Title
Molecular Simulations of Lipid Bilayers and Membrane Helices Interactions

Author
Benjamini, Ayelet

Publication Date
2013

Peer reviewed|Thesis/dissertation

Molecular Simulations of Lipid Bilayers and Membrane Helices Interactions

By

Ayelet Benjamini

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Chemistry

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Berend Smit, Chair
Professor Phillip L. Geissler
Professor George Oster

Fall 2013

Molecular Simulations of Lipid Bilayers and Membrane Helices Interactions

Abstract

Molecular Simulations of Lipid Bilayers and Membrane Helices Interactions

by

Ayelet Benjamini

Doctor of Philosophy in Chemistry

University of California, Berkeley

Professor Berend Smit, Chair

In this work we use mesoscopic simulations to investigate the short- and medium-ranged interactions between transmembrane helices. First, we develop a coarse-grained model of a generic transmembrane α-helix. We use the geometrical constraints of an α-helix, along with a set of homogeneous hydrophobic beads. We discuss the reasoning behind this modeling and explain why various other plausible models have failed in describing the short-range interactions of transmembrane helices.

We subsequently study the effects of changes in the hydrophobic mismatch of these model helices on their tilt angle. Our simulations show that helices with the same mismatch have similar tilt angles. In addition, our calculations show that the crossing angle of two packed transmembrane helices follows directly from the corresponding tilt angles, and hence the hydrophobic mismatch, of the two helices. A similar analysis of a database of experimental structures of proteins shows a similar conclusion.

We further show the effect of hydrophobic mismatch on the potential of mean force between the helices as well as on the thickness and tilt angle of the lipids surrounding the helices. We study a class of especially long helices, termed *super-positive mismatched helices*, to which the response of the lipids differs and we explain this behavior in terms of lipid organization around the helices.

These coarse-grained simulations are performed using the Dissipative Particle Dynamics (DPD) scheme Performing large time and length scale simulations using this scheme is highly CPU-time consuming. To enhance our simulation capabilities, we develop a novel, massively parallel, simulation algorithm. We introduce this algorithm and the original concepts we developed for parallelizing the DPD scheme. We show that this novel approach provides up to 30 times speed up on a Graphical Processing Unit (GPU) over the non-parallel CPU version.

Finally, we use our parallel algorithm to perform simulation on a large time and length scale system. We investigate the effect of Cholesterol on the bending rigidity of lipid bilayers and show the effect of a phase transition on the bending modulus. Extracting the bending

modulus from simulation requires a large system that enables long-range height fluctuations. Using a fast, parallel simulation algorithm for such a task is therefore crucial.

To my beloved family.

Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

This work has been made possible by the support and advice of many people along the way.

First and foremost, I would like to express my sincere gratitude to my advisor, Professor Berend Smit, who has been a dedicated mentor throughout this process. Providing useful scientific advice, caring about my well-being as well as conveying complete belief in me and my abilities, he had enabled me to grow as an independent critical thinker.

I would also like to thank Dr. Jocelyn M. Rodgers, who has been a wonderful mentor and a close friend. Jocelyn has taught me a lot about the path a research project should take, and has been extremely patient and understanding during the early stages of my research.

I am very thankful for the chance I had to collaborate with other scientists in my group and outside it, all providing new insights and allowing me to explore new territories. Among others, these include: Dr. Frederick de Meyer, Shachi Katira, Dr. Jihan Kim, Dr. Oded Farago, Dr. Padmini Rangamani, Michael Webb, Sheeva Haghighat, Dr. Joseph Swisher, Li-Chiang Lin and Forrest Abouelnasr.

None of this work would have been made possible without the support of my friends in the department, who have helped me struggle through the moral challenges as well as shared my excitement along the way. This refers to my dear friends Patrick Shaffer, Todd Gingrich, Anna Schneider, Eric Sundstrom, Dr. Yael Elmatad, Christopher Ryan, David Limmer, Geoff O'donoghue, Doran Bennet, Dr. Patrick Varilly, Dr. Lester Hedges and Katherine Klymko. A special thanks is also in place to the dedicated crew at the coffee lab, who have made it possible for me to get off on the right foot each morning.

Pursuing my research required great sacrifices from my family across the ocean. Nonetheless, they have been extremely supportive of my work and for that I would like to show my great appreciation. This includes my parents Ronit and Ze'ev; my brother Uri and sister Liat; my grandmothers Martha and Pella; my in-laws Yael and Yoav; as well as the rest of my extended family.

Last but not least, I would like to thank my dear husband, Yuval Benjamini, who has helped me in every step of the way, and in many possible aspects. He has been my biggest supporter and I don't think I could have done it without him.

Chapter 1

Introduction

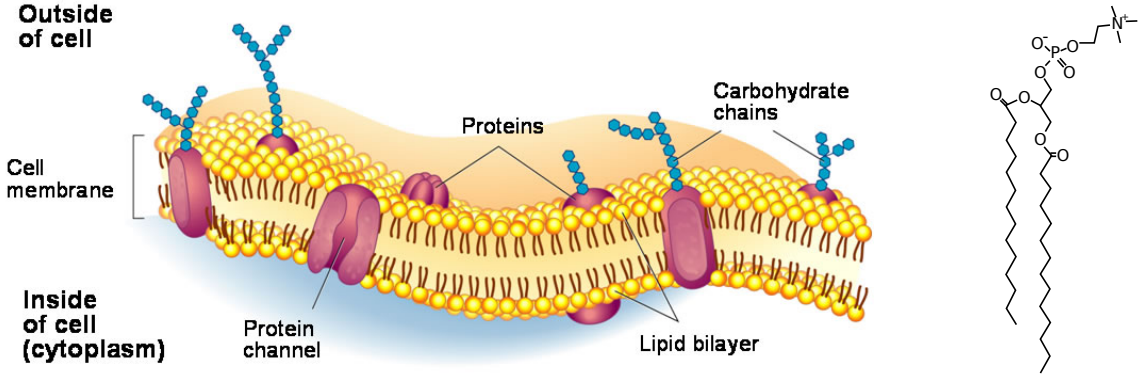
1.1 Biological Membranes

The cell is the elementary structural and functional unit of living organisms. It is typically surrounded by a cell membrane, separating the cell's interior from the outside environment. The basic function of the cell membrane is to protect the cell from its surroundings. It therefore controls the flow of substances in and out of the cell, and serves as a selectively permeable barrier for ions as well as other organic substances. The cell membrane is also involved in many cell processes such as ion conductivity and signaling.

The cell membrane of most living organisms is made of a lipid bilayer. The lipid bilayer consists of two layers of lipid molecules, forming a thin flat sheet (Figure 1.1(a)). Lipids are amphiphilic in nature. They contain a hydrophilic head group and a hydrophobic tail or set of tails (See Figure 1.1(b)). When exposed to water, lipids self-assemble into structures that minimize the exposure of the hydrophobic part of the lipid to water [1]. Lipid bilayer, a structure where two lipid leaflets approach together to create an oily hydrophobic medium in the middle, is one such structure. This oily medium is what limits the transport of many substances through the cell membrane.

The lipid composition of the cell membrane is very diverse. The lipids differ in their head group composition, tail length, and tail saturation. In Eukaryotic cell membranes, for example, lipid head groups are devised of Phosphatidylecholine (PC), Phosphatidylethanolamine (PE), Phosphatidylinositol (PI), Phosphatidylserine (PS), Diphosphatidylglycerol (DPG), Sphyngomyelin (Sph) and others [2]. Cholesterol is also a common lipid in many cell membranes.

The complexity of lipids in cell membranes creates a challenge in studying its function in cellular processes. To that end, a model membrane containing one to three lipid types is often used. Reducing the diversity of lipid composition allows one to better understand the structural and dynamical properties of membrane processes [3, 4]. Both experimental and simulation studies are therefore performed on model membranes. In this work, we focus on a model membrane composed of either one lipid, DMPC (Figure 1.1(b)), or a mixture of two lipids: DMPC and Cholesterol.



(a) (b)

Figure 1.1: (a) Illustration of a lipid bilayer that is part of a cell membrane. The cell membrane separates the cell's interior (cytoplasm) from the outside (extracellular) environment. It may embed different proteins in it. Image from [5]. (b) A typical lipid molecule. Dimyristoylphosphatidylcholine (DMPC) contains a polar phosphate head group and two saturated tails, 18 long carbon chains. DMPC is a bilayer forming lipid.

1.2 Membrane Proteins

In addition to providing separation between the cell exterior and interior, the cell membrane is also the milieu for a vast diversity of proteins. These proteins, termed Membrane Proteins (MPs), are involved in a variety of functions crucial for organism survival [6]. For example, MPs are involved in signaling between the cell's internal and external environment; perform ion transportation across the membrane; allow for cell recognition and adhesion.

Membrane Proteins are classified into three classes based on their topology and interplay with the lipid bilayer [7]: Integral membrane proteins, Peripheral proteins, and Lipid anchored proteins. Integral membrane proteins pass entirely through the membrane. Peripheral proteins are located entirely on one side of the membrane and are non-covalently attached to the polar lipid head groups. Lipid-anchored proteins are covalently bonded to a lipid in the membrane.

Integral membrane proteins constitute a large portion (20-30%) of all encoded proteins [7]. Most of the integral membrane proteins are Transmembrane Proteins (TPs), which span the entire biological membrane and therefore protrude both the extracellular and cytoplasmic sides of the cell membrane. These are classified by their structure into α-helical and β-barrels. In humans, 27% of all proteins are estimated to be α-helical [6]. They are important for many cell processes and include transporters [8], channels [9], and receptors [10].

The functional activity of many membrane proteins depends on the properties of the membrane, such as lipid composition and thickness [11–13]. The membrane environment is therefore speculated to modulate the function of proteins by determining their structure and dynamics.

In this work, we investigate the effects of the lipid bilayer on the structure of embedded proteins. Since α-helical proteins are an important class of TPs, we focus our attention to the effect of the lipid bilayer on the structure of one α-helix, as well as on the structure of paired α-helices.

Previous work [14] showed that the hydrophobic thickness of the lipid bilayer plays an important role in mediating the interactions between integral membrane proteins. Specifically, the difference between the hydrophobic thickness of the membrane and that of the protein was shown to effect the long- and medium-range interaction between proteins in the membrane. In chapters 3 and 4 we extend these concepts by investigating the effect of hydrophobic mismatch on the short-range interaction and configuration of packed α-helices.

1.3 Mesoscopic simulations

To study the interactions of TM proteins within the membrane we use computer simulations. Molecular dynamic simulations use the basic laws of physics to follow the time evolution of investigated structures. It has proven to be a useful tool in many fields of chemistry, biophysics, and material sciences [15–20].

One of the main challenges in biological science is bridging the gap in length and time scales between experimental and computational studies. Chemical and biological processes originate in atomic-level interactions. It would be extremely informative to reach such resolution with experimental techniques. Current experimental techniques, however, reach ms-μs time resolution. On the other hand, molecular dynamic techniques enable atomic-level description of system dynamics. These simulations, however, can only reach up to few hundred ns in time and $\sim 10\ nm$ in length.

One possible method of bridging this gap is the use of Coarse-Grained (CG) or mesoscopic models. CG models treat sets of atoms as one group and waive the atomistic detail in favor of reducing the number of degrees of freedom. This reduction allows faster integration and therefore enables one to reach longer time scales and larger length scales [21–25].

Another advantage of CG model is its generality. A consequence of coarse-graining is the depending on which property one is interested in different degrees of freedom can be important. For example, if one focuses on the structure of a bilayer, diffusion barriers contribute very little and one can develop very successful coarse-grained models that predict the structure of a bilayer well but fail in giving a realistic dynamics. Similarly, a coarse-grained water model will describe real water on a very limited temperature range, hence unless we reparameterize our model for each temperature, such model will show an inconsistent temperature behavior. The goal is to capture only the essential details that will describe the properties we are interested in. Getting a hold on these important features then enables one to tune only the important parameters and investigate the system's response, therefore enabling discovery on the system's biggest leverages.

In this work we make extensive use of CG models. In chapter 2, we build on a previous CG model [14] and change it to include characteristics that are important to short-range

interactions of transmembrane α-helices. In chapter 5, we revise the existing algorithm for CG simulations and present an efficient parallel algorithm for the simulation of CG models on Graphical Processing Units. This enables us to obtain substantial speedup and explore systems of even larger time and length scale than possible by CG models on standard CPUs.

1.4 Overview of this work

In this work, we investigate the short-range interactions between transmembrane helices using mesoscopic simulations. We show our simulation prediction and compare these to results from experimental model membranes as well as to experimental data of natural structures of membrane proteins. Our results show that even the short-range packing of transmembrane α-helices is highly effected by the lipid bilayer it is embedded in. We present a novel algorithm for performing dissipative particle dynamics simulations with substantial speedup. We illustrate the importance of using such algorithm on a model lipid bilayer system with cholesterol.

In Chapter 2 we describe the basic chemistry of the system by presenting the bead types used in our CG model. We further describe the bonded interactions maintaining the structure and topology of this model throughout the simulation. We present the simulation technique used and discuss the details of Dissipative Particle Dynamics and the non bonded forces it employs on the beads in the system. We additionally discuss means to maintain constant surface tension through means of hybrid Monte Carlo - Dissipative Particle Dynamics.

In Chapter 3 we develop a coarse-grained model of an α-helix. This model is developed based on the concepts of describing the minimum-essential characteristics. As such, it contains only two types of beads and the geometry of a typical α-helix. It is therefore a good representation of a generic TM α-helix. We discuss other models that we explored and propose rational for why these models fail to present basic properties of TM helix packing.

In this chapter we survey the concept of hydrophobic mismatch and the different ways it might affect the configuration of a membrane protein as well as the bilayer it is embedded in. We then present the results of our simulations using our novel CG model, revealing how hydrophobic mismatch effects both the single helix configuration as well as the interaction of a pair of TM α-helices. We show that the cross angle (packing angle) between those helices is highly affected by the helices' hydrophobic mismatch. We discuss the origin of this effect as well as deviations from it. We additionally show that hydrophobic mismatch has an important role in determining the potential of mean force between TM helices as well as the tilt of the lipids around embedded helices. This chapter has previously been published in [26].

In Chapter 4 we compare our simulation predictions to experimental membrane protein structures. These structures are expected to be more complex and diverse than model proteins in model membranes. Yet, these experimental packing angles match our predictions and illustrate the effect of hydrophobic mismatch on their distribution. We discuss the analysis of these structures and their comparison to our simulation results. We discuss the similarities as well as the deviations from our predictions and propose potential origins for these deviations.

This chapter has previously been published in [27].

Our simulations in chapters 3 and 4 focus mainly on the interactions of two membrane helices. Yet to fully explore the nature of membrane proteins, the interaction of far more helices must be explored. Currently, we are limited in simulation resources. Chapter 5 is dedicated to the development of a new massively parallel algorithm for Dissipative Particle Dynamics. Employing this algorithm allows for substantial speedup in the simulation technique and for the exploration of larger systems and longer time scales.

In Chapter 5 we discuss the specific challenges of adopting DPD to efficiently run on a GPU, and how those were overcome. The base units (*kernels*) of the algorithm are discussed in detail. We finally present the results for the speedup gained by employing this algorithm in systems of both bonded as well as non-bonded species.

Finally, in Chapter 6, we utilize our novel GPU algorithm to explore the effect of Cholesterol on the bending rigidity of DMPC bilayers. The bending modulus of various DMPC/Cholesterol bilayers is extracted from its undulation spectrum. To obtain the full spectrum a very large system is required, and simulation must be performed for a very large number of time steps to avoid correlations in the lowest mode undulations. Using a fast parallel simulation scheme is therefore crucial for obtaining the bending modulus accurately.

In this chapter we present the classical theory of obtaining the bilayer bending modulus from its undulation spectrum. We additionally derive the theoretical basis for extracting the bending modulus from the simulation of asymmetric (Cholesterol containing) bilayers. We show that these asymmetric bilayers will always display an apparent surface tension higher than its inherit surface tension. We present results for the bending modulus as a function of Cholesterol concentration and show that a sharp increase in the bending modulus is associated with a phase transition of the DMPC/Cholesterol system.

Chapter 2

Mesoscopic Model and Simulation Method

In this chapter we introduce the Coarse-Gained model that was used in our simulation studies. We discuss the types of interactions present in the model and the Hybrid Monte Carlo - Dissipative Particle Dynamics approach that was used and discuss its details and benefits in the context of Lipid Bilayer simulations. Finally, we discuss the use of reduced units in our simulation, and provide estimated conversions to physical units.

2.1 Introduction

The number of simulation studies of biological systems have grown abundantly in the last couple of decades [15–17, 20, 28, 29]. These studies provide insight into the molecular detail of biological building blocks in their native environment. Most of these studies rely on Molecular Dynamics (MD) techniques [30], which follow the evolution of the positions and bonds of atoms in a modeled molecule. Forces between atoms are defined by a force-field, adhering to the chemical interactions of the specific atoms involved. Time propagation is then performed by integrating the equations of motion.

While MD simulations have proven useful in describing short time scale molecular motions and smaller scale structural properties, they are limited in time and length scale. Additionally, as it uses the specific chemical details of the atoms in each molecule, it provides an insight only into the specific system simulated. If a phenomena captured in MD simulations is part of a more robust characteristic of the molecules involved, it would not be directly apparent by performing these molecule-specific simulations.

Coarse-Grained (CG) methods provide an alternative route to deal with these length and time scale issues [21]. In CG simulations, sets of atoms are grouped together to one bead. The bead composition, topology and interaction parameters are chosen to mimic the inherent chemistry of the atoms. Classical time integration can be performed on the resulting forces between the CG beads, when eliminating the stiffest bond motions [31], or alternative

time propagation methods can be employed - such as brownian dynamics [32] or dissipative particle dynamics [33]. Regardless of time propagation methods, bundling atoms together into groups allows for faster computation time, and provides a method of integrating-out the specific details of the molecules and extracting only the characteristics that are relevant for the studied process.

CG models have therefore been extensively used to probe biologically relevant time and length scale phenomena [14, 22, 23, 34]. These models were shown successful in predicting bulk material properties as well as in describing molecular level phenomena.

2.2 Coarse-Grained Model of a Lipid and α-helix

In this work, we model three system components: water, lipids, and α-helices using a coarse-grained representation of the beads. Each bead represents a set of, on average, three heavy atoms (*e.g.* carbon atoms / water molecules), that are bundled together. Four types of beads are considered:

i. a water like bead, ⓦ, representing three water molecules.

ii. a hydrophilic bead, ⓗ, used to model the lipid head group as well as the top and bottom caps of the helix.

iii. a hydrophobic tail bead, ⓣ, used to model the hydrophobic lipid tail.

iv. a hydrophobic protein bead, ⓟ, used to model the hydrophobic core of a TM helix.

The lipid model used in this work was built to represent dimyristoylphosphatidylcholine (DMPC). It was previously shown that this model reproducs the phase diagram of a bilayer-forming lipid [35]. It includes a head group consisting of three ⓗ beads and two tails each containing five ⓣ beads. We therefore label the model $h_3\,(t_5)_2$ (See Fig. 2.1).

The CG model of a lipid was previously parametrized [14, 35]. The CG model of the α-helix is designed and parametrized by the author of this thesis. Its details and motivation are discussed in Section 3.3.

2.2.1 Non-Bonded Interactions

Non-bonded interactions are set to describe the basic interactions between different bead types. Those interactions are set to mimic, for example, the fact that hydrophobic chemical entities repel water, while hydrophilic groups attract water. All non-bonded interactions are modeled as soft-repulsive:

$$\vec{F}^{C}_{ij}(r_{ij}) = \begin{cases} a_{ij}(1 - r_{ij}/R_c)\,\hat{\mathbf{r}}_{\mathbf{ij}}, & r_{ij} < R_c \\ 0, & r_{ij} \geq R_c \end{cases} \tag{2.1}$$

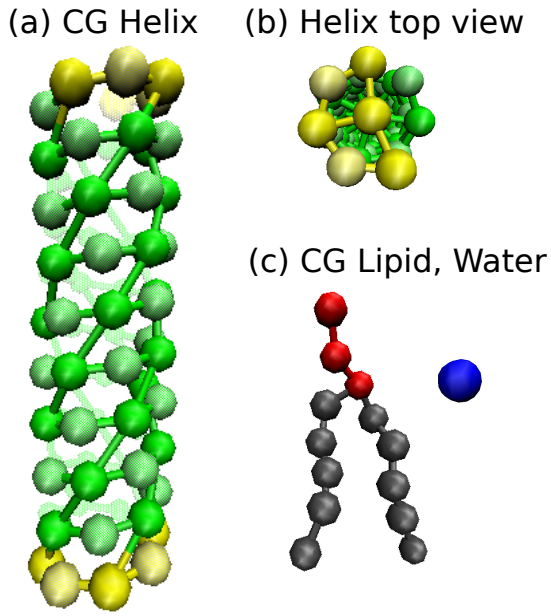


Figure 2.1: Coarse-grained model. (a) CG model of an α-helix assembled from hydrophobic beads at the core and hydrophilic beads at both ends to keep the helix transmembrane. Helical geometry is maintained by harmonic springs, angle springs and dihedral angle potentials of principal helix beads (see Section 3.3.1) (b) Top view of the helix CG model shows the helix central pole that provides excluded volume at the helix interior. (c) CG lipid model includes 3 hydrophilic head groups and two 5 bead hydrophobic tails. Figure 1.1(b) shows the atomistic details. Water is represented explicitly by a single bead. All beads in the system are of the same size and correspond to approximately 3 carbon atoms / water molecules.

Table 2.1: Non-bonded interaction parameters between CG beads.

a	Ⓦ	Ⓗ	Ⓣ	Ⓟ
Ⓦ	25	15	80	120
Ⓗ	15	35	80	80
Ⓣ	80	80	25	25
Ⓟ	120	80	25	25

Note. — Interaction parameters in units of ϵ_0.

Where $r_{ij} = |\vec{r}_i - \vec{r}_j|$ is the distance between beads i and j, and $\hat{\mathbf{r}}_{\mathbf{ij}} = (\vec{r}_i - \vec{r}_j)/r_{ij}$ is a unit vector connecting particle j and particle i. The strength of the repulsion is set by the repulsion parameter, $a_{ij} > 0$. R_c is the cutoff radius for the soft repulsive interaction. It is set to $R_c = 1\ \mathrm{d}_0$ (d_0 is the reduced unit of length, see Section 2.5) in our simulations.

The use of soft repulsive non-bonded interactions is a common practice in Dissipative Particle Dynamics (see Section 2.3). Groot *et al.* [33] introduced a typical value for the self-repulsion parameter, the repulsion parameter between two identical water-like beads, to be $a = 25\ \epsilon_0$ (ϵ_0 is the reduced unit of energy). In our model, we use basic parameterization for most of the interactions. We use a smaller repulsion value for interacting beads that are more attractive than the Ⓦ - Ⓦ interaction. This includes, for example, Ⓦ - Ⓗ interaction ($a = 15\ \epsilon_0$). We set a larger repulsion value for bead interactions that are more repulsive than the Ⓦ - Ⓦ interaction. This includes hydrophobic beads interacting with water, such as Ⓦ - Ⓣ and Ⓦ - Ⓟ interactions. The protein core bead type, Ⓟ, is parametrized to have identical interactions as the lipid tail bead, Ⓣ, within the membrane. We increase its repulsion with the water bead, Ⓦ, (from $a_{\text{Ⓦ}\text{Ⓣ}} = 80\ \epsilon_0$ to $a_{\text{Ⓦ}\text{Ⓟ}} = 120\ \epsilon_0$) to ensure the protein core remains within the membrane.

This parametrization follows previous work by de Meyer *et al.* and others [14]. The complete (symmetric) matrix of non bonded parameters is provided in Table 2.1.

2.2.2 Bonded Interactions

We apply bonded interactions between beads of the same molecule to maintain the topology and structure of the modeled molecule. Bonded interactions are elastic in nature, and can control the strength and flexibility of a given bond. These can be applied to control the distance, angle and dihedral angle of same-molecule beads.

Specifically, we use the following forms of bonded interactions:

$$U_{bond}(r) = \frac{1}{2}K_b(r - r_{eq})^2 \tag{2.2a}$$
$$U_{angle}(\varphi) = \frac{1}{2}K_a(\varphi - \varphi_{eq})^2 \tag{2.2b}$$
$$U_{dihedral}(\chi) = \frac{1}{2}K_d[\cos(\chi) - \cos(\chi_{eq})]^2 \tag{2.2c}$$

Where $U_{bond}(r_{ij})$ describes the harmonic potential controlling the distance between consecutive beads i, j; $U_{angle}(\varphi_{ijk})$ describes the harmonic potential controlling the angle between beads i, j, k; and $U_{dihedral}(\chi_{ijkl})$ describes the dihedral angle potential between beads i, j, k, l, as adopted from the DLPOLY manual [36].

Bonded interaction parameters for the lipid model were obtained from all-atom MD simulations [37]. The parameters for the harmonic distance potential (Eq. 2.2a) were globally set to $r_{eq} = 0.7\,\mathrm{d_0}$ and $K_b = 100\,(\epsilon_0/\mathrm{d_0}^2)$. The parameters for the harmonic angle potential between consecutive bonds in the lipid tails were set to $\varphi_{eq} = \pi\ rad$ and $K_a = 6\,\epsilon_0/rad^2$, favoring the straightening of the lipid tails, whereas $\varphi_{eq} = \pi/2\ rad$ and $K_a = 3\,\epsilon_0/rad^2$ were applied for the junction angle between the lipid tails and head group, maintaining a close-to-right angle.

Bonded interaction parameters for the helix model are described in more detail in Section 3.3.2 and Table 3.1. Bonded interaction parameters are provided in reduced length and energy units, as described in Section 2.5.

2.3 Dissipative Particle Dynamics

Dissipative Particle Dynamics (DPD) has been introduced by Hoogerbrugge and Koelmam [38] as a simulation technique for simulating CG models of complex fluids. It has several advantages over Brownian Dynamics, mostly in reproducing the correct hydrodynamic behavior of fluids. Español and Warren [39] showed that DPD can be represented by a basic Fokker-Planck equation, and that one may chose parameters such that the dissipation-fluctuation theorem is maintained throughout the simulation.

The basic elements of DPD are similar to Molecular Dynamics (MD) simulations. Particles in the system interact and by computing the forces on each particle and solving Newton's equations of motion, we can follow the evolution of a system. Additionally, the use of soft interactions allows for larger integration time steps. Combining that fact with the inherent coarse-graining of each bead, one can reach larger time and length scales in DPD compared with regular MD.

The forces applied on each particle include a conservative force (bonded and non-bonded), $\vec{F}^C$, just like MD. Two other forces combine in DPD to account for the removal of degrees of freedom. One is a dissipative force, $\vec{F}^D$, set to mimic the friction between two beads. The other is a random force, $\vec{F}^R$, representing the possible collisions that may occur between atoms within each bead. All forces represent soft interactions, with finite repulsive barrier at short distances. These interactions mimic the non-zero probability of CG beads to interpenetrate.

While impossible for atoms, coarse-grained beads containing several atoms, may interpenetrate in different configurations.

The dissipative force depends on the velocity difference between the two beads, as well as on their position. Beads that are close together and have a positive relative velocity, will have a higher chance of colliding. The dissipative force represents this possible collision with a larger force acting on the two beads in the direction opposite to their connecting vector (see Eq. 2.3). Both the dissipative and random forces are distance dependent, and are set to zero beyond the beads' inherit cutoff distance, R_c. Their respective equations are:

$$\begin{aligned} \vec{F}_{ij}^{D}(r_{ij}) &= -\eta\omega^{D}(r_{ij})(\hat{\mathbf{r}}_{\mathbf{ij}} \cdot \vec{v}_{ij})\,\hat{\mathbf{r}}_{\mathbf{ij}} \\ \vec{F}_{ij}^{R}(r_{ij}) &= \sigma\omega^{R}(r_{ij})\zeta_{ij}\,\hat{\mathbf{r}}_{\mathbf{ij}} \end{aligned} \tag{2.3}$$

Where $\vec{v}_{ij} = \vec{v}_i - \vec{v}_j$ is the velocity difference between particles i and j. η is the friction coefficient and σ is the noise amplitude. $\zeta_{ij} \sim \mathcal{N}(0,1)$ is a random number normally distributed with mean zero and variance 1.0.

One of the requirements of DPD is the conservation of momentum [39], ensuring the correct hydrodynamic behavior of fluids even in small length scales. This requirement manifest it self in the requirement that:

$$\begin{aligned} \vec{F}_{ij} &= -\vec{F}_{ji} \\ \Rightarrow \zeta_{ij} &= \zeta_{ji} \end{aligned} \tag{2.4}$$

Another key aspect of DPD is the spatial relation between the random and dissipative forces. Español and Warren [39] show that the correct equilibrium density of states that corresponds to the application of the conservative force may be recovered, with a well defined temperature, if the folowing requirements are achieved:

$$\begin{aligned} \omega^{D}(r) &= \left[\omega^{R}(r)\right]^2 \\ \sigma^2 &= 2\eta \mathrm{k_B T} \end{aligned}$$

We set the noise amplitude $\sigma = 3.0$, and set the friction coefficient accordingly to maintain this requirement. For computational convenience we chose the same functional dependence for the dissipative and random forces as for the conservative force. Namely:

$$\omega^{R}(r) = \begin{cases} (1 - r/R_c)\,\hat{\mathbf{r}}_{\mathbf{ij}}, & r < R_c \\ 0, & r \geq R_c \end{cases} \tag{2.5}$$

$$\omega^D(r) = \begin{cases} (1 - r/R_c)^2 \, \hat{\mathbf{r}}_{\mathbf{ij}}, & r < R_c \\ 0, & r \geq R_c \end{cases} \tag{2.6}$$

The propagation of system components in time is done using a modified version of the Velocity-Verlet algorithm [33]. The reason for using a modified version of the common MD integrator arrises from the basic differences in the nature of the force between DPD and MD. In classical MD the force on a given particle depends only on the positions of the particles surrounding it. This allows for a separate integration of the force and velocity on a given particle. In DPD, however, the force on a given particle depends on the positions as well as the velocities of the particles surrounding it. Therefore, one must be careful when performing concurrent time integration on the particles' velocities and forces. The modified version of the Velocity-Verlet algorithm includes the following steps of time integration:

$$\begin{aligned} \vec{r}_i(t + \Delta t) &= \vec{r}_i(t) + \Delta t \vec{v}_i(t) + \frac{1}{2}\Delta t^2 \vec{f}_i(t) \\ \acute{\vec{v}}_i(t + \Delta t) &= \vec{v}_i(t) + \lambda \Delta t \vec{f}_i(t) \\ \vec{f}_i(t + \Delta t) &= \vec{f}_i\left(\vec{r}(t + \Delta t), \acute{\vec{v}}(t + \Delta t)\right) \\ \vec{v}_i(t + \Delta t) &= \vec{v}_i(t) + \frac{1}{2}\Delta t \left(\vec{f}_i(t) + \vec{f}_i(t + \Delta t)\right) \end{aligned} \tag{2.7}$$

These equations are identical to the original Velocity-Verlet algorithm [40] for $\lambda = 0.5$ and $\vec{f}\,(\vec{r}, \vec{v}) \equiv \vec{f}\,(\vec{r})$. The predicted velocity, $\acute{\vec{v}}$, is used to calculate the (velocity dependent) force. The final velocity for each time step is then updated at the end of the time step based on the force.

2.4 Monte-Carlo Approach

Lipid bilayers have several degrees of freedom to adjust to changes in their environment (temperature, composition *etc.*). One of the most common changes observed are in the area per lipid. In nature, a vesicle will expand or contract to allow such changes in areal density. In simulation, we must use an appropriate ensemble to accommodate for such changes as well. We therefore wish to simulate our system in an ensemble that allows for both changes in area as well as in volume, such that the equilibration would mimic natural conditions. This is achieved using the $NP\,\gamma\,T$ensemble.

The free energy differential corresponding to that ensemble is achieved by a simple

Legendre transformation [41]:

$$d\mathcal{F} = -TdS + VdP - \gamma dA + \mu_{lip}dN_{lip} + \mu_w dN_w \tag{2.8}$$

Where γ is the surface tension, and μ_{lip}, μ_w are the chemical potentials for the lipid and water components, respectively, conjugated to the number of lipid and water molecules in the system (N_{lip}, N_w).

Several approaches for simulating this ensemble have been used [42–44]. Rodgers and Smit [41] show that to correctly sample in this ensemble, one must apply two types of moves, in V and in $L_\parallel$ with the following acceptance criteria. (Here we refer to the simulation box length along the surface of the bilayer as $L_\parallel$ and to the box length in the direction perpendicular to the bilayer surface as $L_\perp$).

Move 1: Move in V holding $L_\parallel$ constant

1. Chose a change in the volume: $\Delta V \in [-(V_{step}/2), (V_{step}/2)]$.
2. Calculate the resulting volume, and ensure it is positive: $V_n = V_o + \Delta V$.
3. Set $L_\perp = V_n/L_\parallel^2$ such that $L_\parallel$ and the area, A, are kept constant.
4. Accept this move with probability:

$$\text{acc}\left(V_o \to V_n \,|L_\parallel\right) = \min\left[1, \left(\frac{V_n}{V_o}\right)^{N-1} \exp\left(-\beta P\Delta V - \beta\Delta U\right)\right] \tag{2.9}$$

Here V_{step} is a simulation-specific step size for volume changes; β is the inverse temperature; P is the applied pressure; $N = N_{lip} + N_w$ is the total number of molecules in the system; and ΔU refers to the energy change of the system after rescaling all head groups of all molecules according to the new box size.

Move 2: Move in $L_\parallel$ holding V constant

1. Chose a change in the box length: $\Delta L_\parallel \in [-(L_{step}/2), (L_{step}/2)]$.
2. Calculate the resulting box length, and ensure it is positive: $L_{\parallel,n} = L_{\parallel,o} + \Delta L_\parallel$.
3. Calculate the resulting change in area: $\Delta A = L_{\parallel,n}^2 - L_{\parallel,o}^2$
4. Set $L_\perp = V/L_{\parallel,n}^2$ such that the volume, V, is kept constant.

5. Accept this move with probability:

$$\mathrm{acc}\left(L_{\parallel,o} \to L_{\parallel,n} \,|V\right) = \min\left[1, \left(\frac{L_{\parallel,n}}{L_{\parallel,o}}\right) \exp\left(\beta\gamma\Delta A - \beta\Delta U\right)\right] \tag{2.10}$$

Here L_{step} is a simulation-specific step size for box length changes, and γ is the applied surface tension. Other symbols are defined similarly to **Move 1**.

When performing **Move 1** or **Move 2** we only scale the head groups of each molecule (lipid/ water) while keeping the rest of the beads in the molecule bonded with the same bond lengths and orientations. This results in changes only in the non-bonded interaction at each proposed MC move, and increases the probability that these proposed moves are accepted. Since only one bead out of each molecule is being rescaled, the number of particles, N, corresponds only to the number of molecules in the system.

2.4.1 Hybrid DPD-MC

We combine the dynamical advantages of DPD with the ability of MC to equilibrate the system in the right ensemble using a hybrid approach [25, 44, 45]. Namely, our simulation is a Markov Chain of states. Each state is defined by its system box size. Within each state, the system is propagated through time with the prospects of sampling typical equilibrium configurations within that state. As part of the Markov Chain Monte Carlo (MCMC) scheme [46], this approach, when done correctly, would provide good statistical sampling from the system's equilibrium distribution of configurations.

In practice, at each time step, a Move, **Move** i, is chosen with probability p_i. The possible moves include:

- **Move 1** (Pressure move): Move in V holding $L_\parallel$ constant.
- **Move 2** (Surface tension move): Move in $L_\parallel$ holding V constant.
- **Move 3** (DPD move): A DPD trajectory of N_{DPD} substeps.

Moves 1 and 2 are described above. To maintain detailed balance in performing **Move 3**, we ensure the number of DPD substeps is chosen at random, $N_{DPD} \sim \mathcal{U}[0, 50)$. A typical trajectory in the $NP\,\gamma\,T$ ensemble will perform pressure moves with probability $p_1 = 0.2$, surface tension moves with probability $p_2 = 0.2$ and DPD moves with probability $p_3 = 0.6$.

2.5 Reduced Units

A typical approach within the realm of molecular simulations is to use reduced units [45]. In CG simulations, the conversion between reduced and physical units is non trivial. One origin for this difficulty is the fact that the mapping of atoms into beads is not exact. In our

model, for example, we bundle every three heavy (non-Hydrogen) atoms into a bead. These may result in beads of different masses.

