Calculate the MFPT using the exponential fit.
% The region between T1 and T2 is the region
% fit to the exponential curve.

T1=T*0.8;
% the region between T1 and T2 is the region chosen to fit the expoentila fit
T2=T; deltat=dt*aa; s2=floor(T2/deltat); numbera=ceil(T1/deltat);
Ta=deltat*dsump(1:numbera-1)*(1:numbera-1)'*deltat
fit1=polyfit(deltat*(numbera:s2),log(dsump(numbera:s2)),1);
k1=-fit1(1); c=dsump(numbera)*exp(k1*numbera*deltat); figure(3)
semilogy(deltat*numbera:deltat:T2,dsump(numbera:s2),
deltat*numbera:deltat:T2,c*exp(-k1*(deltat*numbera:deltat:T2)),'r:')
xlabel('t (This fit should be a straight line, otherwise the results
is wrong)')
ylabel('Q(t)')
```

```
Tb=dsump(numbera)*(T1+1/k1)/k1
FPT1=Ta+Tb   % average first passage time ues the new method
sump1=sump;

figure(4)
 semilogy(dt*aa:dt*aa:T,dsump) ylabel('Q(t)') xlabel('t')
```

13.2. Some simple test to show the code can work correctly

To test the Matlab code, I ran it for some simple potentials for which the MFPT can be analytical calculated. Both the analytical results and numerical results are plotted. The numerical results fit the analytical results.

The first case tested is a constant potential. Based on the Einstein Diffusion equation, the MFPT should be proportional to L^2, where L is the diffusion distance. Fig. 13.1 shows that the numerical results fit the analytical results.

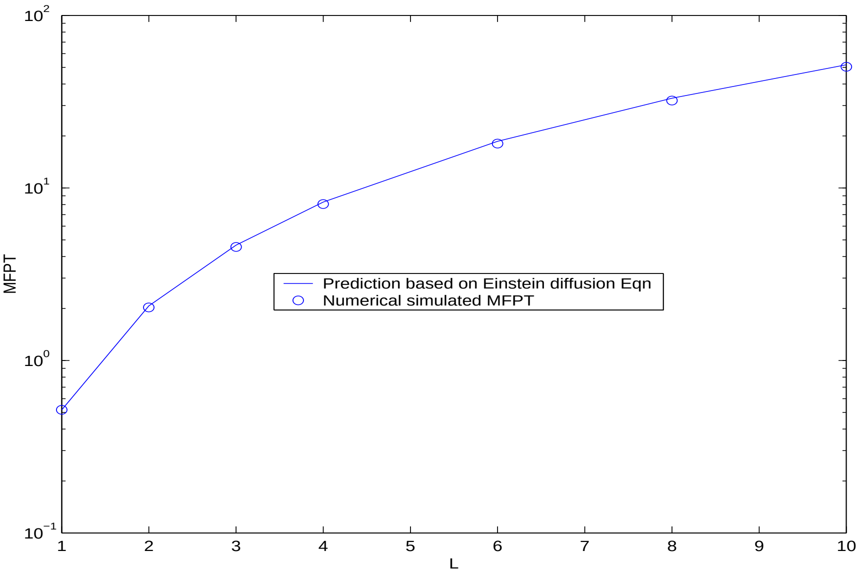


Figure 13.1. Mean FPT for constant potential with different L.

The second test is based on Kramers' harmonic potential.

$$MFPT = C/K\sqrt{\frac{k_BT}{G}}\exp(\frac{G}{k_BT}) \tag{13.1}$$

where C is some constant, and $G = 1/2KL^2$. If we keep L unchanged, then $G \propto K$. Therefore, in this case we should have $MFPT \propto G^{-3/2}\exp(\frac{G}{k_BT})$. Fig. 13.2 shows that numerical results fit the analytical results.

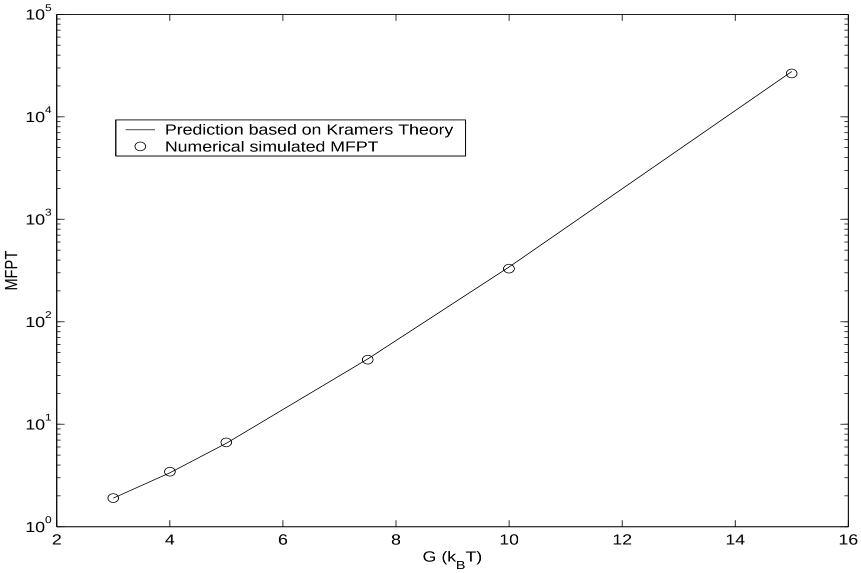


Figure 13.2. Mean FPT for harmonic potential with different K.

13.3. Matlab code: Constant force simulation

The code is used in Sec. 6.10.

```
%This  main code simulates the single-domain folding procedure.


% Define the variables:
% Ah is how many times faster when there is one helix turn formed.
% Ab ...........................................beta sheet.............
% As ...........................................secondary structure.....
% dt is the time interval between two neighbouring steps of the simulation
% time is the total time of the simulation.
% kbt is a constan = k_B*T, where k_T is the Boltzmann constant and T is the
% temperature in Kelvin degree.
% lu is the length of one amino acid in the coild state.
% lf is the length of one amino acid in the helix state.
% f is the external force during the folding procedure in unit of pn.
% hf is the initial high stretching force
% a and b are notations to make writing easier.
% kpu is the rate constant for helix unfolding.
% kpf is the rate constant for helix folding.
% ksu is the rate constant for secondary structures unfolding.
% ksf is the rate constant for secondary structures folding.
% kbf is the rate constant for beta sheet folding.
% avelf is the average length of one amino acid in the coil state.
% avelu is the average length of one amino acid in the helix state.

clear; Ah=50; Ab=1000; As=100; dt=0.01; time=10; kbt=300*0.0138;
lu=0.375; lf=0.15; f=15; hf=100; a=f*lf/kbt; b=f*lu/kbt; kpu=0.0;
kpf=0.03; ksu=0.0; ksf=100; kbf=0.1; kbftext=num2str(kbf);
k12=ksf;k23=ksf;k13=ksf;
avelf=lf/a^2*(a*cosh(a)-sinh(a))/(sinh(a)/a);
avelu=lu/b^2*(b*cosh(b)-sinh(b))/(sinh(b)/b);
number_step=floor(time/dt);

sumlength=[];




% Sequence includes the structure information of one domain.
% Each row represents one secondary structure of the domain.
% The first number of each row represents the number of amino acids in the
% secondary structure.
% The second number of each row represents the type of the secondary structure:
% 1 represents beta sheet, 2 represents loop region, 3 represents helix structure.
% To simulate the folding procedure for a different kind of protein,
% just change this part.
sequence=[
    7 1;
    2 2;
    8 1;
    5 2;
    12 3;
    5 2;
    6 1;
    2 2;
    3 1;
    5 2;
    4 3;
    4 2;
    10 1;
    3 2;
]; parts=[3 1 1 1 8]; nparts=max(size(parts));
partslength=zeros(1,nparts);
nsecondary=max(size(sequence)); % number of secondary structure

% helix is the matrix to record the states of all the helix amino acids.
% ah is the number of helix secondary structure.
% nh is a vector to record the number of amino acids of each helix secondary
% structure.
% The variables sheet, as and ns means similarly for beta sheet structure.
helix=[];ah=0;nh=[]; sheet=[];as=0;ns=[]; loop=[];al=0;nl=[];
helixpointer=[]; allpl=[]; allparts=[];

for i=1:nsecondary
    if sequence(i,2)==1
        as=as+1;
        ns=[ns sequence(i,1)];
    end
    if sequence(i,2)==2
        al=al+1;
        nl=[nl sequence(i,1)];
    end
    if sequence(i,2)==3
```

```
        ah=ah+1;
        nh=[nh sequence(i,1)];
        helixpointer=[helixpointer sequence(i,1)];
    end
end
  helix=zeros(ah,max(nh));

  sheet=zeros(as,max(ns));
  loop=zeros(al,max(nl));
  betafold1=0;
  betafold2=0;
  tfold=[0 0 0];
  extension=[];
  allbfold1=[];
  allbfold2=[];
  helix1=[];

% Here we simulate the folding procedure. The details can be seen in the
% section of "Simulation Procedure"
for i=1:number_step
     C=sum(sum(helix))/sum(nh);
     allc=[allc C];
     C=floor(sum(helix(1,:))/nh(1));
     [helix]=fhelix(helix,Ah,ah,nh,kpu,kpf,dt,f,kbt,f,lu);
     hLength=findls(helix,lf,lu,f,ah,nh,kbt);

     for j=1:nparts
         if j==1&betafold1==1
             parstlength(1)=randomfjc(lu*max(sequence(
             1:parts(1),1)),kbt,f,1);
         end
         if j==5&betafold2==1
             partslength(5)=randomfjc(lu*max(sequence(
             (sum(parts(1:j-1))+1):
             sum(parts(1:j)))),kbt,f,1);
         end

         if j==1&betafold1==0
              partslength(1)=0;
             for k=1:parts(1)
                 if rand<kbf*dt*C
                     betafold1=1;
                     kbf=kbf*Ab;
```

```
               % parstlength(1)=0.375*max(sequence(1:parts(1),1));
                parstlength(1)=randomfjc(lu*max(sequence(
                1:parts(1),1)),kbt,f,1);
            elseif betafold1==0
                partslength(1)=partslength(1)+findlength(
                sequence(k,:),f,kbt,lu,helix);
            end
        end
    elseif j==5&betafold2==0
       partslength(5)=0;
        for k=sum(parts(1:j-1))+1:sum(parts(1:j))
            if rand<kbf*dt*C
                betafold2=1;
                kbf=kbf*Ab;
               %partslength(5)=0.375*max(sequence(
                (sum(parts(1:j-1))+1)  :
                sum(parts(1:j))));
               partslength(5)=randomfjc(lu*max(sequence(
               (sum(parts(1:j-1))+1):
               sum(parts(1:j)))),kbt,f,1);
           elseif betafold2==0
                partslength(j)=partslength(j)+findlength(
                sequence(k,:),f,kbt,lu,
                helix,helixpointer, hLength);
            end
        end
    elseif j~=1&j~=5
         partslength(j)=0;
        for k=sum(parts(1:j-1))+1:sum(parts(1:j))
            templ=findlength(sequence(k,:),f,kbt,lu,helix,
            helixpointer, hLength);
           partslength(j)=partslength(j)+templ;
        end
    end

end

[tlength, tfold]=tertiary(partslength,tfold,k12*betafold1*C*betafold2,
k23*betafold2*C*betafold1,k13*betafold2*betafold1*C,As,dt);
%[tlength, tfold]=tertiary(partslength,tfold,k12,k23,k13,As,dt)
extension=[extension tlength];
allpl=[allpl sum(partslength)] ;
allbfold1=[allbfold1 betafold1];
allbfold2=[allbfold2 betafold2];
```

```
    allparts=[allparts;  partslength];
    allhl=[allhl; hLength];
    helix1=[helix1; helix(1,:)];
    end


t=dt:dt:time; hf=100; b=hf*lu/kbt;
avelu=lu/b^2*(b*cosh(b)-sinh(b))/(sinh(b)/b);

initial_time=0:dt:1; initial_length=[]; for
i=1:max(size(initial_time))
  % initial_length=[initial_length fjc(lu,kbt,hf,74)];
  initial_length=[initial_length avelu*76];
end iex=initial_length; nt=[initial_time t+1]; nextension=[iex
extension]; sumlength=[sumlength;nextension]; size(sumlength)
numberofdomain



%Plot the results.

figure(1)
%subplot(1,2,2)
plot(nt,(sumlength))

dateaxis=axis;
text(0.9*dateaxis(2)+.1*dateaxis(1),1.1*dateaxis(3)-0.1*dateaxis(4),date)
Nptext=num2str(sum(sequence(:,1))); Aptext=num2str(Ah);
Astext=num2str(As); kpftext=num2str(kpf);

kputext=num2str(kpu); ksftext=num2str(ksf); k12text=num2str(k12);
k23text=num2str(k23); k13text=num2str(k13);

ksutext=num2str(ksu); lftext=num2str(lf); lutext=num2str(lu);
dttext=num2str(dt); titletext=['Np=' Nptext ' dt=' dttext 's' '
lu=' lutext 'nm'  ' lf=' lftext 'nm'  '  Ah=' Aptext '  As='
Astext '  kbf=' kbftext '/s' sprintf('\n')   '  khf=' kpftext '/s'
'  khu=' kputext '/s'  '  ksf=' ksftext '/s' '  k13=' k12text '/s'
'  k35=' k23text '/s'  '  k15=' k13text '/s' ]; title(titletext)
xlabel(['time(s) ']) ylabel(['extension(nm) ' ])



% The followings are the functions used in the code
% Function 1.
% This code is to simulate the folding of one helix secondary structure.
% The inputs are the states of all the amino acids in one secondary structure
% After time dt, the amino aicds can change state.
% The outputs are the new states of all the amino acids.

function [number_fold ]=primary(number_fold,A,kpu,kpf,Ns,ns,dt);
for i=1:ns
    if number_fold(i,1)==0
        tempkpf=kpf;
        if number_fold(i,2)==1
            tempkpf=tempkpf*A;
        end
        if rand<tempkpf*dt
            number_fold(i,1)=1;
        end
    else
        if rand<kpu*dt
            number_fold(i,1)=0;
        end
    end


    for n=2:Ns(i)-1
        if number_fold(i,n)==0

            if number_fold(i,n-1)==1
                tempkpf=tempkpf*A;
            end
            if number_fold(i,n+1)==1
                tempkpf=tempkpf*A;
            end
            if rand<tempkpf*dt
                number_fold(i,n)=1;
            end
        else
            if rand<kpu*dt
                number_fold(i,n)=0;
            end
        end
    end

    if number_fold(i,Ns(i))==0
        tempkpf=kpf;
        if number_fold(i,Ns(i)-1)==1
            tempkpf=tempkpf*A;
        end
        if rand<tempkpf*dt
            number_fold(i,Ns(i))=1;
        end
    else
        if rand<kpu*dt
            number_fold(i,Ns(i))=0;
        end
    end
end


% Function 2.
% This code is to simulate the folding of the secondary structures.
% The varialbe second_fold represents whether the secondary structure has
% interacted with some other one. If it equals one, the secondary structure
% has interacted. i.e., it is in the folded state. Otherwise it is in the
% unfolded state.

function [second_fold ]=secondary(second_fold,A,ksu,ksf,ns,dt);

    if second_fold(1)==0
        tempksf=ksf;
        if second_fold(2)==1
            tempksf=tempksf*A;
        end
        if rand<tempksf*dt
            second_fold(1)=1;
        end
    else
        if rand<ksu*dt
            second_fold(1)=0;
        end
    end


    for n=2:ns-1
        if second_fold(n)==0
            tempksf=ksf;
            if second_fold(n-1)==1
                tempksf=tempksf*A;
            end
            if second_fold(n+1)==1
                tempksf=tempksf*A;
            end
            if rand<tempksf*dt
                second_fold(n)=1;
            end
        else
            if rand<ksu*dt
                second_fold(n)=0;
            end
        end
    end

    if second_fold(ns)==0
        tempksf=ksf;
        if second_fold(ns-1)==1
            tempksf=tempksf*A;
        end
        if rand<tempksf*dt
            second_fold(ns)=1;
        end
    else
        if rand<ksu*dt
            second_fold(ns)=0;
        end
    end

% Function 3.
% This code calculates the total length of one domain.

function [length,
tfold]=tertiary(partslength,tfold,k12,k23,k13,As,dt)
%tfold(1): describe folding between 1 and 2
%tfold(2):2&3
%tfold(3):1&3;
if sum(tfold)==2|sum(tfold)==3
    length=max(partslength);
    tfold=[1 1 1];

elseif sum(tfold)==1
    if tfold(1)==1
        length=max(partslength(1:3))+partslength(4)+partslength(5);
        if rand<dt*k23*As
            tfold(2)=1;
        end
        if rand<dt*k13*As
            tfold(3)=1;
        end
    end

     if tfold(2)==1
        length=max(partslength(3:5))+partslength(1)+partslength(2);
        if rand<dt*k12*As
            tfold(1)=1;
        end
        if rand<dt*k13*As
            tfold(3)=1;
        end
    end

    if tfold(3)==1
        length=max(partslength(1:5));
        if rand<dt*k23*As
            tfold(2)=1;
        end
        if rand<dt*k12*As
            tfold(1)=1;
        end
    end
elseif sum(tfold)==0
    length=sum(partslength);
     if rand<dt*k13
            tfold(3)=1;
     end
     if rand<dt*k23
            tfold(2)=1;
     end
     if rand<dt*k12
            tfold(1)=1;
     end
 end


% Function 4.
% This code is to simulate the folding of all helix secondary structures.
% The inputs are the states of all the amino acids in secondary structures.
% helix is a matrix, each row represents one secondary structure.
% Every element of one row represents one amino acid
% After time dt, the amino acids can change state.
% The outputs are the new states of all the amino acids.

function [helix]=fhelix(helix,A,ah,nh,kpu,kpf,dt,f,kbt,lf,lu);
tempkpf=0; for i=1:ah
    if helix(i,1)==0
        tempkpf=kpf;
        if helix(i,2)==1
            tempkpf=tempkpf*A;
        end
        if rand<tempkpf*dt
            helix(i,1)=1;
        end
    else
        if rand<kpu*dt
            helix(i,1)=0;
        end
    end


    for n=2:nh(i)-1
        if helix(i,n)==0

            if helix(i,n-1)==1
                tempkpf=tempkpf*A;
            end
            if helix(i,n+1)==1
                tempkpf=tempkpf*A;
            end
            if rand<tempkpf*dt
                helix(i,n)=1;
            end
        else
            if rand<kpu*dt
                helix(i,n)=0;
            end
        end
    end

    if helix(i,nh(i))==0
        tempkpf=kpf;
        if helix(i,nh(i)-1)==1
            tempkpf=tempkpf*A;
        end
        if rand<tempkpf*dt
            helix(i,nh(i))=1;
        end
    else
        if rand<kpu*dt
            helix(i,nh(i))=0;
        end
    end

end

% Function 5
% This code calculates the total length of one structure if
% there n amino acids under force f.
% The length of each amino acid is the average length.
function [l]=fjc(lu)

f=30; kbt=3*1.38; a=f*lu/kbt;
l=lu/a^2*(a*cosh(a)-sinh(a))/(sinh(a)/a);

% Function 6.
% This code calculates the total length of one structure if
% there n amino acids under force f.
% The length of each amino acid is determined by random number.
% Details can be seen in seciton "Use random number to determine
% the projected length of a freely rotating bar in external force "
function [suml]=randomfjc(l,kbt,f,n) a=f*l/kbt; suml=0; for i=1:n
    costheta=1/a*log(exp(a)-rand*(exp(a)-exp(-a)));
    templ=l*costheta;
    suml=suml+templ;
end


% Function 7.
% This code is to find the length of one helix strucure.
% The length of one amino acid is the average length.
function [s]=findls(number_fold,lf,lu,f,ns,Ns,kbt)

a=f*lu/kbt;

avelu=lu/a^2*(a*cosh(a)-sinh(a))/(sinh(a)/a); for i=1:ns
    s(i)=0;
    temp=0;
    for j=1:Ns(i)
```

```
        if number_fold(i,j)==0
            s(i)=s(i)+avelu;
            if temp==0

            else
                b=temp*f*lf/kbt;
                avelf=lf*temp/b^2*(b*cosh(b)-sinh(b))/(sinh(b)/b);
                s(i)=s(i)+avelf;
                temp=0;
            end
        else
            temp=temp+1;
        end
    end
    if temp~=0
     b=temp*f*lf/kbt;
     avelf=lf*temp/b^2*(b*cosh(b)-sinh(b))/(sinh(b)/b);
     s(i)=s(i)+avelf;
    end

end

% Function 8.
% This code is to find the length of one helix structure.
% The length of one amino acid is determined by random numbers.

function [s]=findls2(number_fold,lf,lu,f,ns,Ns,kbt)

a=f*lu/kbt;

avelu=lu/a^2*(a*cosh(a)-sinh(a))/(sinh(a)/a); for i=1:ns
    s(i)=0;
    temp=0;
    for j=1:Ns(i)
        if number_fold(i,j)==0
            s(i)=s(i)+randomfjc(lu,kbt,f,1);
            if temp==0

            else
                s(i)=s(i)+randomfjc(lf*temp,kbt,f,1);
                temp=0;
            end
        else
            temp=temp+1;
        end
    end
    if temp~=0
     b=temp*f*lf/kbt;
     avelf=lf*temp/b^2*(b*cosh(b)-sinh(b))/(sinh(b)/b);
     s(i)=s(i)+randomfjc(lf*temp,kbt,f,1);
    end

end

% Function 9.
% This code is to find the length of one secondary structure.
% The length of one amino acid is the average length.
function [length]=findlength(sequence,f,kbt,lu,helix,helixpointer,
hLength); if sequence(2)~=3
    b=f*lu/kbt;
    avelu=lu/b^2*(b*cosh(b)-sinh(b))/(sinh(b)/b);
    length=sequence(1)*avelu;
else
    a=find(helixpointer==sequence(1));
    length= hLength(a);
end

% Function 10.
% This code is to find the length of one secondary structure.
% The length of one amino acid is determined by random numbers.
function
[length]=findlength2(sequence,f,kbt,lu,helix,helixpointer,
hLength); if sequence(2)~=3
    b=f*lu/kbt;
    avelu=lu/b^2*(b*cosh(b)-sinh(b))/(sinh(b)/b);
    length=randomfjc(lu,kbt,f,sequence(1));
else
    a=find(helixpointer==sequence(1));
    length= hLength(a);
end
```

13.4. Matlab code: IB model for RNase H

This is the code to simulate the folding procedure of RNase H based on IB model. This code is used in Sec. 11.

```
% Shows the sequence at which bonds forms showing their initial contour
% length expressed in terms of sequence length.
% Modified from simumodel6PLUSJuly29 to have annotation.
% August 2008
clear all; close all;
% This code is used to show how the folding sequence change
% close;
 allT=[];
 % Sequence_record stores Lchange_f for all the forces,
 % each in its own row.
 sequence_record=[];
 % All_contact stores the Contact_order for all the forces,
 % each in its own row.
 all_contact=[];

 ft=[];
 for f=1:10
     f
     allT=[];
% Lchange_f stores the changes in contour length in order of bonding,
% for each force in its own column.
% Note that it stores the Lchanges for all repeats as a long row.
    Lchange_f=[];
% Contact_order is a row vector of the Sequence Distance of bonds, in order
% of bonding.
     contact_order=[];
 for repeats=1:2
     repeats
% Lchange stores the changes in contour length in order of bonding.
% Note that it resets at each repeat.
  Lchange=[];
N=155;
contact=zeros(N,N);
distance=zeros(N,N);
foldingrates=zeros(N,N);
unfoldingrates=zeros(N,N);
aminoacidL=0.38;
```

```
helixL=0.15;
betaL=0;
foldrate=1.5;
contour=0.38*ones(1,N);
alphafoldingrate  = foldrate;
alphaunfoldingrate = 0.00000001;
%alphaunfoldingrate = 0.00000001*0.01;
alphaunfoldingx = 0.1;
betafoldingrate  = foldrate;
%betaunfoldingrate = 0.00000001;
betaunfoldingrate = 0.00000001*0.01;
betaunfoldingx = 0.1;
dt=0.01;
df=0.03;
highforce=30;
lowforce=df;
kbt=3*1.38;
p=0.7;

% Input the protein structure.
% First column, length of structure.
% Second column, type: 0 means loop, 1 means alpha helix, 2 means beta sheet.
protein=[3 0;
         10,2;
         7, 0;
         8, 2;
         2, 0;
         6, 2;
         7, 0;
         15,1;
         5, 0;
         5, 2;
         3, 0;
         7, 1;
         2, 0;
         8, 1;
         12,0;
         11,1;
         2, 0;
         9, 2;
         5, 0;
         14,1;
         14,0;
        ];
```