We approach this difficulty by setting a standard bead mass, and aligning all other units accordingly. In that, we follow Groot and Rabone [47], and set our mass and length reduced units according to the size of three water molecules. Yet the reader must remember that small discrepancies in the units conversion may arise because of slightly different choices in the coarse-graining procedure. These differences are usually later revisited when matched with experimental thermodynamic data.

Mass

The reduced unit for mass is set to be the mass of one bead. The standard bead is defined to be the one representing $N_m = 3$ water molecule. This therefore sets the reduced unit of mass, m^* to be $N_m \cdot m_{H_2O} \simeq 9 \cdot 10^{-26}\ kg$.

Length

The length scale for the simulation is determined by the volume occupied by the beads at a fixed temperature $\mathrm{T}^* = 1.0$. A cube of unit volume, $1\ \mathrm{d}_0^3$, represents ρN_m water molecules, where ρ is the number density (number of beads per unit of volume). We set ρ to be 3.0 in our simulations. Using the fact that a water molecule occupies an approximate volume of $30\ \text{Å}^3$ we have:

$$\begin{aligned} \mathrm{d}_0^3 &= \rho N_m \cdot 30\ \text{Å}^3 \\ \Rightarrow \mathrm{d}_0 &= 6.4633\ \text{Å} \end{aligned} \tag{2.11}$$

Temperature

There are different methods to derive the mapping of reduced temperature to physical units scale. As we are interested in providing qualitative structural description of biological bilayers, mapping the temperature scale to the thermodynamic properties of bilayers would be the best approach. Here we assume linear scaling between the reduced and physical temperature, and estimate the regression parameters from the main and pre-transition temperatures of DMPC bilayer. See more details in ref. [14].

The resulting temperature scale is:

$$T(^\circ C) = 108.75\ \mathrm{T}^* - 8.7 \tag{2.12}$$

We set the mapping of energy $1\ \epsilon_0 = 1\ \mathrm{k_B T}$ to physical units, to follow from this temperature conversion. See Table 2.2.

Table 2.2: Conversion of reduced units to physical units.

Unit Symbol	Unit Name	Value	Conversion to SI units
m^*	Mass	$3 \cdot m_{H_2O}$	$9 \cdot 10^{-26}\ kg$
d_0	Length	6.46 Å	$6.46 \cdot 10^{-10}\ m$
ϵ_0	Energy	$1\ \mathrm{k_B T}$	$5.15 \cdot 10^{-21}\ J$
τ	Time	$\cdots$	$\sim 2.7 \cdot 10^{-12}\ sec$

Time

One may conceive different mapping criteria for the unit of time. The straight-forward approach would be to derive it based on its inherit relationship with other units of the system. Meaning:

$$\tau = \mathrm{d}_0 \sqrt{\mathrm{m}^*/\epsilon_0} \tag{2.13}$$

The advantages of defining the unit of time in such a way is the consistency it provides between all the different reduced units. We use this method to determine the value of τ in Table 2.2. The caveats in this approach originate in the mismatch between dynamical and structural properties. It has been shown that using CG model parametrized by thermodynamic data results in faster diffusion time compared to the times measured in physical systems [48]. Another approach for setting the mapping of the reduced time is therefore using the diffusion constants of the system components. Groot *et al.*, used the experimental value for the self diffusion of water to set the time unit [33]. This yields a reduced time unit of $\tau \sim 5\,ps$, almost twice as fast.

Chapter 3

Molecular Simulation study of Membrane Helices Interactions

In this chapter we study the influence of a membrane environment on the packing of Transmembrane (TM) helices. Using a coarse-grained (CG) model that captures the essential structure and interactions of membrane α-helices we show that the inter-helical crossing angle is mostly impacted by the tendency of the membrane to decrease the distance of the two helix tips in the plain of the membrane. Combined with a hydrophobic related driving force to tilt the helix in the membrane, a non-trivial crossing angle distribution is formed. As we use a model that only captures the structure of a helix and its hydrophobic nature, we gain insight into the influence of the membrane on the interaction of two helices, without the effect of specific residues.

3.1 Introduction

Association of Transmembrane (TM) α-helices is the basis of many cell processes. Among the membrane proteins (MPs) that require helix-assembly to function are ion channels [49], GPCR membrane receptors [50], and integrins [51]. To probe the complex association of many α-helices, the dimerization of two α-helices is typically used as a case study. Understanding the factors determining helix-helix packing has therefore been the aim of several studies [52–60].

Most models developed to explain helix-helix packing rely on specific residue interactions between the helices [56–60]. These theories predict that specific residue interactions control the packing angle between the two α-helices in a TM helix pair. In most of these models the role of the membrane is limited to constraining the proteins diffusion in the two-dimensional plane of the membrane. It is, however, well-known that the membrane plays an important role determining the tilt of TM helices [11, 34, 61–65] and also in mediating the interactions between TM helices [14, 63, 66–70]. It is reasonable to expect that the membrane also plays a role in the helix-helix packing.

Previous CG models developed in our group showed the importance of lipid mediated interactions in controlling the long and medium range interaction of TM proteins [14]. These models were able to predict the potential of mean force between a pair of TM proteins embedded in a lipid bilayer based on the protein's hydrophobic length. Using these model for short range interactions might however lead to false conclusions on the packing of TM helices, due to the simplified nature of these models.

Here, we develop a more detailed CG model of an α-helix, that will allow us to simulate the short range interactions of TM helices. Our model relies on the previous model by de Meyer *et al.* [14] in its basic parameterization, while providing more geometric details. This type of modeling enables us to capture both the short range packing of TM helices as well as longer range lipid mediated interactions.

3.2 Hydrophobic Mismatch

TM α-helices are characterized by a long sequence of hydrophobic residues. The interaction of TM α-helices with the membrane they are embedded is influenced by the hydrophobic core of the helix. To a first order, the hydrophobicity of the helix core is what enables it to be a TM protein, and distinguishes it from a water-soluble protein. There is a strong tendency to shield the hydrophobic helix residues from the water environment on both sides of the lipid bilayer, and therefore there is a driving force for the helix to be embedded in the bilayer.

Those same driving forces that cause the helix to be a TM helix (*i.e.* shielding its hydrophobic core from the water) are causing other, second order effects, which may be thought of as a size dependent perturbation. These effects relate to the *Hydrophobic Mismatch* between a helix and the surrounding membrane. The hydrophobic mismatch is defined as the difference between the hydrophobic length of the helix, d_{H}, and the hydrophobic thickness of a membrane, d_{L}:

$$\Delta d = d_{\mathrm{H}} - d_{\mathrm{L}} \tag{3.1}$$

The concept of hydrophobic mismatch as well as the helix hydrophobic length and membrane hydrophobic thickness are illustrated in Figure 3.1. When $d_{\mathrm{H}} = d_{\mathrm{L}}$ ($\Delta d = 0$), we say that there is *zero* hydrophobic mismatch. The size of the helix then fits the membrane perfectly (Figure 3.1(b)). When $d_{\mathrm{H}} > d_{\mathrm{L}}$ ($\Delta d > 0$), we say that there is a *positive* hydrophobic mismatch. In positive mismatch situations, as a generalization, the system would search for alternative means to shield the hydrophobic parts of the helix core that are exposed to water (Figure 3.1(c)). When $d_{\mathrm{H}} < d_{\mathrm{L}}$ ($\Delta d < 0$), we say that there is a *negative* hydrophobic mismatch. In negative mismatch situations, the system would search for means to shield the

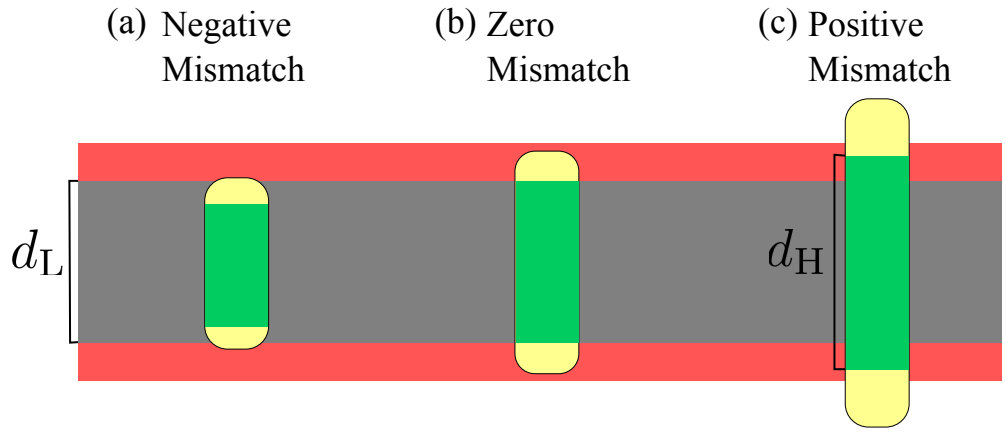


Figure 3.1: Illustration of Hydrophobic Mismatch. (a) Negative mismatch, $d_H < d_L$ (b) Zero Mismatch, $d_H = d_L$ (c) Positive Mismatch, $d_H > d_L$. The membrane is represented by a continuous hydrophobic layer of tails (grey) and two layers of hydrophilic head groups (red). The helices are represented by a hydrophobic core (green) and two hydrophilic caps (yellow).

lipid tails that are exposed to the water (Figure 3.1(a)). These three distinct situations have a great effect on the configurations of TM proteins.

Hydrophobic mismatch has been studied extensively, both in experiments of synthetic peptides and in simulations. It was shown to be an important factor in helical tilt within the membrane [11, 61–65] and have an impact on oligamerization of helical peptides [14, 63, 66–70]. In general, three major degrees of freedom have been shown to be important in the presence of hydrophobicly mismatched helices:

(a) Changes in single helix configuration. A helix might tilt, bend, stretch, kink, or otherwise change its orientation in the membrane as to better fit the mismatched conditions. (Figure 3.2(b,c,d,h))

(b) Helix aggregation. Helices may bundle together into bigger clusters. (Figure 3.2(e,i))

(c) Bilayer deformation. The lipid bilayer around a helix might stretch or compress to accommodate different mismatch conditions. (Figure 3.2(f,j))

As a schematic representation, we distinguish between several major effects of hydrophobic mismatch. Based on this schematic differentiation we can predict a-priori which of the above mentioned conditions might come into play in each case. We describe these prediction for each mismatch case in the following sections 3.2.1, 3.2.2, and 3.2.3.

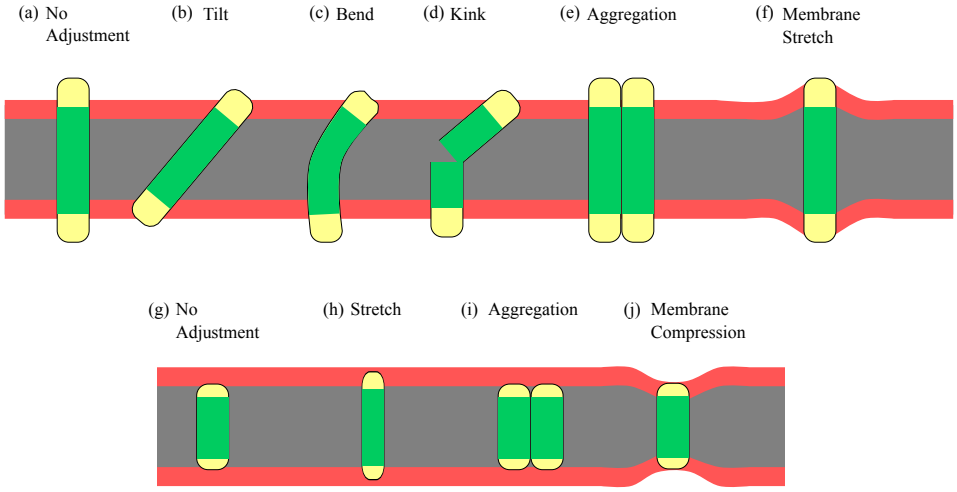


Figure 3.2: Possible adjustments to hydrophobic mismatch for positive (a-f) and negative (g-j) hydrophobic mismatched helices. (a) Shows the configuration of a positive mismatched helix with no adjustments. In response to the mismatch, this helix might tilt (b), bend (c), form a kink (d), or aggregate with other helices (e). Additionally, the membrane can stretch around it (f). (g) Shows the configuration of a negative mismatched helix with no adjustments. In response to the mismatch, this helix might stretch (h) or aggregate with other helices (i). Additionally, the membrane can compress around it (j). In practice, a combination of several possible adjustments might be the resulting configuration

3.2.1 Zero Hydrophobic Mismatch

Zero mismatched helices are illustrated in Figure 3.1(b). In this scenario a helix is embedded in a membrane whose hydrophobic thickness matches the hydrophobic thickness of the helix. There is therefore perfect shielding of the helix core from the hydrophilic head groups of the membrane and the water. The lipid tails are also shielded from the water by the presence of a hydrophobic helix. We therefore expect none of the three accommodation methods described above to be important.

In experiments and simulations of zero mismatched helices, very small tendency towards helix tilt is observed [11, 62]. Sparr *et al.* [68] has also shown that there is very small tendency towards oligamerization in near-zero mismatched synthetic peptides.

3.2.2 Positive Hydrophobic Mismatch

Positive mismatched helices are illustrated in Figure 3.1(c). A positively mismatched helix is embedded in a membrane whose hydrophobic thickness is smaller than the hydrophobic length of the helix. As a result, part of the helix' hydrophobic core is exposed to the hydrophilic head groups of the lipids and to water. Several of the mechanism described above might resolve this unfavorable exposure:

1. The helix might tilt. Tilting may decrease the length of the helix that is exposed to water. Helix tilt angles were indeed observed to be correlated with positive mismatch in several synthetic peptides [11, 65].

2. The helix might kink or bend to decrease its length. Kinks are more common in TM helices than they are in water soluble helices, and are often produced by a Proline residue [71].

3. Helices may aggregate. Aggregation of helices may decrease the area of exposed hydrophobic helix cores. It may also decrease the line tension between helices and the surrounding bilayer. This process is envisioned to be similar to the well known *Hydrophobic Effect* [72] in which hydrophobic substances aggregate in water driven by the tendency to decrease the area of exposed hydrophobic surface. Aggregation of synthetic WALP peptides has been experimentally observed for positive mismatched peptides [68].

4. The membrane may thicken. The lipids around a positively mismatched helix may change their configuration to better shield the exposed areas of the helix.

3.2.3 Negative Hydrophobic Mismatch

Negative mismatched helices are illustrated in Figure 3.1(a). A negatively mismatched helix is embedded in a membrane whose hydrophobic thickness is larger than that of the helix. Several hydrophobic lipid tails might therefore become exposed to hydrophilic residues

or water. Several hydrophilic helix residues might additionally become exposed to the hydrophobic core of the membrane. We envision several accommodation solutions out of the possible arrangements described :

1. The helix might stretch. Artificial lengthening of the helix core might reduce the mismatch with the surrounding lipids. However, the secondary structure of an α-helix tends to be fairly stable and not much stretching is typically observed.

2. Helices might aggregate. As with positively mismatched helices, aggregation of negative mismatched helices might localize the perturbation and minimize the line tension with the surrounding lipids.

3. The membrane may shrink. The lipids around a negatively mismatched helix may change their configuration to shield their tails from being exposed to water.

In our model described below, we take into account the importance of hydrophobic mismatch and study its effect on the direct and lipid mediated packing of TM helices. Most existing models for studying helix-helix packing focus solely on the interacting residues on the surface of the two interacting helices. Our approach focusses on obtaining an understanding of the interplay between helix interactions and the embedded bilayer.

3.3 Mesoscopic Model of α-helix

A mesoscopic model of an α-helix was derived based on the reductionist approach, starting with the minimum building blocks that describe the characteristics of a TM helix. A cylindrical model by Venturoli *et al.* [25] successfully described medium and long range interactions between TM helices [14]. The model includes a representation of each TM α-helix as a cylinder, with hydrophilic core and two hydrophilic caps. This model, however, is not successful in describing the short range interactions between TM helices. In section 3.7 we describe in more detail the results that led us to modify the cylinder model. Here, it is suffice to say that we have learned that the helical geometry is a crucial factor in the short range interactions of TM α-helices. Our mesoscopic model was therefore extended beyond the cylindrical model to include hydrophobic beads laid out in a helical geometry at the core of a modeled α-helix.

3.3.1 Helix Model Topology

Our transmembrane helix model contains three structural features:

i. N_{p} principal beads (P) that follow the outer radius structure of an α-helix backbone, represented in Figure 2.1(a) by opaque beads. All structural bonds are associated with those beads.

ii. $N_{\mathrm{s}} = N_{\mathrm{p}} - 1$ secondary beads (S) located in between every adjacent pair of principal beads and provide excluded volume along the helix exterior. These are represented in Figure 2.1(a) by transparent beads.

iii. $N_{\mathrm{c}} = N_{\mathrm{p}}/3$ central beads (C) located along the central axis of the helix and provide excluded volume at the helix interior. These are shown in the center of Figure 2.1(b).

To build our TM helix model we use the geometric parameters of a typical α-helix, by positioning our beads at the outer core of the helix. This mimics the helical structure by focussing on the surface accessible for interaction with other helices. We use a typical outer radius of $R_{\mathrm{h}} = 6$ Å [73]. The angle and height pitch between consecutive principal beads follow the typical α-helix parameters: $\Delta\eta = 100\,^{\circ}$; $\Delta Z = 1.5$ Å [74].

To generate the positions of N_{p} principal helix beads we chose an initial angle η_0 at random. We then place the first bead of the helix at position

$$\vec{r_0} = \Big(R_{\mathrm{h}} \times \cos(\eta_0), R_{\mathrm{h}} \times \sin(\eta_0), 0\Big).$$

The position of each principal bead thereafter, $\mathsf{P}_{i=1\ldots N_{\mathrm{p}-1}}$, is subsequently determined by

$$\vec{r_i} = \Big(R_{\mathrm{h}} \times \cos(\eta_0 + \Delta\eta \cdot i), R_{\mathrm{h}} \times \sin(\eta_0 + \Delta\eta \cdot i), \Delta Z \cdot i\Big).$$

We next position the secondary beads in between each pair of adjacent principal beads such that for each bead, $\mathsf{S}_{j=0\ldots N_{\mathrm{s}}-1}$;

$$\vec{r_j} = \Big(R_{\mathrm{h}} \times \cos(\eta_0 + \Delta\eta \cdot (j + \frac{1}{2})), R_{\mathrm{h}} \times \sin(\eta_0 + \Delta\eta \cdot (j + \frac{1}{2})), \Delta Z \cdot (j + \frac{1}{2})\Big)$$

Lastly, the central beads are positioned along the main axis of the helix. A central bead's z-component is determined by the average position of the three principal beads it is adjacent to. This yields for each central bead $\mathsf{C}_{k=0\ldots N_{\mathrm{c}}-1}$

$$\vec{r_k} = \Big(0, 0, \Delta Z \cdot (3k + 1)\Big)$$

Helix bead types are assigned such that the core of the helix consists of hydrophobic ⓟ beads, and its edges are hydrophilic ⓗ beads. This assignment ensures that both edges of the protein remain in the water or membrane head group phase, thus preventing unphysical configurations such as a helix lying within the membrane perpendicular to the membrane normal. Each hydrophilic edge of the helix consists of three principal beads, two secondary beads, and one central bead.

3.3.2 Helix Model Bonded Interactions

Harmonic bond forces, derived from U_{bond} (2.2a), describe the force used to control the inter-bead distance. These are applied between:

i. each principal bead P_i and its two adjacent secondary beads $\mathsf{S}_{j=i-1,i}$

ii. each two adjacent central beads C_k and C_{k+1}

iii. each central bead C_k and its three adjacent principal beads $\mathsf{P}_{i=3k,3k+1,3k+2}$, and

iv. between each principal bead P_i and its consecutive P_{i+4} thus mimicking α-helix hydrogen bonding.

Harmonic bond constants are globally assigned to $K_b = 100\,\epsilon_0/\mathrm{d_0}^2$ continuing our assignment of bonded interaction parameters in the lipid model (reduced units of energy (ϵ_0) and length ($\mathrm{d_0}$) are defined in Section 2.5). Equilibrium distances are derived from the helical structure and can be found in Table 3.1. Since the role of central beads is to provide excluded volume and not constrain structure, interactions coupling the central pole and the helical structure have a decreased force constant of $K_b = 20\,\epsilon_0/\mathrm{d_0}^2$. The equilibrium distance for this interaction changes based on the position of the principle bead, as described in Table 3.1.

Harmonic-angle forces, derived from U_{angle} (2.2b), are used to control the stiffness of the helical structure. These are applied between:

i. each three consecutive principal beads: $(\mathsf{P}_i, \mathsf{P}_{i+1}, \mathsf{P}_{i+2})$

ii. each three principal beads surrounding a hydrogen bond: $(\mathsf{P}_i, \mathsf{P}_{i+4}, \mathsf{P}_{i+5})$, and

Table 3.1: Bonded interaction parameters in the coarse-grained TM helix model.

Interaction	Interacting Beads	Force constant	Equilibrium value
U_{bond}		$K_b\ (\epsilon_0/\mathrm{d_0}^2)$	$r_{eq}\ (\mathrm{d_0})$
	P_i S_{i-1}	100	0.7936
	P_i S_i	100	0.7936
	P_i P_{i+4}	100	1.1241
	C_k C_{k+1}	100	0.6960
	C_k P_{3k}	20	0.9575
	C_k P_{3k+1}	20	0.9290
	C_k P_{3k+2}	20	0.9575
U_{angle}		$K_a\ (\epsilon_0/rad^2)$	$\varphi_{eq}\ (^\circ)$
	P_i P_{i+1} P_{i+2}	600	81.2
	P_i P_{i+4} P_{i+5}	100	108.9
	C_k C_{k+1} C_{k+2}	20	180.0
$U_{dihedral}$		$K_d\ (\epsilon_0)$	$\chi_{eq}\ (^\circ)$
	P_i P_{i+1} P_{i+2} P_{i+3}	1000	21.7

Note. — Interaction energies are calculated as described in Eq. 2.2.

iii. each three consecutive central beads: $(\mathsf{C}_k, \mathsf{C}_{k+1}, \mathsf{C}_{k+2})$.

Harmonic angle force constants are assigned to a large $K_a = 600\ \epsilon_0/rad^2$ for angles between consecutive principal beads as those are the most important for maintaining the helical structure. Supporting angle contributions, such as angles around hydrogen bonds are assigned an intermediate value of $K_a = 100\ \epsilon_0/rad^2$. Angles involving central beads, which should only have a minor contribution to structural stability, are assigned a small value of $K_a = 20\ \epsilon_0/rad^2$. Values for φ_{eq} can be found in Table 3.1.

Dihedral angle forces, derived from $U_{dihedral}$ (2.2c), regulate distortions in helix structure and prevent the helix from unfolding. Therefore, changes in the protein's secondary structure are not described by this model. Those dihedral forces are applied between all sets of consecutive principal beads: $(\mathsf{P}_i, \mathsf{P}_{i+1}, \mathsf{P}_{i+2}, \mathsf{P}_{i+3})$. Note that helix structure features are controlled almost exclusively by interactions of principal beads. secondary and central beads are positioned to serve as exclusive volume and are only weakly bonded to ensure they stay in close proximity to the helical structure.

3.4 Simulation Details

We study the configurations of TM helices using the hybrid Dissipative Particle Dynamics - Monte Carlo (DPD-MC) simulation technique described in Section 2.4.1. To simulate the tensionless state of unconstrained lipid bilayers [75] we set the bilayer surface tension, γ to be zero while the normal pressure, $P_\perp$, is set to equal the bulk water pressure ($P^* = 22.3$ in reduced units). The temperature used throughout the simulations is $k_B T^* = 0.7\,\epsilon_0$ in which the lipid bilayer is well within the L_α phase ($\sim 65°C$, see temperature scale in Eq. 2.12).

The number of lipids was approximately 2000 lipids. Initially, we equilibrate a bilayer of 2024 lipids. We then make room for each helix (either one or two helices were simulated) by removing any lipids within R_h (helix radius) distance from the point of helix insertion. To avoid unphysical biases in surface tension between the leaflets we remove an equal number of lipids from both bilayer leaflets. We then insert a helix of a given size into the bilayer (helix major axis parallel to bilayer normal). For a system with only a single helix, we simulate for $8 \cdot 10^4$ cycles, equivalent to $\sim 100\,ns$. For a system of two helices we prepare 30 separate copies of the system, to gain a larger ensemble of configurations to sample from. We simulate each copy of the system for $2 \cdot 10^4$ cycles.

3.5 Sampling helix properties

We sample various characteristics of the helix and the helical-dimer structure. For most properties investigated (*e.g.* cross angle, tilt angle) we need to define the helix major axis. That allows us to treat the helix as one unit and consistently measure its properties. In this section we discuss how the helix major vector is determined and how other properties are determined thereafter.

3.5.1 Helix Main Axis

Our goal was to determine an instantaneous, per time-frame, vector that will describe the orientation of the helix. Several characteristics of our model has made this pursuit non-trivial. First, the fact that the helix is not a rigid body, but rather a collection of bonded beads, means that there will be inherit fluctuations in this property. Second, a helix is by its nature non symmetric around its main axis. The mass distribution of beads around the major axis is uneven, and only approaches an even distribution at infinite length helices. This means there is an inherit bias from the *true* major axis of the helix.

We have considered and compared several approaches for constructing the helix major axis, described below.

Principal axis analysis

A rigid body's principal axis may be constructed by finding the eigenvectors of its moment of inertia tensor. The principal axis then defines the best linear representation for the axis

around which the mass is evenly distributed.

We compute the moment of inertia tensor from all backbone beads, assuming all beads have an equal mass. The individual tensor components can be computed by summing over the mass distributions around the different axes. N_p is the number of principal beads:

$$\mathcal{I}_{xx} = \sum_{i=1}^{N_\mathrm{p}} y_i^2 + z_i^2 \qquad \mathcal{I}_{yz} = \mathcal{I}_{zy} = -\sum_{i=1}^{N_\mathrm{p}} y_i z_i$$
$$\mathcal{I}_{yy} = \sum_{i=1}^{N_\mathrm{p}} x_i^2 + z_i^2 \qquad \mathcal{I}_{xz} = \mathcal{I}_{zx} = -\sum_{i=1}^{N_\mathrm{p}} x_i z_i$$
$$\mathcal{I}_{zz} = \sum_{i=1}^{N_\mathrm{p}} x_i^2 + y_i^2 \qquad \mathcal{I}_{xy} = \mathcal{I}_{yx} = -\sum_{i=1}^{N_\mathrm{p}} x_i y_i$$

The helix vector $\vec{H}$ is then defined as the eigenvector of $\mathcal{I}$ corressponding to the highest eigenvalue. $\mathcal{I}\vec{H} = \lambda_1 \vec{H}$ where $\lambda_1 \geq \lambda_l, \forall l = 1 \cdots 3$.

Mean End Points

Another approach is to determine the helix axis based on its two extreme ends. We can find the center of mass of the n extreme principal beads at both ends of the helix. We can then define the helix vector, $\vec{H}$, as the vector connecting those top and bottom end points (T and B, respectively) at each step in time.

In determining n, one must take into account a balanced distribution of beads around the helix. An α-helix contains 3.6 residues (beads) per span. We therefore tried using both $n = 3$ and $n = 4$ as the number of average principal beads at each end point.

Comparison of Methods

We created a model system of helices to test the different methods for determining helix axis. We test several helix sizes. For each helix size, we build the helix around the main axis: $(0, 0, 1)$. We then tilt the helix by a random predetermined angle. We use the methods described above to find the resulting helix axis and estimate the current tilt angle of the helix. A good method for determining the helix main axis is expected to yield small differences between the applied and estimated tilt angle of a helix.

Figure 3.3 shows the results for selected helix sizes 10-27. Each figure shows the difference between the applied tilt angle (x-axis) and the estimated tilt angle (y-axis) in any one of the three method described above. One may observe that none of the methods yields a perfect

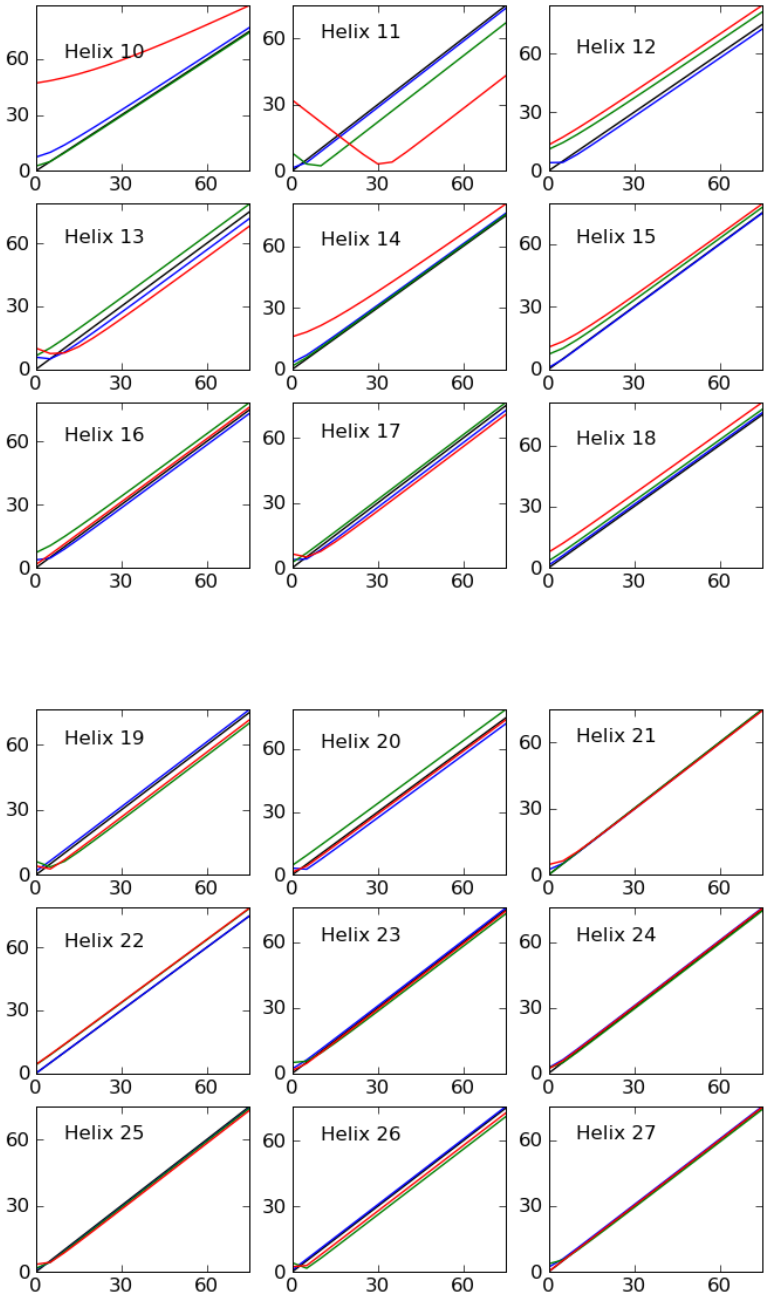


Figure 3.3: Comparison of predicted tilt angle (y-axis) with applied helix tilt angle (x-axis) in three various methods: Principal axis analysis (red), 3-Bead Mean End Points (green) and 4-Bead Mean End Points (blue). Black lines represent slope of 1, for convenience. Number of principal beads in each helix is described by the title.

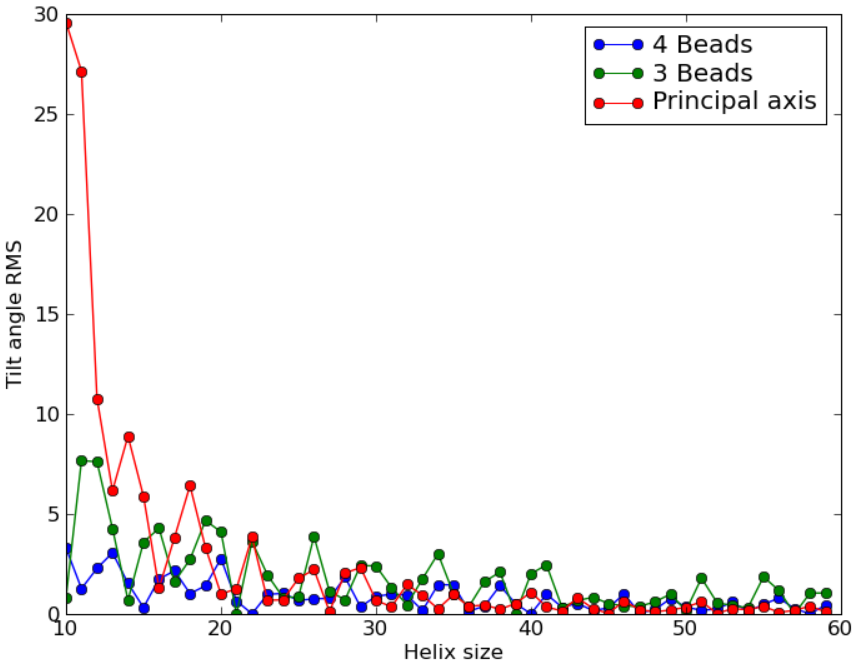


Figure 3.4: Comparison of methods to determining helix main axis as a function of helix length. Each point shows the root mean square error in tilt angle estimation for a given method and helix length.

match with the applied tilt angle, but all method converge to the a match as the helix size increases.

We calculated the root mean square error (RMS) of each method and each helix size and present these results in Figure 3.4. The comparison shows that calculating the mean end points, with $n = 4$ beads defining the center of mass of each end point, gives the least errors. It is interesting to note that the principal axis approach yields the minimum number of errors for large helix sizes. However, since the typical sizes of helices are within the $N_{\mathrm{p}} = 15 \cdots 45$ range, we prefer the method that consistently performs better in that range.

We therefore chose to define the helix vector, $\vec{H}$ as the vector connecting the bottom and top end points, B and T. These are defined as the center of mass of the first (or last) 4 principal beads of the helix. We use this definition of the helix axis from now on.

3.5.2 Helix Radius

We calculate the instantaneous radius of the simulated helix to monitor the stability of the helical structure. The instantaneous helix radius, $\mathcal{R}_{\mathrm{h}}$, is defined as the average distance between each principal bead and the helix main axis. The distance between the beads and the helix main axis is calculated based on the canonical formula [76]:

$$\begin{aligned} d_i &= \frac{|(B - \vec{r}_i) \times (T - B)|}{|(T - B)|} \\ &= \frac{|(B - \vec{r}_i) \times \vec{H}|}{|\vec{H}|} \\ \mathcal{R}_{\mathrm{h}} &= \frac{1}{N_{\mathrm{p}}} \sum_{i=1}^{N_{\mathrm{p}}} d_i \end{aligned}$$

A helix with a well maintained structure is expected to maintain its radius, such that $\langle \mathcal{R}_{\mathrm{h}} \rangle \simeq R_{\mathrm{h}}$ throughout the simulation.

3.5.3 Helix Order Parameter

The helix model is not a rigid body, but rather a network of harmonic bonds. We wish to probe its structural rigidity by determining how well it maintains its order along the helix main axis. For that end we calculate an order parameter for the helix. A good order parameter should distinguish between helices that bend during a simulation and ones who remain elongated along the helix main axis.

We use a common order parameter from the field of liquid crystals [77] to determine the helix order parameter. The order parameter is based on the average of the second Legendre Polynomial: $S = \langle P_2^0(\cos\theta) \rangle = \left\langle \frac{3\cos^2\theta - 1}{2} \right\rangle$.

To obtain the helix order parameter we divide the helix into vertebrae. For an ordered helix, the vector connecting each vertebra to the next is expected to align with the helix

main axis. The instantaneous order parameter of a helix is therefore an average over the individual order parameter of each of the vertebrae connecting vectors.

Each vertebra, $\vec{V}_k$, in the helix is defined by the position of the central bead it corresponds two. The vector connecting adjacent central beads, $\vec{V}_k = (\mathsf{C}_{k+1} - \mathsf{C}_k)$ is used for the calculation of the instantaneous order parameter.