```
temp=size(protein);
% Define the number of secondary structures.
Nsecond=temp(1);
% Define beta sheet in details
% 1st and 2nd column:
% sequence positions of the first bond of beta sheet:
% If anti-parallel, then 1st column is at base (of arrow) of first strand,
% and 2nd column is at head (of arrow) of second strand.
% If parallel, then both columns are at base of sequence.
% The 3rd column: length of the beta sheet.
% The 4th column: type of beta sheet: -1 means anti parallel, 1 means parallel.
beta=[5, 28, 8, -1;
      23,36, 6, -1;
      4, 64, 5, 1;
      64, 115,5, 1;
  ];

% Once we meet a alpha helix structure, we know those bonds need
% to be formed inside the helix.
 if ~isempty(find(protein(:,2)==1))
    pointer=0;
    for i=1:Nsecond
        tempsize=protein(i,1);
        if protein(i,2)==1       % Find the alpha helices.
            for j=1:tempsize-4
                contact(pointer+j,pointer+j+4)=-1;
                foldingrates(pointer+j,pointer+j+4)=alphafoldingrate;
                unfoldingrates(pointer+j,pointer+j+4)=alphaunfoldingrate;
            end
        end
        pointer=pointer+tempsize;
    end

end

% Automatically set the bonds between beta sheet
if ~isempty(beta)
    temp=size(beta);
    Nbeta=temp(1);
    for i=1:Nbeta
        tempsize=beta(i,3);
        direction=beta(i,4);
        temp1=beta(i,1);
        temp2=beta(i,2);
```

```
         % The following 'for' loop has been corrected,  July 29, 2008.
         for j=1:tempsize
             contact(temp1+j-1,temp2+(j-1)*direction)=-2;
             foldingrates(temp1+j-1,temp2+(j-1)*direction)=betafoldingrate;
             unfoldingrates(temp1+j-1,temp2+(j-1)*direction)=betaunfoldingrate;
         end
     end
 end



 secondaryfoldrate=foldrate * 1;


 %%%%%%%%%%%%NEW ENTRIES%%%%%%%%%%%%%
 % first column, second column, third column:
 % amino acid 1, amino acid 2, bond length between 1 & 2
second_bonds=[7,55,3.3;
    7,55,3.3;
    7,56,3.5;
    8,133,3.5;
    10,134,2.9;
    10,130,3.3;
    10,136,3.4;
    10,137,3.4;
    10,133,3.5;
    11,137,3.2;
    12,137,2.9;
    12,141,3.3;
    13,44,3.3;
    21,47,3.0;
    22,47,3.1;
    22,50,3.1;
    22,51,3.5;
    22,54,3.3;
    26,50,3.1;
    26,55,3.1;
    22,48,3.1;
    25,129,3.5;
    34,136,3.3;
    35,58,3.1;
    36,136,3.1;
    36,139,3.2;
    45,73,2.7;
    46,103,3.2;
    49,73,2.8;
    49,104,3.4;
    49,107,3.1;
    53,103,3.1;
    53,106,3.5;
    56,107,3.3;
    56,114,3.4;
    57,106,2.6;
    73,104,3.2;
    75,120,3.5;
    75,122,3.4;
    82,104,3.5;
    110,114,3.4;
    112,116,3.2;
    ];
bonds_length=3.5;
sbonds_number=max(size(second_bonds));
for i=1:sbonds_number
    if second_bonds(i,3)<=3.5
        contact(second_bonds(i,1),second_bonds(i,2))=-2;
        foldingrates(second_bonds(i,1),second_bonds(i,2))=secondaryfoldrate;
        unfoldingrates(second_bonds(i,1),second_bonds(i,2))=betaunfoldingrate;
    end
end


% Contact now holds -1's and -2's.
% -1 for an amino acid pair which will form an alpha-helix bond.
% -2 for an amino acid pair which will form a beta sheet or
% secondary structure bond.
 %%%%%%%%%%%%%%%%%%%%%%%%
 % Now begin to simulate the folding procedure from the high force to
 %the low force region.
 % The initial conditions are set for the totally unfolded state.
 % If want to use this code to simulate the unfolding procedure,
 % you need to set a different initial condition.
 % The simulation procedure do not need to change.
 % (Nothing need to change in the big "for" loop.)
 fitxL=0.01:0.01:0.99;
 fitF=kbt/p*(1./4./(1-fitxL).^2-1/4+fitxL);
 extension=[];
 Lc=[];
 time=0;
% how many bonds we have:
BondCount = max(size(find(contact~=0)));
 while max(size(Lchange))<BondCount
     time=time+dt;
     % Get the x/L at this force using Marko-Siggia
     xpercentageL=interp1(fitF,fitxL,f);
     for m=1:N-1
         for n=m+1:N
             if contact(m,n)==-1
                 Ldistance=sum(contour(m+1:n));
                 tempx=xpercentageL*Ldistance;
                 if rand<dt*foldingrates(m,n)*exp(-tempx*f/kbt)
                     contact(m,n)=1;
                     Lchange=[Lchange Ldistance];
                     contact_order=[contact_order n-m];
                     for i=m:n
                         if contour(i)~=betaL
                         contour(i)=helixL;
                         end
                     end
                 end
             elseif contact(m,n)==1
                     if rand<dt*unfoldingrates(m,n)*exp(alphaunfoldingx*f/kbt)
                          disp(’break’)
                          m
                          n
                     contact(m,n)=-1;

                     for i=m:n
                         contour(i)=aminoacidL;
                     end
                     end
             elseif contact(m,n)==-2
                    Ldistance=sum(contour(m+1:n));
                    tempx=xpercentageL*Ldistance;
                     if rand<dt*foldingrates(m,n)*exp(-tempx*f/kbt)
                         contact(m,n)=2;
                         Lchange=[Lchange Ldistance];
                         contact_order=[contact_order n-m];
                         for i=m:n
                             contour(i)=betaL;
                         end
                     end
             elseif contact(m,n)==2
                 if rand<dt*unfoldingrates(m,n)*exp(betaunfoldingx*f/kbt)
                     contact(m,n)=-2;
                     contour(m:n)=aminoacidL;
                     disp('break')
                     disp('warning!!  beta structure break!!!')
                     disp('warning!!  beta structure break!!!')
                     disp('warning!!  beta structure break!!!')
                     m
                     n
                     for s1=m:n-1
                         for s2=s1+1:n
                             if contact(s1,s2)==1
                                 for s3=s1:s2
                                     if contour(s3)==aminoacidL
                                         contour(s3)=helixL;
                                     end
                                 end
                             elseif contact(s1,s2)==2
                                 contour(s1:s2)=betaL;
                             end
                         end
                     end

                 end
             end
         end
     end
     extension=[extension sum(contour)*xpercentageL];
     Lc=[Lc sum(contour)];
 end
 allT=[allT time];
 Lchange_f=[Lchange_f Lchange];
end
sequence_record(f,:)=Lchange_f;
ft(f,:)=allT;
all_contact(f,:)=contact_order;
end
 deltaL=1;
 LN=20;
 contact_bar=zeros(f,BondCount);
 for i=1:f
     for j=1:repeats
         for k=1:BondCount
```

```
% Corrected August 4, 2008: '48' changed to 'BondCount'
             contact_bar(i,k)=all_contact(i,BondCount*(j-1)+k)+contact_bar(i,k);
         end
     end
 end
  figure(1)
 bar3(1:BondCount,contact_bar')
 ylabel('Folding order')
 zlabel('Sequence distance')
 xlabel('Force (pN)')
```

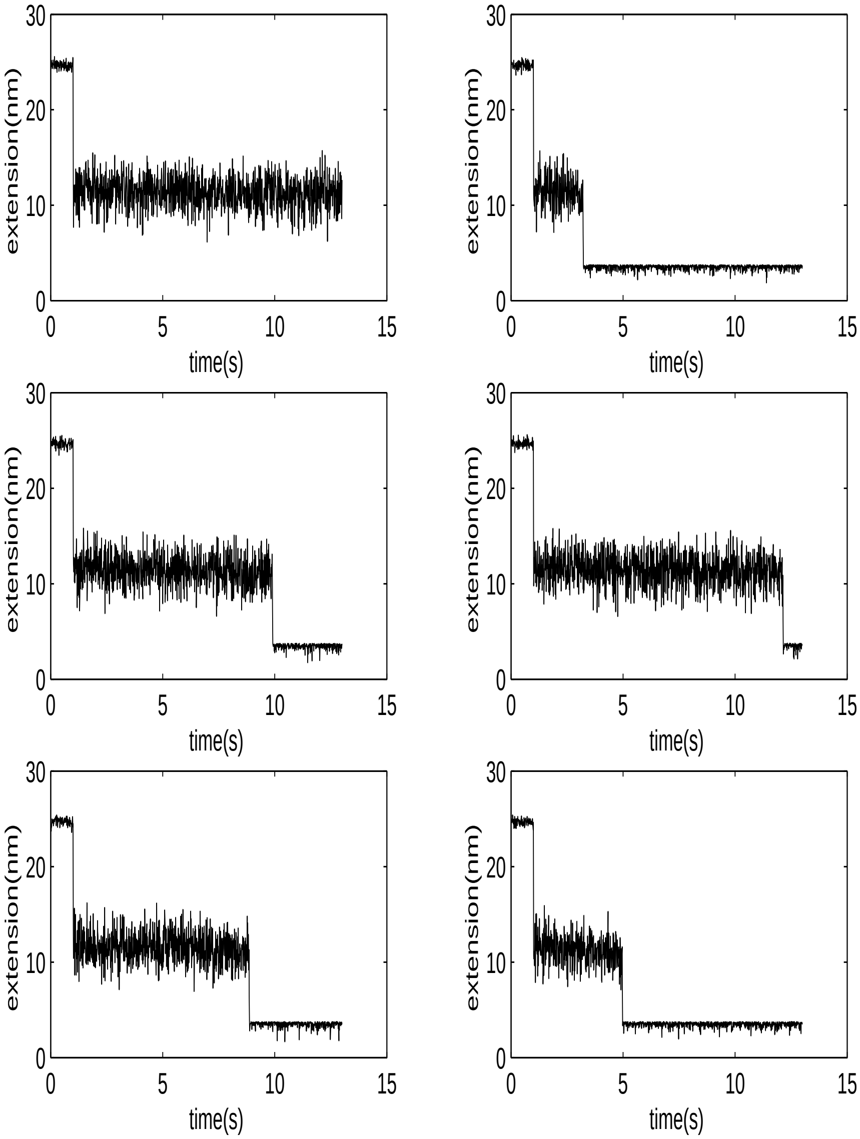


Figure 13.3. The simulation results for one domain. The parameters are given in Sec. 13.5. These results are for Sec. 6.17.

13.5. Simulation results without interactions between domains

The simulation results for different number of domains. The parameters used are the same: k_{pf}=0.03/s, k_{bf}=0.1, k_{pu}=0/s, k_{bu}=0 /s, k_{su}=0 /s, A_h=50,A_s=1000, and A_s=100. These results are for Sec. 6.17.

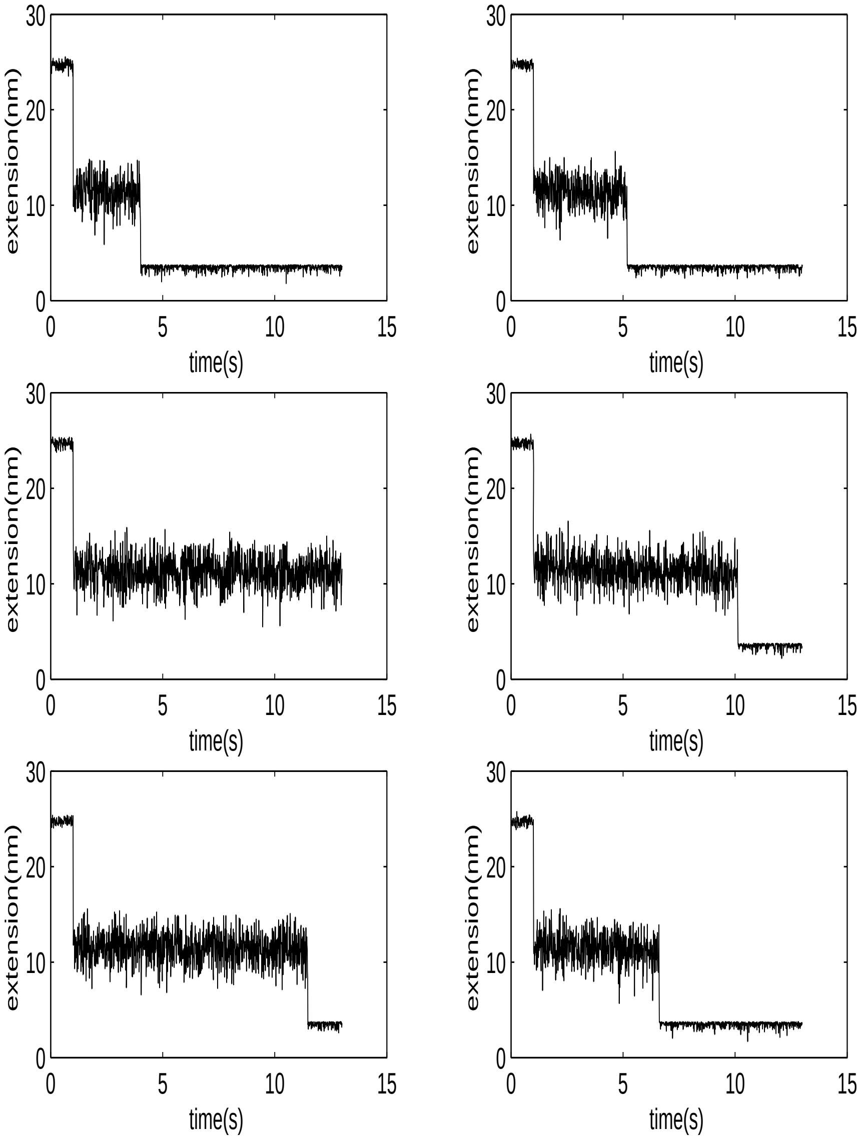


Figure 13.4. The simulation results for one domain. For the same parameters as in Fig. 13.3, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

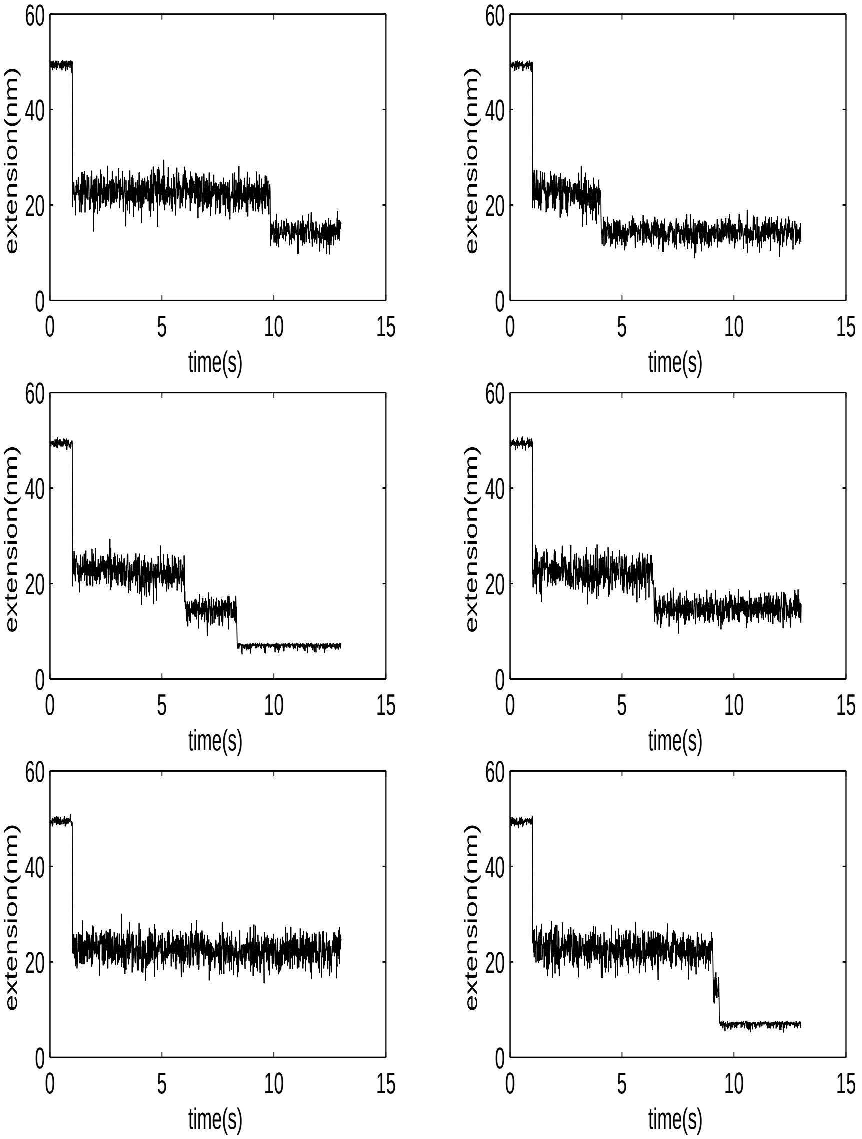


Figure 13.5. The simulation results for two domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

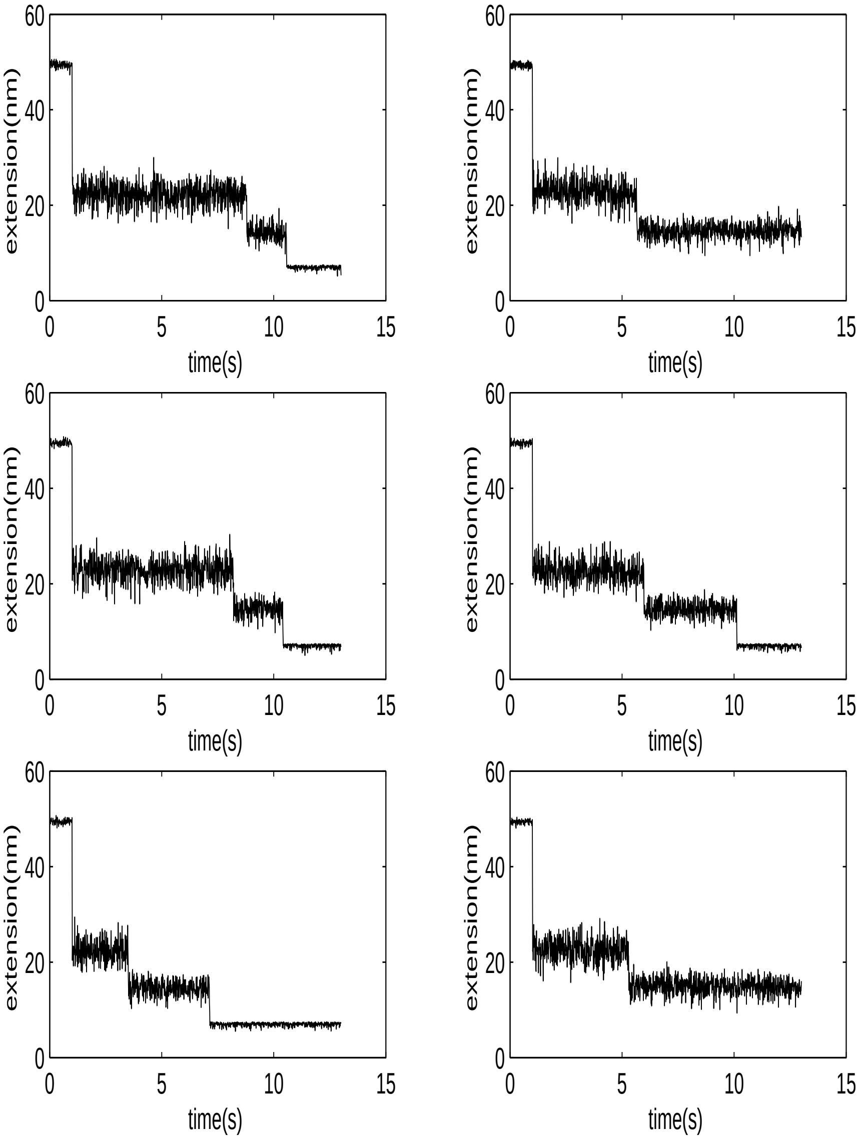


Figure 13.6. The simulation results for two domains.For the same parameters as in Fig. 13.5, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

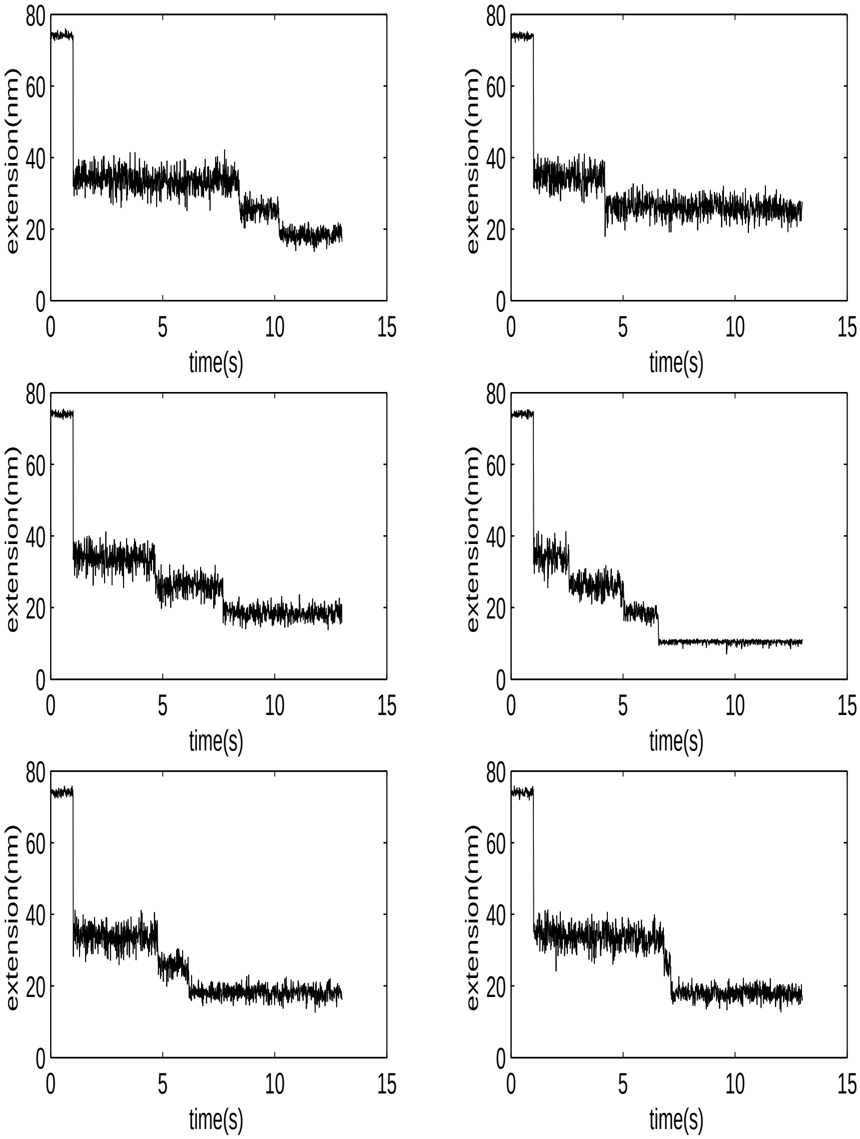


Figure 13.7. The simulation results for three domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

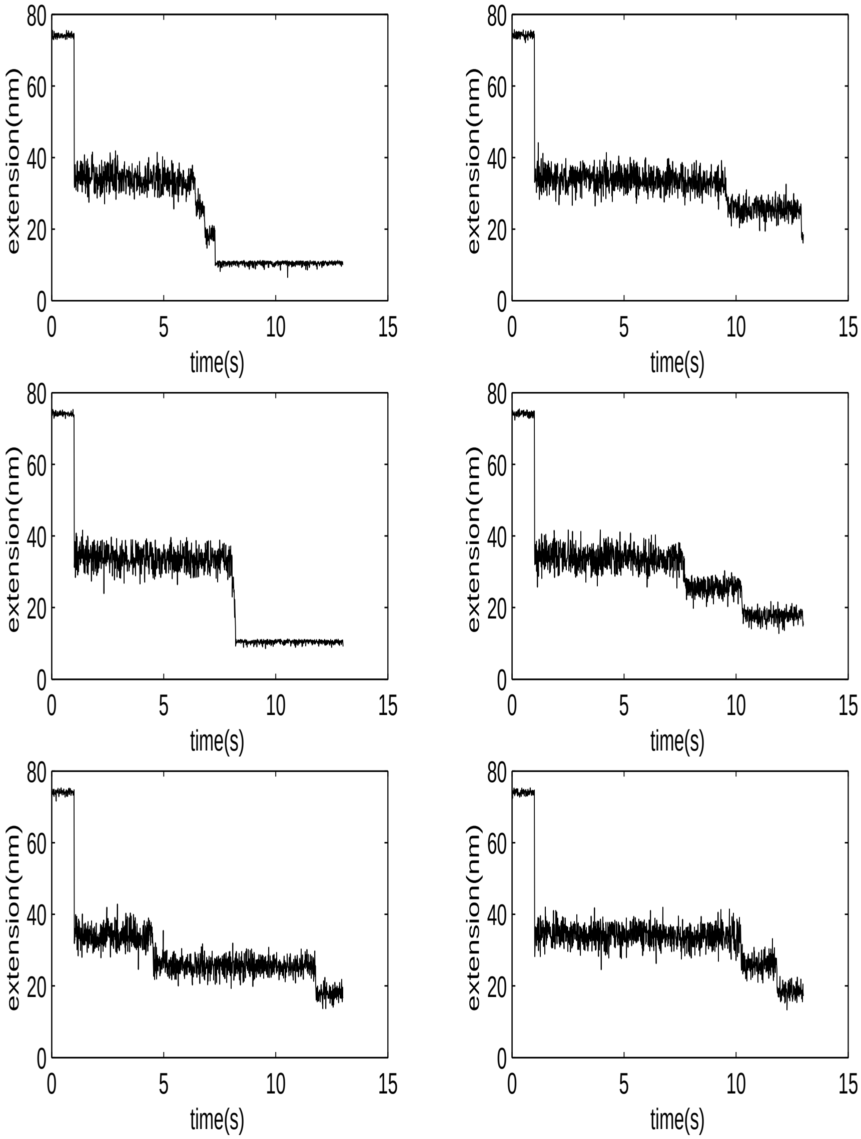


Figure 13.8. The simulation results for three domains. For the same parameters as in Fig. 13.7, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

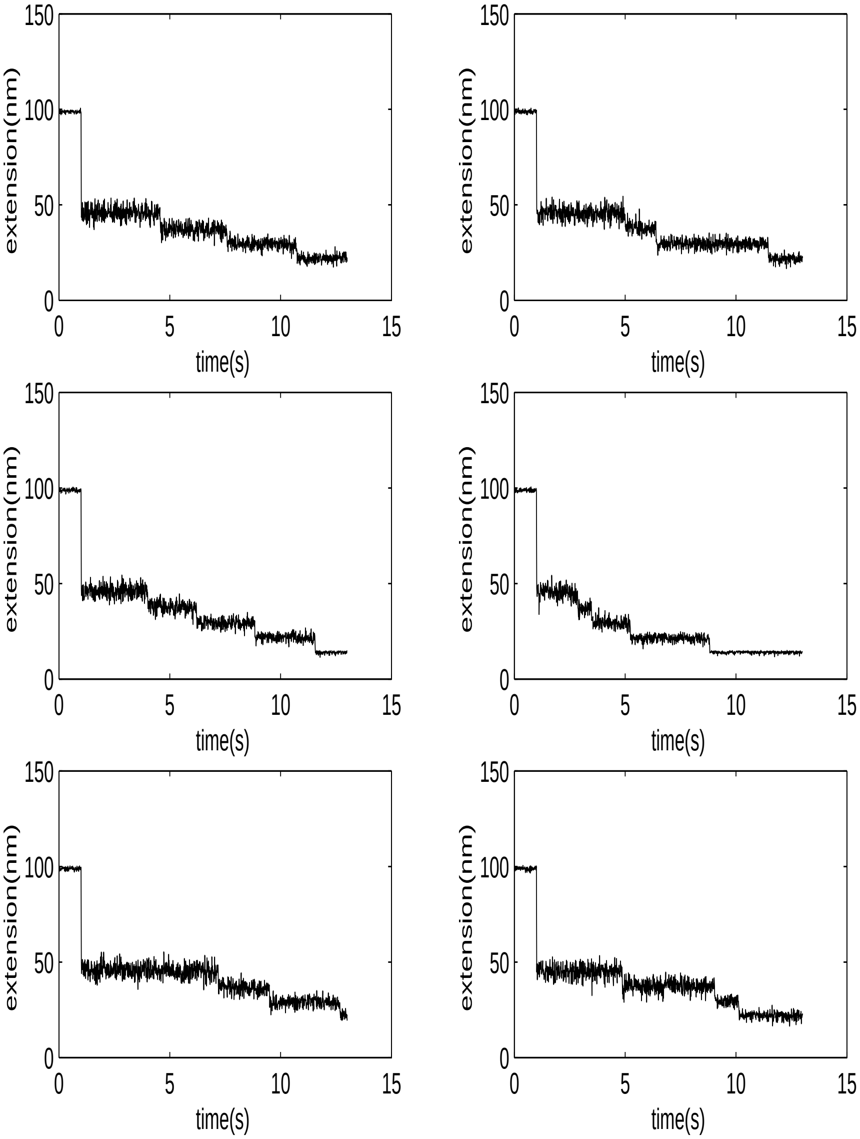


Figure 13.9. The simulation results for four domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

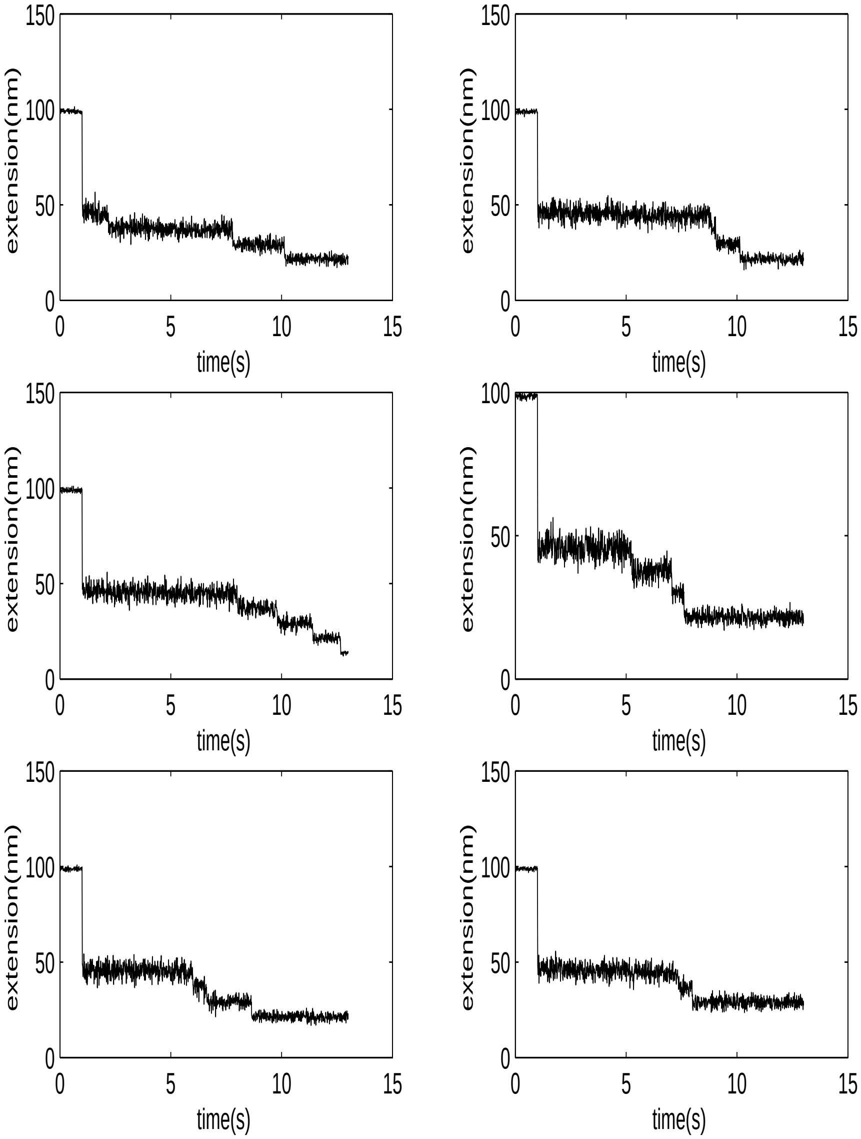


Figure 13.10. The simulation results for four domains. For the same parameters as in Fig. 13.9, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

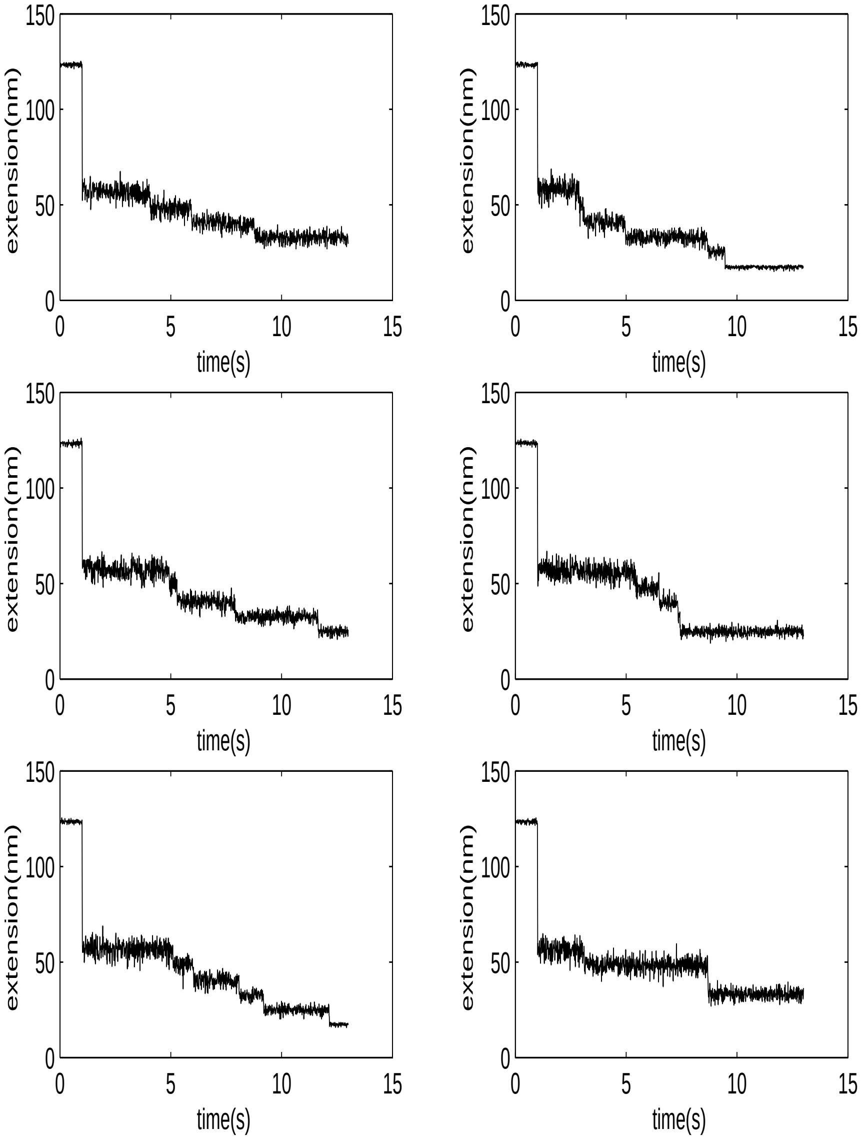


Figure 13.11. The simulation results for five domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

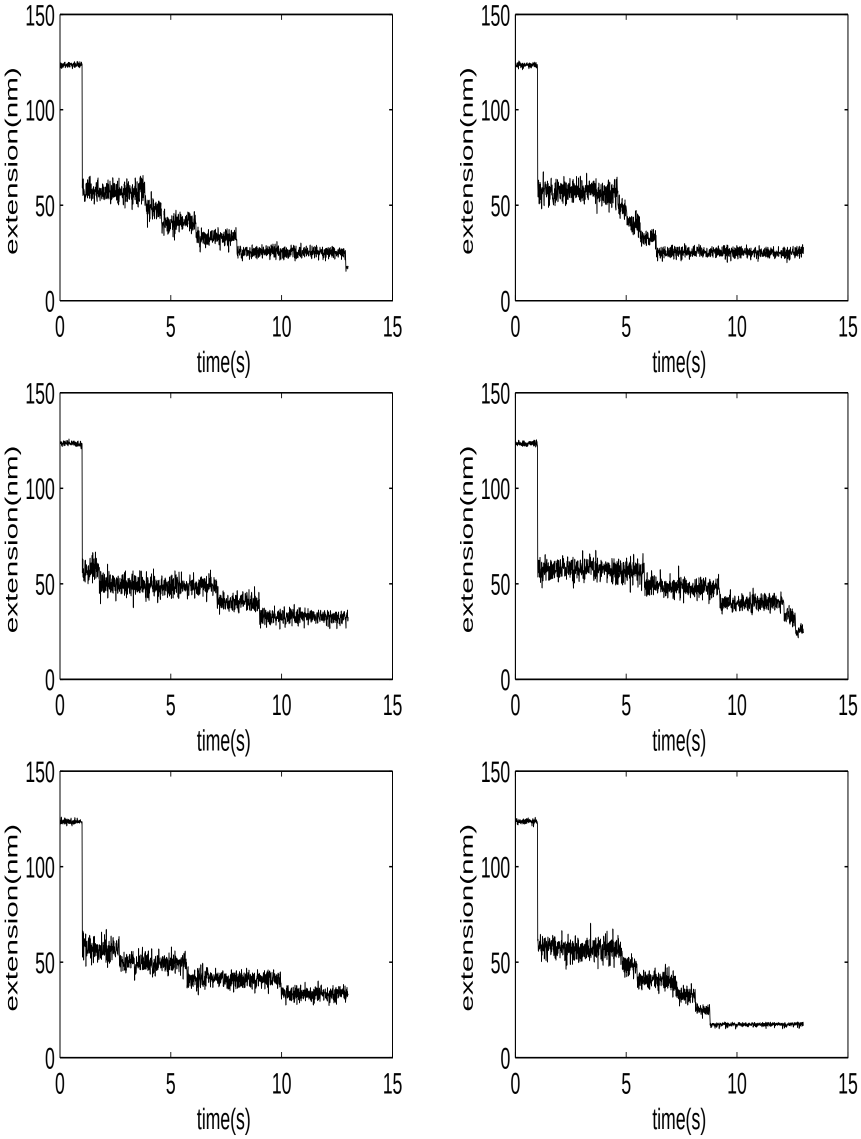


Figure 13.12. The simulation results for five domains. For the same parameters as in Fig. 13.11, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