The helix order parameter is therefore calculated as follows:

$$\cos^2(\theta_k) = \frac{(\vec{V}_k \cdot \vec{H})}{(\vec{V}_k \cdot \vec{V}_k)(\vec{H} \cdot \vec{H})}$$
$$S_k = \frac{1}{2}\Big(3\cos^2(\theta_k) - 1\Big)$$
$$S = \frac{1}{N_\mathrm{c} - 1}\sum_{k=1}^{N_\mathrm{c}-1} S_k$$

If the helix was to maintain an ordered structure, such that the central beads are aligned with the helix main axes, we expect the order parameter to yield $S \sim 1.0$. If, however, the helix was to bend and obtain an unordered structure, its order parameter will average at $S \sim 0.0$. For an anti-ordered structure, we expect to find $S = -1/2$.

3.5.4 Distance between helices

Calculating the distance between helices can be done in several ways. One canonical approach is to calculate the distance between the center of mass of both helices. This approach gives a clear estimate on the distance of the helices if all helices are parallel. The method becomes problematic, however, when treating helices of different sizes that might tilt within the membrane. In that scenario, we might calculate a large distance between centers of mass of helices, when in fact those helices are tilted in the membrane and are in contact (see illustration in Figure 3.5).

Instead, we calculate the distance between a pair of helices using the Point of Closest Approach algorithm following the procedure provided by Sunday [78]. In this method we first find a vector, $\vec{w}_c$, connecting the two helices lines:

$$\mathbf{L_1}(s) = B_1 + s \cdot \vec{H}_1$$
$$\mathbf{L_2}(t) = B_2 + t \cdot \vec{H}_2$$

As many such connecting vectors exist we chose the vector with the shortest size connecting the points of the helices that are in closest approach. Namely, we find the parameters (s_c, t_c) for which $|\vec{w}_c| = |\mathbf{L_1}(s_c) - \mathbf{L_2}(t_c)|$ is minimized. The distance between the helices is then defined as the size of the connecting vector, $r_{1,2} = |\vec{w}_c|$. We confine (s_c, t_c) to the range $0 \leq s_c, t_c \leq 1$, forcing the point of closest approach to be a physical point on each helix. This ensures the calculated distance is the closest physical distance between the helices.

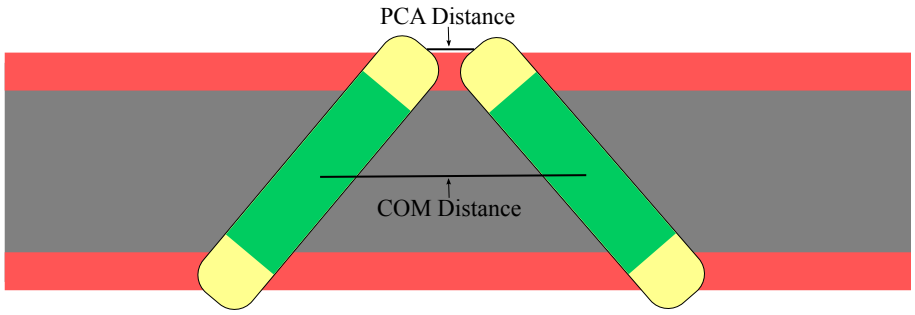


Figure 3.5: Comparison between the Center-of-Mass-Distance (COM Distance) approach, in which the distance between helices is defined as the distance between their respective centers of mass, and the Point-of-Closest-Contact-Distance (PCA Distance) approach, in which the distance is defined at the point in which the helices' major axes are in closest approach. Large discrepancies between the two approaches might arise when the helices are tilted in the membrane.

3.5.5 Helix Angles

Tilt angles for each TM helix, as well as cross and projection angle for each helix pair were defined based on the helices major axes.

Tilt Angle

The tilt angle, θ, is defined as the angle between the helix major axis and the normal to the bilayer, $+\hat{\mathbf{z}}$. The helix major axis is always defined as pointing towards the $+\hat{\mathbf{z}}$ direction, therefore limiting θ to the range $[0°, 90°)$.

Cross Angle

The inter-helical cross angle, Ω, is defined as the 3D angle between the two helices major axes, $\Omega = \arccos(\vec{H}_1 \cdot \vec{H}_2)$. It is limited to the range $[0°, 180°)$.

Projection Angle

The inter-helical projection angle, γ, is defined as the 2D angle between the two helices major axes along the plane of the membrane $(\hat{\mathbf{x}} - \hat{\mathbf{y}})$. It is limited to the range $[0°, 180°)$.

These angles are displayed in Figure 3.6

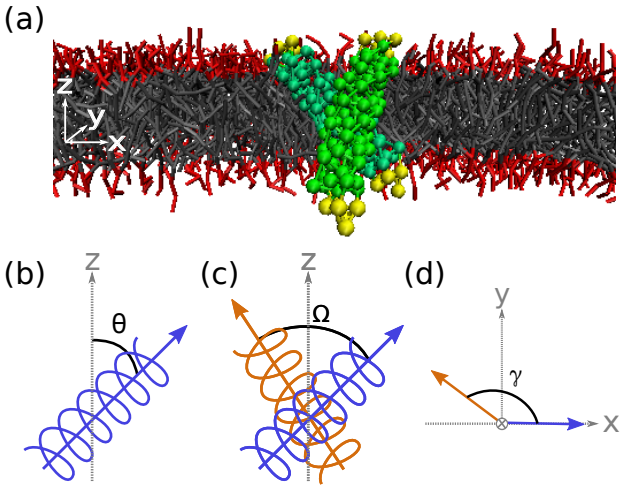


Figure 3.6: Coarse-grained helix pair and angle definitions. (a) Shows two positive mismatched helices in a typical crossed configuration. Water particles are not displayed for clarity. (b) The tilt angle, θ, of a helix is defined as the angle between the helix major axis (blue arrow) and the bilayer normal, $+\hat{\mathbf{z}}$. (c) The cross angle, Ω, is defined as the dihedral angle between the major axes of the two helices (blue and orange arrows). (d) The projection angle, γ, is defined as the angle between the major helix axes along the plain of the bilayer ($\hat{\mathbf{x}} - \hat{\mathbf{y}}$).

3.5.6 Helix Hydrophobic Mismatch

The hydrophobic mismatch of each helix was calculated based on the hydrophobic length of the helix (d_{H}) and that of the membrane (d_{L}), using Eq. 3.1. The number of CG hydrophobic beads determines the hydrophobic length of the helix. Each hydrophobic bead corresponds to one helix residue with a distance of 1.5 Å between each consecutive residues. To calculate the hydrophobic length we add half a bond length (0.75 Å) in each direction such that the cutoff of hydrophobic residues is in the middle of the bond between the hydrophobic and hydrophilic size of the membrane. d_{H} is therefore equal to $d_{\mathrm{H}} = N_{HB} \cdot 1.5$ Å where N_{HB} is the number of hydrophobic beads in the helix, or $d_{\mathrm{H}} = (N_{\mathrm{p}} - 6) \cdot 1.5$ Å, where N_{p} is the number of principal beads in the helix.

We calculate the membrane hydrophobic thickness (d_{L}) by fitting the density profile of the hydrophobic beads to a Gaussian error function. This has become the standard method for determining the membrane hydrophobic thickness [79]. The bilayer thickness in our model was calculated to be $d_{\mathrm{L}} = 26.77$ Å or 28.12 Å for a lipid model of 5 or 6 beads, respectively. Density profile along with the fitted Gaussian error function is displayed in Figure 3.7.

We simulate 11 different helix sizes. The number of principal beads in each helix as well as the corresponding hydrophobic mismatch can be found in Table 3.2.

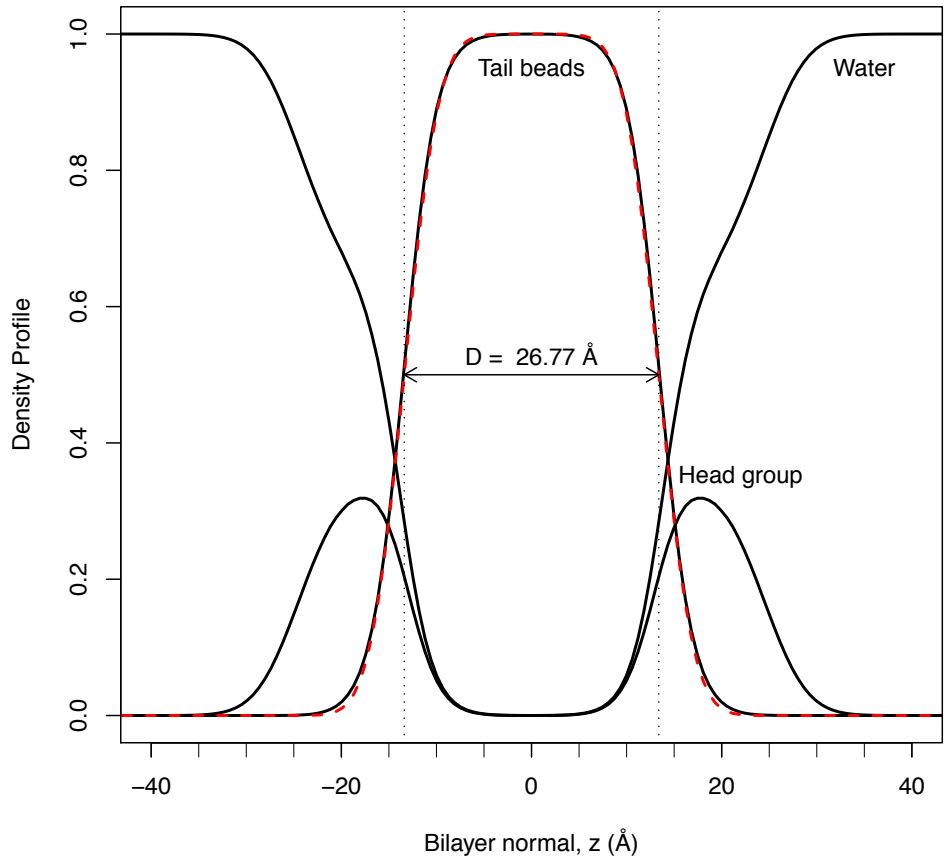


Figure 3.7: Density profile of a pure bilayer simulation. Solid lines show the volume fraction of the various components in the simulation: lipid head beads (head group), lipid tail beads and water. Red dashed line corresponds to the symmetric error function: $P_{HC}(z) = \frac{1}{2}\left[\mathrm{erf}(z, -z_{HC}, \sigma_{HC}) - \mathrm{erf}(z, z_{HC}, \sigma_{HC})\right]$ as defined by Eq. 3 in Kucerka *et al.* [79]. Fit values correspond to $z_{HC} = 13.385$ and $\sigma_{HC} = 2.85$ yielding a hydrophobic thickness of $d_{\mathrm{L}} = 26.77$.

# Principal Beads	Hydrophobic Mismatch (Å)
15	-13.27
18	-8.77
21	-4.27
24	0.23
27	4.73
30	9.23
33	13.73
36	18.23
39	22.73
42	27.23
45	31.72

Table 3.2: Number of principal beads (N_{p}) and the hydrophobic mismatch (Δd) of each of the simulated helices.

3.6 Simulation Results

3.6.1 Helix Structure

We have performed several test simulations using different parameters for the bond length, angle and dihedral angles. At certain parameter sets the helix structure starts falling apart and does not stay true to its helical nature. In all other parameter sets explored, which maintain the helix structure stability, we see consistent results for the tilt and cross angle distributions, in response to varying the hydrophobic mismatch.

To further test the validity of our bonded interaction parameters, we compared the fluctuation in our model helix's length to that of a natural helix, simulated in a membrane environment using all atoms simulations. To that end we use the helix structure 1SPF, provided in the RCSB database [80]. We inserted the helix to a pre-equilibrated bilayer of DPPC lipids and preformed initial equilibration using the CHARMM-GUI program [81]. We then simulated the membrane embedded helix for a total of 5ns in the $NP\gamma T$ ensemble, and obtain a histogram of its natural length fluctuations. Ensemble parameters were chosen to match those in our CG simulation. These fluctuations in helix length were then compared to the length fluctuations of a CG helix of similar size (containing 27 residues). Results are presented in Figure 3.8.

We do not expect a perfect match between these distributions, for several reasons: 1. These are not identical helices; 2. The nature of CG models is such that finer structural details cannot be resolved, and so wider fluctuations are expected; and finally 3. The helix simulated in all-atom simulations included hydrophilic residues only on one end, while the CG model contains hydrophilic residues on both ends. Different driving forces for stretching and shrinking are therefore expected. Despite all these differences, we see a reasonable agreement

in the distribution of helix length, with more flexibility in the CG helix model. The all-atoms helix length corresponds to $H_L^{AA} = 35.7\text{Å} \pm 0.54\text{Å}$, while the coarse grained helix length corresponds to $H_L^{CG} = 34.2\text{Å} \pm 0.71\text{Å}$. In Figure 3.8 we present the deviations around the mean helix length for easier comparison.

Another test to check the structural stability of the helix model is probing the helix radius and order parameter. A high order parameter ($S \to 1$) suggests a highly ordered structure approaching a rigid body. We present the simulated order parameter in Figure 3.9 as a function of helix size. The order parameter is on average 0.94 with a small fluctuation of 0.02. This suggest a stable rather rigid structure and ensures our helix does not change its helicity during the simulation.

The helix radius was also measured throughout the simulation. We see that that radius remains fluctuating around an average of 5.5 Å, with a standard deviation of 0.2 Å. The small deviation (3.4%) suggests again that the helix structure remains rather stable throughout the simulation, and for different helix sizes. The average radius is smaller than the imposed $R_{\text{h}} = 6$ Å, as a consequence of the other imposed structural bonds. Nonetheless, the radius is very close to representing the radius of the helix residues.

3.6.2 Tilt angles And Effective Hydrophobic Mismatch

To understand the configuration of packed helices, one must first understand the configuration of a single TM helix imbedded in a lipid bilayer. The tilt angle of a helix, θ, is an important characteristic of single helix configuration. We measured the tilt angle for each helix as a function of its hydrophobic mismatch, Δd. Mean and standard deviation values of tilt angles are shown in Fig. 3.10(a). We observe three regimes in the behavior of tilt angles with respect to hydrophobic mismatch, which we refer to as: Negative, Positive, and Super-Positive mismatches. It is interesting to compare the observed results in light of our estimated means of adjustments to hydrophobic mismatch (Section 3.2).

For negative mismatched helices ($\Delta d < 0$ Å), we observe a plateau of $\langle\theta\rangle \sim 10^\circ$ in the tilt angle. This effect was previously seen for synthetic peptides [34, 62, 65]. In those helices, tilting would decrease the effective hydrophobic size of the helix along the membrane normal, making it energetically unfavorable. The helix is therefore expected to tilt as little as entropically possible [62].

For positive mismatched helices, $\Delta d \in (0, 20]$ Å, we observe an increase in the tilt angle of the helices as a function of hydrophobic mismatch. This increase matches previously reported results [34, 62]. In super-positive mismatched helices ($\Delta d \gtrsim 20\text{Å}$) we observe a change in the slope of tilt angle, as helices tend to adopt a higher tilt angle. This effect has not been observed before as, to the best of our knowledge, this range of hydrophobic mismatches had not been explored previously. See Section 3.6.8 for further discussion on this regime.

The tilt of positive mismatched helices is the response of the system to decrease the hydrophobic mismatch. If a helix of hydrophobic size d_{H} adopts a non-zero tilt angle, θ, its effective hydrophobic length along the bilayer normal direction would decrease to $d_{\text{H}}^{\text{eff}} = d_{\text{H}} \cdot \cos(\theta)$, approaching the membrane hydrophobic thickness, d_{L}. We define the

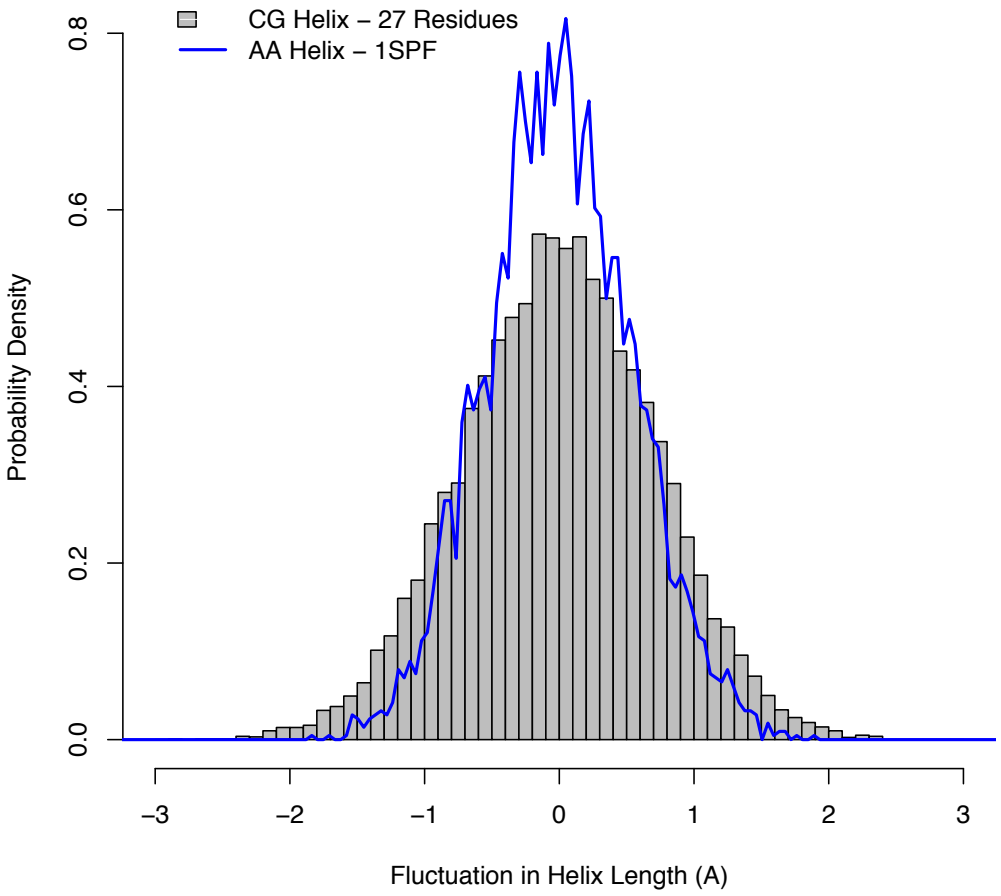


Figure 3.8: Comparison of helix length distribution of a model coarse grained (CG) helix of 27 residues (grey boxes) and a helix structure from the PDB database (1SPF), simulated with all-atoms (AA) simulation technique (blue line). The histograms are aligned around the mean value of each simulated helix, which correspond to $H_L^{AA} = 35.7$Å for the all-atom helix and $H_L^{CG} = 34.2$Å for the coarse grained helix. These distributions show the fluctuations in helix length in both simulation techniques are substantial. The CG model captures fairly well the inherent fluctuations in helix length, showing a tendency towards a broader distribution.

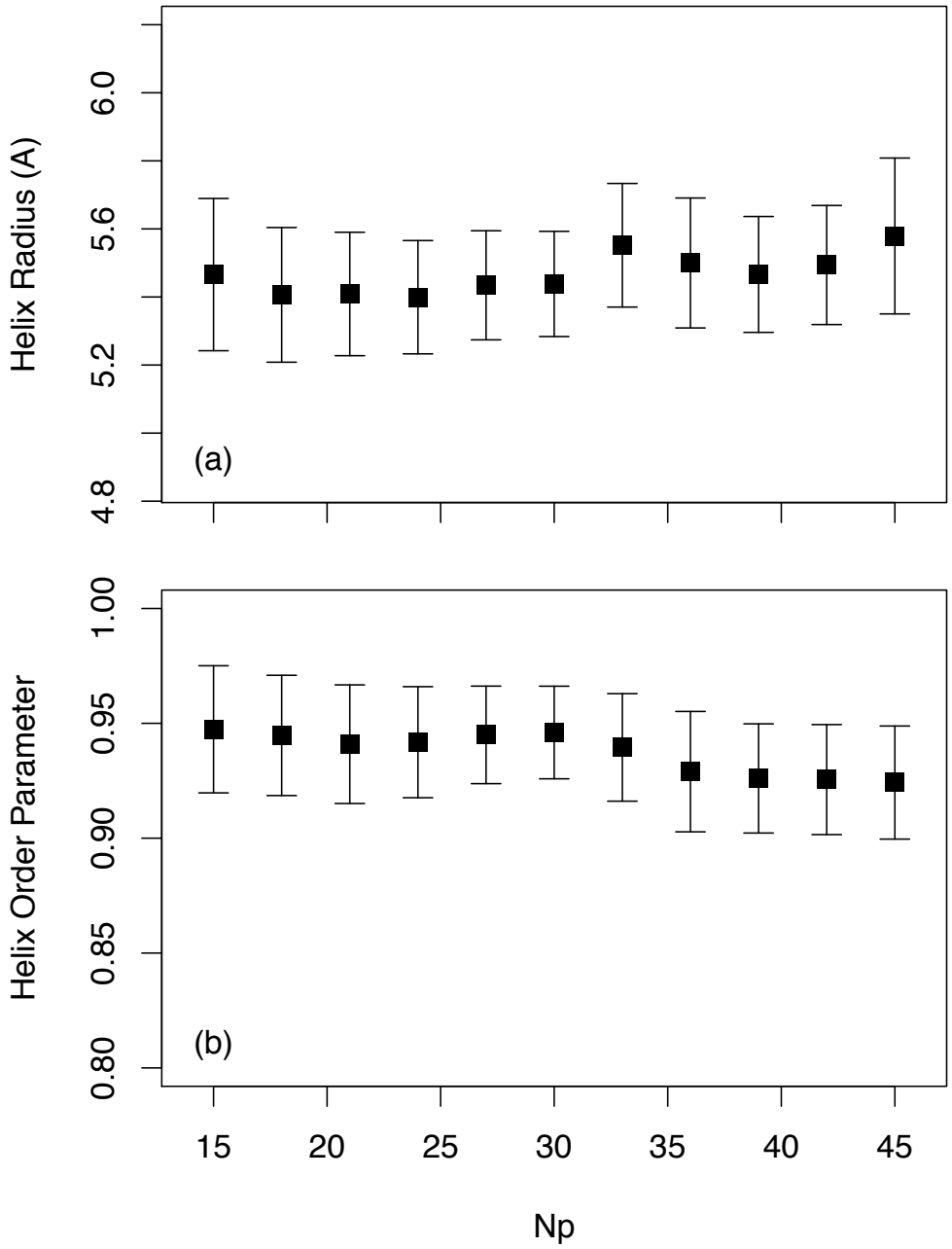


Figure 3.9: Helix radius and order parameter for various helix lengths. Helix radius (a) remains fluctuating around an average $\langle \mathcal{R}_{\rm h} \rangle = 0.85 \pm 0.03$ $d_0 = 5.5 \pm 0.2$ Å for all helix sizes. The helix order parameter (b) remains very close to 1.0 ($\langle S \rangle = 0.94 \pm 0.02$), suggesting a very ordered structure. The x-axis represents the number of principal beads in the helix, $N_{\rm p}$.

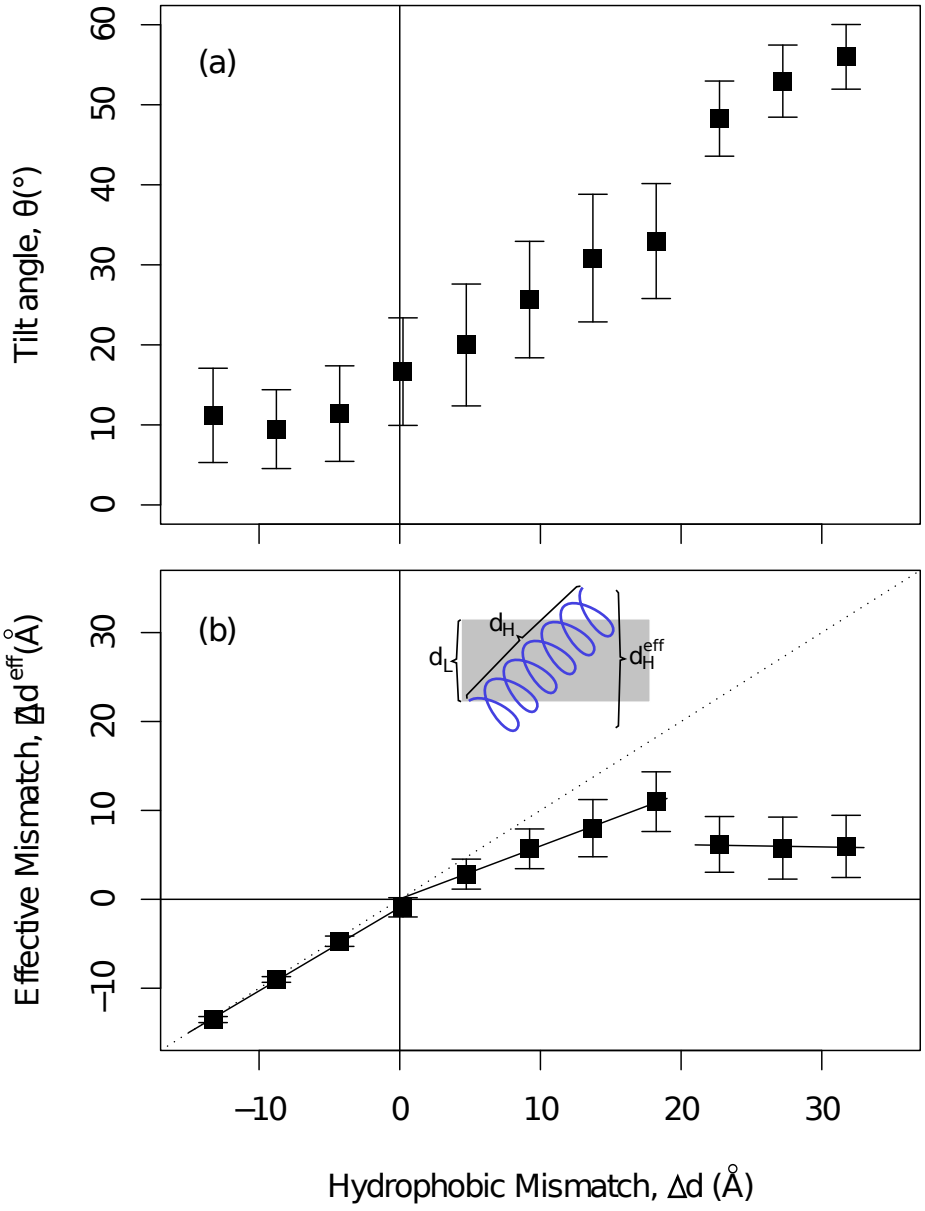


Figure 3.10: Helix tilt angle (a) and effective hydrophobic mismatch (b) as a function of hydrophobic mismatch for single helices. We observe three regimes of hydrophobic mismatch. For $\Delta d < 0$Å, the tilt angle plateaus and the effective hydrophobic mismatch matches the hydrophobic mismatch, with a slope of 0.94 (dashed line in (b) represents a slope of 1.0, for reference). For $\Delta d \in (0, 20]$ Å the tilt angle increases monotonically, and the slope of the effective mismatch decreases to 0.59; For $\Delta d > 20$Å the effective mismatch stays constant at $\Delta d^{\text{eff}} \simeq 6$Å while tilt angles increase. Solid lines in (b) represent linear fit of effective mismatch for each of the three regimes. Whiskers in both (a) and (b) represent one standard deviation.

effective hydrophobic mismatch as the difference between the helix effective hydrophobic length and the bilayer hydrophobic thickness, $\Delta d^{\text{eff}} = d_{\text{H}}^{\text{eff}} - d_{\text{L}} = d_{\text{H}} \cdot \cos(\theta) - d_{\text{L}}$. It is a measure of the degree to which the hydrophobic mismatch strain is reduced by the tilting of a helix. If the helix was to tilt fully to match the membrane hydrophobic thickness, its effective mismatch would be $\Delta d^{\text{eff}} = 0$Å, yielding a tilt angle of $\theta = \arccos(d_{\text{L}}/d_{\text{H}})$.

The effective hydrophobic mismatch of our simulated TM helices is displayed in Fig. 3.10(b). Again, we observe differences between the three major regimes. For negative-mismatched helices, the effective mismatch is nearly identical to the hydrophobic mismatch ($\Delta d^{\text{eff}} \propto 0.94 \cdot \Delta d$). It is energetically unfavorable for these helices to decrease their hydrophobic length and so they preserve their length almost fully. Deviations from a slope of 1.0 originates in entropic contributions.

For positive-mismatched helices we observe a smaller slope, $\Delta d^{\text{eff}} \propto 0.59 \cdot \Delta d$. In this range, there is an energetic balance between the adjustment of the helix, by tilting, and that of the membrane, by expanding its hydrophobic thickness around the helix (see Section 3.6.5). We observe that the degree to which the helix reduces its hydrophobic size is fixed (60% of its mismatch) in the positive mismatch regime.

For super positive-mismatched helices, though, we observe a significant drop in the effective mismatch. Irrespective of the hydrophobic mismatch, the helices tilt to reach an effective mismatch of $\Delta d^{\text{eff}} \simeq 6$Å. We observe a complimentary trend in the membrane thickness around super positive mismatched helices as well. The membrane thickness linearly increases with the hydrophobic mismatch of the embedded helix until it drops to a plateau for helices of super positive mismatch (see Section 3.6.5). This suggests a shift in the balance of energetic contributions, and might point to a different arrangement of lipids around a super-positive mismatched helix. We discuss these results further in Section 3.6.8.

3.6.3 Cross Angles

Traditionally, the cross angle of packed helices is thought to be determined by specific residue interactions. The fit between 'groves' of one helix and the 'ridges' of its counter pair is believed to lock the packed configuration into the 'correct' cross angle. In contrast, in our model specific residues are not present as all beads are equivalent. This allows us to explore the role of the membrane in determining the cross-angle distribution in the absence of specific residue interactions.

We calculated the distribution of cross angle, (Ω), defined as the angle between the two helices major axes (see Section 3.5.5), for helix pairs. Our results show a relation between cross angle and hydrophobic mismatch. Cross-angle distributions of homogenous helix pairs are presented in Fig. 3.11. These show clear dependence of the cross-angle distribution on the helices hydrophobic mismatch. The mean cross angle value as well as the width of the distribution increases with increasing mismatch.

The relation between cross angle and hydrophobic mismatch might be surprising at first sight. Careful consideration though leads to the realization that the cross angle, Ω, and the helices tilt angle θ_1, θ_2 are **not independent**, even if the two helices do not interact with

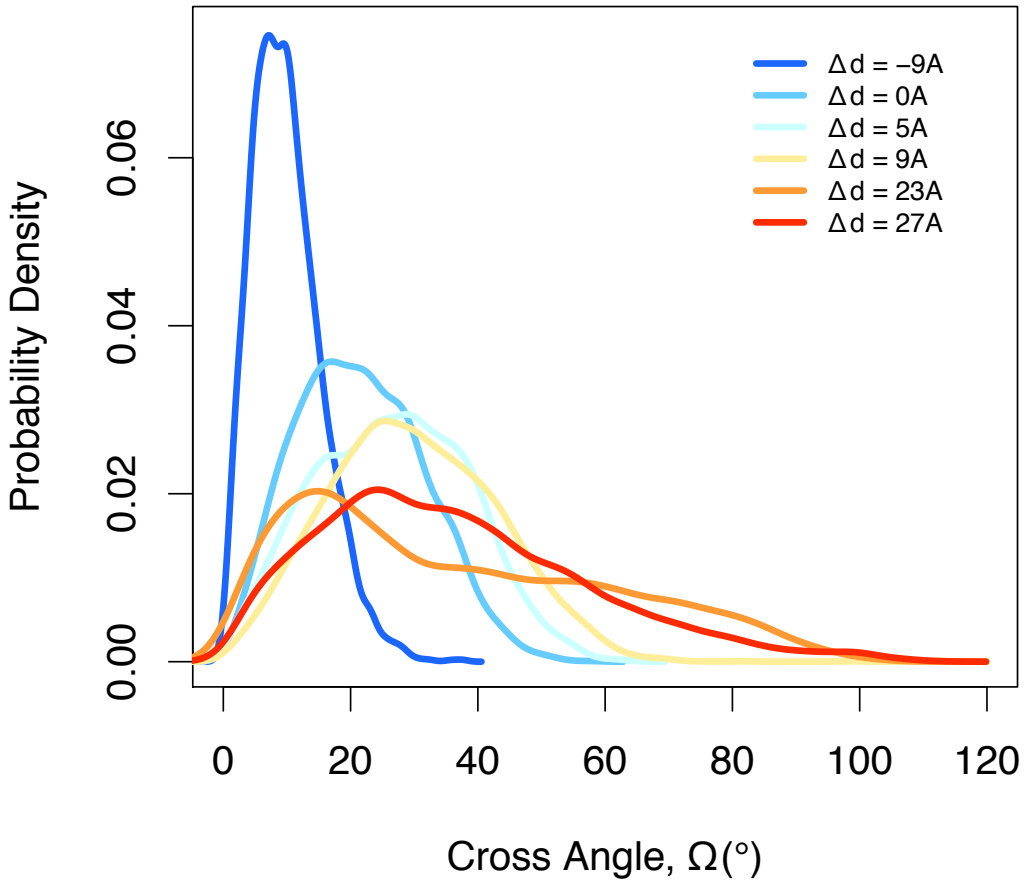


Figure 3.11: Cross-angle distribution of homogeneous helix pairs. Each line represents the probability density of cross angles for one of six characteristic helix mismatches (see legend). Cross angles were collected only from packed configurations (inter-helical distance < 15Å).

each other. Simple geometric consideration lead to the following relationship between the aforementioned angles along with γ, the projection angle (see angle definitions in Section 3.5.5):

$$\cos\Omega = \frac{(1+\cos\gamma)\cdot\cos(\theta_1-\theta_2)+(1-\cos\gamma)\cdot\cos(\theta_1+\theta_2)}{2} \tag{3.2}$$

This is a key relationship determining the cross angle of TM helix pairs. According to this relationship, for fixed tilt angles, θ_1', θ_2', the cross angle, Ω, would be **confined** to the range $|\theta_1' - \theta_2'| \leq \Omega \leq |\theta_1' + \theta_2'|$. The distribution of cross angles within that range would be non-uniform arcsine-like (see Fig. 3.12, dashed line).

TM helices do not have a fixed tilt angle, but rather a distribution of tilt angles. The overall cross-angle distribution of two such helices is therefore expected to be a weighted average of fixed tilt angle histograms. One such representative *reference distribution* is shown in Fig. 3.12 (solid line).

Based on the reference distribution, we expect the cross-angle distribution of two helices, even when far apart in the membrane, to display non-uniform distribution. Since the cross angle is confined by the sum and difference of the individual tilt angles, we expect the cross-angle distribution to be wider for helices with large tilt angles and less wide for helices with small tilt angles. Additionally, for pairs of helices with very different tilt angles ($\theta_1 \nsim \theta_2$) we expect to see fewer small cross angles.

Taking the correlation between hydrophobic mismatch and tilt into account, many characteristics of the distributions in Fig. 3.11 follow these expected behaviors. This suggests that the main dependence of the cross angle in the hydrophobic mismatch is through its effect on the tilt angles. We hypothesize that the hydrophobic mismatch controls the tilt of the individual helices, and that in turn confines the cross angle between those helices. This view matches our observations that the mean and spread of cross angles of homogeneous helix pairs grows with hydrophobic mismatch.

The effect of hydrophobic mismatch on the cross-angle distribution can be further observed in Fig. 3.13. Here we compared the cross-angle distribution of both homogenous and heterogenous pairs of helix mismatch with their reference distribution. The reference distribution for each helix pair was obtained by random sampling of tilt angles and projection angles. The resulting cross angle was calculated according to Eq. 3.2. Tilt angles were sampled from a normal distribution, with mean and standard deviation corresponding to the helix hydrophobic mismatch (as presented in Fig. 3.10(a)). Projection angles were sampled uniformly in the range $[0°, 180°)$. The reference distribution is therefore the distribution of cross angles for non-interacting helices, where each helix samples configurations as if it was alone in the membrane.

Figure 3.13 shows a surprisingly good match between the simulation and reference cross-angle distribution. In both cases, reference and simulation, it is clear that the spread of the

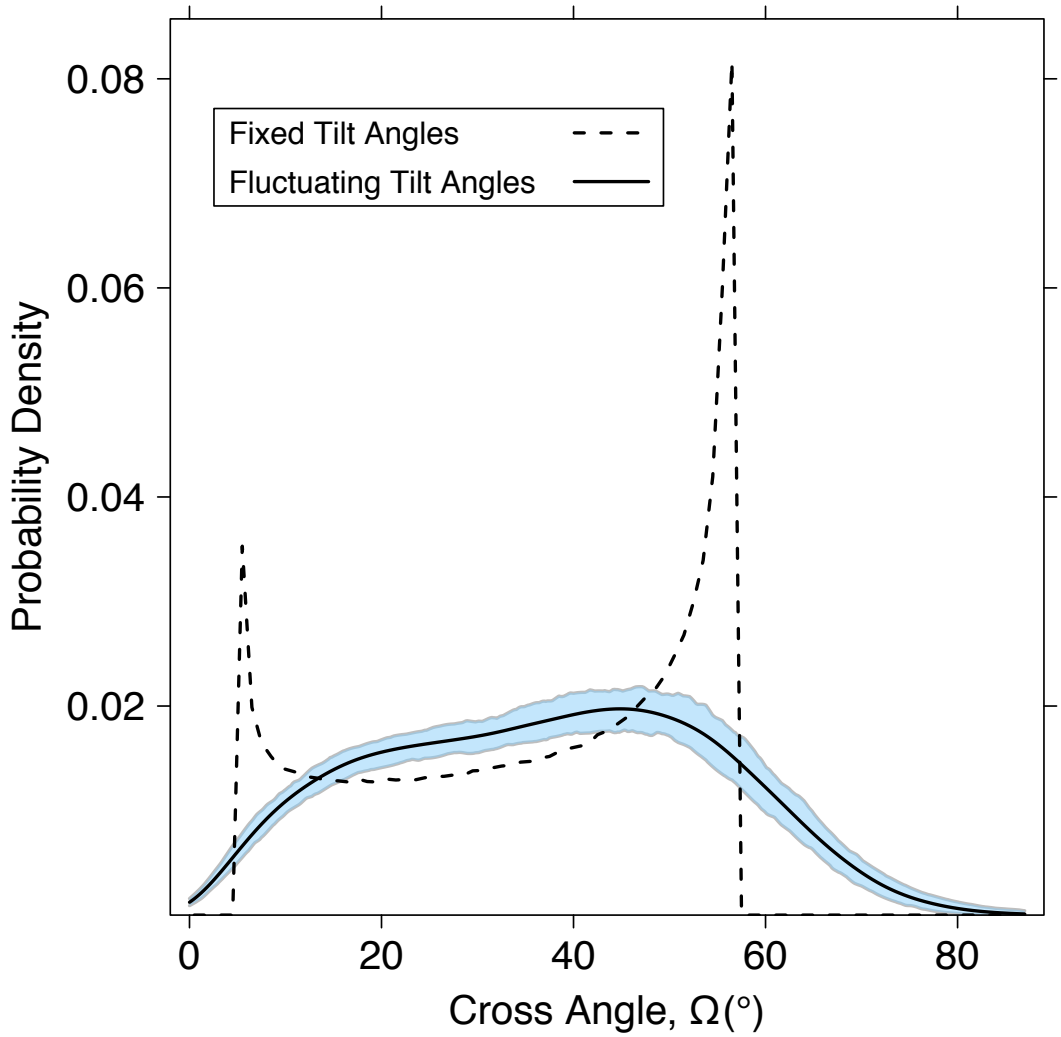


Figure 3.12: Reference cross-angle distribution. Dashed line corresponds to the reference cross-angle distribution of two non-interacting helices with fixed tilt angles, θ_1' and θ_2', and random projection angle, γ; Solid line corresponds to the reference cross-angle distribution of two helices with normally distributed tilt angles with mean θ_1', θ_2' and standard deviation σ_1, σ_2, respectively. These mean and standard deviation values correspond to the observed tilt angles for helices with hydrophobic mismatches $\Delta d_1 = 9\text{Å}, \Delta d_2 = 14\text{Å}$, respectively. These amount to $\theta_1' = 25.7°, \theta_2' = 30.8°, \sigma_1 = 5.9°$ and $\sigma_2 = 6.5°$. The shaded area around the reference distribution represents 90% confidence intervals.

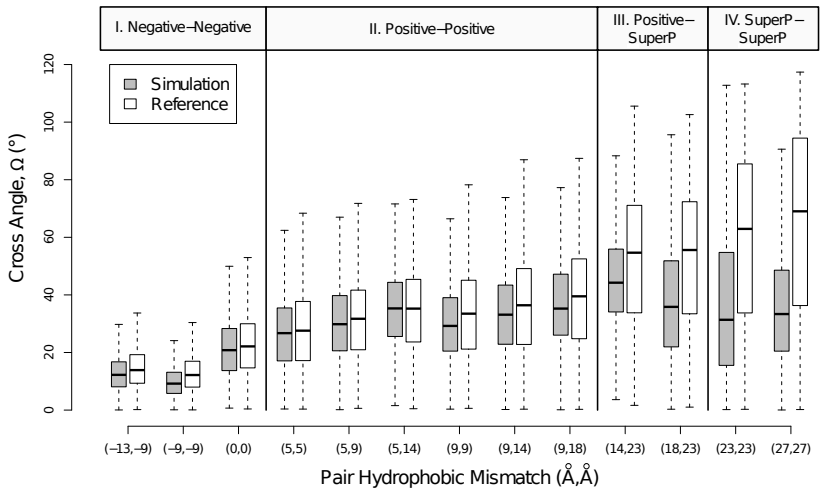


Figure 3.13: Comparison of simulated cross-angle distributions (filled box-and-whisker diagrams) with their corresponding reference distributions (empty box-and-whisker). Each box represents the lower and upper quartiles around the median cross angle (horizontal black line). See SI for details on box-and-whisker representations. Simulated cross angles were collected only from packed configurations (inter-helical distance < 15Å). Helix pairs are segregated by vertical lines into four groups, depending on the mismatch of each helix in the pair, as represented in the group titles ('SuperP' refers to super-positive mismatched helices).

distribution, as well as the median cross angle, increases with mismatch. We define four groups of helix pairs, based on the hydrophobic mismatch of each of the paired helices (see Fig. 3.13 titles for group definitions). The agreement with the reference case is particularly good in the first two groups, I and II, of helix pairs who are both negative mismatched (or zero) and helix pairs who are both positive mismatched. The difference in the median cross angle (represented by the thick horizontal lines) from the reference distributions, is on average only 2.3°.

The spread of the simulated cross angles, as represented by the total height of the box, also seems to follow the same trend as the reference distribution, though it is consistently smaller in our simulated results (average 20% decrease). This effect is coupled with the bias in projection angles discussed in Section 3.6.4.

This particularly good agreement with reference distributions in the majority of helices (negative and positive mismatched) suggests the following. As the tilt angle is a dominant factor in determining the cross angle, and the tilt angle is set by the hydrophobic mismatch, in the majority of helices it is sufficient to understand the effect of the hydrophobic mismatch on the individual tilt angles of the two helices, to obtain the effect on the cross-angle distribution.

For the latter groups (III and IV), containing super-positive mismatched helices, the agreement is not as good. The spread of the simulated distribution decreases by 33% on average compared to the reference distributions, and the difference in the median cross angle increases to 19.7°. This suggests a different response of the super-positive helices to the presence of another helix. We discuss this effect further in the Section 3.6.8.

These effects, where the cross-angle distribution depends on the hydrophobic mismatch of individual helices suggests a robust mechanism for determining the cross angle between TM helices. The hydrophobic mismatch of a helix in a membrane will determine to a large extent its relative orientation to other helices. Experimentally it has been observed that changes in membrane thickness lead to changes in protein structure and function [13]. While the cross angle of two helices is a very simplistic model for protein structure, our observations show that changes in the membrane hydrophobic thickness can modulate the cross angle of packed helices. Combining these observations is a tempting suggestion, which we feel merits further studies.

3.6.4 Deviations from the reference distribution

The reference distribution of cross angles corresponds to a configuration of non-interacting, far-away helices. Deviations from the reference distribution are expected when helices are allowed to interact. In particular, three assumption are made when constructing the reference distribution, and each can be invalidated for interacting helices:

(a) Projection angles are uniformly distributed between 0° and 180°;

(b) Tilt angles of both helices are uncorrelated;

(c) The distribution of helix tilt angles is equivalent to that of a single helix in membrane.

These assumptions are not independent, but the extent to which each one holds in a given pair can shed light on the way these helices interact. In this section we discuss the validity of each of these assumptions in simulated helix pairs. The results presented here refer only to packed configurations where the inter-helical distance is less than 15Å.

Projection angle

For a pair of helices with fixed tilt angles, the projection angle, γ, is the independent variable determining the cross angle of the pair. A-priori, there is no reason to expect one projection angle to be more favorable than others. The reference distribution of projection angles is therefore a uniform distribution: $\gamma \sim \mathcal{U}(0°, 180°)$.

We compared the distribution of simulated helix pairs to a reference uniform distribution in Fig. 3.14(a). The results vary based on the pair mismatch group. For a pair of two positive mismatched helices (group II), as well as for the zero-mismatched pair, we see good agreement with the reference distribution, with a tendency towards marginally higher projection angles. The average median projection angle among this group is $\langle\gamma\rangle = 94.7° \pm 5.8°$ (compared to a reference $\langle\gamma\rangle = 90°$). For other helix pairs (groups I, III, IV), we observe the opposite trend. The projection angle is biased towards lower values, with a median projection angle of $\langle\gamma\rangle = 60.0° \pm 15.1°$. The spread of the distribution, measured here by the size of the box is similar in all groups and amounts to $\langle\Delta\gamma\rangle = 80.8° \pm 4.5°$ (excluding the outlier (27Å, 27Å) pair). This value is somewhat smaller than that of the reference distribution ($\langle\Delta\gamma\rangle = 90°$).

One explanation for non-uniform projection angles and the tendency towards $\langle\gamma\rangle < 90°$ could originate from the frustration of lipid configurations. The number of lipids found between the ends of the two helices in a pair is directly proportional to γ. We show in Section 3.6.7, that the presence of a helix in the membrane is energetically unfavorable for the lipids. When confined by two helices, the lipids can be more constrained, which results in a driving force towards minimizing the number of lipids confined between the helices. A smaller projection angle decreases the number of lipids between helix ends. This might therefore explain the observed tendency towards lower projection angles.

Tilt Correlation

Another means of measuring the deviation from the reference, non-interacting case is by comparing the tilt angles of the two helices in a pair. The non-interacting case assumes the tilt angles of both helices are independent, with correlation coefficient of 0. We calculated the correlation of tilt angles throughout the simulation and present the results in Fig. 3.14(b) (open circles). For groups I - III the correlation coefficient is smaller than 0.2, with an average of 0.07, suggesting no significant correlation between the tilt of both helices. For group IV, containing pairs of helices who are both super-positive mismatched, the correlation is much larger, reaching 0.37 ± 0.15 for pair (23Å, 23Å) and 0.50 ± 0.22 for pair (27Å, 27Å). This suggests cooperative tilting for pairs of super positive mismatched helices.

Another interesting observation relates to the average correlation coefficient in all pairs considered. Although in theory the full range of tilt correlation $[-1.0, 1.0]$ is accessible, in

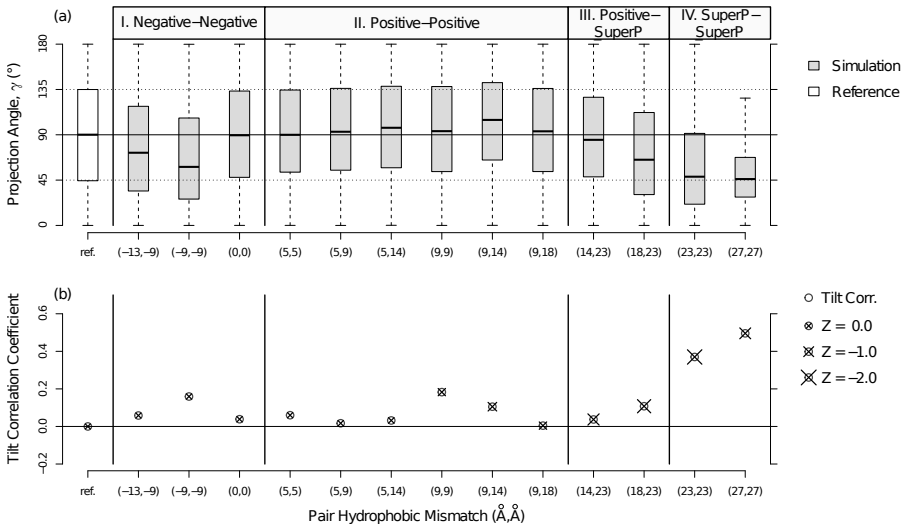


Figure 3.14: Deviations from the reference distribution. (a) Shows the distribution of projected angles. Simulated helix pairs are represented by grey filled box-and-whisker diagrams. The white diagram represents the reference distribution of a uniform projection angle. Horizontal solid line corresponds to the mean projection angle of the reference distribution, $\gamma = 90.0°$; horizontal dotted lines corresponds to the 25% and 75% percentiles of the reference distribution $\gamma = 90.0° \pm 45°$. (b) Shows the tilt correlation and change in tilt distribution in helix pairs. Empty circles represent the correlation in the tilt angles between both helices in the simulated pair, with standard error of 0.03. The size of the x symbol on each circle is proportional to the scaled mean distance (Z) of helix tilt angles, based on their distribution when isolated in the membrane (see legend and Table 3.3). The bigger the symbol, the larger the difference is between the tilt angles of helices in a packed configuration compared to their individual tilt angle distributions. Group definitions are identical to Fig. 3.13.

practice, the average correlation was positive in all pairs considered. Even though the average correlation in group I-III was rather small, the overall effect of having only positive correlation coefficient suggests somewhat cooperative tilting of all helices. Namely, if one helix tilted to a greater extent than its average tilt angle, so did the other helix, and vice versa.

Tilt Distribution

When calculating the reference distribution we assume each helix samples tilt angles based on its single-helix distribution. This distribution corresponds to results presented in Figure 3.10(a) where each helix is simulated alone in the membrane. In practice, the average tilt angles observed for each helix in a packed configuration are always *smaller* than their respective single-helix distribution. Detailed values are presented in Table 3.3.

We calculate the differences in the distribution of tilt angles in paired versus single helix configuration, by determining the scaled mean distance (Z) for each helix pair (see caption of Table 3.3 for details). A value of $Z = 0$ means the distribution of tilt angles in the paired configuration is identical to the single-helix distribution. The larger the absolute value of Z, the larger the difference is between these distributions.

Results are presented by the x symbols in Figure 3.14(b). These show small differences in tilt angles for groups I-II, with average Z-score of -0.25 (in units of standard deviation). This suggests that although tilt angles are smaller, the deviation from the single-helix histogram is not very large. For helix pairs with super-positive mismatched helices the deviation from the single-helix distribution is much larger. The average Z-score in groups III-IV, comes out to -1.5 (in units of standard deviation).

The observations on tilt correlation and changes in tilt angle distribution when helices are paired present the following picture. The helices are by-enlarge independently tilting, with a small tendency towards cooperative tilting. This results in an overall small but positive tilt correlation coefficient throughout all simulations. The presence of another helix shields some of the constraint originating in the hydrophobic mismatch, causing the helices to tilt less than when isolated in the membrane. When super-positive helices are present, the proposed shielding is enhanced, and both helices tilt significantly less than their respective single helix distribution. If two super-positive helices are in a packed configuration, the change in helix tilt becomes noticeably more cooperative. Combining these findings with the tendency towards small projection angles suggests that group IV helices adopt a tight, closer to parallel configuration where they behave as a single larger peptide and tilt together.

3.6.5 Membrane Thickness Surrounding Helices

In Section 3.2 we envision several adjustments that might be caused by hydrophobic mismatch. So far we have focus on the helix itself, and its response to hydrophobic mismatch conditions. The system, however, includes both the helix and the surrounding lipids. It is fair to expect changes in the lipids surrounding the inclusion of a hyrophobically mismatched helix.

Pair	Mismatch (Å)	Single $\langle\theta\rangle$ (°)	Single σ_θ(°)	Pair $\langle\theta\rangle$ (°)	z-value	Z-Score
(-13,-9)	-13.27	11.19	5.89	11.19	0.00	
	-8.77	9.47	4.93	9.08	-0.08	-0.04
(-9,-9)	-8.77	9.47	4.93	8.98	-0.10	
	-8.77	9.47	4.93	8.95	-0.11	-0.10
(0,0)	0.23	16.66	6.72	15.96	-0.10	
	0.23	16.66	6.72	15.78	-0.13	-0.12
(5,5)	4.73	19.99	7.61	19.55	-0.06	
	4.73	19.99	7.61	19.69	-0.04	-0.05
(5,9)	4.73	19.99	7.61	19.34	-0.09	
	9.23	25.66	7.28	24.55	-0.15	-0.12
(5,14)	4.73	19.99	7.61	19.20	-0.10	
	13.73	30.84	7.97	29.31	-0.19	-0.15
(9,9)	9.23	25.66	7.28	22.27	-0.47	
	9.23	25.66	7.28	22.06	-0.49	-0.48
(9,14)	9.23	25.66	7.28	20.61	-0.69	
	13.73	30.84	7.97	24.55	-0.79	-0.74
(9,18)	9.23	25.66	7.28	20.19	-0.75	
	18.23	32.97	7.18	31.19	-0.25	-0.50
(14,23)	13.73	30.84	7.97	20.28	-1.32	
	22.73	48.27	4.69	42.61	-1.21	-1.27
(18,23)	18.23	32.97	7.18	24.59	-1.17	
	22.73	48.27	4.69	38.09	-2.17	-1.67
(23,23)	22.73	48.27	4.69	39.22	-1.93	
	22.73	48.27	4.69	39.45	-1.88	-1.90
(27,27)	27.23	52.96	4.51	47.58	-1.19	
	27.23	52.96	4.51	47.54	-1.20	-1.20

Table 3.3: Tilt angles of interacting helices. For each pair in Fig. 6 of the main text, we compare the helix average tilt angle when paired to another helix (Pair $\langle\theta\rangle \equiv \langle\theta\rangle^P$) to the helix average and standard deviation in tilt angle when isolated in the membrane (Single $\langle\theta\rangle \equiv \langle\theta\rangle^S$, Single $\sigma_\theta \equiv \sigma_\theta^P$, respectively). Based on the single helix distribution of tilt angles, we calculate the z-value of the observed mean tilt angle in the paired configuration ($z_i = \frac{\langle\theta_i\rangle^P - \langle\theta_i\rangle^S}{\sigma_{\theta_i}^S}$, $i = 1, 2$). We then obtain the scaled mean distance of the pair, $Z = \frac{1}{2}(z_1 + z_2)$. This Z-Score provides a measurement of how likely is it, in units of standard deviation, that the tilt angles of helices in a pair were obtained from the single helix tilt angle distribution. Pair configurations where the tilt angle of each helix remains the same as when isolated correspond to small absolute value Z-score. Pair configurations where the tilt angle of each helix differs greatly from the single helix configuration correspond to large absolute value Z-score.

We sample the membrane thickness around a single TM helix. Thickness was calculated based on the positions of the second head-group bead in the lipid model, in both top and bottom leaflets. This is equivalent to calculating the membrane Phosphorous-to-Phosphorous distance.

Figure 3.15 shows the change in membrane thickness as a function of the distance from the helix center. These results show that the membrane thickness remains mostly unperturbed for zero mismatch helices. For all other helices, negative and positive mismatched, we see a long range adjustment of the membrane thickness to the helix perturbation.

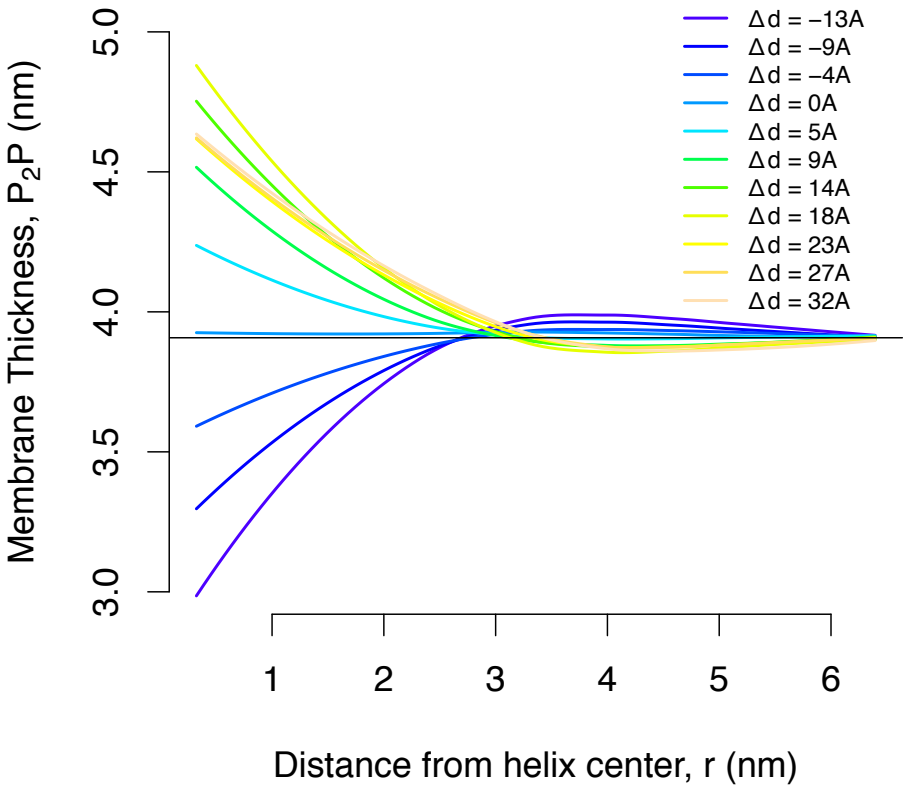


Figure 3.15: Membrane thickness as a function of distance from the helix center. Helix hydrophobic mismatch represented by line color - see legend. Horizontal black line represents the unperturbed membrane thickness (P_2P).

From Figure 3.15 we can extrapolate the thickness of the membrane at the helix center, as a function of the hydrophobic distance. Results are shown in Figure 3.16. These results

show that the membrane thickness at $r = 0$ is roughly linearly increasing with hydrophobic mismatch. A change in this trend is apparent for super-positive mismatched helices, where the membrane thickness reaches a plateau of $d_{\mathrm{M}} = 46.25\text{Å} \pm 0.09\text{Å}$. This value corresponds to $d^0_{\mathrm{M}} + 7.2\text{Å}$.

The linear increase in membrane thickness for negative, zero and positive mismatched helices suggests a robust response. The membrane responses linearly to the perturbation. The lipids surrounding the helix stretch or shrink to accommodate that perturbation, but in a non-cooperative way (hence the linearity). The rest of the adjustment has to come from the helix, that must now change its configuration to be shielded.

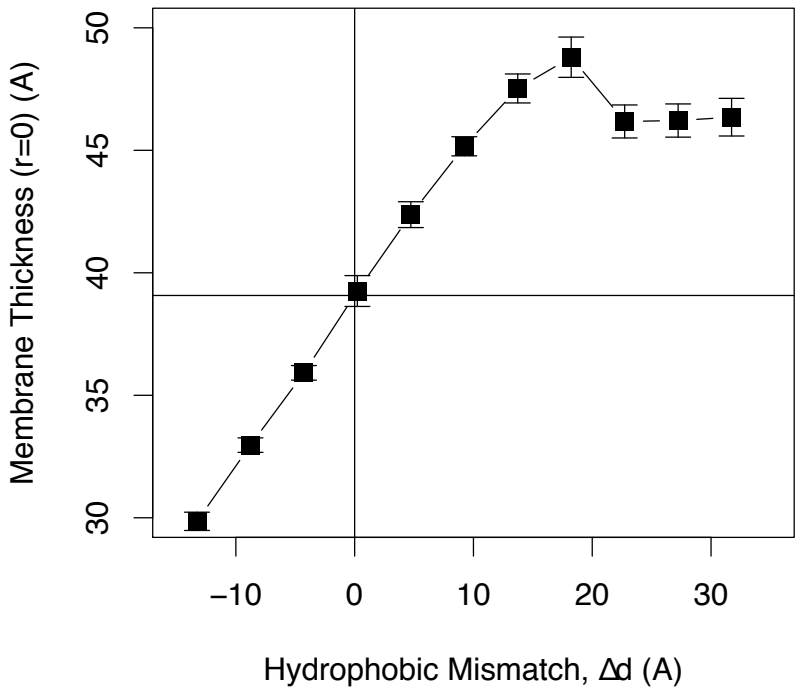


Figure 3.16: Membrane thickness (d_{M}) around a single TM helix. Filled squares represent extrapolated membrane thickness at zero distance ($r = 0$) from the helix. Whiskers represent one standard deviation in thickness. The unperturbed membrane thickness (d^0_{M}) is represented by the solid horizontal line at 39.1Å.

It is interesting to compare the change in membrane thickness (Figure 3.16) with the change in effective hydrophobic mismatch (Figure 3.10(b)). While the effect in the membrane is linear with respect to the hydrophobic mismatch, the effective hydrophobic mismatch shows

distinct regimes (negative and positive) in which the response changes. This originates in the fact that while the lipids have degrees of freedom to both decrease and increase the membrane thickness, the helix is more limited in the available degrees of freedom to accommodate hydrophobic mismatch conditions.

The comparison of these two complimentary results additionally shows why positive mismatched helices do not tilt fully to maintain a 0 effective hydrophobic mismatch. Rather, they display a slope of 0.59 between the effective and physical hydrophobic mismatch. This slope is explained by the fact that the rest of the shielding comes from the membrane linearly increasing its thickness around the helix. Therefore, $\sim 60\%$ of the adjustment in positive mismatched helices originates in the helix and $\sim 40\%$ originates in the membrane.

In super-positive mismatched helices this balance breaks. The membrane maintains a fixed thickness while the helices maintain a fixed effective hydrophobic mismatch. We discuss this effect further in Section 3.6.8.

Interestingly, Figure 3.15 shows that in all helix mismatches we see a similar distance from the helix is needed for the membrane to recover its unperturbed thickness. The membrane thickness seem to universally follow an initial decay, followed by a recoil. We can therefore fit the membrane thickness trend to an underdamped oscillator curve (here r refers to the radial distance from the helix center):

$$d_{\mathrm{M}} = d_{\mathrm{M}}^{0} + A \times e^{-k \cdot r} \cos(\omega r + \phi) \tag{3.3}$$

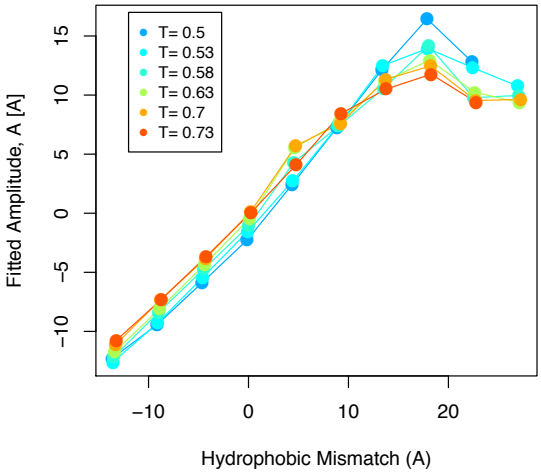
For zero mismatched helices, the underdamping behavior disappears and we see only damping. We can therefore fit those results to the following curve:

$$d_{\mathrm{M}} = d_{\mathrm{M}}^{0} + A \times e^{-k \cdot r} \tag{3.4}$$

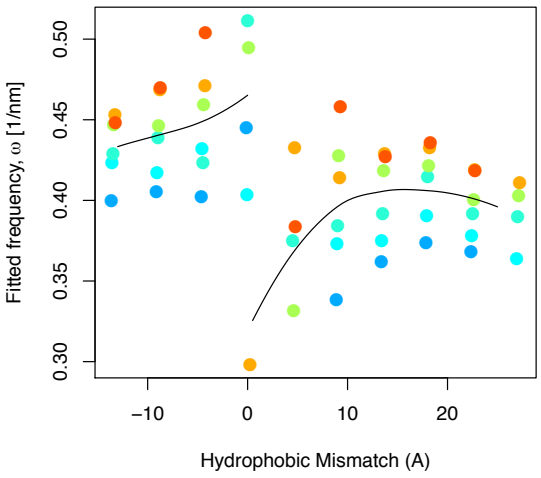
To determine which curve fits better, we use the Akaike's Information Criteria [82]. This criteria takes into account the residuals of the fit as well as the difference in the number of fitting parameters.

We have fitted the curve of membrane thickness as a function of distance from helix center to either Eq. 3.3 or Eq. 3.4 for several different temperatures. The unperturbed membrane thickness, d_{M}^{0}, was determined separately for each temperature, by the average plateau value of the thickness of all helix sizes.

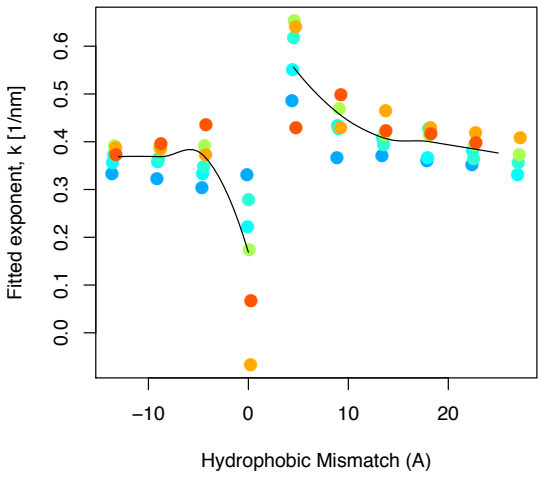
Figure 3.17(a) shows the fitted A parameters for a simulated system of one helix at different temperatures, as a function of hydrophobic mismatch. We see that the trend



(a)



(b)



(c)

Figure 3.17: Fitted amplitude, frequency and exponent of membrane thickness as a function of hydrophobic mismatch. We fit the curve of membrane thickness as a function of distance from helix center to equations 3.3, 3.4 for several simulation temperatures. The fitted parameters of each system is represented by a single point. (a) shows the fitted amplitude, A. (b) shows the fitted frequency, ω. (c) shows the fitted exponent, k. Black lines represent the LOESS [83] fit of negative and positive mismatch values, respectively.

observed by a simple fit of the thickness at $r = 0$ persists even when fitted more rigorously to the equations 3.3, 3.4, and across different temperatures. The thickness amplitude matches the hydrophobic mismatch in a linear trend that breaks for super positive mismatched helices.

Figures 3.17(b), 3.17(c) shows the fitted frequency, ω, and exponent, k, for various helix sizes and temperatures, respectively. Both parameters follow a tangent-like curve and diverge at zero mismatch. The mean values of both parameters are similar ($\langle\omega\rangle = 0.415 \pm 0.04\ {}^{1}/_{nm}$, $\langle k\rangle = 0.385 \pm 0.11\ {}^{1}/_{nm}$). These parameters are influenced by the response distance within the membrane. The fact that those average match and are of small spread suggests that the responses distance is robust and is affected mainly by the lipids, robust to the size of the embedded helix. The response distance is therefore the inverse of those mean values ($\frac{1}{\langle\omega\rangle} = 2.4\ nm$, $\frac{1}{\langle k\rangle} = 2.6\ nm$).

3.6.6 Lipid Tilt Surrounding Helices

Another mean of investigating the influence of embedding a helix in a lipid bilayer is by retracting the changes in lipid tilt. In the course of our CG simulation, we sample the tilt of lipids in the vicinity of TM helices. The lipid tilt angle is defined by the position of lipid tails, as shown in Figure 3.18(b). We define a vector connecting the center-of-mass of the first two tail beads (first out of five beads in each tail) to the center-of-mass of the two last tail beads (fifth out of five beads in each tail). The lipid tilt is then defined as the angle between the constructed vector and the the $+\hat{z}$ direction.

To obtain the tilt angle of lipids around one inserted TM helix, we sample lipids as a function of their radial distance from the helix (see Figure 3.18(c,d)). In each time frame, we loop over several radii away from the helix center-of-mass ($r_i = \{1, 2, \ldots, 10\}\,\mathrm{d}_0$). For each radii bin, i, we maintain all lipids within the range $r \in [r_i - 1.0, r_i)$ of the helix center-of-mass. As the number of lipids in each bin grows with the radius, we randomly chose a subset of the lipids in each bin, such that all bins will contain approximately the same number of samples. To that end we chose to sample a lipid with probability $P(r_i) = 1/(2r_i - 1)$. Note that all the lipids in the bin $r_i = 1.0\,\mathrm{d}_0$ are sampled, and all consequitive bins will contain a similar number of lipids.

Results for the distribution of lipid tilt angle as a function of distance away from the TM helix center are presented in Figure 3.19. For negative mismatch helices we observe a large lipid tilt angle close to the helix center, followed by an exponential decay to a plateau level of $\zeta_{r\to\infty} = 24.1°$.

For positive mismatched helices we observe a different behavior. Close to the helix center ($r \lesssim 3\ nm$) we note an exponential decay. Interestingly we see a decay even when the initial value of the lipid tilt angle is small than the plateau value, $\zeta_{r=0} < \zeta_{r\to\infty}$. The range of this initial decay matches the range of recross to the plateau value in membrane thickness, as seen in Figure 3.15. This initial decay is followed by an increase in the average lipid tilt angle, until an intermediate distance $r \simeq 5\ nm$. This trend suggests a different lipid response in the short ($r \lesssim 3\ nm$) and intermediate ($3\ nm \lesssim r \lesssim 5\ nm$). The intersection between these lipid tilt regions may result in a line tension.

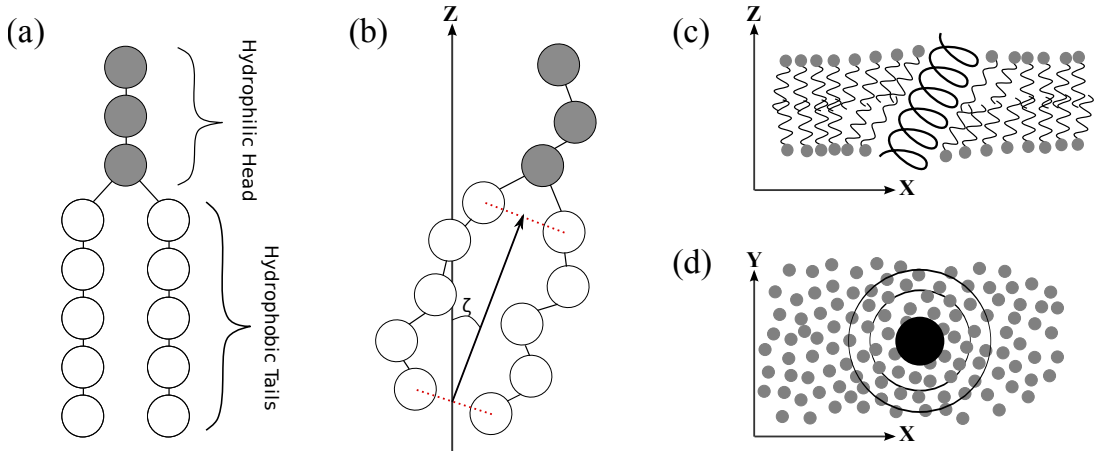


Figure 3.18: Schematic representation of lipid tilt calculation. (a) A model lipid is represented by equally sized beads. Three hydrophilic beads assemble the lipid head group, bonded to two 5-bead hydrophobic tails. (b) The lipid tilt angle, ζ, is definesd as the angle between the vector connecting the first two tail beads with the last two tail beads and the $+\hat{z}$ direction. A schematic picture of a membrane helix inserted into a bilayer is shown in subfigure (c). We analyze the tilt angle of lipid around the inserted helix as a function of the radial distance from the helix center, as shown in (d). The helix core is represented by a black circle while the lipid head groups are represented by grey circles.