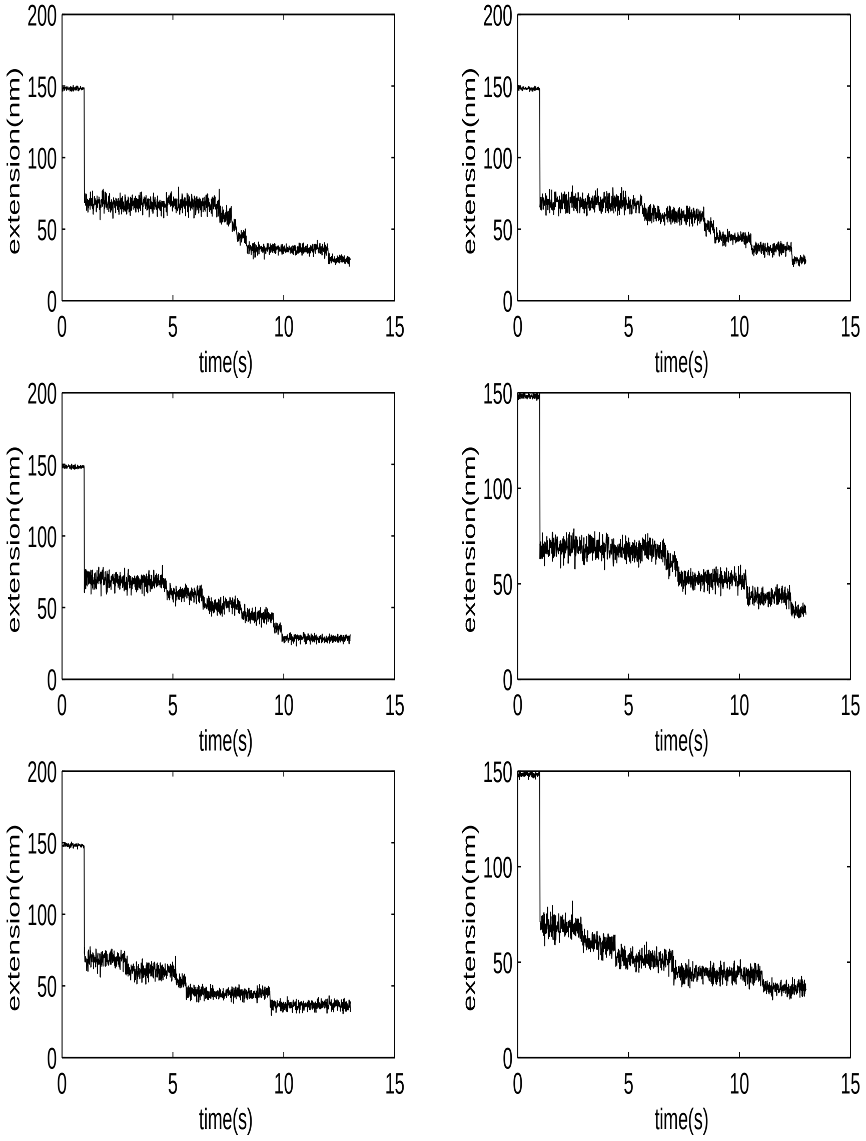


Figure 13.13. The simulation results for six domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

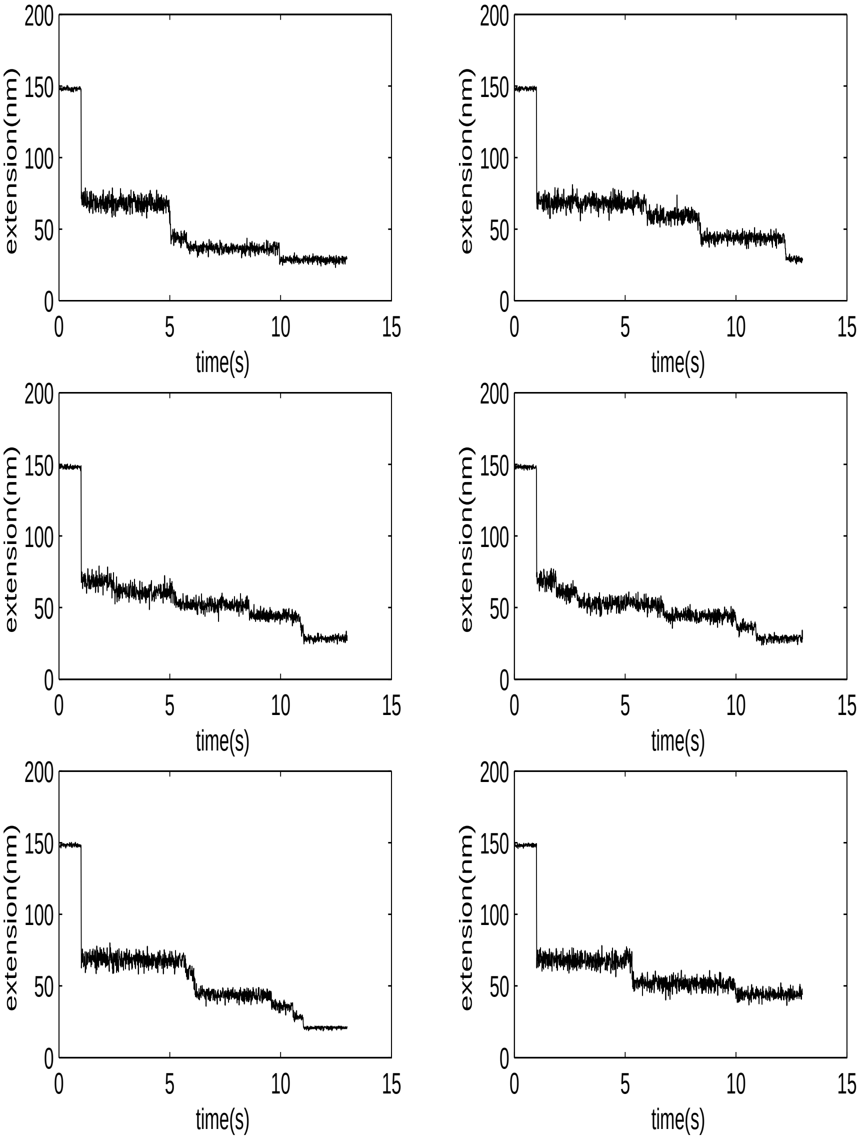


Figure 13.14. The simulation results for six domains. For the same parameters as in Fig. 13.13, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

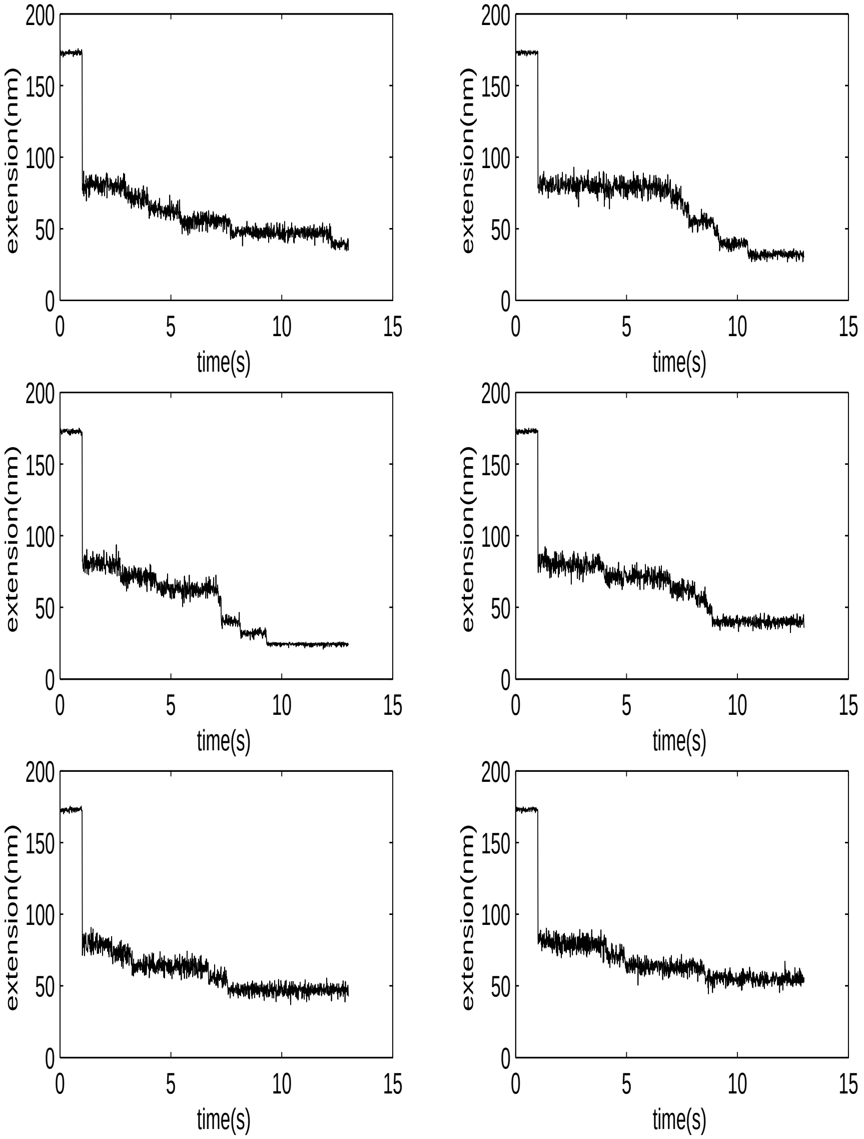


Figure 13.15. The simulation results for seven domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

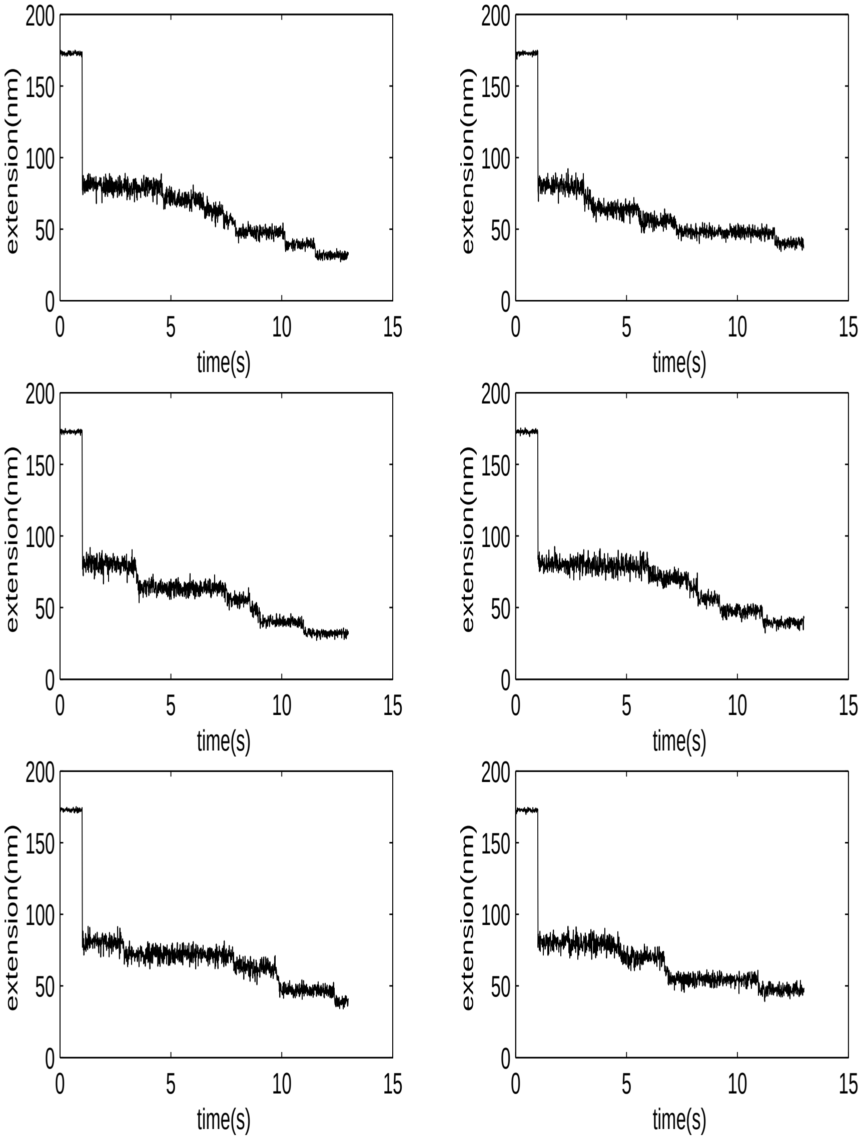


Figure 13.16. The simulation results for seven domains. For the same parameters as in Fig. 13.15, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

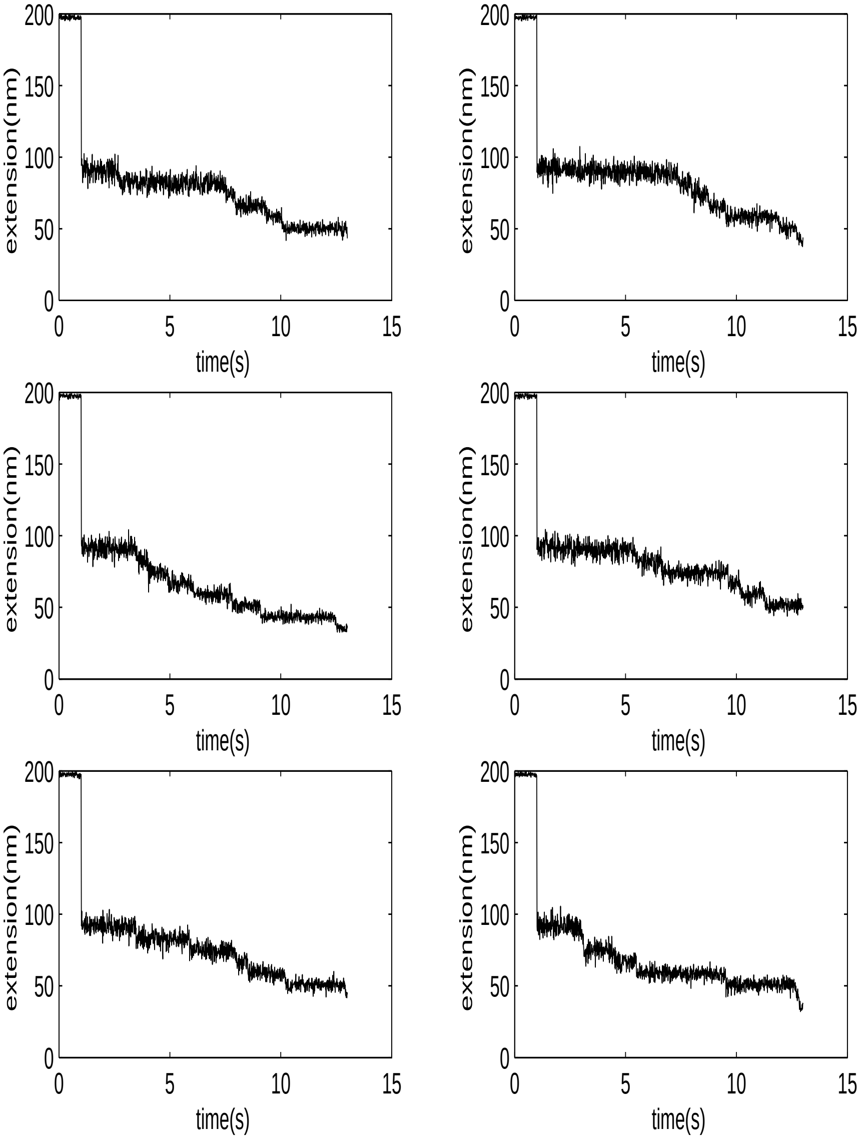


Figure 13.17. The simulation results for eight domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

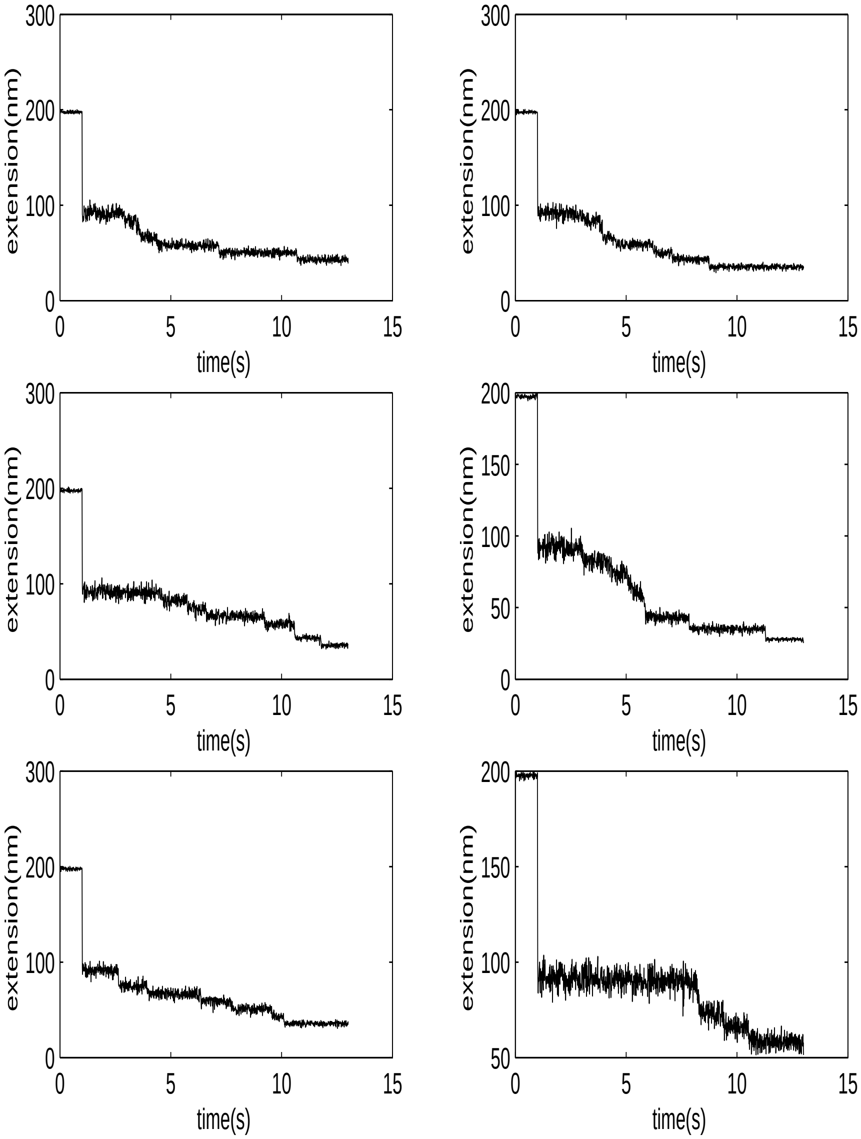


Figure 13.18. The simulation results for eight domains. For the same parameters as in Fig. 13.17, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

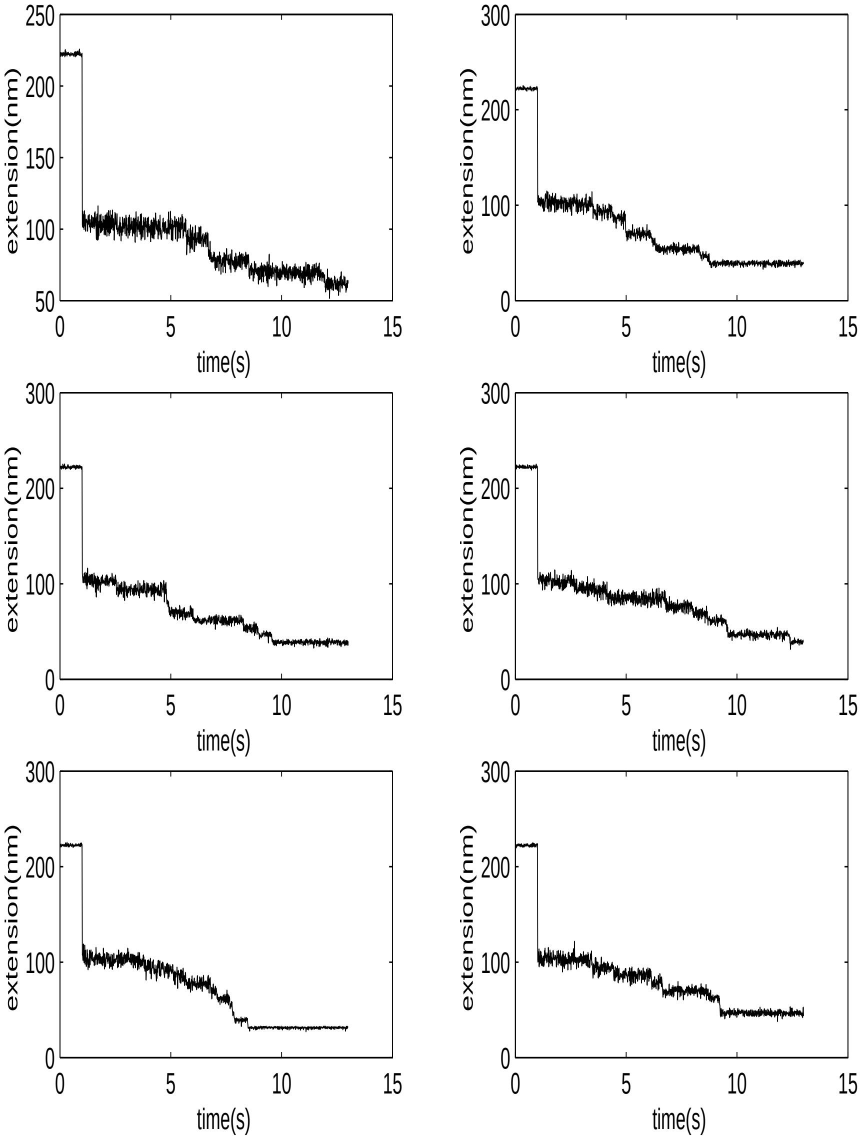


Figure 13.19. The simulation results for nine domains. The parameters are given in Sec. 13.5. These results are for Sec. 6.17. Different domains fold independently.

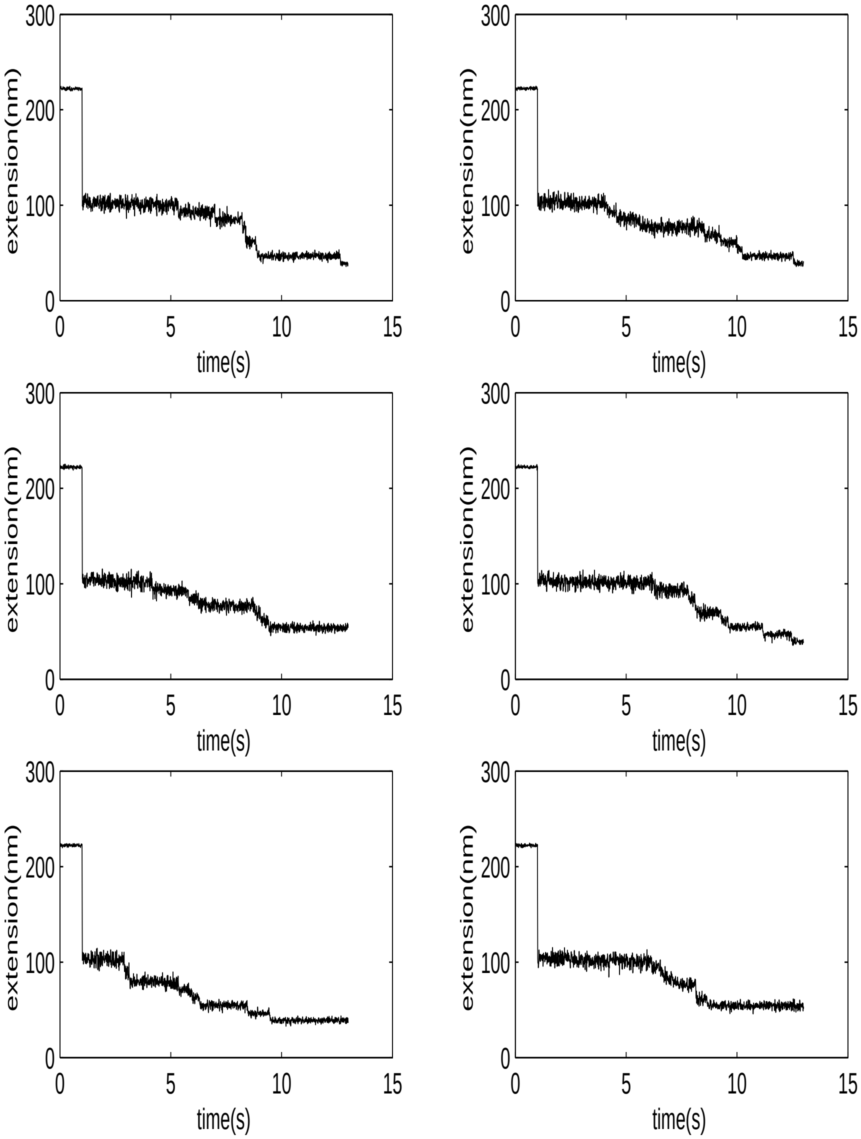


Figure 13.20. The simulation results for nine domains. For the same parameters as in Fig. 13.19, additional realizations. These results are different due to the inherent stochastic nature of the dynamics.

References

[1] R. Afrin, M. T. Alam, and A. Ikai. Pretransition and progressive softening of bovine carbonic anhydrase II as probed by single molecule atomic force microscopy. *Protein Sci.*, 14:1447–1457, 2005.

[2] D. Baker. A surprising simplicity to protein folding. *Nature*, 405:39–42, 2000.

[3] Protein Data Bank. http://www.rcsb.org/pdb.

[4] G. I. Bell. Models for the specific adhesion of cells to cells. *Science*, 200:618 – 627, 1978.

[5] R. B. Best and G. Hummer. Protein folding kinetics under force from molecular simulation. *J. Am. Chem. Soc.*, 130:3706–3707, 2008.

[6] R. B. Best, B. Li, A. Steward, V. Daggett, and J. Clarke. Can non-mechanical proteins withstand force? Stretching barnase by atomic force microscopy and molecular dynamics simulation. *Biophys. J.*, 81:2344–2356, 2001.