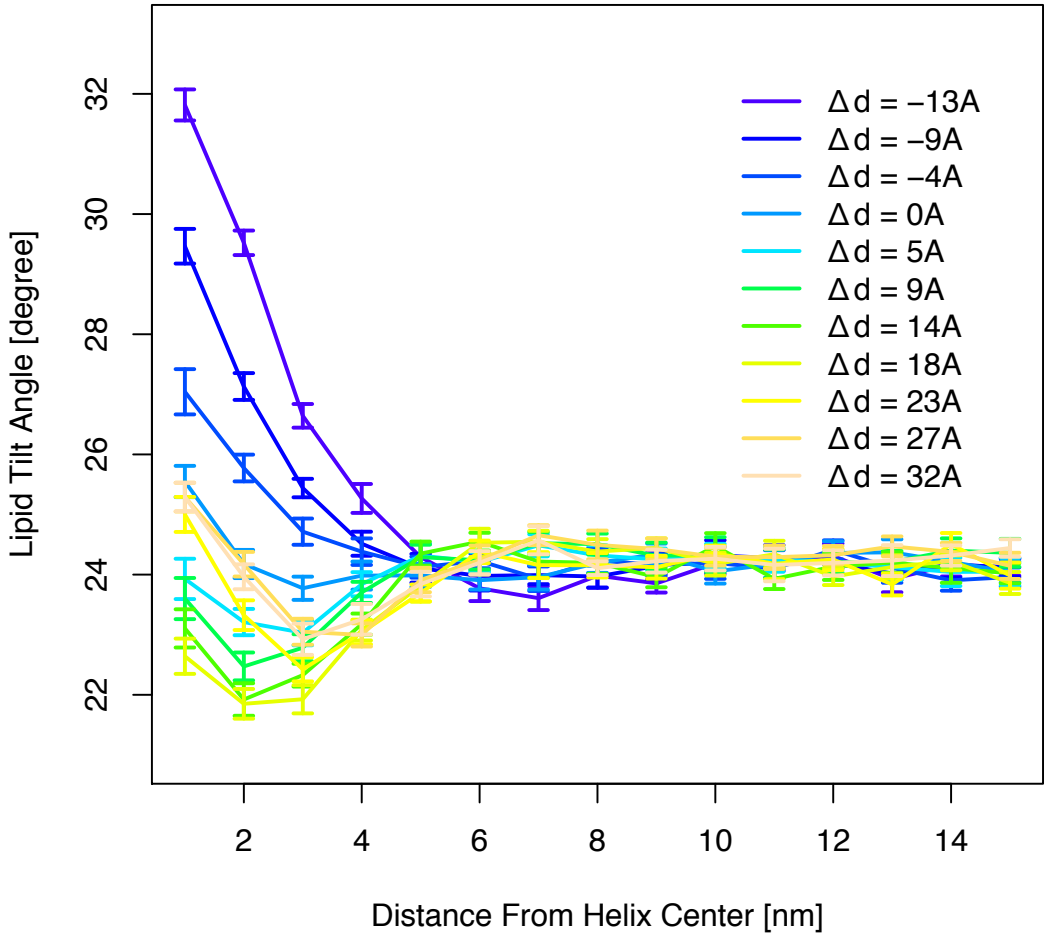


Figure 3.19: Average lipid tilt angle as a function of distance from the center of an embedded helix. Lipid tilt angle is shown for different helix hydrophobic mismatches (see legend).

In super positive mismatched helices we see similar trend as in positive mismatched helices. Namely, an initial decay in lipid tilt angle followed by an increase to the plateau level at intermediate distances. The initial tilt angle, however, is higher. While for positive mismatched helices $\zeta_{r=0}$ decreases with increasing mismatch, this trend breaks when super positive mismatched helices are embedded. We discuss these results further in Section 3.6.8.

3.6.7 Potential of Mean Force

We calculate the potential of mean force (PMF) between pairs of TM helices as a function of the inter-helical distance. To that end we use the traditional umbrella sampling method [84] with a heavy-harmonic biasing potential. That is, we split the entire examined range into overlapping windows specified by a range (ξ_1, ξ_2). The biasing potential for each window corresponds to:

$$U_{bias}(\xi) = \begin{cases} \frac{1}{2}K(\xi - \xi_1)^2 & \xi < \xi_1 \\ 0 & \xi_1 \leq \xi \leq \xi_2 \\ \frac{1}{2}K(\xi - \xi_2)^2 & \xi > \xi_2 \end{cases} \tag{3.5}$$

Since we sample in the $NP\,\gamma\,T$ ensemble, in which the area constantly changes, we preform umbrella sampling in scaled units, *i.e.* $\xi = r/L_x$ where $L_x = L_y$ is the instantaneous box-size along the plane of the membrane and r is the inter-helical distance. We simulate ξ values ranging from $\xi_{min} = 0.035$ up to $\xi_{max} = 0.5$ in windows of size $\Delta\xi = 0.006$. For some helix pairs, containing super positive mismatched helices, a smaller $\xi_{min} = 0.029$ was used, as these helices tend to approach closer to one another. In real units this roughly corresponds to range (8 Å, 120 Å) with windows of size ~ 1.5 Å. The biasing potential constant is set to $K \simeq 50.0\ \epsilon_0/{\mathrm{d}_0}^2 = 78500\ \epsilon_0/[\xi]^2$.

We simulate the system in each window for 20000 cycles. We then unbias and combine the results of each window using the Weighted Histogram Analysis Method (WHAM) [85] to obtain the PMF. We normalize the results to account for the metric jacobian. We reiterate this procedure by adding the inverse of the PMF we obtained in the previous iteration to the biasing potential, until we obtain uniform sampling throughout the entire range. Finally we convert the scaled ξ coordinates to the reduced units by multiplying over by the average box-size: $r = \xi \cdot \langle L_x \rangle$.

The results depend greatly on the mismatch of the two helices in the pair, as shown in Figures 3.20, 3.21. For negative mismatched helices (Figure 3.20(a)), we observe a short-range attraction followed by an intermediate-range repulsion. The strength of attraction, as well as the size of the barrier in intermediate distances, decreases with increasing mismatch (decreasing absolute mismatch). When the two helices are zero-mismatched the barrier diminishes and we observe only a very shallow ($\sim 0.6\ \mathrm{k_B T}$) attraction well.

In positive mismatched helices we observe the reverse trend (Figure 3.20(b)). The depth

of the attractive well increases with growing mismatch. This can be viewed as the same effect as in negative mismatched helices, with respect to growing further from zero mismatch. Namely, the strength of attraction grows with the absolute value of the mismatch. The barrier in intermediated distances, however, is noticeably smaller for positive mismatched helices reaching a shear ~ 0.6 k_BT. Additionally, the width of the attractive wells is wider for positive mismatched pairs, crossing the 0.0 potential line only at $r_0 \sim 33$Å for pair $(9\text{Å}, 9\text{Å})$ compared to a smaller $r_0 \sim 28$Å for pair $(-9\text{Å}, -9\text{Å})$.

Previous theoretical and simulation studies [14, 86–88] showed that the range of attraction between TM proteins is fairly large, extending over several layers of lipid solvation shells, followed by an intermediate range repulsion. This characteristic potential profile is hypothesized to originate in the line-tension forces of the liquid medium (lipid bilayer). Namely, the presence of proteins perturbs the membrane and creates a curved thickness profile. If the proteins are fairly close, it will be favorable to bring them together than to have two centers of perturbation.

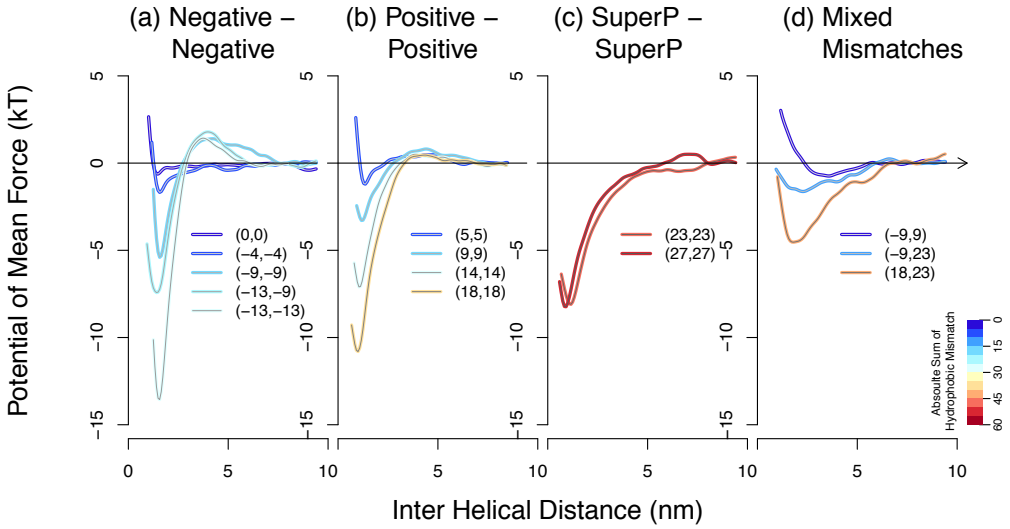


Figure 3.20: Potential of mean force (PMF) between TM helices. Each line corresponds to a pair of helices. Mismatch values for both helices are provided in the legend in units of (Å, Å). Line colors correspond to the absolute sum of hydrophobic mismatch of both helices (see legend in bottom right corner) (a) Shows the PMF of helix pairs who are both negative or zero mismatched; (b) both positive mismatched; (c) both super-positive mismatched, and (d) pairs of mixed mismatch groups. See Fig. 3.21 for detailed error bars.

We see evidence for this behavior in our model as well. The fact that the attraction strength increases with the absolute mismatch, corresponding to larger deformation in the membrane thickness (Fig. 3.16), enforces that hypothesis. If the helices are at intermediate ($\sim 40 - 50$Å) distance from one another, the membrane thickness deformation at the intersection between the two helices has decreased. Bringing the helices closer together in this configuration will

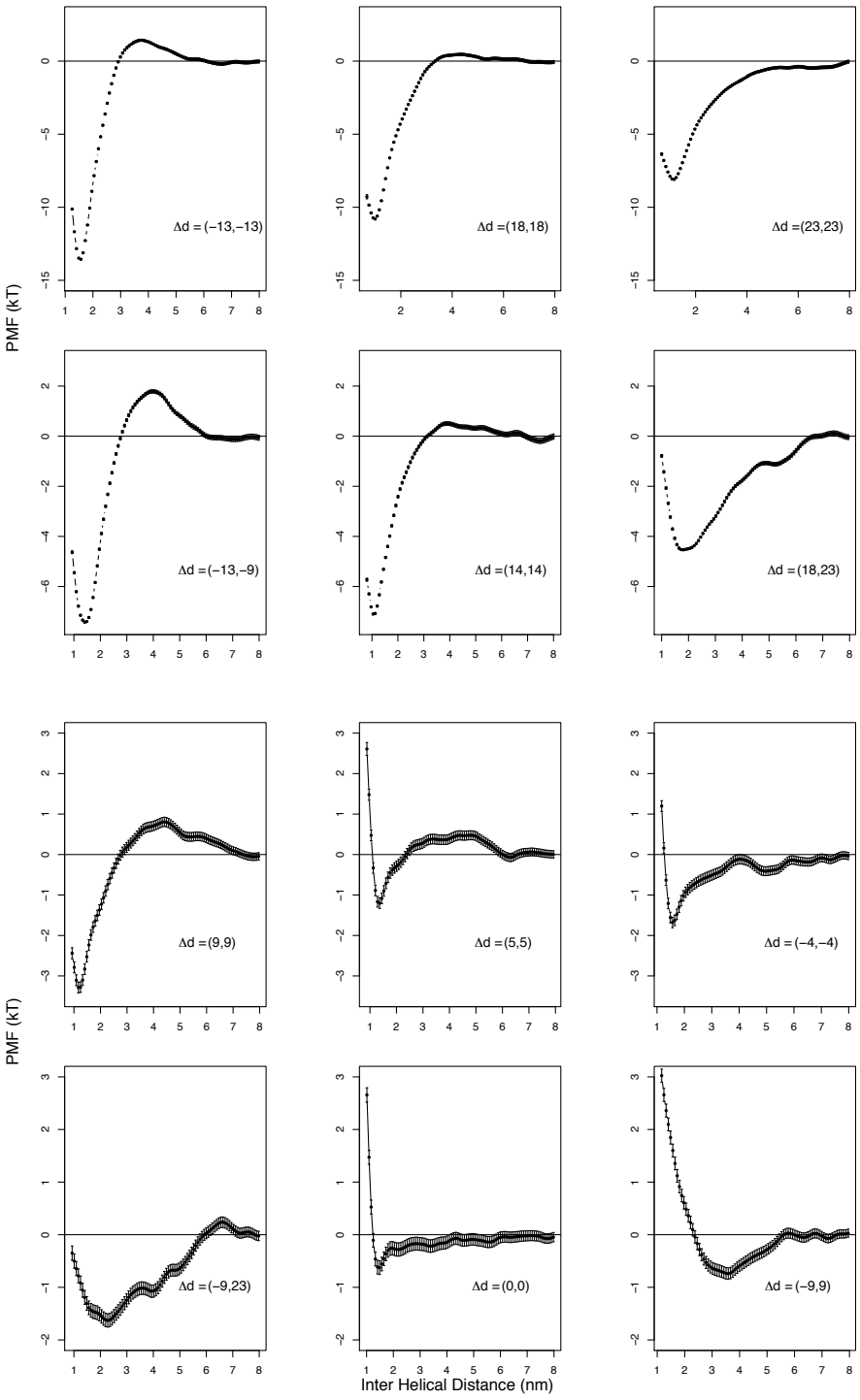


Figure 3.21: Potential of Mean Force for helix pairs with detailed error bars. Each helix pair is displayed in a separate plot with the according error bars. The hydrophobic mismatch values of the helices in the pair are displayed in units of $(\text{Å}, \text{Å})$ in the legend of each plot.

create larger line tension and would therefore be unfavorable, resulting in a potential barrier. At large distances the membrane thickness completely relaxes and so the helices do not feel each other's presence. We observe a plateau at the PMF in those ranges.

The potential of mean force for super-positive mismatched helices is provided in Figure 3.20(c). The strength of attraction for these helices does not grow with mismatch, and is smaller than for the largest positive mismatched pair considered ((18Å, 18Å)). This observation fits the assumption that the potential of mean force between TM helices is greatly determined by the membrane thickness. As we show in Figure 3.16, the membrane thickness at $r = 0$ distance from the helix decreases and plateaus for super-positive mismatched helices. It is therefore expected that the strength of attraction between a pair of such helices, will be smaller than that of positive mismatched helices.

In the last group considered, heterogenous pairs of TM helices (Fig. 3.20(d)), we observe a wide and shallow attractive well. In all three cases examined the depth of the attractive well is smaller than that of the corresponding homogenous pairs. This suggests that TM helices tend to associate with helices of similar mismatch. If the helices are of different mismatch, causing different deformations to the bilayer, there would still be a driving force to localize this disturbance. But, if the deformations are different in nature (negative mismatched helix with positive or super-positive mismatched helix, for example), the attraction would not be very large, reaching at most -1.6 k_BT for pair $(-9Å, 23Å)$.

3.6.8 Super-Positive Mismatched Helices

The results presented in previous sections imply on the existence of a special class of helices, super-positive mismatched. Though rare; some natural TM helices with hydrophobic mismatch greater than 19Å can be found (to be precise, 21 TM helices out of the currently resolved structures, see Table 3.4). Additionally, understanding the driving forces that control the configurations of super-positive mismatched helices can shed light on the forces acting on regular positive mismatched helices. It is therefore important to understand the characteristics of this class of helices.

For super-positive mismatched helices we observe a break-down of the trend between mismatch and tilt angle. It seems that when helices reach that large of a mismatch, the balance between the membrane and the helix energetics is disrupted. It becomes more favorable for the helix to tilt to a greater extent (Figure 3.10(a)), while the membrane thickness drops to a smaller value, closer to the bulk thickness (Figure 3.16) and the lipid tilt angle remains closer to its bulk value (Figure 3.19). The picture that emerges from our simulation analysis is presented in Fig. 3.22. For positive mismatched helices (Fig. 3.22(a),(c)), the lipids accommodate the tilting helix by shielding its hydrophobic core from both top and bottom leaflets. This creates a frustrated lipid configuration as the density of lipid tails and heads deviates from its bulk value. As the helix grows larger so does its tilt angle, and the lipids on both leaflets get further frustrated. When the helix reaches a certain mismatch, it becomes more favorable for the helix to tilt to a greater degree, while the lipids shield it only from one leaflet. Namely, the face of the super-positive mismatched helix pointing

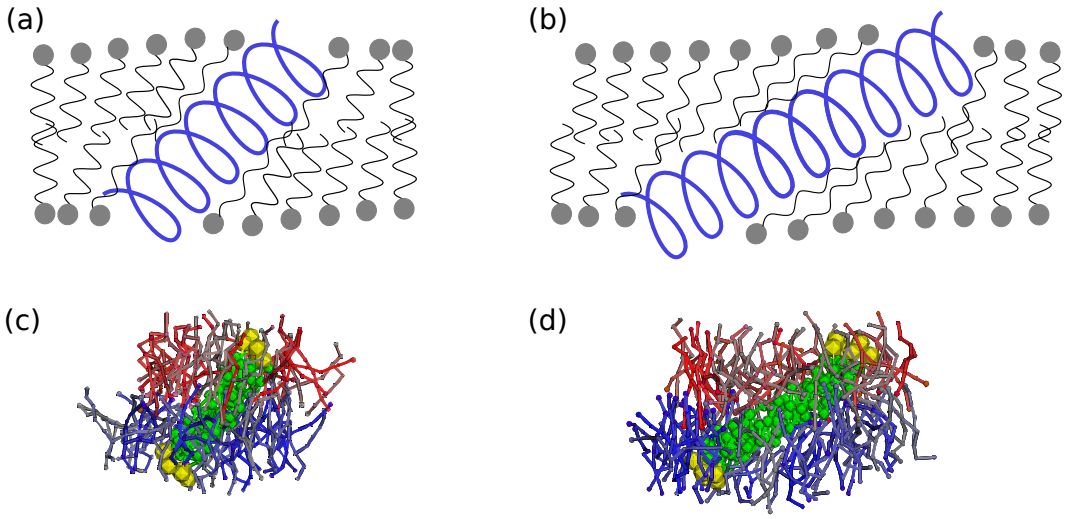


Figure 3.22: Lipid configurations around positive mismatched ((a),(c)) and super-positive mismatched ((b),(d)) helices. (a),(b) Shows a cartoon of lipid orientation around TM helices. (c),(d) Shows snapshots from simulations of helices with mismatch $\Delta d = 13.73\text{Å}, 27.23\text{Å}$, respectively. Lower leaflet lipids are shaded blue and upper leaflet lipids are shaded red or clarity.

'down' ($-\hat{\mathbf{z}}$) would be mostly shielded by lipids from the bottom leaflet; the face of the helix pointing 'up' ($+\hat{\mathbf{z}}$) would be mostly shielded by lipids from the top leaflet. This alleviates the tension between lipid heads and tails.

Super-positive mismatched helices also tend to display a small projection angle and a large correlation of tilt angles when paired with other helices. This indicates a tendency towards tighter configurations, where the helices are bundled together to form one larger peptide.

These phenomena imply an entropic origin. We hypothesize the following scenario. For super-positive mismatched helices, representing a large volume exclusion, it is more favorable for the lipids to minimize the volume of inclusion and maximize the volume available for the lipids. This is done by pushing super-positive mismatched helices closer together into a tight configuration. In positive mismatched helices, the balance is reversed. The entropy of the helices, represented by their ability to explore a larger range of configurations (and projection angles), is greater than that of the lipids. This results in a configuration of rather-freely moving helices displaying a large set of projection angles and rather frustrated lipids in the vicinity of the helix pair.

3.7 Other Protein Models

3.7.1 Cylinder Model

We explored the cross angles of a cylindrical model identical to what is described by de Meyer *et al.* [14]. The peptide is modeled by seven bonded chains, each containing n_{tp} hydrophobic beads at the core and 3 hydrophilic beads at each end. The radius of the cylinder is 6.8Å which corresponds to the radius of an α-helix and to the radius of our CG helical model. This cylindrical model of TM peptides was shown to produce realistic tilt angles [25] and a PMF similar in trends to the one observed here [14, 86].

We have sampled the cross angle in homogeneous pairs of cylinders. The results for cross angle distribution in pairs of $n_{\text{tp}} = 4, 5, 8, 9$ and 10 are presented in Figure 3.24. These chain lengths correspond to hydrophobic mismatch $\Delta d = -10.0$Å, -5.5Å, 8.0Å, 12.6Å and 17.1Å, respectively. Pairs of $n_{\text{tp}} = 6, 7$ ($\Delta d = -1.0$Å, 3.5Å) did not stay in a packed configuration and are therefore not presented. The results show that in all hydrophobic mismatches considered, the mean cross angle distribution was smaller than that of the helical model, averaging on only $\langle\Omega\rangle \sim 14.7°$. This behavior is in contrary to the cross angles typically observed in natural helices [60], where $\langle\Omega\rangle \sim 32°$ and can reach as high as $\Omega = 120°$. The cylinders tend to adopt a tight parallel configuration and not cross. This model is therefore not suitable to describe the short range packing of TM helices.

Typical packed configurations are presented in Figures 3.23(a),3.23(b).

PDB	Chain	Residue Range	Length (Å)	Mismatch (Å)
2jln	A	296 : 329	51.64	21.04
3kcu	A	247 : 278	50.89	20.19
2bl2	B	90 : 122	56.43	20.03
2bl2	D	90 : 122	57.14	20.74
2bl2	G	90 : 122	56.04	19.64
2bl2	H	90 : 122	55.90	19.50
2xq2	A	348 : 380	49.89	19.29
1ots	A	33 : 69	56.39	25.89
1ots	B	33 : 69	56.22	25.72
2iqv	A	120 : 152	56.15	23.45
2a65	A	89 : 124	54.76	24.16
2a65	B	89 : 124	54.76	24.16
3hqk	A	83 : 113	52.32	21.72
3din	D	30 : 64	51.71	24.01
1kpl	A	32 : 66	57.08	26.98
1kpl	B	33 : 67	53.14	23.04
3org	A	90 : 127	58.54	29.04
3org	D	90 : 127	58.53	29.03
2wsx	A	35 : 70	50.14	20.74
2wsx	C	35 : 70	50.11	20.71
2wsx	B	35 : 70	49.93	20.53

Table 3.4: Natural super-positive mismatched helices. Natural helices with hydrophobic mismatch greater than 19Å are described by the PDB entry and chain they are present in, at a certain residue range. Helix hydrophobic length and hydrophobic mismatch are calculated following the analysis in Section 4.2.

3.7.2 Charged Cylinders

We expected the existence of a permanent dipole moment in α-helices to support a crossed configuration of the peptides, rather than a parallel one [89]. We therefore further modified the cylinder model to include partial charges. These charges were chosen to mimic the dipole moment in α-helices, which is a result of all carbonyl groups pointing in the direction of the helix major axis. To that end we added partial charges to beads in the cylindrical model (see Figure 3.23(c)). In DPD, soft repulsive forces are applied between the CG beads. There is therefore no explicit hard core interaction and special care needs to be taken when applying electrostatic forces to avoid divergence at zero distances. We followed the method developed by Groot [90] for adding electrostatics in DPD simulations, and use a smeared-out charge in the center of charged beads.

As we were interested in exploring the local effect of charges on the packing of a pair of peptides, we made some simplifications in our simulation. We applied charges only on a small set of beads along the surface of the cylinder model. We did not model charges on other system components such as water and lipid head groups as these are accounted for explicitly in the un-bonded interaction parameters of those beads [25]. Additionally, we did not account for interaction with the nearest image as we focused on the local interaction of peptides. We therefore calculated the force resulting in electrostatic interactions explicitly by adding the short range contribution. Furthermore, we have assumed that all charges interact through the hydrophobic membrane medium (dielectric constant $\epsilon = 2$ as per Groot[90]), and not screened through the water medium. This assumption represents the extreme case of stronger electrostatic interactions. Based on these simplifications, we are hesitant to say that this model captures the full effect of partial charges along the surface of the peptide. However, we do believe that we have captured the first-order effect of such charges and we don't expect our results to change qualitatively had we done more extensive changes.

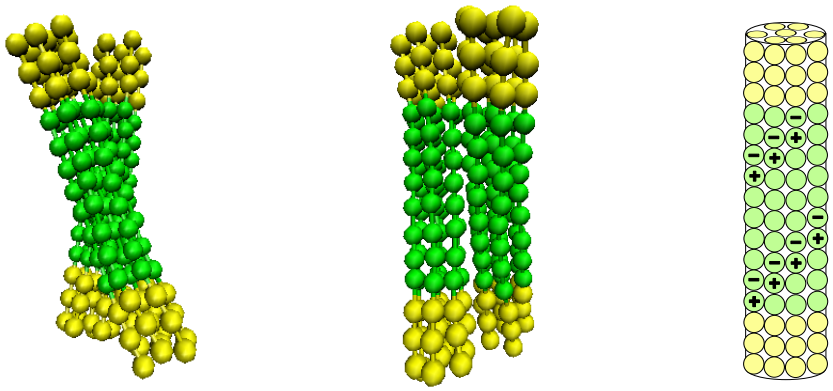
We added partial charges along the surface of the cylinder model, in a helical manner (see Figure 3.23(c)). The charges were added such that the total dipole of 0.5 electron charge [91, 92] is spread out along the surface of the helix. We sampled the cross angle distribution of homogenous pairs of charged cylinders with $n_{\text{tp}} = 5, 6, 8, 9$ and 10. Charged cylinders with $n_{\text{tp}} = 4, 7$ did not remain in packed configurations and were therefore not accounted for in cross angle distribution calculations.

Results for these simulations are presented by the blue filled box-and-whisker diagrams in Fig. 3.24. Even in the presence of partial charges the configuration of paired peptides remained roughly parallel. We observe a slight increase in the average cross angle to $\langle\Omega\rangle \sim 15.6°$, but cross angle values were still much lower than typical cross angles in natural TM helices. The configuration remained parallel, shifting the peptide surface such that positively partially charged beads of one peptide faced the negatively partially charged beads of the paired peptide.

3.7.3 Model: Conclusions

The cylinder model, even in the presence of a dipole moment, did not display any crossed configurations. The helical model, containing the same non-bonded interaction parameters but different geometry, however, did display configurations with a cross angle close to the experimentally observed angles (see for example Figure 3.6(a)). These observations suggest that the helical geometry of beads is crucial for displaying a crossed configuration of TM peptides.

These observations present the following picture. The interface between two cylinders allows 'locking' of the beads in both cylinders. Since the surface of these cylinders is smooth, this locking mechanism is a cumulative sum of contributions from all beads along the surface. It therefore favors a parallel configuration where all beads can be 'locked', over a crossed configuration. In a helix, on the other hand, a similar 'locking' mechanism is frustrated. The packing of beads in both helices is only apparent in a small part of the helix. The helices can not, by geometric constrains, be locked to each other through more than a few residues, unless the helix flexes (as in coiled coils). We therefore hypothesize that the geometry of a helix, in opposed to a cylinder, supports the crossing of helices by *minimizing* the number of locked residues along the surface of the packed helices. In the remainder of this work we focus on the helical model.



(a) (b) (c)

Figure 3.23: Cylinder model of TM peptide. (a) and (b) Show a typical crossed configuration of cylinder model peptides from front and side view, respectively. A crossed configuration of cylinder peptides presents a much lower cross angle than that of helical peptides, as can be seen in Figure 3.24. (c) Shows the distribution of charges along the surface of a cylinder model.

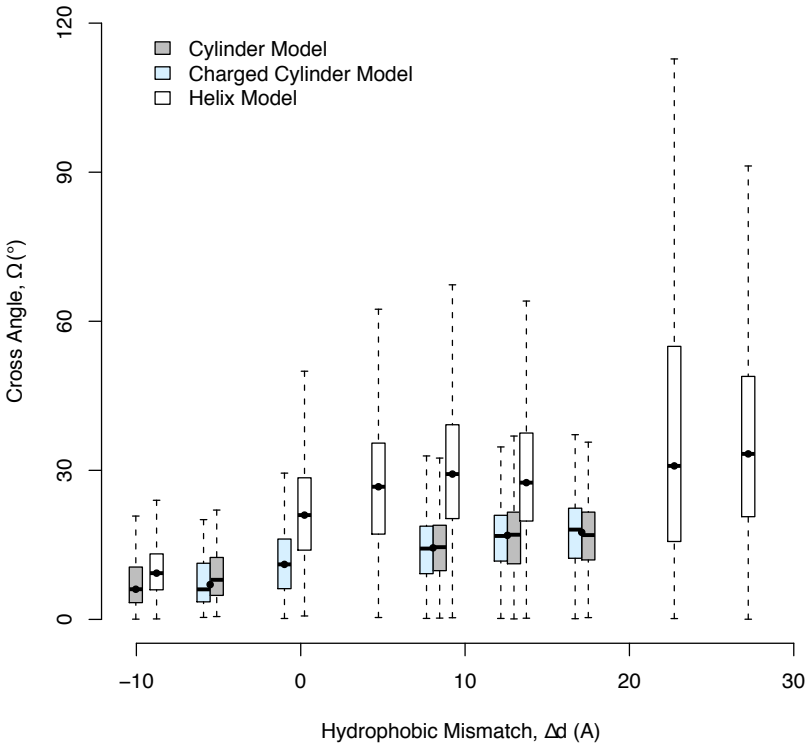


Figure 3.24: Cross angle distribution of homogenous TM peptides as a function of hydrophobic mismatch. Only packed configurations (inter-peptide distance < 15Å) were considered. Filled box-and-whisker diagrams represent the cross angle distribution in the cylinder model of de Meyer *et al.* [14], with (blue) and without (grey) partial charges. Empty box-and-whisker diagrams represent the cross angle distribution in the helix model presented in this work. Black points represent the hydrophobic mismatch for each box-and-whisker diagrams. Charged and uncharged cylinder model have identical hydrophobic mismatch.

3.8 Conclusions

In this work we explored the effect of hydrophobic mismatch on the cross-angle distribution, as well as other configurational parameters, of TM helices. We show that, even in the full range of hydrophobic mismatches, the cross-angle distribution of model TM helices is non-uniform and depends, both in mean and in spread, on the hydrophobic mismatch of helices. We additionally show that the cross-angle distribution of simulated helices matches to a great extent that of two non-interacting helices. Deviations from that distribution are hypothesized to result from membrane mediated forces pushing the helices towards smaller projection angle, for example. Membrane mediate forces also induces association of TM helices.

These results have important implications on understanding the driving forces for helical packing. In our model, we do not define any specific residues. Yet we get a large deviation in cross-angle distribution when presented a different mismatch. This observation is in contrary to current belief that specific residue interaction determine the cross angle. Our results suggest that when attributing a cross angle between helices to a specific residue interaction, one must account for all system components. Hydrophobic mismatch, through its effect on the tilt angle of both helices, might to a large extent explain the cross-angle distribution.

Chapter 4

Membrane Helices Interactions - Experimental Results

The packing structures of transmembrane helices are traditionally attributed to patterns in residues along the contact surface. In this view, besides keeping the helices confined in the membrane, the bilayer has only a minor effect on the helices structure. In this chapter, we show that the lipid environment has a crucial effect in determining the cross angle distribution of packed helices, even in natural helices. We analyzed structural data of a membrane proteins database. We show that the distribution of cross angles of helix pairs in this database is statistically indistinguishable from the cross angle distribution of two non-interacting helices imbedded in the membrane. These results support our previous chapter conclusions that the cross angle is, to a large extent, determined by the tilt angle of the individual helices. Our results indicate that hydrophobic mismatch is the dominant factor guiding the transmembrane helix packing. Other short-range forces might then fine-tune the structure to its final configuration.

4.1 Introduction

Membrane Proteins (MPs) are crucial players in many cellular processes, including the transport of ions across cell membranes as well as numerous signaling pathways. The folding of membrane proteins into functioning units has been proposed to proceed in two stages [93]. In the first stage, secondary structure elements of the protein, comprising mostly of stable transmembrane (TM) α-helices, are generated by ribosomes and inserted into the membrane through the translocon complex [94]. The second stage involves packing of these α-helices into their tertiary and quaternary structures inside the membrane environment.

One of the simplest structural characteristics that form in the second stage of the folding process is the helix-helix pairwise packing. Several studies have shown that the packing of TM helices could have functional consequences for membrane proteins [53–55], which has motivated theoretical research to give a molecular explanation of this packing. Most

theoretical models explain the observed packing in terms of steric "surface matching"; the residues of one α-helix fit into the 'holes' left between the residues of the paired α-helix [95, 96]. This steric fit determines the cross angle between the helices' major axes. Motifs of repeating small residues along the surface of TM helices, which would provide a characteristic pattern of 'holes', were thought to support this idea [56–59]. Walters *et al.* [60] further showed that two-thirds of TM helix pairs found in the protein database classify into a small set of subcategories, with each subcategory having a characteristic side-chain packing motif. On the basis of this observation, Walters *et al.* concluded that specific residue interactions control the helix -helix cross angle.

These previous theories explain the TM helix packing angles on the basis of protein-protein interactions alone and assume that the surrounding membrane has little to no role in shaping the structure of MPs. Recent studies have indicated, however, that the functional activity of many membrane proteins depends on the properties of the membrane, such as lipid composition and thickness [11, 12]. Examples include the nicotinic acetylcholine receptor [97], the thermosensor protein DesK [98, 99], the cation/sugar symporter MelB [100] as well as other proteins (see Table 1 in Andersen *et al.* [13]). These results suggest that various characteristics of the membrane environment could modulate the function of proteins by determining their structure and dynamics. Based on the examples presented in [97–100], Holt & Killian [11] propose that MPs have an optimal bilayer thickness for their functioning. A non-optimal thickness, hey hypothesize, can lead to changes in the structure and orientation of transmembrane units, which can, in turn, lead to changes in protein functionality.

At present, we have very little understanding of the role of the membrane on helix-helix packing. Hydrophobic mismatch, the difference in length between the hydrophobic medium of the lipid bilayer and the hydrophobic span of the protein, has been proposed as a major determinant of TM helix configurations. In general, this feature determines the extent to which the membrane is perturbed by the presence of a protein. If a protein in the membrane has a positive hydrophobic mismatch, for example, either the membrane locally thickens or the protein tilts to minimize the hydrophobic interactions. Studies have shown that hydrophobic mismatch is also a good indicator of the driving force of the oligomerization of helical peptides [14, 63, 66–70]. In addition, hydrophobic mismatch is also an important determinant of the helical tilt within the membrane [11, 61–65]. Studies show that proteins with a large hydrophobic mismatch have a larger tilt angle with respect to the normal of the bilayer. Proteins with a negative mismatch behave differently. In these cases it is important to distinguish between proteins that are truly transmembrane, *i.e.*, the hydrophilic ends are in contact with water, and those that are fully in the hydrophobic region of the membrane. For the first group the tilt will be close to zero [62], while for the latter the orientation will be random.

These findings show that lipid-protein interactions can have a large effect on the tilt angles and interactions of membrane α-helices. At present, however, we have very little insight regarding the role of the membrane in the details of the folding process. In this article we study the role of the surrounding membrane in the formation of the TM helix-helix packing. Inspired by the importance of hydrophobic mismatch in describing helix-helix interactions, we

analyze an extensive database (Orientations of Proteins in Membranes, OPM) [101] containing a large number of experimental MP structures to investigate whether or not helix packing angles are correlated to the hydrophobic mismatch. Surprisingly, our analysis shows that packing angles of TM helices can be explained sufficiently by the hydrophobic mismatch. This indicates that even in the absence of any specific protein-protein interactions the TM helix packing is dictated by the hydrophobic mismatch of the proteins.

We compare these experimental structures with our CG model of TM helices described in Chapter 3. By comparing our simulation results with the experimental results of the OPM database, we obtain more insights on the relative importance of these membrane mediated interaction and the direct interactions between the proteins.

4.2 OPM Database Analysis

For the experimental TM helix packing we use the recently established Orientations of Proteins in Membranes (OPM) database [101]. This database provides information not only on the protein structures, but also on the thickness of the membranes in which they were embedded (see Figure 4.1).

4.2.1 Extracting experimental helices

Our analysis of the OPM database consists of the following steps. For each protein we first identify all transmembrane segments. For those TM segments that have a helical structure we then define their orientation and the hydrophobic mismatch. Finally, we identify those transmembrane helices that are in direct contact (neighboring pairs). For these pairs we can then determine the cross angle. From a total of 261 MPs we determined a little over 2300 cross angles.

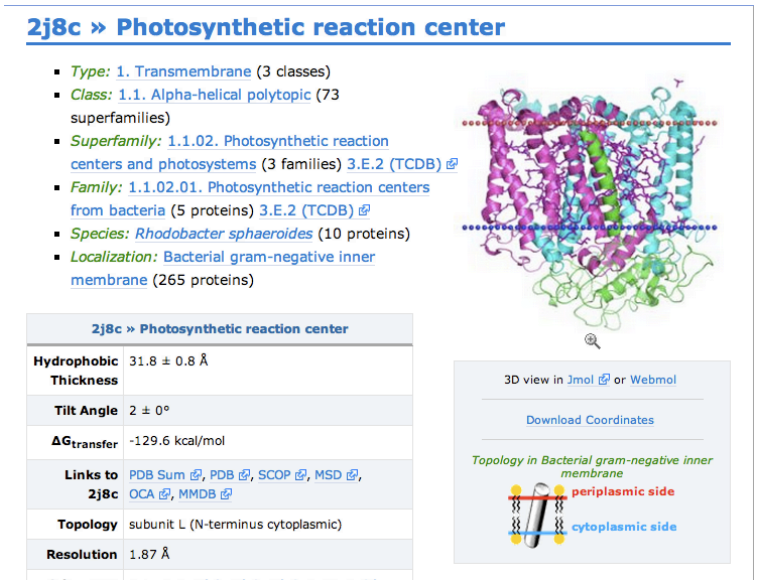
In more details, the extraction of experimental helix structure data from the OPM database included the following steps:

1. We refine the database to include only proteins with TM helices. This includes *alpha-helical polytopic* and *alpha-helical bitopic* OPM classes as of September 2011 (357 TM proteins, 261 unique). The full list of MPs is available in Table 4.1.

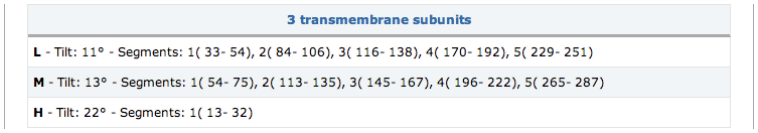
2. We extract the structure of each such protein from the database.

3. We extract other supporting information on the protein such as the hydrophobic thickness of the membrane surrounding it (d_{L}) and the list of the protein's transmembrane segments.

4. For each TM segment of the protein we analyze the secondary structure of its residues using DSSP [103] to determine whether it has a range of residues with helical geometry.



(a)



(b)

Figure 4.1: A Sample of structure 2J8C (Photosynthetic reaction center from Rhodobacter sphaeroides) in the OPM Database. Structure was obtained by 1.87 Å resolution X-Ray diffraction [102]. OPM shows information on the hydrophobic thickness of the membrane the protein is embedded in (a) as well as on the transmembrane subunits the protein is constructed from (b). We extract this information automatically from the OPM database to retrieve the set of TM helices and their hydrophobic mismatch.

5. For each TM helix we determine the range of residues that span its hydrophobic core. The original TM segment is therefore extended if it includes more hydrophobic-scored helical residues at either end or shortened if it includes hydrophilic-scored helical residues at either end. Details on the hydrophobicity scoring function is available in Section 4.2.2.

By the aforementioned process we obtained a list of hydrophobic helix cores for each membrane protein. We further analyze each hydrophobic helix core with the following steps:

1. We refine the set of TM core helices to include only helices with more than 9 residues, as was done by Walters *et al.* [60].

2. We analyze the structure of each helix core to define its major axis using the HELANAL [104] algorithm available through the MDAnalysis [105] package.

3. We obtain the length of the hydrophilic part of the helix, d_{H}, by projecting the C_α positions (origins) onto the obtained helix axis vector.

4. We define the hydrophobic mismatch of each helix by subtracting the thickness of the membrane ($\Delta d = d_{\mathrm{H}} - d_{\mathrm{L}}$) for each helix.

5. We refine the set to analyze helices that are in the bulk range (96%) of hydrophobic mismatch (-10 Å $\leq \Delta d \leq 35$ Å) and have a reasonable length per residue (in range [1.25,1.8] Å/residue; theoretical length per residue is 1.5 Å/residue).

6. We eliminate kinked helices by screening helices that are not centered around the bilayer center. Helices whose center-of-mass distance from the bilayer center ($z = 0$) in the bilayer normal direction (z) exceeded 20% of the helix length were defined as part of a kinked helix and were not used in further analysis.

7. We define each helix as *unique* / *non-unique* based on its similarity in sequence and TM segment range to other helices in the same MP. Therefore if a protein was in fact a dimer of two smaller subunits, only helices of the first monomer were defined as *unique* while the other helices were tagged as *non-unique*.

Contacts between same-protein helices were calculated following the analysis in Gimpelev *et al.* [106]. Two helices were defined to be a neighboring pair if they contained at least three residues in contact. Residues were defined to be in contact if the distance between any two of their atoms was within 0.6 Å of the sum of their van der Waals radii [107], as calculated by Li & Nussinov [108]. Helix pairs were restricted to pairs in which at least one of the helices is *unique*, thereby eliminating over-counting of helix pairs. The number of contacts each helix has with other same-protein helices was also used as a measure of how influenced the helix is by other surrounding helices. Helices with less than one contact per helix residue (average 0.57 contacts / residue) were defined as *peripheral helices* with minimal helix-helix interactions.

By the aforementioned process we extract 3002 helices from 261 unique transmembrane proteins in the OPM database. Of this set, 1542 were defined as unique with 224 of these defined as peripheral helices. We obtained a total number of 17695 pairs of helices, with 2328 of them (13%) as neighboring pairs.

4.2.2 Hydrophobic mismatch calculations

We define hydrophobic mismatch, Δd, as the difference between the hydrophobic length of a helix, d_{H}, and the hydrophobic thickness of a membrane, d_{L}, following Equation 3.1.

For all MPs obtained from the OPM database, we used the effective membrane hydrophobic thickness as reported in the database. We therefore rely on the membrane thickness estimations made by Lomize *et al.* [109, 110]. In their calculations, Lomize *et al.* used the free energy of transferring a protein between water and a hydrophobic region to construct the membrane hydrophobic thickness around a given protein. It is important to note that most experimental structures for MPs required the usage of a detergent that is necessary for the crystallization process. These detergents are assumed to maintain the protein structure. We therefore used the membrane hydrophobic thickness as reported in OPM to determine the effective membrane hydrophobic thickness. This provides a reasonable representation of the membrane environment of the protein at equilibrium, rather than the temporary detergent environment.

To determine the helix hydrophobic length of OPM extracted helices, we use a non-local hydrophobicity score. First, for each residue i in a TM helix we calculate a local hydrophobicity score, s_L^i, based on the Kyte and Doolittle hydrophobicity scale [111]. We then calculate the non-local hydrophobicity score for each residue by a weighted average of the local scores in a window of five surrounding residues (including the current residue and the two adjacent residues on either side). The weights were calculated by a Gaussian of mean zero and standard deviation of one residue. The non-local score of residue i, s_{NL}^i, is therefore computed by $s_{NL}^i = 0.054 \cdot s_L^{i-2} + 0.242 \cdot s_L^{i-1} + 0.399 \cdot s_L^i + 0.242 \cdot s_L^{i+1} + 0.054 \cdot s_L^{i+2}$. This non-local score prevents a single isolated hydrophilic residue in the middle of the helix from being seen as a hydrophilic end.

We define the hydrophobic helix core by extending or decreasing the range of helical residues to include only hydrophobic-scored ($s_{NL}^i > -0.02$) residues at both ends. The total range of residues in the hydrophobic core of the helix determines the helix hydrophobic length, d_{H}. This length is determined by the HELANAL algorithm for analyzing helix structures [104].

We note here that the membrane hydrophobic thickness, d_{L}, is different for each protein and therefore the hydrophobic mismatch of each helix, Δd, differs from the helix length, d_{H}.

4.3 Database Analysis Results

4.3.1 Tilt angles

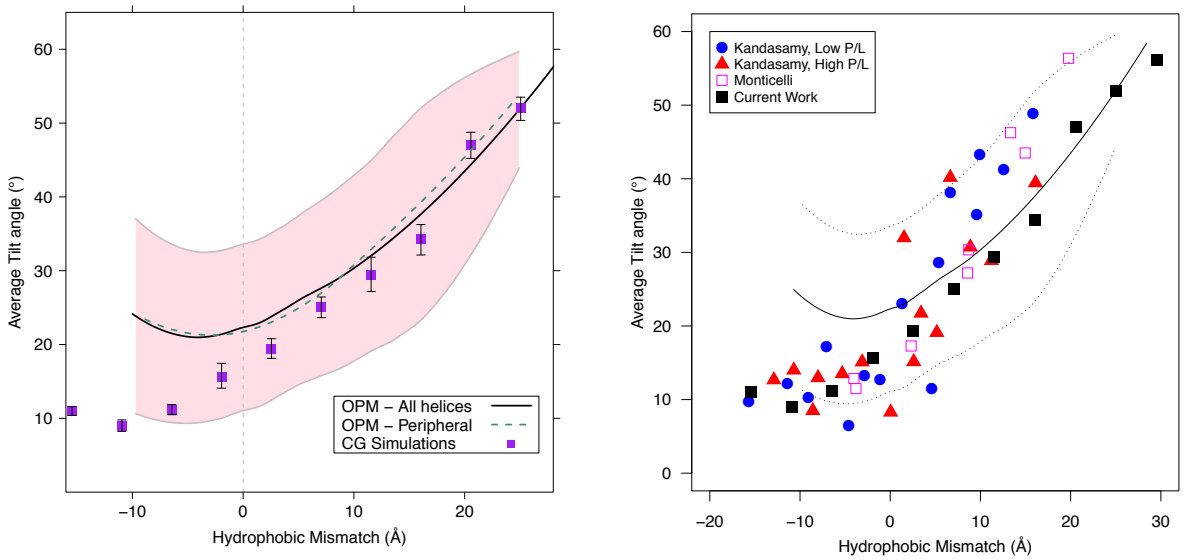
For the 1542 unique helix structures in the OPM database we computed the tilt angle as a function of the hydrophobic mismatch. The tilt angle, θ, was calculated as the angle between the helix major axis and the bilayer normal vector, $+\hat{\mathbf{z}}$ (see Figure 3.6(b)). For a given value of the hydrophobic mismatch we observe a distribution of different angles. In Figure 4.2(a) we represent this distribution by plotting the average and standard deviation. We additionally compare these result with the result of our CG model, described in the previous chapter.

For positive mismatch we observe that the tilt increases with hydrophobic mismatch. Such a trend has also been observed for WALP and KALP peptides [11, 34, 62] and in simulations of our course grained model (see Figure 4.2).

For negative mismatch the OPM data shows a minima for a mismatch of $\Delta d \approx -2\text{Å}$. These results disagree with our CG simulations and simulation of KALP peptides for negative mismatch. At this point it is important to recall that in our CG model, the interactions are chosen such that even for negative mismatch the hydrophilic ends of the helix are in contact with the hydrophilic part of the membrane (or water). Hence, for these systems tilt will be energetically unfavorable [11, 62]. Our model is consistent with the results for WALP and KALP peptides, for which the hydrophilic part of a helix, even for negative mismatch, is in contact with the hydrophilic part of the membrane or in the water phase [34, 64]. Most helices with negative mismatch in the OPM database, however, are not capped by hydrophilic regions, but rather are continued by hydrophobic non-helical chains. Hence, for these helices changing the tilt does not change the energy and the orientation of such helices is expected to be random. Because of these differences we focus in the remainder of this article on the cross angle of positive hydrophobic mismatch helices.

The results for KALP peptides, as seen in Figure 4.2(b), have a larger slope of tilt angle versus hydrophobic mismatch compared to the OPM data and our CG simulations. In our CG model, we chose the size of the hydrophilic part of the helix (3 residues) to match the hydrophilic part of the membrane. Natural helices from the OPM database often have a chain of several hydrophilic residues at both sides of the helix. KALP peptides, on the other hand, contain only two hydrophilic residues. In the absence of hydrophilic residues to match the hydrophilic part of the membrane, the energetic balance between the hydrophobic mismatch and the counter hydrophilic mismatch is shifted such that the hydrophobic mismatch might be more pronounced. In that case we expect to observe a larger tilt angle with similar hydrophobic mismatch.

While analyzing the database we observed a strong 'mirror effect' for parallel versus anti parallel helices. Namely, the distribution of tilt angle and cross angle was similar for all helices, regardless of their C to N terminus orientation. Evidence for this effect is shown in Figure 4.3. Therefore, we follow the definition given by Chothia *et al.* [112] and treat all helices as vectors pointing towards the $+\hat{\mathbf{z}}$-direction along the calculated helix axis. All our results are reported using this convention.



(a) (b)

Figure 4.2: Average tilt angle as function of hydrophobic mismatch. (a) Compares tilt angle for all unique TM helices (black line), peripheral unique TM helices (dashed green line, see text) as obtained from the experimental data in the OPM database with CG simulation of hydrophobic helices in a lipid bilayer (purple squares). The experimental data in the OPM database gives a distribution of tilt angles, which is represented by the solid line, for the average, and the pink shading, for one standard deviation (as obtained by the LOESS method [83]). The simulated CG results are an average from system of both 5-bead and 6-bead tail lipids, representing two membranes with different hydrophobic thicknesses, with error-bars representing the error in average values of both systems. (b) Further compares our simulation (black filled squares) and OPM experimental data (solid black line, standard deviation represented by dashed lines) with previously published simulation results for KALP peptides. Results by Kandasamy *et al.* [62] are shown for low (blue filled circles) and high (red filled triangles) protein to lipid ratio. Results of coarse grained simulations by Monticelli *et al.* [34] are shown with magenta empty squares.

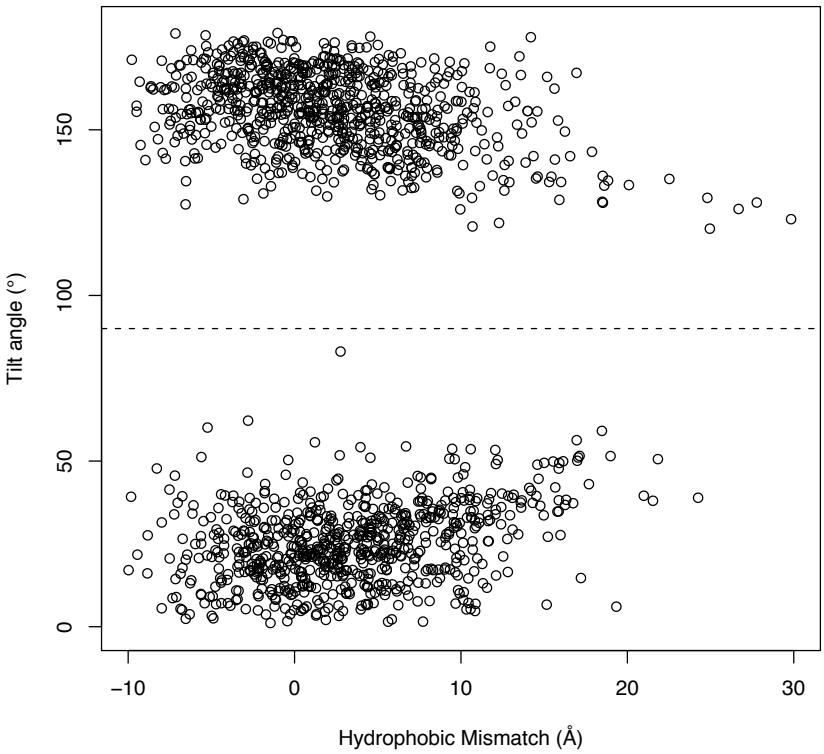


Figure 4.3: Unadjusted tilt angles versus hydrophobic mismatch scatter plot shows a mirror effect around $\theta = 90°$ (dashed horizontal line).

We find a surprising trend in the subset of peripheral helices (14% of helices). As these helices are mainly at the outer regions of the protein, one would expect the effect of the lipid bilayer on the tilt to be more pronounced, while for the non-peripheral helices the helix-helix interactions would dominate. Instead, we observe the tilt of these peripheral helices to be similar (Figure 4.2(a), dashed green line). This suggests that the presence of another helix nearby is of a lesser importance to the tilt than the presence of the surrounding bilayer.

We note here that our results for the tilt angle of OPM helices depend on the membrane thickness as determined in the OPM database (see Section 4.2.2). As the definition of the membrane thickness is somewhat arbitrary, a different definition of the membrane thickness could lead to quantitative changes in the tilt angle trend. Yet we expect that the qualitative behavior of the tilt angle with respect to hydrophobic mismatch would not change.

4.3.2 Cross angle distributions

Reference Cross Angle Distribution

Bowie [113] showed that there are significant statistical biases that have to be taken into account when referring to inter-axial angles of packed α-helices. Namely, using geometric considerations one can show that the number of configurations available to a pair of helices with a fixed cross angle depends on the value of that cross angle. This leads to a non-flat cosine-like distribution of cross angles. Bowie [114] further explored this effect by analyzing the packing of membrane proteins. However, at that time the number of resolved membrane protein structures was too small to study this effect.

When studying the cross angle distribution we also have to account for a statistical bias that originates from the fact that the cross and tilt angles are not independent. The relation between these angles is described in Section 3.6.3 and Equation 3.2.

As mentioned, this relation confines the cross angle to the range $|\theta_1 - \theta_2| \leq |\Omega| \leq |\theta_1 + \theta_2|$ and displays a non-uniform Arcsine-like distribution within that range. This distribution displays two sharp peaks at $|\Omega| = |\theta_1 - \theta_2|$ and $|\Omega| = |\theta_1 + \theta_2|$ (see Figure 4.4, dashed line). The importance of this relation is that even for a pair of helices that are so far apart in the membrane that they don't interact, we see a limited set of possible cross angles between these helices. For helices with smaller tilt angles we will observe a smaller cross angle and for a pair of helices with larger tilt angles we will observe a larger one. The cross angle is also expected to be larger for a case of large difference in tilt angles ($|\theta_1 - \theta_2| \gg 0$) and to display a smaller value for similar tilt angles ($|\theta_1 - \theta_2| \sim 0$).

For fixed tilt angles of the two helices, the projection angle between the helices, γ, is the only independent variable controlling the cross angle of the two helices. For non-interacting helices, the tilt angles are uncorrelated and this projection angle is uniformly distributed. Taking this into consideration, we calculate the distribution of cross angles in the OPM database for the case that the two helices would not interact. For a given mismatch, we assume a Gaussian distribution of tilt angles, where the mean and variance of the distribution follow from the data in Figure 4.2(a). We sample a large number of tilt angles from these

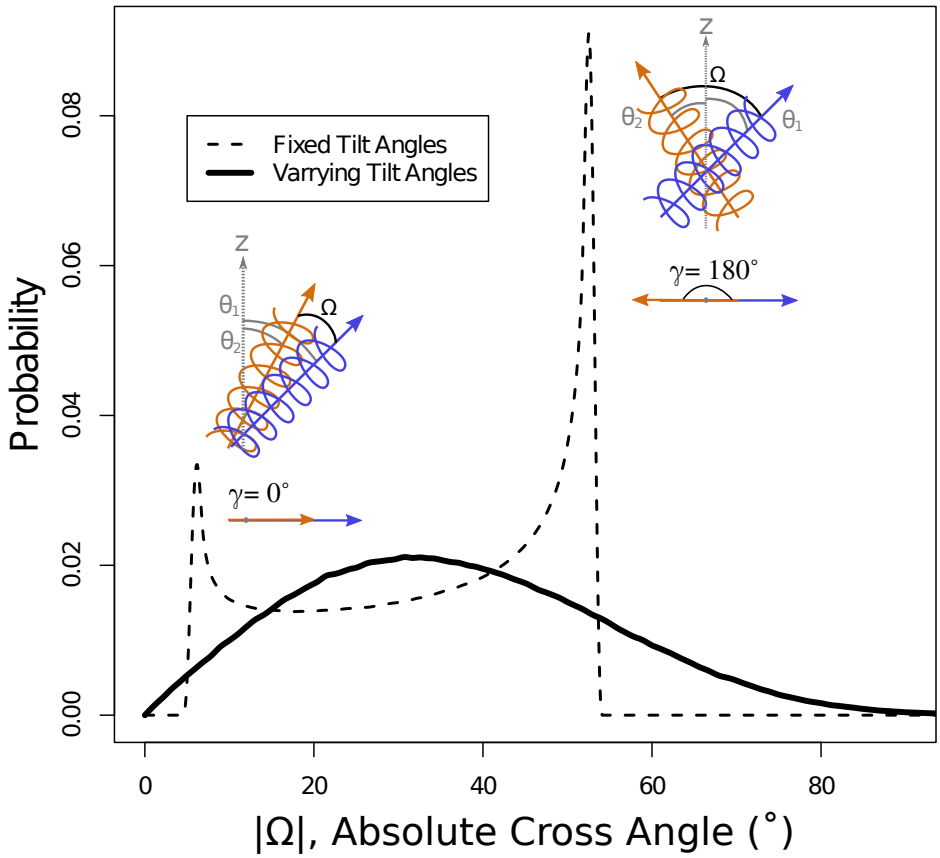


Figure 4.4: Reference Cross angle (Ω) Distribution. Dashed line corresponds to the reference cross angle distribution of two helices with fixed tilt angles, θ_1' and θ_2', and random projection angle, γ; Solid line corresponds to the reference cross angle distribution of two helices with normally distributed tilt angles with mean θ_1', θ_2' and standard deviation σ_1, σ_2, respectively. These mean and standard deviation values correspond to the experimental tilt angles at hydrophobic mismatch ranges $\Delta d_1 = 3.0$ Å and $\Delta d_2 = 10.0$ Å respectively, as extracted from Fig. 4.2(a). These amount to $\theta_1' = 23.6°, \theta_2' = 29.4°, \sigma_1 = 11.3°$ and $\sigma_2 = 12.4°$. A schematic sketch of helices is provides for both extremes of $\gamma = 0°$ and $\gamma = 180°$.

distributions for each helix and uniformly sample the projection angle. We calculate the resulting cross angles using Equation 3.2.

Each pair of fixed tilt angles will impose a cross angle distribution as shown in Figure 4.4 (dashed line), and the overall distribution will be a weighted average of these individual distribution. The solid line in Figure 4.4 is an example of such a distribution for a pair of helices that are drawn from the distribution corresponding to hydrophobic mismatches of 3 Å and 10 Å. This figure illustrates that, even in absence of any interactions between the helices, we obtain a dependence of the cross angle on the hydrophobic mismatch. To compute the reference distribution of cross angles for all neighboring helix pairs in the OPM database, we repeat this procedure for all possible mismatches of pairs of helices, $(\Delta d_1; \Delta d_2)$.

Experimental Cross Angle Distribution

We calculated the cross angle between all pairs of helices within the same protein in the OPM database. In determining the cross angle one can distinguish between a right- and left- handed angles [112]. The histograms of left and right handed cross angles are close in absolute value (see Figure 4.5) and there is are no major differences in preference for a right or left handed orientation. Therefore in the remainder of this section we use absolute cross angle values, which allows us to improve the statistics.

To determine the extent to which the OPM distribution differs from the one for non-interacting helices, we first divided the pairs of helices into groups based on their hydrophobic mismatch. We then analyzed the cross angle distribution of two representative groups of neighboring helix pairs: (a) $\Delta d_1 \in (2, 5]$ Å , $\Delta d_2 \in (4, 7]$ Å containing 175 helix pairs, and (b) $\Delta d_1 \in (0, 3]$ Å , $\Delta d_2 \in (0, 3]$ Å containing 135 helix pairs.

In Figure 4.6 the OPM data are compared with the non-interacting reference curve. Surprisingly, this comparison shows no significant differences between the two distributions. The notion that the cross angle distribution is not determined by the helix-helix interaction but determined by the orientation of the helices in the membrane is further illustrated by a comparison with the distribution obtained from our molecular simulations using a similar mismatch ((a) $\Delta d_1 = 3.2$ Å, $\Delta d_2 = 7.7$ Å and (b) $\Delta d_1 = 3.2$ Å , $\Delta d_2 = 3.2$ Å). Figures 4.7, 4.8 further illustrates the change in mean cross angle as a function of the hydrophobic mismatch and tilt of both helices.

To construct the reference distribution of each cross angle, we sampled the reference distribution by a Monte Carlo procedure for all neighboring helix pairs in the OPM database. For each pair, we extract the hydrophobic mismatches of both helices, $(\Delta d_1; \Delta d_2)$. This imposes the expected tilt angle distribution of each helix, based on the information displayed in Figure 4.2(a), $(\langle\theta_1\rangle, \sigma_{\theta_1}; \langle\theta_2\rangle, \sigma_{\theta_2})$. Note that we do not explicitly use the helices tilt angle but rather the average distribution based on their mismatch, therefore obtaining the reference distribution in the case that hydrophobic mismatch would be the only factor determining the helix tilt. We then produce a set of $N = 1000$ values of tilt angles for each helix in the pair, drawn from the imposed distribution, $\theta_1^i \sim \mathcal{N}(\langle\theta_1\rangle, \sigma_{\theta_1})$ and $\theta_2^i \sim \mathcal{N}(\langle\theta_2\rangle, \sigma_{\theta_2})$, for $i = 1 \ldots N$. For each i, a set of $j = 1 \ldots M = 1000$ uniformly distributed projection γ_j^i

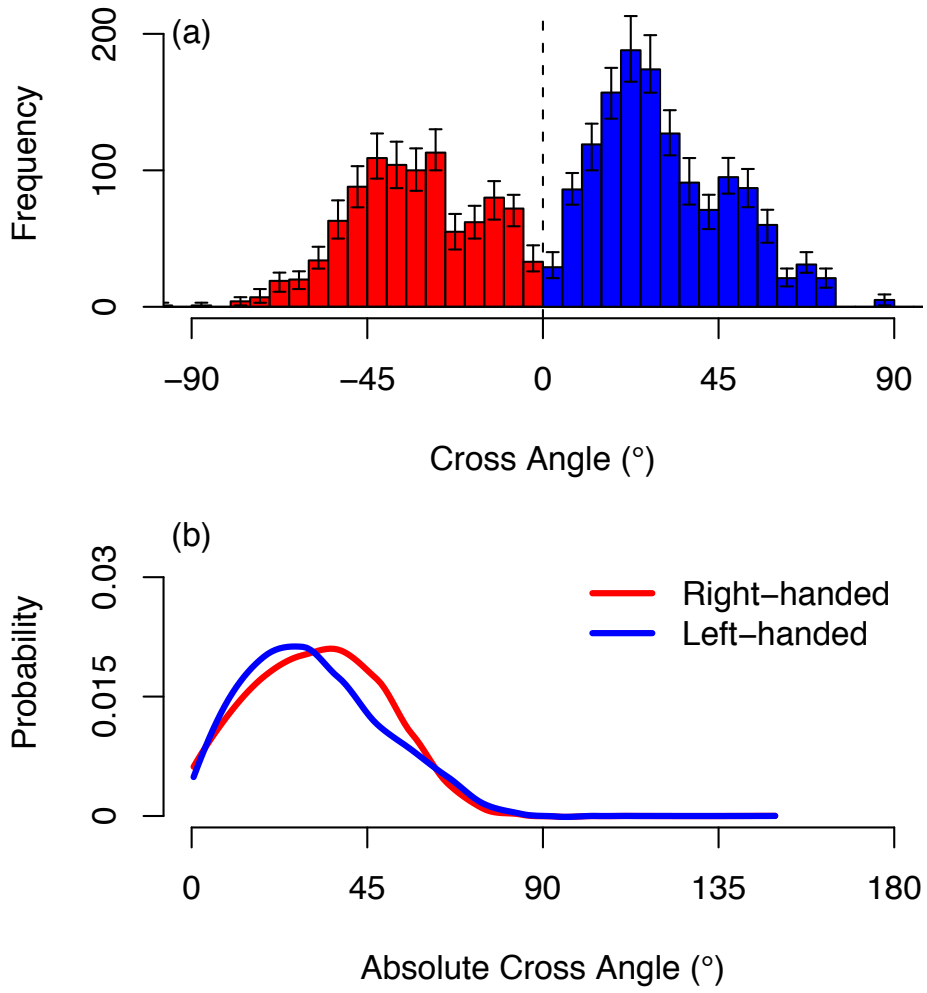


Figure 4.5: Comparison of right and left handed experimental cross angles. (a) Shows the frequency of cross angles along the entire range $\Omega \in [-90°, 90°]$ with colors corresponding to helix orientation. Error-bars represent 0.9 confidence intervals. (b) Shows the absolute cross angle distribution in which all helix vectors are treated as pointing towards $+\hat{\mathbf{z}}$ direction and the angle between those vectors is calculated by a simple dot product (see Methods Section). The mean standard deviation around the density lines is 0.006.

angles are drawn at random, and M cross angle values, Ω_j^i, are then calculated based on Equation 3.2 by plugging in θ_1^i, θ_2^i, γ_j^i. We therefore obtain $MxN = 10^6$ independent cross angle values which we bin into a histogram of narrow 0.1° bins. This histogram represents the reference discrete distribution for this specific helix pair.

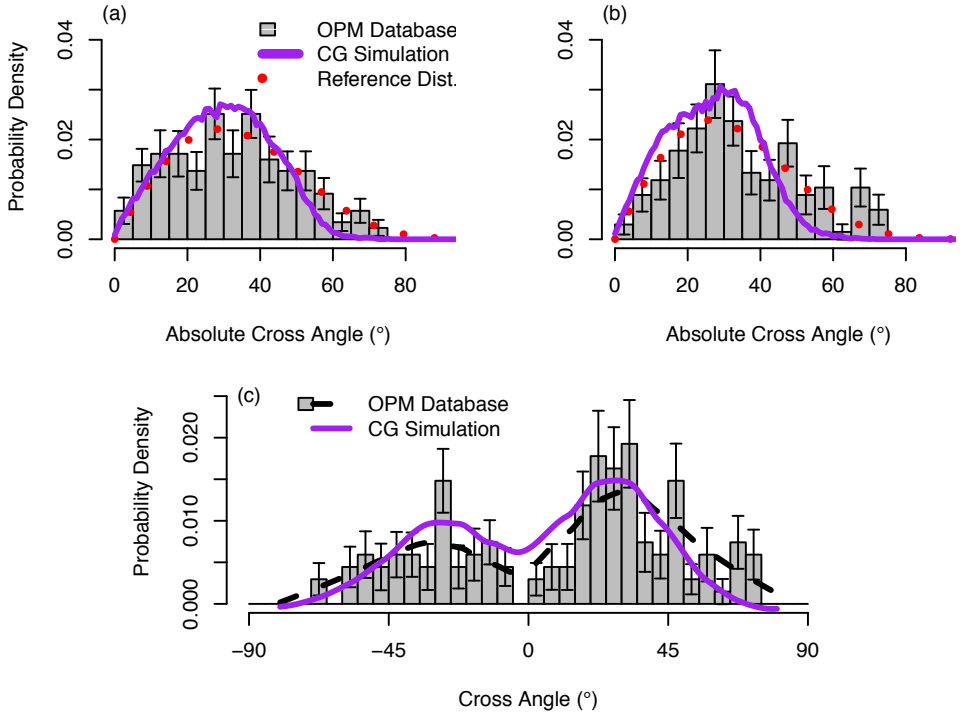


Figure 4.6: Comparison of cross angle distributions. The experimental cross angle distribution is shown in grey boxes; purple curves show the results of cross angle between simulated helices and red dotted lines represent the reference cross angle distribution. The reference distribution is based on the tilt angle distribution of the helices' mismatch as extracted from Fig. 4.2(a). (a) Shows absolute cross angle of pairs of neighboring helices with hydrophobic mismatches in the ranges $\Delta d_1 \in (2, 5]$ Å and $\Delta d_2 \in (4, 7]$ Å compared to simulated helices of mismatch $\Delta d = 3.2$ Å ,7.7 Å respectively. (b) Shows pairs in ranges $\Delta d_1 \in (0, 3]$ Å and $\Delta d_2 \in (0, 3]$ Å compared to simulated helices of mismatch $\Delta d = 3.2$ Å ,3.2 Å respectively. (c) Shows the full cross angle distribution (and not the absolute one) for the same set used in plot (b). Error-bars in all three plots represent 0.9 confidence intervals.

We note that the choice of neighboring helix groups in Fig 4.6 was made by balancing two factors: In order to display a representative cross angle distribution, a large enough number of data points (neighboring pairs) should be accounted for, implying the use of a large range of hydrophobic mismatch. The maximum density of helix pairs is 33 pairs/Δd^2 (see Figure 4.9) and so a range of several Å in hydrophobic mismatch is needed to include enough data points

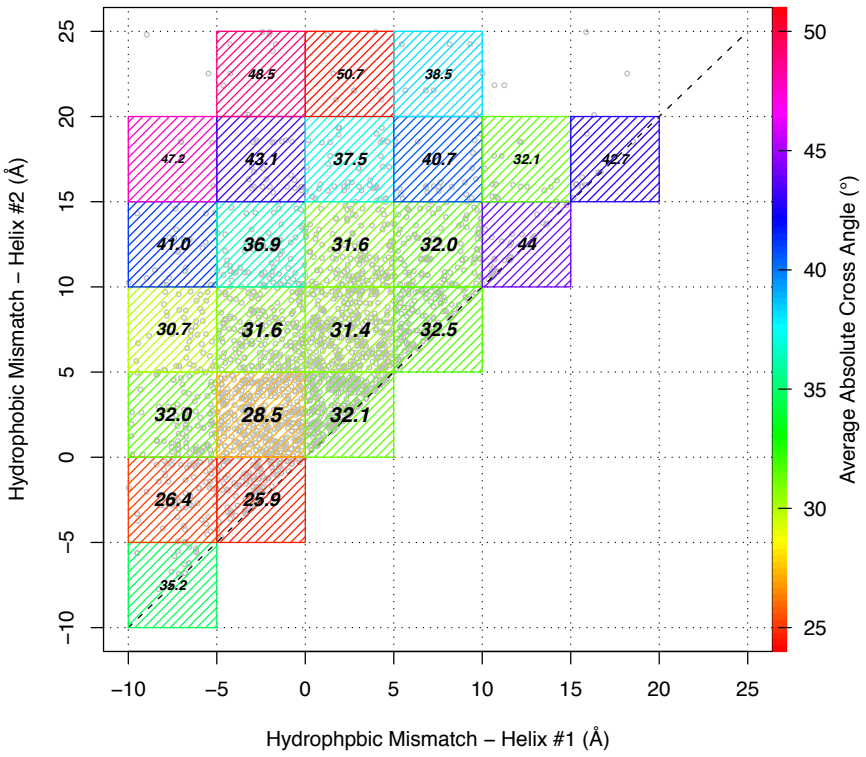


Figure 4.7: Scatter plot of neighboring helix pairs as function of helix hydrophobic mismatch. Available data points represented by grey circles. Colors and text correspond to average absolute cross angle for pairs in each mismatch range. Text size scales with the number of data points in each range, representing the accuracy in average value calculation. Larger text therefore corresponds to smaller error in average calculation. The plot shows the dependence of cross angle on the hydrophobic mismatch of both helices. We observe smaller cross angle values for helices with small hydrophobic mismatch, and larger cross angle values for helices with large hydrophobic mismatch. We additionally observe larger cross angle values for helices whose mismatch ranges differ (increase in average cross angle towards upper left corner of the plot). Both phenomena are expected from our reference model.