[7] C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Publishing, Inc., 1998.

[8] D. J. Brockwell et al. Pulling geometry defines the mechanical resistance of a β-sheet protein. *Nat. Struct. Biol.*, 10:731–737, 2003.

[9] B. Bullard et al. The molecular elasticity of the insect flight muscle proteins projectin and kettin. *Proc. Natl. Acad. Sci. USA*, 103:4451–4456, 2006.

[10] R. W. Carrell and B. Gooptu. Conformational changes and disease-serpins, prions and alzheimer's. *Curr. Opin. Struct. Biol.*, 8:799–809, 1998.

[11] M. Carrion-Vazquez et al. Mechanical and chemical unfolding of a single protein: A comparison. *Proc. Natl. Acad. Sci. USA*, 96:3696–3699, 1999.

[12] M. Carrion-Vazquez et al. The mechanical stability of ubiquitin is linkage dependent. *Nat. Struct. Biol.*, 10(9):738–743, 2003.

[13] C. Cecconi, E. A. Shank, C. Bustamante, and S. Marqusee. Direct observation of the three-state folding of a single protein molecule. *Science*, 309:2057–2060, 2005.

[14] H. S. Chung, M. Khalil, A. W. Smith, Z. Ganim, and Tokmakoff. Conformational changes during nanosecond-to-millisecond unfolding of ubiquitin. *Proc. Nat. Acad. Sci. U.S.A.*, 102:612–617, 2005.

[15] T. E. Creighton. *Proteins: Structures and Molecular Properties.* W. H. Freeman & Co., New York, 1993.

[16] P. G. de Gennes. *J. Phys. Lett.*, 46(L639), 1985.

[17] C. Dobson, A. Sali, and M. Karplus. Protein folding: A perspective from theory and experiment. *Chem. Int. Ed.*, 37:868–893, 1998.

[18] Y. Duan, L. Wang, and P.A. Kollman. The early stage of folding of villin headpiece subdomain observed in a 200-nanosecond fully solvated molecular dynamics simulation. *Proceedings of the National Academy of Sciences 95*, pages 9897–9902, 1998.

[19] P. A. Ellison and S. Cavagnero. Role of unfolded state heterogeneity and en-route ruggedness in protein folding kinetics. *Protein Sci.*, 15:564–582, 2006.

[20] E. Evans and K. Ritchie. Dynamic strength of molecular adhesion bonds. *Biophys. J.*, 72:1541–1555, 1997.

[21] J. M. Fernandez and H. Li. Force-clamp spectroscopy monitors the folding trajectory of a single protein. *Science*, 303(5664):1674–1678, 2004.

[22] T. Frisch and A. Verga. Slow relaxation and solvent effects in the collapse of a polymer. *Phys. Rev. E*, 65(041801), 2002.

[23] C. W. Gardiner. *Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences.* Springer, 1983.

[24] M. Gruebele. Protein folding: the free energy suface. *Curr. Opin. Struct. Biol.*, 12:161–168, 2002.

[25] J. Howard. *Mechanics of Motor Proteins and the Cytoskeleton.* Sinauer Associates, 2001.

[26] A. Irbäck, S. Mitternacht, and S. Mohanty. Dissecting the mechanical unfolding of ubiquitin. *Proc. Natl. Acad. Sci. USA*, 102:13427–13432, 2005.

[27] D. N. Ivankov, S. Garbuzynskiy, E. Alm, K. Plaxco, and D. Baker. Contact order revisited: Influence of protein size on the folding rate. *Protein Sci.*, 12:2057–2062, 2003.

[28] M. Jacob, M. Geeves, G. Holtermann, and F. X. Schmid. Diffusional barrier crossing in a two-state protein folding reaction. *Nat. Struct. Biol.*, 6(10):923–926, 1999.

[29] H. A. Kramers. Brownian motion in a field of force and the diffusion model of chemical reactions. *Physica*, 7(284), 1940.

[30] G. Lee et al. Nanospring behaviour of ankyrin repeats. *Nature*, 440:246–249, 2006.

[31] M. S. Li, M. Kouza, and C.-K. Hu. Refolding upon force quench and pathways of mechanical and thermal unfolding of ubiquitin. *Biophys. J.*, 92:547–561, 2007.

[32] J. Liphardt, B. Onoa, S. B. Smith, I. Tinoco Jr., and C. Bustamante. Reversible unfolding of single RNA molecules by mechanical force. *Science*, 292:733–737, 2001.

[33] J. F. Marko and E. D. Siggia. Statistical mechanics of supercoiled DNA. *Macromolecules*, 28:8759–8770, 1995.

[34] P. Marszalek. Get the experimental data by personal communication.

[35] C. L. Masters and K. Beyreuther. Spongiform encephalopathies: tracking turncoat prion proteins. *Nature*, 388:228–229, 1997.

[36] B. Nolting and K. Andert. Mechanism of protein folding. *Proteins*, 336(41):288–298, 2000.

[37] T. G. Oas and P. S. Kim. A peptide model of a protein folding intermediate. *Nature*, 336:42–48, 1988.

[38] A. F. Oberhauser, P. E. Marszalek, H. P. Erickson, and J. M. Fernandez. The molecular elasticity of the extracellular matrix protein tenascin. *Nature*, 393:181–185, 1998.

[39] E. Paci and M. Karplus. Unfolding proteins by external forces and temperature: The importance of topology and energetics. *Proc. Natl. Acad. Sci. USA*, 97:6521–6526, 2000.

[40] M. J. Parker and S. Marqusee. The cooperativity of burst phase reactions explored. *J. Mol. Biol.*, 293:1195–1210, 1999.