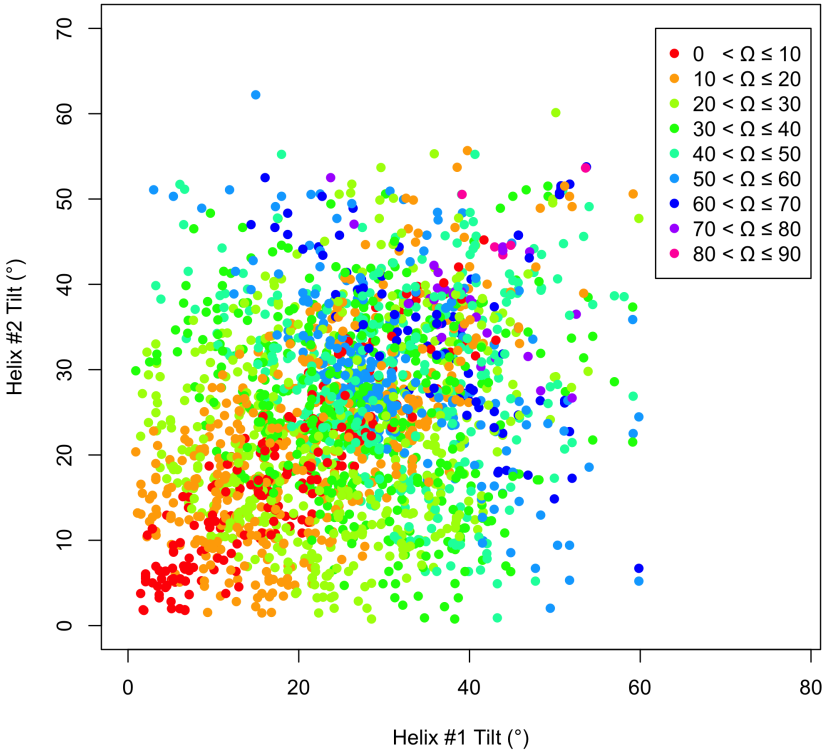


Figure 4.8: Scatter plot of tilt angles for both helices across all neighboring helix pairs. Colors represent the cross angle of the pair, according to the legend. The plot shows the dependence of cross angle on the tilt angle of both pairs, as expected by Eq. 3.2. Small cross angles ($\Omega \leq 10$, red circles) are available only to pairs with a small difference in tilt angles (along the diagonal). While moving away from the diagonal, the cross angles are limited to growingly larger values. The dependence of cross angle value on the sum of tilt angles is also apparent. Each value of cross angles is limited in the bottom left direction of the plot, suggesting a minimal limit on the sum of tilt angles.

in the distribution of pair cross angles. On the other hand, using a too-large hydrophobic mismatch range would include large variability in the tilt angle distribution of the helices (observed change of up to $1.3°/\Delta d$ in average tilt angle) and therefore a variability in the resulting cross angle distribution. We therefore chose windows of an intermediate size of 3Å in mismatch ranges that include a large enough number of data points. We focus on neighboring helix pairs where both helices have a positive mismatch for reasons described in the main text. Figure 4.9 shows the scatter-plot of neighboring helix pairs and the chosen ranges are shown for comparison.

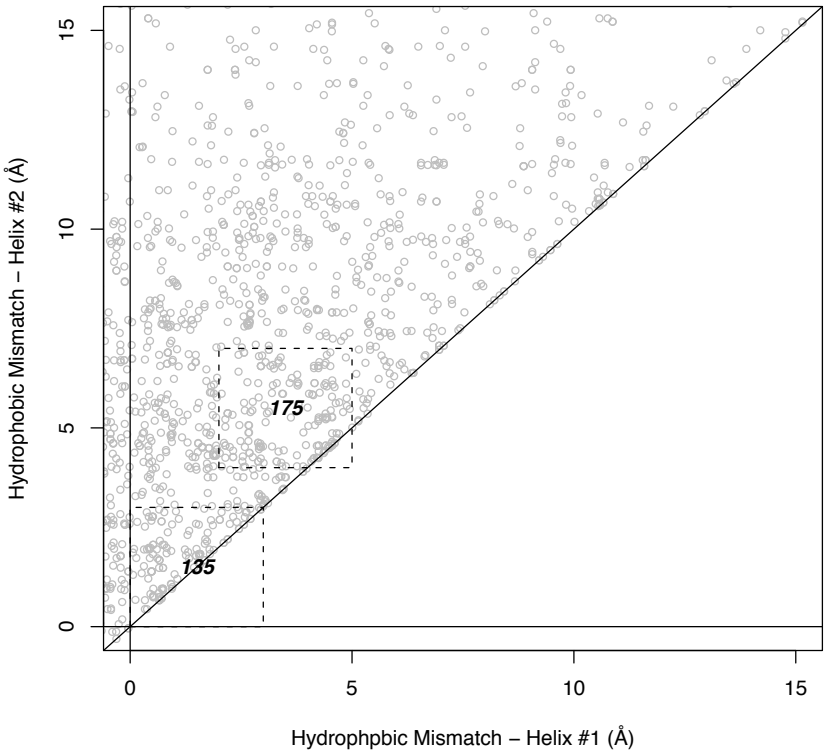


Figure 4.9: Scatter plot of neighboring helix pairs with positive hydrophobic mismatch as a function of helix hydrophobic mismatch. X-axis represents the hydrophobic mismatch of the helix with the lower mismatch value in the pair. Y-axis represents the hydrophobic mismatch of the helix with the higher mismatch value in the pair. Diagonal line represents pairs with equal mismatch values for both helices. Dashed rectangles correspond to mismatch ranges used in Figure 4.6. The number on each rectangle represents the number of pairs in that rectangle.

To quantitatively test that the OPM distribution and our reference distributions are

statistically equivalent over all hydrophobic mismatch ranges, we used our generated reference distributions for each helix pair to obtain the overall expected histogram of all helix pairs. We calculated the overall p-value of all helix pairs. This provides a measurement of how probable it is that the observed cross angles originate in the reference cross angle distribution. Comparing the overall p-value to a p-value obtained for a set of random cross angles sampled directly from the reference distribution shows that the experimental p-value (0.497) is well within the standard error of the random p-value (0.500 ± 0.006). This suggests that it is statistically probable that the experimental cross angle distribution was in fact generated by the model in Eq. 3.2.

To obtain the overall expected cross angle histogram of all helix pairs we combine the reference histograms by adding the counts of each bin for every helix pair and normalizing the resulting distribution at the end of the procedure. As the reference distribution does not differentiate between negative and positive cross angles we calculate the cross angle value in its positive form, and set its sign randomly such that the percentage of parallel pairs will be identical to its value in the data (58.5%).

We further use the reference histograms to calculate the overall p-value of the experimental results. For each helix pair ($k = 1 \dots 2328$) we calculate the probability of the pair's cross angle, $\Pr^k(\Omega_k)$, according to the pair's corresponding reference histogram. We then sum over the probabilities of all the other bins in that reference histogram with lower probability to obtain the k'th p-value, $p^k = \sum_x \Pr^k(x) \mid \Pr^k(x) \leq \Pr^k(\Omega_k)$. To verify that the discrete binning does not bias the p-value and to get an estimate on the variance of this result, we calculate the same score on $R = 500$ random numbers sampled directly from the histogram $p^{k,l}$ for $l = 1 \dots R$. We then obtain an experimental overall p-value $p = \sum_{k=1}^{2328} p^k = 0.497$ and R random p-values $p^l = \sum_{k=1}^{2328} p^{k,l}$. The average random p-value is calculated by averaging these R values. As expected, it comes out to be 0.500. The standard deviation is also obtain from these R values as 0.006.

Using the overall reference histogram, one can obtain a direct comparison for cross angle values that are over- or under- represented in the experimental results. The degree to which each cross angle probability differs from its probability in the reference distribution can be used as a measure for what specific interactions and other membrane effects are imposing upon the helices. Results for the comparison of the overall cross angle are presented for both the probability density (Fig. 4.10(a)) and the difference in log of probabilities (Fig. 4.10(b)). Since most helix pairs (97%) are from structures with reported crystallization or experimental temperature in the small range $T = 275°K - 300°K$ (see Table 4.1), the difference in log probabilities is close to the difference in free energy, up to a factor of k_BT. These results show that the differences in log(P) are rather small, and show an over-representation of experimental cross angles in range $-10 \leq \Omega \leq 30$ and an under-representation in ranges $-25 < \Omega < -15$, $37 < \Omega < 47$, $58 < \Omega < 65$ over the entire error bar span.

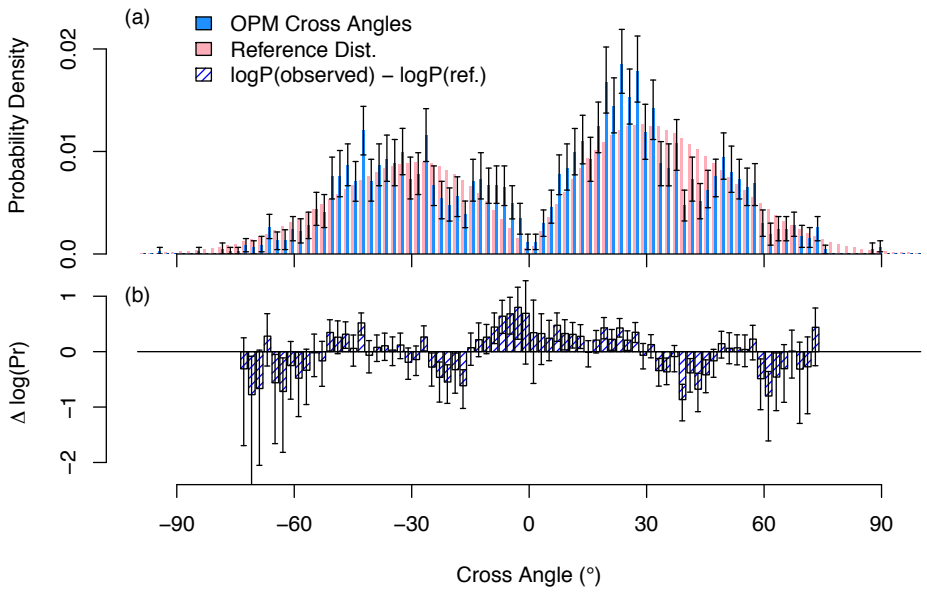


Figure 4.10: Comparison of cross angle distribution of all neighboring helix pairs with the reference distribution based on their respective mismatches. (a) Shows the distribution of all neighboring pairs (blue) compared to the overall reference distribution (pink, see text). (b) Shows the difference in log of probabilities between the two sets. It is equivalent to difference in estimated free energies. The plot shows higher tendency for small cross angles ($-10^\circ < \Omega < 30^\circ$) than expected by the reference distributions and lower than expected tendency for larger cross angles absolute values ($-25^\circ < \Omega < -15^\circ$), ($37^\circ < \Omega < 47^\circ$), ($58^\circ < \Omega < 65^\circ$). Error-bars represent 0.9 confidence intervals obtained by a 500 resample bootstrap [115] on the values of experimental cross angles. Data is omitted for $|\Omega| > 75^\circ$ due to large uncertainty.

4.3.3 Deviations from the reference distribution

Deviations from the reference distribution can arise from a non-uniform distribution of projection angles. In the OPM data, we observe a tendency towards small projection angles ($\gamma \leq 100$) as presented in Fig. 4.11. This tendency was also observed in our CG simulations and was seen to vary with hydrophobic mismatch. This trend toward smaller projection angles corresponds to a trend towards smaller cross angles. We hypothesize that these deviations are due to the membrane applying effective force to minimize the amount of constrained lipids between the two helices. Since we observe similar deviations in both experimental and simulated results, it is unlikely that this effect originates in specific residue interactions since they are not included in our simulations. Other deviations from the reference distribution can arise from correlations in tilt angle between the two helices. In our CG simulations we observe a time averaged correlation coefficient ranging from 0.03 for negative mismatch ($\Delta d = -15, -15$ Å) and up to 0.52 for extremely positive mismatch ($\Delta d = 26, 26$ Å). We note that theses two effects are not independent, but as both effects are present in our CG model that does not resolve specific residues, it is unlikely that they originate in specific residue interactions.

4.4 Discussion

Our findings show that even in the absence of any specific interactions, the cross angle distribution of TM helices is not uniform. Some angles are more preferable than others. The distribution of cross angles depends on the hydrophobic mismatch of the individual helices, through its effect on the tilt angle. Therefore, special care has to be taken when considering a set of TM helix pairs and attributing their cross angle distribution to specific interactions. The observed distribution of cross angles should first be compared with the reference distribution. Only statistically meaningful deviations from the reference distribution point towards direct influence of specific interactions on the packed configuration.

It is interesting to discuss the consequences of our observations in the context of the folding mechanisms of membrane proteins. Previous studies on mesoscopic models show that membrane mediated interactions lead to long-range attractive forces between helices. These forces depend only on the hydrophobic mismatch [14] and do not rely on specific interactions. The importance of these membrane mediated interactions is that they are very robust and, unlike specific interactions, do not rely on the helices to be in an ideal orientation such that the surface motifs can interact. The surprising result of this current study is that the same hydrophobic mismatch can guide the cross angle. Hence, the forces that constrain the cross angle of the helices are also robust. We therefore see the membrane playing a role in all aspects of the second stage of protein folding - from helix association to tertiary structure determination.

It is important to emphasize that our results do not suggest that specific interactions are not important in the folding of a membrane protein. The picture that emerges from this study is that membrane mediated interactions play an important role in keeping the

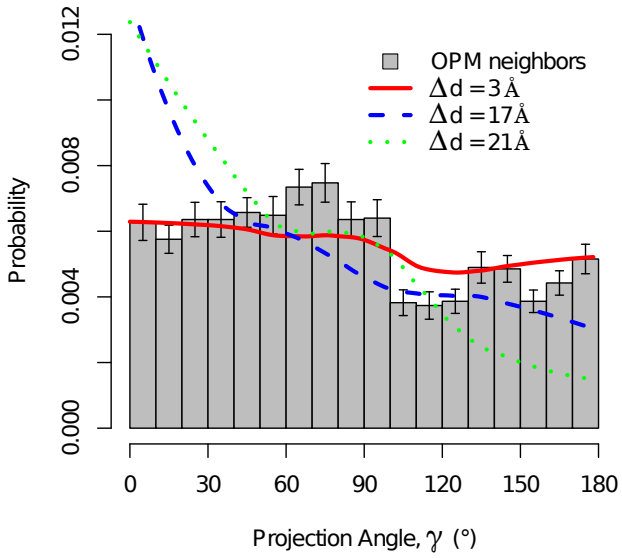


Figure 4.11: Histogram of experimental projection angles of neighboring helices (grey boxes) shows tendency towards lower ($\gamma < 100°$) angles with p-value < 0.0001 (based on a binomial test). Representative projection angle histograms from CG simulations are shown in lines. Each line corresponds to a different pair of same-mismatch helices: $\Delta d = 3$ Å (red solid), 17 Å (blue dashed), 21 Å (green dotted). These show that the trend of over-population of lower projection angles is observed in simulations as well and varies in strength and shape with mismatch. Error-bars represent 0.9 confidence intervals in experimental projection angle histogram.

helices together and ensuring the helices have a stable cross angle. Hence, we hypothesize that membrane mediated interactions put the helices in an ideal position for the short-ranged specific interactions to take over and finalize the folding.

4.5 Appendix

4.5.1 List of Experimental Structures

Table 4.1: Experimental Structures

PDB	Experimental Method	Temperature	# Unique Helices	# Neigh. Pairs
1A11	SOLUTION NMR	313	1	0
1A91	SOLUTION NMR	300	2	1
1AFO	SOLUTION NMR	313	1	1
1AR1	X-RAY DIFFRACTION	N/A	12	18
1DXR	X-RAY DIFFRACTION	N/A	9	8
1E12	X-RAY DIFFRACTION	N/A	6	13
1EHK	X-RAY DIFFRACTION	293	13	21
1EYS	X-RAY DIFFRACTION	277	11	15
1FFT	X-RAY DIFFRACTION	277	13	12
1FJK	SOLUTION NMR	300	1	0
1H2S	X-RAY DIFFRACTION	N/A	9	18
1H6I	ELECTRON CRYSTALLOGRAPHY	N/A	6	15
1HGZ	FIBER DIFFRACTION	300	2	2
1IFP	FIBER DIFFRACTION	298	1	1
1IJD	X-RAY DIFFRACTION	291	1	2
1J4N	X-RAY DIFFRACTION	277	6	15
1JB0	X-RAY DIFFRACTION	277	26	25
1KF6	X-RAY DIFFRACTION	295	6	8
1KPL	X-RAY DIFFRACTION	293	8	11
1KQF	X-RAY DIFFRACTION	277	4	6
1L9B	X-RAY DIFFRACTION	292	11	14
1LDF	X-RAY DIFFRACTION	298	6	19
1LGH	X-RAY DIFFRACTION	N/A	1	2
1M0L	X-RAY DIFFRACTION	295	6	11
1M56	X-RAY DIFFRACTION	277	21	35
1MHS	ELECTRON CRYSTALLOGRAPHY	277	7	6
1N7L	SOLUTION NMR	323	1	0
1NEK	X-RAY DIFFRACTION	293	6	7
1NKZ	X-RAY DIFFRACTION	289	1	3
1OKC	X-RAY DIFFRACTION	N/A	6	7
1OQW	X-RAY DIFFRACTION	293	1	0
1OTS	X-RAY DIFFRACTION	298	8	10
1P49	X-RAY DIFFRACTION	298	2	1
1PP9	X-RAY DIFFRACTION	277	11	18
1PW4	X-RAY DIFFRACTION	293	11	18
1PY6	X-RAY DIFFRACTION	310	6	7
1Q90	X-RAY DIFFRACTION	277	13	25

PDB	Experimental Method	Temperature	# Unique Helices	# Neigh. Pairs
1QL1	FIBER DIFFRACTION	283	2	0
1R3J	X-RAY DIFFRACTION	293	2	3
1RC2	X-RAY DIFFRACTION	300	6	15
1RKL	SOLUTION NMR	298	1	0
1RWT	X-RAY DIFFRACTION	291	3	1
1S5H	X-RAY DIFFRACTION	293	2	3
1SU4	X-RAY DIFFRACTION	N/A	7	8
1T5S	X-RAY DIFFRACTION	292	8	7
1U7G	X-RAY DIFFRACTION	298	9	28
1UAZ	X-RAY DIFFRACTION	280	6	7
1V55	X-RAY DIFFRACTION	N/A	22	47
1VGO	X-RAY DIFFRACTION	283	6	7
1WPG	X-RAY DIFFRACTION	N/A	9	12
1WU0	SOLUTION NMR	298	2	1
1XIO	X-RAY DIFFRACTION	293	7	12
1XL6	X-RAY DIFFRACTION	293	2	5
1XRD	SOLUTION NMR	298	1	0
1YCE	X-RAY DIFFRACTION	290	2	7
1YEW	X-RAY DIFFRACTION	298	7	5
1YMG	X-RAY DIFFRACTION	298	6	17
1YQ3	X-RAY DIFFRACTION	278	6	8
1ZCD	X-RAY DIFFRACTION	279	7	9
1ZLL	SOLUTION NMR	303	1	2
1ZOY	X-RAY DIFFRACTION	290	5	6
1ZRT	X-RAY DIFFRACTION	277	10	13
2A0L	X-RAY DIFFRACTION	293	2	3
2A65	X-RAY DIFFRACTION	291	9	18
2A9H	SOLUTION NMR	315	2	3
2AGV	X-RAY DIFFRACTION	283	5	4
2B2F	X-RAY DIFFRACTION	298	9	28
2B5F	X-RAY DIFFRACTION	N/A	5	9
2B6O	ELECTRON CRYSTALLOGRAPHY	300	6	17
2BBJ	X-RAY DIFFRACTION	297	1	0
2BG9	ELECTRON MICROSCOPY	277.20	16	28
2BHW	X-RAY DIFFRACTION	N/A	1	0
2BL2	X-RAY DIFFRACTION	N/A	4	11
2BS2	X-RAY DIFFRACTION	N/A	5	9
2BS3	X-RAY DIFFRACTION	N/A	4	6
2E74	X-RAY DIFFRACTION	N/A	10	20
2F2B	X-RAY DIFFRACTION	295	6	19
2FYN	X-RAY DIFFRACTION	288.20	10	15
2GFP	X-RAY DIFFRACTION	N/A	10	15
2H8A	ELECTRON CRYSTALLOGRAPHY	277	3	4
2HIL	ELECTRON MICROSCOPY	368	1	0
2HYD	X-RAY DIFFRACTION	N/A	4	4
2IFO	FIBER DIFFRACTION	298	2	0
2IQL	THEORETICAL MODEL	298	6	9
2IQO	THEORETICAL MODEL	298	6	9
2IQR	THEORETICAL MODEL	298	5	6

PDB	Experimental Method	Temperature	# Unique Helices	# Neigh. Pairs
2IQV	THEORETICAL MODEL	298	6	9
2JLN	X-RAY DIFFRACTION	N/A	7	6
2JWA	SOLUTION NMR	313	2	1
2K1K	SOLUTION NMR	313	1	1
2K74	SOLUTION NMR	313	2	1
2K9P	SOLUTION NMR	320	2	1
2K9Y	SOLUTION NMR	313	1	1
2KA2	SOLUTION NMR	313	1	1
2KB7	SOLID-STATE & SOLUTION NMR	N/A	1	0
2KDC	SOLUTION NMR	318	1	0
2KIX	SOLUTION NMR	305	1	2
2KNC	SOLUTION NMR	298	2	1
2KQT	SOLID-STATE NMR	243	1	2
2KSD	SOLUTION NMR	318	2	1
2KSE	SOLUTION NMR	313	2	1
2KSF	SOLUTION NMR	318	3	2
2KSR	SOLUTION NMR	313	4	3
2KV5	SOLUTION NMR	303	1	0
2KWX	SOLUTION NMR	303	1	2
2KYH	SOLUTION NMR	318	3	2
2L35	SOLUTION NMR	303	3	3
2L9U	SOLUTION NMR	313	1	1
2LAT	SOLUTION NMR	298	1	0
2LCK	SOLUTION NMR	306	4	2
2NQ2	X-RAY DIFFRACTION	277	8	12
2NR9	X-RAY DIFFRACTION	298	4	4
2NRF	X-RAY DIFFRACTION	295	4	4
2NS1	X-RAY DIFFRACTION	293	9	27
2NWL	X-RAY DIFFRACTION	277	5	3
2O01	X-RAY DIFFRACTION	278	6	0
2OAR	X-RAY DIFFRACTION	N/A	2	5
2OAU	X-RAY DIFFRACTION	277	1	0
2Q7R	X-RAY DIFFRACTION	298	2	2
2QI9	X-RAY DIFFRACTION	293	14	18
2QKS	X-RAY DIFFRACTION	293	2	5
2QTS	X-RAY DIFFRACTION	277	2	4
2RDD	X-RAY DIFFRACTION	293	10	13
2RH1	X-RAY DIFFRACTION	293	7	10
2RLF	SOLUTION NMR	303.10	1	2
2UUH	X-RAY DIFFRACTION	N/A	3	8
2V50	X-RAY DIFFRACTION	N/A	22	41
2V8N	X-RAY DIFFRACTION	N/A	5	2
2VL0	X-RAY DIFFRACTION	N/A	2	0
2VPZ	X-RAY DIFFRACTION	N/A	14	25
2VT4	X-RAY DIFFRACTION	N/A	7	11
2W2E	X-RAY DIFFRACTION	N/A	5	18
2W5J	X-RAY DIFFRACTION	N/A	1	2
2WCD	X-RAY DIFFRACTION	N/A	1	2
2WIT	X-RAY DIFFRACTION	N/A	12	26

PDB	Experimental Method	Temperature	# Unique Helices	# Neigh. Pairs
2WLL	X-RAY DIFFRACTION	N/A	2	5
2WSW	X-RAY DIFFRACTION	N/A	7	8
2WSX	X-RAY DIFFRACTION	N/A	10	19
2WWB	ELECTRON MICROSCOPY	357	7	4
2XKM	SOLID-STATE NMR; X-RAY DIFF.	N/A	1	0
2XOK	X-RAY DIFFRACTION	N/A	2	7
2XQ2	X-RAY DIFFRACTION	N/A	11	13
2XQU	X-RAY DIFFRACTION	N/A	2	7
2XUT	X-RAY DIFFRACTION	N/A	8	9
2XZB	ELECTRON CRYSTALLOGRAPHY	277.50	7	4
2YDV	X-RAY DIFFRACTION	N/A	7	12
2YL4	X-RAY DIFFRACTION	N/A	5	7
2YVX	X-RAY DIFFRACTION	293	3	3
2YXR	X-RAY DIFFRACTION	277	8	7
2ZBD	X-RAY DIFFRACTION	283	8	12
2ZJS	X-RAY DIFFRACTION	293	6	5
2ZT9	X-RAY DIFFRACTION	N/A	11	22
2ZW3	X-RAY DIFFRACTION	277	4	9
2ZXE	X-RAY DIFFRACTION	298	6	3
2ZZ9	ELECTRON CRYSTALLOGRAPHY	293	6	15
3A0B	X-RAY DIFFRACTION	293	24	21
3A7K	X-RAY DIFFRACTION	293	6	11
3AM6	X-RAY DIFFRACTION	293	6	7
3AQP	X-RAY DIFFRACTION	293	12	23
3AR8	X-RAY DIFFRACTION	283	5	4
3AR9	X-RAY DIFFRACTION	283	7	8
3ARC	X-RAY DIFFRACTION	285	27	32
3B4R	X-RAY DIFFRACTION	295	4	2
3B60	X-RAY DIFFRACTION	277	5	7
3B8C	X-RAY DIFFRACTION	N/A	2	0
3B8E	X-RAY DIFFRACTION	292	5	1
3B9B	X-RAY DIFFRACTION	292	8	7
3B9Y	X-RAY DIFFRACTION	298	9	22
3BEH	X-RAY DIFFRACTION	298	5	9
3C02	X-RAY DIFFRACTION	291	5	14
3CAP	X-RAY DIFFRACTION	277	7	13
3CHX	X-RAY DIFFRACTION	293	7	3
3CX5	X-RAY DIFFRACTION	277	10	14
3D31	X-RAY DIFFRACTION	293.15	10	16
3D9S	X-RAY DIFFRACTION	281	5	11
3DDL	X-RAY DIFFRACTION	295	7	10
3DIN	X-RAY DIFFRACTION	298	10	12
3DWW	ELECTRON CRYSTALLOGRAPHY	373	3	4
3E86	X-RAY DIFFRACTION	293	2	3
3EAM	X-RAY DIFFRACTION	298	3	6
3EML	X-RAY DIFFRACTION	293	6	8
3F5W	X-RAY DIFFRACTION	298	2	1
3F7V	X-RAY DIFFRACTION	298	2	3
3F7Y	X-RAY DIFFRACTION	298	2	1

PDB	Experimental Method	Temperature	# Unique Helices	# Neigh. Pairs
3FB5	X-RAY DIFFRACTION	298	2	3
3G5U	X-RAY DIFFRACTION	278	12	18
3GD8	X-RAY DIFFRACTION	298	6	17
3GIA	X-RAY DIFFRACTION	293	8	11
3H1J	X-RAY DIFFRACTION	277	9	9
3H9V	X-RAY DIFFRACTION	277	1	2
3HD6	X-RAY DIFFRACTION	277	6	12
3HD7	X-RAY DIFFRACTION	293	2	1
3HFX	X-RAY DIFFRACTION	289	7	8
3HQK	X-RAY DIFFRACTION	293	9	19
3HZQ	X-RAY DIFFRACTION	277	1	2
3J01	ELECTRON MICROSCOPY	N/A	6	2
3JYC	X-RAY DIFFRACTION	277.10	2	5
3K03	X-RAY DIFFRACTION	293	2	3
3K3F	X-RAY DIFFRACTION	293	6	11
3KBC	X-RAY DIFFRACTION	277	4	1
3KCU	X-RAY DIFFRACTION	291	6	23
3KG2	X-RAY DIFFRACTION	277	3	5
3KLY	X-RAY DIFFRACTION	291	13	48
3KP9	X-RAY DIFFRACTION	298	5	6
3KZI	X-RAY DIFFRACTION	291	28	35
3L1L	X-RAY DIFFRACTION	291.15	15	18
3LBW	X-RAY DIFFRACTION	N/A	1	2
3LDC	X-RAY DIFFRACTION	293	2	1
3LLQ	X-RAY DIFFRACTION	295	6	16
3LUT	X-RAY DIFFRACTION	N/A	5	6
3M73	X-RAY DIFFRACTION	298	9	20
3MK7	X-RAY DIFFRACTION	294	10	13
3MKT	X-RAY DIFFRACTION	295	10	11
3MP7	X-RAY DIFFRACTION	273	9	11
3MRA	SOLUTION NMR	303	1	0
3N23	X-RAY DIFFRACTION	292	6	3
3ND0	X-RAY DIFFRACTION	293	7	9
3NE2	X-RAY DIFFRACTION	N/A	6	23
3NE5	X-RAY DIFFRACTION	298	6	6
3O0R	X-RAY DIFFRACTION	277	12	23
3O7Q	X-RAY DIFFRACTION	291	12	24
3OAX	X-RAY DIFFRACTION	283	7	11
3OB6	X-RAY DIFFRACTION	298	8	12
3ODU	X-RAY DIFFRACTION	293	5	6
3ORG	X-RAY DIFFRACTION	298	6	7
3OUF	X-RAY DIFFRACTION	293	2	3
3P03	X-RAY DIFFRACTION	291	19	32
3P0G	X-RAY DIFFRACTION	293	7	12
3P5N	X-RAY DIFFRACTION	298	4	4
3PBL	X-RAY DIFFRACTION	293	6	9
3PCQ	X-RAY DIFFRACTION	277	24	23
3PJZ	X-RAY DIFFRACTION	298	5	2
3PL9	X-RAY DIFFRACTION	291	3	1

PDB	Experimental Method	Temperature	# Unique Helices	# Neigh. Pairs
3PQR	X-RAY DIFFRACTION	277	6	8
3PUY	X-RAY DIFFRACTION	293	11	18
3PUZ	X-RAY DIFFRACTION	293	10	16
3PXO	X-RAY DIFFRACTION	277	7	13
3Q7K	X-RAY DIFFRACTION	293	6	23
3QE7	X-RAY DIFFRACTION	291	9	11
3QNQ	X-RAY DIFFRACTION	293	7	14
3RCE	X-RAY DIFFRACTION	298	6	4
3RFU	X-RAY DIFFRACTION	293	5	3
3RHW	X-RAY DIFFRACTION	277	3	4
3RKO	X-RAY DIFFRACTION	296	41	67
3RLB	X-RAY DIFFRACTION	279	4	3
3RVY	X-RAY DIFFRACTION	298	8	16
3RZE	X-RAY DIFFRACTION	293	7	11
3S0X	X-RAY DIFFRACTION	298	4	2

Table 4.1: PDB entries of all experimental structures used in the OPM database analysis. Experimental Method and Temperature was obtained for each PDB entry from the RCSB protein data bank [80]. In X-ray diffraction experiments the temperature refers to the crystallization temperature. The two right-most columns provide the number of unique helices and the number of neighboring helix pairs extracted for each protein structure by our method.

Chapter 5

Acceleration of Molecular Simulations using Graphical Processing Units

Recent developments in computer science and architecture has made it possible for massively parallel, multithread algorithms to be executed. Graphical Processing Units (GPUs) represent the state-of-the-art of parallel computing, allowing significant speedup in general purpose computer programs in various fields. Dissipative Particle Dynamics (DPD), is an established simulation technique in the field of computational chemistry. It allows simulation of coarse-grained models while still remaining true to the dynamical nature of the system. In this chapter we present a novel algorithm for highly parallel DPD simulations on a GPU. Our algorithm shows substantial speedup over the canonical CPU approach. Our GPU algorithm gives an additions speedup of a factor 30, which comes on top of the factor gained by using a coarse-grained model. This combined approach allows us to access biologically relevant time and length scale.

5.1 Introduction

The field of Computational Chemistry has always been in search for the fastest implementations to complex calculations. Acceleration in computation time allows for simulation of larger system sizes for longer times. This may have a crucial effect when simulating biologically relevant processes, which can typically take place in the μs - sec time scale. Recent technological advancements have made Graphical Processing Units (GPUs) affordable and easily programmable for general purpose computations [116, 117]. Allowing for massively parallel computation, it has become a popular tool in scientific computing [118–120].

The conversion from serial CPU-based algorithms to parallel GPU-based algorithms is, however, not straightforward. To accelerate an algorithm, special attention needs to be given to minimizing communication and to parallelizing serial instructions. Failing to do so will result in very small speed up, if any.

Our goal is to accelerate the implementation for Dissipative Particle Dynamics (DPD)

by making it GPU compatible. GPU based implementations of molecular dynamics (MD) have been developed in recent years [121–123] and have shown speed-up of several orders of magnitude over conventional serial algorithms. Nevertheless, several characteristic make the implementation of DPD on a GPU a non-trivial extension to these GPU-based MD implementations. First, DPD requires the momentum to be conserved (Eq. 2.4). Second, the non-bonded force acting between every two particles contains a random component (Eq. 2.3). These two requirements combined require a different strategy to parallelize the force calculation, since, unlike in MD simulation, in DPD for each particle the forces with all its neighbors cannot be calculated independently. In DPD we need to ensure that for each particle i the force calculated for the interaction with particle j, including its random component, $\vec{F}_{ij}^{R}(r_{ij})$, is identical and opposite to the force calculated simultaneously for particle j.

Philips *et al.* [124] have bypassed some of these difficulties and extended HOOMD-blue, a commonly used GPU implementation of MD [121], to be suitable for DPD by implementing a thread based random number generator. Wu *et al.* [125] and Rożen *et al.* [126] disregarded the conservation of momentum all together, overlooking its importance to the system's hydrodynamic behavior. We take a different approach, in which we calculate the interactions of each particle only with particles in half the space. In this approach, we perform much fewer calculations, but need to perform more communications between threads, compared to regular GPU-based MD. In the following section we describe the details of our implementation.

5.2 CUDA overview

GPUs were originally devised for use in graphic cards, as the gaming industry required extremely fast and realistic computer graphics. In order to allow for wider and easier development, NVIDIA (`http://www.nvidia.com`) created a platform for parallel computing programming, called CUDA. CUDA, written similar to C/C++, allows a developer to access the memory and operate instructions directly on the GPU device. Below we briefly review those aspects of CUDA that are important for our implementation. For a more detailed discussion on CUDA we refer to [127].

Within the CUDA framework, algorithms distribute calculation over multiple *threads*. Each thread executes the same set of instructions, but operates on different data. Threads are bundled into different *blocks*, defined by the instruction *kernel*. A block may contain any number of threads up to 1024. Given a block index and a thread index, each thread operates the exact same set of instructions, on the data determined by the indices. The number of blocks that may run in parallel depends on the resource limit of the hardware. Each block is executed on a Streaming Multiprocessor (SM). Only 32 SMs can run in parallel on a GeForce GTX 590 card. Each SM can run up to 8 blocks / 1536 threads at a time, and up to 48 KB of block-wide memory (*shared memory*) can be accommodated [128].

The basic execution unit in CUDA is called a *warp*. A warp contains 32 threads that execute the exact same instruction at the same time. Branches in the code, such as if-

statements or loops, may cause different threads to execute different instructions. For efficient parallelism, it is important to ensure that entire warps follow the same branch. This may be done by branch predication [127], in which all threads follow the branching, but only some have the instruction meaningfully performed.

This execution hierarchy is complimented by memory hierarchy. Each thread can have a limited set of registers that can save data locally (*local memory*), for the use of that specific thread only. In GeForce GTX 590, this memory is limited to 63 registers. Threads that are within the same block can share memory with low latency at a rather fast pace. This type of memory is referred to as *shared memory* and it is limited to small memory sizes (48 KB per SM) [128]. A larger, long latency memory is available to all threads via *global memory*. During the time it takes to access global memory, hundreds of instructions may be performed on a multiprocessor. It is therefore advised to minimize the communication with global memory, and make sure other warps can access their data while a given warp is waiting on its data to come from global memory. It is important to note, however, that only global memory keeps the data after a kernel has finished its instructions. Local and shared memory have the life time of the kernel they are executing - any unsaved information at the end of a kernel will be lost. Therefore, these memory types are only used for data manipulation, and do not replace read / write to global memory at the beginning and end of each kernel.

Memory collisions, where two or more threads are accessing the same address in memory, is one of the obstacles to be avoided when developing parallel code. CUDA allows for *atomic operations*, to help prevent memory collisions [128]. Atomic operations perform writing to memory addresses by principles of mutual exclusion. Each memory writing is preceded by a locking of the address and proceeded by an unlocking. This is far slower than a non-atomic write to memory. Avoiding memory collisions in the algorithmic level is therefore far more desirable.

5.3 Implementation Details

The limiting step in DPD, as well as other MD simulations, is typically the non-bonded force calculation. The complexity of this step, if done naively, scales as $\mathrm{O}(N^2)$, N being the number of particles in the system. Several algorithms for reducing this time complexity exist, including Cell List [129] and Neighbor List [130], which reduce the complexity to $\mathrm{O}(N)$. Even so, it still remains the bottleneck in any given simulation. Improving the efficiency of the DPD algorithm is therefore first and foremost focused on accelerating the non-bonded force calculation, as well as its supporting cell or neighbor list.