[41] S. E. Radford and C. M. Dobson. From computer simulations to human disease: emerging themes in protein folding. *Cell*, 97:291–298, 1999.

[42] S. E. Radford, C. M. Dobson, and P. A. Evans. The folding of hen lysozyme involves partially structured intermediates and multiple pathways. *Nature*, 358:302–307, 1992.

[43] M. Rief, J. M. Fernandez, and H. E. Gaub. Elastically coupled two-level systems as a model for biopolymer extensibility. *Phys. Rev. Lett.*, 81(21):4764–4767, 1998.

[44] M. Rief, M. Gautel, F. Oesterhelt, J. M. Fernandez, and H. E. Gaub. Reversible unfolding of individual titin immunoglobulin domains by AFM. *Science*, 276:1109–1112, 1997.

[45] M. Rief, F. Oesterhelt, and H. E. Gaub. Single molecule force spectroscopy on polysaccharides by atomic force microscopy. *Science*, 275:1295, 1997.

[46] M. Rief, J. Pascual, M. Saraste, and H. E. Gaub. Single molecule force spectroscopy of spectrin repeats: Low unfolding forces in helix bundles. *J. Mol. Biol.*, 286:553–561, 1999.

[47] M. Rief, J. Pascual, M. Saraste, and H. E. Gaub. Single molecule force spectroscopy of spectrin repeats: Low unfolding forces in helix bundles. *J. Mol. Biol.*, 286:553–561, 1999.

[48] H. Roder, G. A. Elove, and S. W. Englander. Structural characterisation of folding intermediates in cytochrome c by h-exchange labelling and proton NMR. *Nature*, 335:700–704, 1988.

[49] M. Schlierf, F. Berkemeier, and M. Rief. Direct observation of active protein folding using lock-in force spectroscopy. *Biophys. J.*, 93:3989–3998, 2007.

[50] M. Schlierf, H. Li, and J.M. Fernandez. The unfolding kinectics of ubiquintin captured with single-molecule force-clamp techniques. *PNAS*, 101(19):7299–7304, 2004.

[51] J. B. Udgaonkar and R. L. Baldwin. NMR evidence for an early framework intermediate on the folding pathway of ribonuclease a. *Nature*, 335:694–699, 1998.

[52] C. Wagner and T. Kiefhaber. Intermediates can accelerate protein folding. *Proc. Natl. Acad. Sci. USA*, 96:6716–6721, 1999.

[53] T. R. Weikl, M. Palassini, and K. A. Dill. Cooperativity in two-state protein folding kinetics. *Prot. Sci.*, 13:822–829, 2004.

[54] R. Zwanzig. Two-state models of protein folding kinetics. *Proc. Natl. Acad. Sci. USA*, 94:148–150, 1997.