5.3.1 Cell List Structure

In our simulation, we define a fixed size *Cell List Structure* that holds all the information on the current position, velocity and force of each particle. Each particle is assigned into a specific cell, based on its location. All non-bonded force interactions are calculated based on

this structure. A few factors have been considered when constructing this structure: Cell list based algorithms rely on dividing the space onto a grid. The complexity of calculating forces with neighboring particles is then reduced from enumerating all pairs of particles, to searching only particles that reside in neighboring cells. Choosing the grid size is typically done based on the underlying force that has to be calculated. In DPD, the non-bonded force is only non-zero up to a certain cutoff, R_c. Therefore, choosing the grid spacing (or cell-size) to be $r_{cell} \geq R_c$ ensures that all possible interacting particles with a particle in a given cell will be found in the nearest 27 neighboring cells, and nowhere else.

To minimize branching of threads we impose that the total number of particles in the list structure is the same for all cells (N_{pc}). Empty spots in the cell lest are therefore occupied with *dummy* particles. These *dummy* particles have no interactions with the real particles and therefore their presence does influence the behavior of the real system. Force calculations between adjacent cells are therefore always of fixed complexity. In determining the desired N_{pc}, we take into account the particle density. In DPD, the mean density of particles is typically $\rho = 3$ particles per R_c^3 [33]. In a cell of size $r_{cell} = R_c$ we would therefore expect on average 3 particles. We set the number of particles in each cell to be larger than that, such that the probability of finding a cell with more than N_{pc} particles would be small. We chose $N_{pc} = 8$, such that $P(N > N_{pc}) \simeq 0.012$ for cells of size $r_{cell} = R_c$ in all directions.

We note here that special care needs to be taken when dealing with particles that are left out of the basic cell list structure. In practice, we leave room for many more particles in each cell ($N'_{pc} = 16$) such that the probability of having no spot available is decreased to $P(N > N'_{pc}) \simeq 1.25 \cdot 10^{-7}$ for cells of size $r_{cell} = R_c$ in all directions. Particles that exceed the original N_{pc} occupancy are treated differently, however, and do not contribute to the fixed complexity of force calculation. We further discuss this issue in Section 5.3.6.

As mentioned in Section 5.2, the basic unit of instruction in CUDA is a *warp*, containing 32 threads. Threads within each warp should follow the same set of instructions and not branch . It is therefore preferred that each unit in the cell list structure corresponds to at least one warp, where each thread is in charge of one particle. To reach the minimal unit size of 32 threads per warp, we bundle 8 adjacent cells into a *super-cell*, redefining it to be the basic cell list structure unit. Each super-cell may contain up to 64 particles, 8 particles in each cell. The super-cell size in each direction ($\alpha = \hat{x}, \hat{y}, \hat{z}$) corresponds to $r^{\alpha}_{\mathrm{SC}} = 2 \cdot r^{\alpha}_{cell}$.

Within one super-cell, each thread will be assigned a specific particle out of the 64 possible particles, based on the thread index. Given the occupancy of the super-cell, this particle may either be a real particle in the system or a *dummy* particle.

One additional factor that plays a role when constructing the cell list structure is minimizing memory collisions. The details of this scheme will be discussed in the next sections. To minimize memory collisions we use a generalized 3D checkerboard decomposition [131] in which we run in parallel only super-cells that are not spatially adjacent. This requires the number of super-cells along each axis to be even, and the number of cell along each axis to be devisable by 4.

The size of each super-cell in each dierction ($\alpha = \hat{x}, \hat{y}, \hat{z}$) is therefore given by:

$$
\begin{aligned}
N^{\alpha}_{rep} &= \left\lfloor \frac{B^{\alpha}}{4 \cdot R_c \cdot \varepsilon} \right\rfloor \\
N^{\alpha}_{\mathrm{SC}} &= 2 \cdot N^{\alpha}_{rep} = 2 \cdot \left\lfloor \frac{B^{\alpha}}{4 \cdot R_c \cdot \varepsilon} \right\rfloor \\
r^{\alpha}_{\mathrm{SC}} &= \frac{B^{\alpha}}{N^{\alpha}_{\mathrm{SC}}} \geq 2 \cdot R_c \cdot \varepsilon \qquad (5.1) \\
N^{\alpha}_{cell} &= 2 \cdot N^{\alpha}_{\mathrm{SC}} = 4 \cdot \left\lfloor \frac{B^{\alpha}}{4 \cdot R_c \cdot \varepsilon} \right\rfloor \\
r^{\alpha}_{cell} &= \frac{B^{\alpha}}{N^{\alpha}_{cell}} \geq R_c \cdot \varepsilon \qquad (5.2)
\end{aligned}
$$

where N_{rep} is the number of super-cell repeats in each direction (corresponding to the number of black tiles in the checkerboard representation); N_{SC} and N_{cell} are the number of super-cells and number of cells in each direction, respectively. $\vec{B} = (B^x, B^y, B^z)$ is the 3D box size and ε is a small *cell boundary factor*, always set to be $\varepsilon \geq 1$. We discuss the functionality of the cell boundary factor in Section 5.3.9.

Based on their location, the checkerboard decomposition divides the system into 8 distinguished super-cell *types*. Each pair of super-cells that share the same super-cell type are necessarily non-adjacent. Determining the super-cell type ($T = 0, \cdots, 7$) is done based on its 3D location (c^x, c^y, c^z) in the following way:

$$
\vec{C} = \left(\left\lfloor c^x \cdot \frac{r_{\mathrm{SC}}}{B^x} \right\rfloor, \left\lfloor c^y \cdot \frac{r_{\mathrm{SC}}}{B^y} \right\rfloor, \left\lfloor c^z \cdot \frac{r_{\mathrm{SC}}}{B^z} \right\rfloor \right) \qquad (5.3)
$$

$$
\begin{aligned}
T &= 4 \cdot (C^x \,\&\, 1) \\
&\quad + 2 \cdot (C^y \,\&\, 1) \\
&\quad + 1 \cdot (C^z \,\&\, 1)
\end{aligned} \qquad (5.4)
$$

Here, $\vec{C}$ is a vector of integers representing the super-cell location in 3D and T is the resulting super-cell type. It is based on the super-cell's even / odd location along each axis. Therefore, super-cells $\vec{C}_1 = (1, 2, 3)$ and $\vec{C}_2 = (5, 4, 1)$ will both share the same type ($T = 5$) since they have the same evenness / oddness in all three axes. For visual representation, see Figure 5.1.

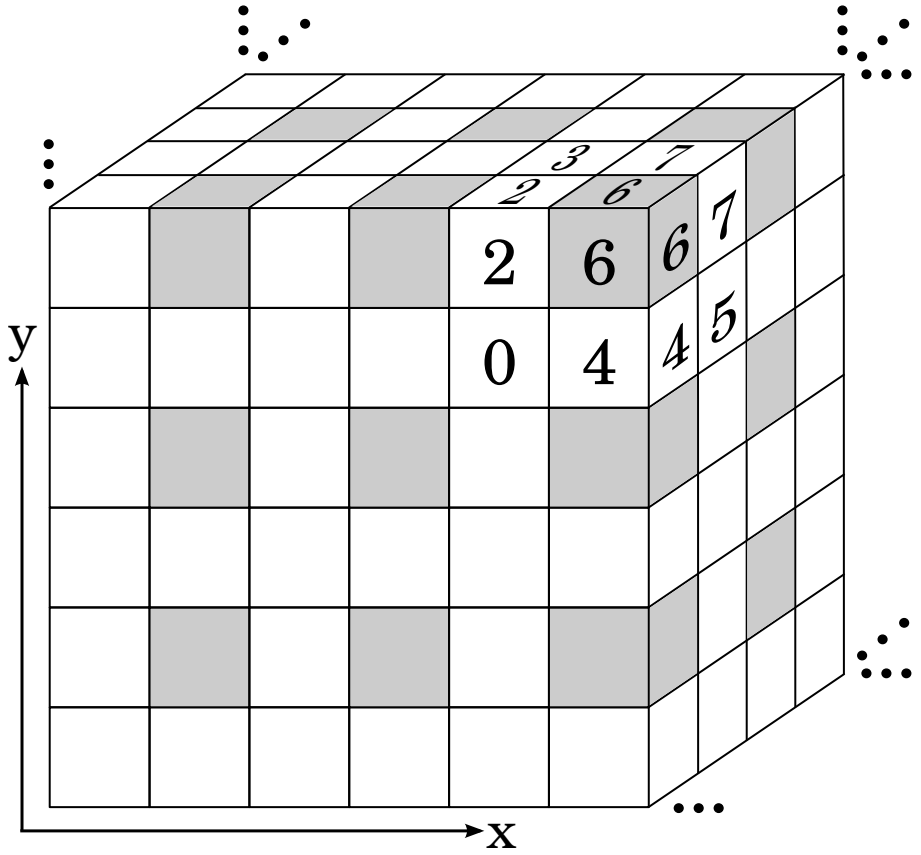


Figure 5.1: The super-cell list structure in our GPU based implementation of DPD. The system is divided to super-cells, each containing 8 cells. Each super-cell is assigned a different type ($T = 0, \cdots, 7$) based on its spatial position, as indicated on the top right corner. Grey coloring shows the repeated pattern in type definition. In non-bonded force calculation we employ the checkerboard decomposition, such that only super-cells of the same type are being calculated at any given time. In this figure, for example, non-bonded forces for particles in super-cells of type $T = 6$ (colored grey) are being calculated.

5.3.2 Algorithm overview

Before we discuss the details of our algorithm, we provide here an overview of our algorithm (Algorithm 1). The algorithm is naturally divided into DPD moves, in which the system evolves dynamically according to applied forces (Algorithm 2) and MC moves, in which the system size samples the desired ensemble (Algorithm 3).

The GPU algorithm is devised of several kernels (GPU function). Each kernel is compiled of a different number of blocks and threads. The number of blocks and threads typically match either the total number of particles (N), the total number of molecules (N_{mol}), or the total number of super-cells (N_{SC}), based on its designation. A complete list of the kernels discussed in the previous section is provided in Table 5.4.

```
input : Initial configuration, r⃗^0_{i=0...N−1}
        Initial box size, B
        number of simulation steps, Nstep
        Ensemble parameters: Temp, Gamma, Pressure.

// Initialize super-cell list, velocities and forces
SCL ← bucketSort(B, r⃗^0);
v⃗^0 ← initVelocities(Temp);
f⃗^0 ← initForces(0.0);

// Enumerate simulation steps
for step ← 0 to Nstep do
    Move ← pickMove();                          // Pick a move - DPD or MC
    if Move is DPD then                                          // DPD move
        Nsub ∈ U(0, 50];                  // Choose number of DPD sub-steps
        for substep ← 0 to Nsub do
            DPD(s);
            s ← s +1;
        end
    else                                                          // MC move
        MC(s,Move,Gamma,Pressure);
    end
end

// Finalize simulation
```

Algorithm 1: Hybrid DPD-MC algorithm with GPU implementation

```
Subprocess DPD(s) // Perform a single DPD substep
    r^{s+1}, v^{s+1} ← moveParticles(r^s, v^s, f^s);        // First integration step
    SCL ← updateCells(SCL, r^{s+1}, OutOfCell);             // Update super-cell list
    for T ← 0 to 8 do                                       // Calculate Non-Bonded Forces
        nonBondedForce(SCL, T);
        nonBondedForceTerminal(SCL, T);
    end
    if any particles out-of-cell then nonBondedForceOOC(OutOfCell);
    f^{s+1} ← extractForces(SCL);                           // Extract forces from SCL
    f^{s+1} ← f^{s+1} + bondedForce(Nmol);                  // Add bonded forces
    v^{s+1} ← updateVelocities(v^{s+1}, f^{s+1});           // Second integration step
end
```

Algorithm 2: GPU implementation of DPD substep, as part of Algorithm 1

```
Subprocess MC(s,Move,Gamma,Pressure) // Perform an MC move
    E ← nonBondedEnergy(SCL);                               // Calculate non-bonded energy
    B' ← proposeMCMove(Move,B);
    // Check cell size.  See Section 5.3.9 for details
    Rcell ← setBoxSize(B');
    if Rcell too small then stop and restart simulation;
    // Scale all coordinates.  Bonded energy remains constant
    r'^s ← scaleNonMolecular(r^s, B');
    r'^s ← scaleMolecular(r^s, B', Nmol);
    SCL ← updateCells(SCL, r'^s, OutOfCell);                // Update super-cell list
    E' ← nonBondedEnergy(SCL);                              // Calculate non-bonded energy
    accepted ← testMCMove(E-E',Move,Gamma,Pressure);
    if accepted then
        r^s ← r'^s;                                         // Save positions
        E ← E' ;                                            // Save energy
        B ← B' ;                                            // Save box size
    else
        Rcell ← setBoxSize(B);                              // Restore cell size
        SCL ← updateCells(SCL, r^s);                        // Restore super-cell list
    end
end
```

Algorithm 3: GPU implementation of MC move, as part of Algorithm 1

```
Function updateCells(SCL,r⃗,OutOfCell) // Update super-cell list
    input  : Current super-cell list, SCL
             updated positions, r⃗
             Array of out-of-cell particles, OutOfCell
    output : Updated super-cell list, SCL

    // Update all particles that stayed in the same super-cell
    SCL ← updateSuperCell(SCL,r⃗);
    // Update all particles that moved between super-cells
    SCL ← moveSuperCell(SCL,r⃗);
    // Accommodate any out-of-cell particles in the super-cell list.
    SCL ← updateOutOfCell(SCL,OutOfCell);

    return SCL
end
```

Algorithm 4: GPU implementation of super-cell list update, as part of Algorithms 2, 3

5.3.3 Cell List Initialization and Update

Cell List Initialization

To initialize the cell list structure, we follow the modified bucket sort algorithm by Rożen *et al.* [126]. The classical bucket sort algorithm [132] achieves sorting of elements in linear time by distributing them to different 'buckets'. In the context of the cell list structure, the elements are particles and the 'buckets' are the cells. The sorting of particles within each cell, in this particular case, is not important.

The serial implementation of the bucket sort algorithm must be modified when using GPUs. We must guarantee that no memory clashes occur when two particles are assigned to the same cell at the same time. Rożen *et al.* revise the bucket sort algorithm to a GPU implementation. Based on the revised bucket sort implementation, we distribute the particles in a designated kernel. Each thread is assigned to a different particle and a different super-cell. An array of 8 counters in the super-cell list (one per cell) is initially set to zero. This counter keeps count of the number of particles already assigned to the current 'bucket' (cell).

The particle's desired super-cell and cell index are pre-determined by their initial location. As the kernel proceeds, each thread is trying to access the counter at the specific super-cell and cell index it belongs to (s and c, respectively). Accessing this counter is done using CUDA's atomic operations, such that only one thread at a time can increase the counter - an operation that results in obtaining the latest value of the counter, p.

In the typical case that the latest value of the counter is less then the maximal limit of particles per cell, $p < N'_{pc}$, the particle's information is safely added to the super-cell list in super-cell index s, cell index c and particle index p. If, however, p is greater than the maximal value, the particle's index is added to the out-of-cell list (see Section 5.3.6).

cell	counter	In-cell particles								Terminal particles							
0	3	8405	211	2997	-	-	-	-	-	-	-	-	-	-	-	-	-
1	1	4135	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
3	5	2360	5846	1646	7293	5261	-	-	-	-	-	-	-	-	-	-	-
4	9	7978	3009	2010	9808	2932	3182	2817	9362	5092	-	-	-	-	-	-	-
5	2	5126	3950	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	2	2989	2317	-	-	-	-	-	-	-	-	-	-	-	-	-	-
7	4	8916	9205	5250	6840	-	-	-	-	-	-	-	-	-	-	-	-

Table 5.1: Sample of super-cell data. An super-cell contains 8 cells. Each cell can contain up to $N'_{pc} = 16$ particles. A particle position in the cell list can either be occupied by a true particle index, or unoccupied and assigned a *dummy* particle (denoted in the table by '-'). The first $N_{pc} = 8$ particle positions are termed *in-cell*, and are more likely to be occupied, while the rest are termed *terminal* and are less likely to be occupied. The cell list implementation procedure (Section 5.3.3) ensures continuous occupation of particles in each cell, such that *dummy* particles are always pushed towards the higher particle positions.

Cell List Update

The advantage of the bucket sort algorithm is in allowing for simple resorting in the super-cell list in the following steps. The update is done in two stages. In the first stage, the super-cell list is scanned for particles that have moved to a new super-cell, and updates within the super-cell are performed. In the second stage, all particles that were indicated as 'moved' in the first stage, are being reassigned to their new location.

The first update stage is being done in a kernel that has number of threads matching the number of slots in the super-cell list (named **updateSuperCell**). Each particle in the super-cell is tested for its location. If the particle's location is such that it belongs to a different super-cell (and it is not a *dummy* particle), a flag is set for the particle index. The flag indicates the particle should be assigned to a new super-cell, and its information is being cleared from the super-cell list. The counter of the cell that contained this particle is subsequently decreased by one.

If the particle has moved to a new cell within the same super-cell, we update this change of cell number in the same kernel as well. While updating other super-cells requires slow global memory access, updating cells within the same super-cell can be done via the fast shared memory. We remove the particle's information from the old cell, and decrease the old cell's counter by one. We then enumerate through the particle indices of the new cell, trying to locate an empty position (indicated by a *dummy* particle). If one is found, we assign the particle to that location. If no empty position is found, we save the particle to the out-of-cell array of particles (see Section 5.3.6).

Finally, within the first stage kernel, we ensure that all the non-*dummy* particles are found first in the cell list (See Table 5.1). This will guarantee that, if we do find a *dummy* particle while enumerating particles in a given cell, all the particles following this *dummy* particle are

also *dummy* particles. Such ordering will allow us to terminate enumerations in the force calculation much sooner. To ensure this ordering of particles we first synchronize all the threads in the block. As each block is assigned to a different super-cell, this ensures that all particles that needed to be *removed* from the super-cell have been removed and replaced with dummy particles. Then, for each particle having an index that exceeds the current counter, we find a new, empty, location within the cell list. The final result is a well sorted cell list in which the last non-*dummy* particle is found in a position identical to the current cell counter.

In the second stage of updating, we treat particles that have moved to a new super-cell (kernel **moveSuperCell**). This is a by-particle kernel where the number of threads is identical to the number of particles in the system. Only particles that are flagged to have moved a super-cell continue in the kernel. Their location in the new super-cell can readily be found by using the cell counter. Updating the particle's information in the super-cell list is done through global memory, and therefore this kernel update is rather slow, even if it involves only a small number of particles.

5.3.4 Non-Bonded Forces

The essence of the force calculation algorithm is to allow each particle to calculate its interactions with particles in half the space surrounding it, and to update the force on them as well (See Figure 5.2). This would be done in parallel, such that each thread will be assigned to a given particle and will have access to the variables (position, force *etc.*) associated with it. By allowing each particle to calculate interactions with only a pre-determined half of the space ($+\hat{x}$), we ensure that the interaction between any given pair of particles i, j would be calculated only once. This assurance is especially important in DPD, where the force is comprised of a random component. If we were to allow a force to be calculated more than once, different random numbers might have been drawn for the random force, resulting in a different total force.

Challenges arise when applying the opposite force. Say a particle i has calculated its interaction with particle j and the result was a non bonded force, $\vec{F}_{ij}(r_{ij})$, applied on particle i. An opposite force $\vec{F}_{ji}(r_{ij}) = -\vec{F}_{ij}(r_{ij})$ must now be applied on particle j (See Eq. 2.4). This requires additional communication, allowing the thread in charge of particle i to update the forces of particle j. We can minimize this communication overhead by using the faster *shared memory* as much as possible, and by avoiding CUDA's clash safe atomic operations.

The checkerboard decomposition [131] helps ensure force updates can be made non-atomically. When calculating non-bonded forces we serially enumerate super-cell types ($T = 0, \cdots, 7$). For each super-cell of same type we are calculating non-bonded forces in parallel, assigning a GPU block for each super-cell. Since super-cells of the same type are guaranteed to be non-adjacent, we are guaranteed that each cell, and therefore each particle, will be handled by only one block.

After assigning each super-cell to a GPU block, the **nonBondedForce** kernel prefetches data from global memory. The information on each local particle in the super-cell (See Table 5.2 for a detailed list of variables), as well as on all cells adjacent to the current

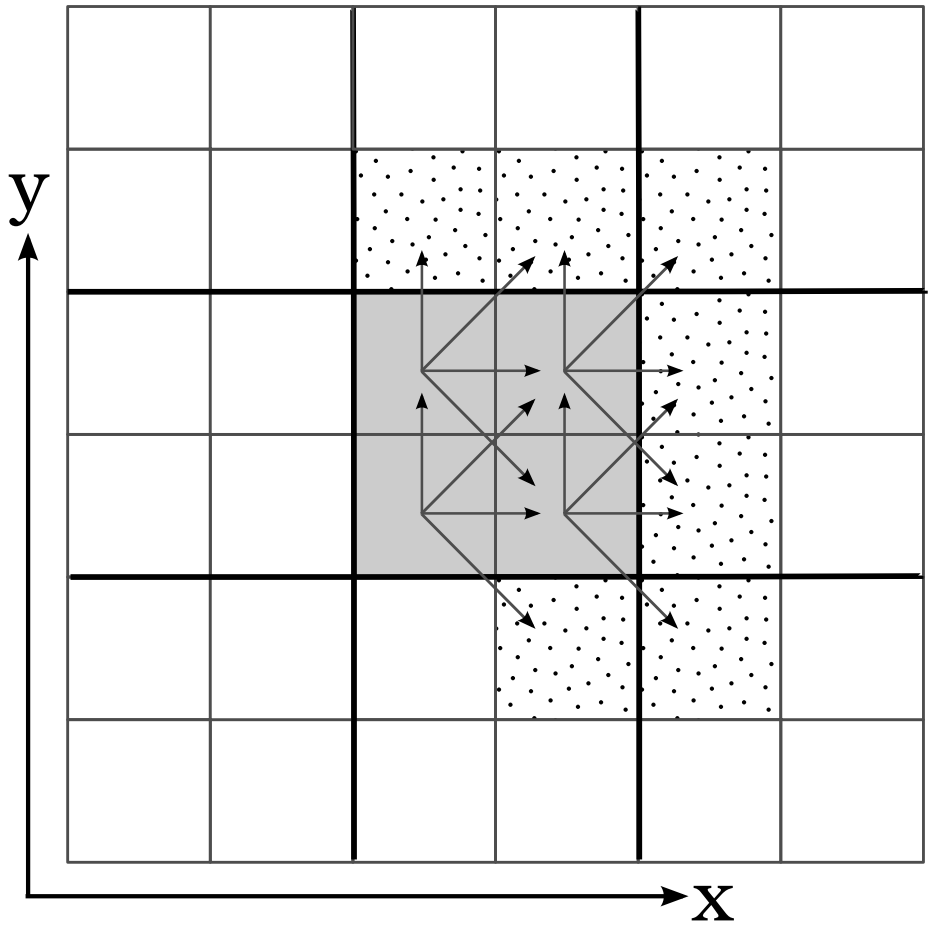


Figure 5.2: Illustration of the force calculation. The 3D system is presented in 2D here for clarity. The local super-cell being calculated is shaded grey (compare to super-cell in Figure 5.1). It contains 8 cells, with up to N_{pc} from which we can calculate the interactions in each cell. To ensure each pair of particles is calculated only once, each cell is calculating interaction with only half the space (denoted by arrows). The local cells allows us to calculate interactions with particles located in either other neighboring local cells (shaded grey), or in neighboring non-local cells (dotted). These non-local cells will be loaded into shared memory while their interaction is being calculated, to minimize read/write to global memory. Super-cells in this figure are separated by thick black lines. Cells are separated by thin black lines. In 3D, there are 35 non-local (dotted) neighboring cells.

super-cell is loaded into shared memory. This will allow faster writes to update forces on neighboring particles. Then, the forces on each local particle in the given super-cell are being calculated. In total, each block will be comprised of $N_{\text{threads per block}} = 64$ threads. The total number of blocks will be determined by the system size, and will be equal to $N_{blocks} = N^x_{rep} \times N^y_{rep} \times N^z_{rep} = N_{\text{SC}}/8$.

Forces with adjacent cells

The force calculation between the local particle i and all neighboring particles in adjacent cells is comprised of several steps:

1. Enumerate adjacent cells - Each particle calculates its interaction with particles in 13 adjacent cells to cover half the space. The direction of the adjacent cells is fixed such that the same half space is accessed each time.

2. Enumerate particles within the adjacent cell - Each of the $N_{pc} = 8$ possible particles in the adjacent cell is being accessed. The loop is done with an offset based on the local thread index, such that each particle is trying to access a *different* neighboring particle (avoiding bank conflicts).

3. Calculate the distance between the local particle and the neighboring particle. Here we ensure that the periodic boundary conditions are applied.

4. Calculate the force between the two particles only if the calculated distance is less the R_c. This might create warp divergence but has proved more efficient than non-branching, when profiling the program.

5. If any of the particles is a *dummy* particle, set the force to zero.

6. Add the calculated force on the local particle to a local register.

7. Add the force on the neighboring particle to its shared memory variable. In principal, this should be done atomically, to make sure no other thread is updating that same particle in the same time. In practice, we ensure algorithmically that no such collisions are possible. Each particle is accessed only once at each given time. See Section 5.3.5 for more details. The force update on the neighboring particle is therefore done by a faster non-atomic operation.

8. Copy back to global memory - When all threads are done updating the forces on neighboring cells, forces of all particles in all neighboring cells are copied back to global memory. Due to the use of the checkerboard decomposition, no two blocks can write back to the same cell and therefore this copy can be done non-atomically.

Variable name	Variable Size (bytes)	Definition
x, y, z	$3 \times 4 \times 128 = 1536$	Particle position along the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$ axes
vx,vy,vz	$3 \times 4 \times 128 = 1536$	Particle velocity along the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$ axes
fx, fy, fz	$3 \times 4 \times 128 = 1536$	Force on particle along the $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$, $\hat{\mathbf{z}}$ axes
type	$4 \times 128 = 512$	Particle type. Determines the force coefficient
index	$4 \times 128 = 512$	Particle index
Total Size	5632 bytes = 5.5 KB	

Table 5.2: Each variable is loaded for 64 local (same super-cell) particles as well as for 64 neighboring particles from neighboring cells.

Forces within the same cell

Forces within the same cell are calculated by enumerating over half of the other particles in the cell. We ensure that the interaction of all pairs of particles is being calculated, while the load is distributed evenly between the threads. To that end use an offset based looping scheme as described by Table 5.3.

The force is the being calculated based on the distance and velocity difference between the two particles. The force on the local particle is saved to the local register while the force on the neighboring particle is saved to shared memory. The offset based enumeration ensures that each particle is being updated by only one thread at any given time and so updates to shared memory are done non-atomically.

After all forces, within the cell and with adjacent cells, have been calculated, we copy them back to global memory. First, we add the forces saved in local registers to the forces on the local particle that were saved through shared memory. These include any forces applied on the local particle by particles in the same super-cell. Then, each thread adds the forces saved in shared memory to the corresponding forces saved in global memory.

Random number generator

Each particle-particle interaction requires a random number. To minimize communication between threads, each thread maintains its own Random Number Generator (RNG), a CURAND [133] object. The state of all RNGs is being passed to the kernel, and each thread initializes its local RNG to the last seed saved for this thread index. After all force calculations are completed, the state (current seed) of each RNG is saved back to global memory. This ensures the full cycle of the RNG will be used. In total, we use $N_{blocks} \times N_{\text{threads per block}} = 8N_{\text{SC}}$ RNGs.

5.3.5 Optimizing Loads to Shared Memory

In calculating non-bonded forces, we need to access all the local variables of the particles in the current super-cell (64 particles), as well as the variables of particles in adjacent cells. In

Particle Index	0	1	2	3	4	5	6	7
0	-	x	x	x	x			
1		-	x	x	x	x		
2			-	x	x	x	x	
3				-	x	x	x	x
4					-	x	x	x
5	x					-	x	x
6	x	x					-	x
7	x	x	x					-

Table 5.3: Each thread, in charge of one particle index in a given cell (column heading) will calculate interaction with at most 4 other particles in the same cell (row heading), to distribute the load equally among all threads.

total, each super-cell is bordered by 35 non-local neighboring cells in the half space we wish to calculate interaction with (See Figure 5.2). Each of these cells include $N_{pc} = 8$ particles. Loading all information on all these cells simultaneously would result in a large use of shared memory ($\sim$ 15 KB per cell). Since only 48 KB of shared memory can fit in on SM (see Section 5.2), loading all 35 cells to shared memory at once would result in a reduction from the optimal 8 blocks per SM, and consequently in a slow down of the entire program. One can only load up to 8 adjacent cells (64 particles, see Table 5.2) at once to shared memory without changing the blocks per SM load.

Additionally, there is no need to load all 35 cells, information into shared memory simultaneously. Parallelization occurs on the super-cell level, by allowing each block to treat a different super-cell. Within the *kernel*, each block would have to enumerate adjacent cells to obtain all the different interactions. If such enumeration takes place either way, it means loading to shared memory can be done sequentially as well.

The question is how one can distribute these 35 cells, among different loads to shared memory, while still maintaining the following criteria:

1. Only 8 neighboring cells are loaded into shared memory at any given moment
2. No more than the minimal $\lceil 35/8 \rceil = 5$ loads to shared memory occur.
3. The load is being distributed equally, such that each thread is calculating interaction with exactly the same number of neighboring cells. Since we are loading only the $+\hat{x}$ half of space, cells on the $+\hat{x}$ side of a super-cell would have more non-local neighboring cells to calculate interactions with than cells on the $-\hat{x}$ side of a super-cell. Load should nonetheless be distributed such that all threads perform the same number of calculations.
4. Each non-local cell is being accessed by only one local cell at any point in time, to minimize memory collisions.

To achieve these goals we need to optimize the loads to shared memory. Each load of 8 neighboring cells to shared memory would be followed by a few iteration of force calculation with the loaded cells.

In the following paragraphs we describe the full optimization process for the distribution of non-local neighboring cells into shared memory loads.

Level 1: The naive approach

We begin with the naive approach, in which we iterate over 13 different directions. In each iteration $k = 0 \dots 12$, a direction $\vec{s_k}$ is being chosen. Each of the 8 local cells would calculate interaction with the cell in its $\vec{s_k}$ direction. For some this would be a local cell (within the same super-cell neighbor), while for others it will be a non-local neighboring cell.

This approach will result in inefficient loads to shared memory. Up to 16 non-local cells would be loaded twice and up to 7 non-local cells would be loaded 4 times in this scheme. This would result in 13 loads to shared memory, much larger than the optimal 5 loads that could in principal be achieved.

Level 2: Optimizing sets

The main problem with the naive approach is that there are non-local cells that are needed more than once (up to 4 times) in calculating interaction with local cells. It would be more beneficial to group these cells together, such that the loading to shared memory will happen only once for each non-local cell.

To that end, for each of the $k = 0 \dots 12$ iterations mentioned in the previous section, we define a *set*, $P_k = \{P_{k,0}, \dots, P_{k,7}\}$, which contains all the neighboring cells calculated in iteration k. We additionally define a scoring method for each set:

$$S^1(P_k) = \#\ of\ unique\ non{-}local\ cells\ in\ set\ P_k$$

We then try and minimize this score by swapping elements between sets. Swaps are only allowed between cells of the same index. For example, if local cell 0 is calculating interactions with non-local neighboring cell j in iteration 5 ($P_{5,0} = j$), and then local cell 3 is calculating interactions with that same non-local cell in iteration 6 ($P_{6,3} = j$), it would be better if these were done in the same iteration. Swapping $P_{5,3}$ with $P_{6,3}$ would result in lowering the score $S^1(P_5)$.

We perform the optimization process with regular MC. We define a sensitivity 'temperature' that determined the tolerance level for switches that increase the total score. We propose random swaps which we accept or reject based on the change in the total score $S^1 = \sum_{k=0}^{12} S^1(P_k)$.

The naive approach yields the following scores

$$S^1_{before} = \sum\{4, 6, 4, 6, 7, 6, 7, 6, 4, 6, 7, 6, 7\} = 76$$

Which after optimization can be lowered down to

$$S^1_{after} = \sum\{1,1,1,2,2,2,3,3,4,4,4,4,5\} = 36$$

Level 3: Optimizing loads

Now that we have minimized each set separately, we wish to bundle them sets together into shared memory loads. Each load into shared memory, L_i should include exactly 8 non-local cells. After these cells have been loaded, we iterate the different sets in each load, and calculate their interactions, before moving to the next batch of non-local cells to be loaded.

We preform a similar optimization scheme to distribute the sets $P_{k=0...12}$ into loads $L_{i=0...4}$. Our goal is to reach a distribution in which each load contains exactly 8 non-local cells. This means that sets that share same non-local cells would be encouraged to be part of the same load. Sets that have a high number of unique non-local cells would be encouraged to be part of the same load as sets that contain a small number of unique non-local cells.

We begin with a naive grouping, with loads of size $4,3,2,2,2$ sets each:

$$\begin{aligned} L_0 &= \{P_0, P_1, P_{11}, P_{12}\} \\ L_1 &= \{P_2, P_3, P_{10}\} \\ L_2 &= \{P_4, P_5\} \\ L_3 &= \{P_6, P_7\} \\ L_4 &= \{P_8, P_9\} \end{aligned}$$

We define a second scoring function:

$$S^2(L_i) = \Big|(\#\ of\ unique\ non{-}local\ cells\ in\ all\ sets\ of\ load\ L_i) - 8\Big|$$

and perform MC minimization by proposing random swaps of sets P_k, P_l between loads L_i, L_j. We accept or reject such swaps based on the total score $S^2 = \sum_{i=0}^{4} S^2(L_i)$. For example, the naive grouping produces the following number of non-local cells in each load: $\{11,7,4,6,7\}$, with a score of $S^2_{before} = \sum\{3,1,4,2,1\} = 11$. After minimization, we can find a distribution with exactly 8 non-local cells in each load ($\{8,8,8,8,8\}$) and a score of $S^2_{after} = 0$.

The result is powerful. We have found a way to distribute the 35 non-local neighboring cells among 5 different loads to shared memory. We can now have a 5-fold iteration where at each step we load 8 non-local neighboring cells into shared memory (each thread loads exactly one non-local particle), calculate ALL possible interactions with particles in those cells, and save the updated forces back to shared memory. This is already a notable improvement over the initial 13 loads in the naive approach.

Level 4: Optimizing memory clashes

The final optimization deals with memory clashes. So far, in the previous levels, we have tried to group interactions with the same non-local cell into the same set and load. This allowed us to achieve a distribution of cells that minimizes the number of shared memory loads. Accessing the same non-local cell by multiple local cells simultaneously has its disadvantages, however. It creates memory clashes while multiple threads try to read / write to the same address in memory.

In this level, after the loads have been set up, we wish to redistribute the cells within the different iterations of the load. After each load of neighboring cells to shared memory, several iterations will occur (4 for L_0, 3 for L_1 and 2 for L_2, L_3, L_4). If non-local cell j is being accessed by local cell 0 and by local cell 1 in the same iteration, memory clashes will occur. If however non-local cell j will be accessed by local cell 0 in the first iteration and then by local cell 1 at the second iteration, memory clashes will be alleviated.

We therefore make another optimization step. For each load we found in Level 3, we try and redistribute the cells between the different iterations such that each non-local cell is being accessed exactly once per iteration. If we could not achieve this goal, Level 3 was repeated until a no-memory clash distribution was achieved.

The resulting scheme can be found in Tables 5.5 and 5.6. With this scheme, one can calculate interactions of 8 local cells (and the 8 particles within them), with all 35 surrounding non-local cells (and the 8 particles within them), in an efficient, evenly distributed and memory-clashes-free way. This scheme does not need to be found repeatedly. We determine the optimal scheme once, and use the same scheme in all of our simulations. The scheme itself, as well as the optimization process leading to it, can be useful in other parallel applications as well.

5.3.6 Terminal and Out-of-cell particles

The scheme described in Section 5.3.4 provides substantial speed-up in force calculation, given the number of particles in a cell is less than N_{pc}. The complexity of these operations is $\mathrm{O}\big(N\big)$. This is true for a great majority of cells and particles (98.8% of particles), and so one might envision performing our fast scheme on only N_{pc} particles per cell, and using the naive approach for all particles that cannot be accommodated in the current cells. However, as the complexity of the naive approach scales as $\mathrm{O}\big(N^2\big)$, for larger system size these calculations will become the simulation's bottleneck.

Another approach would be to use a larger cell, allowing $N'_{pc} = 16$ particles per cell, and performing the optimized scheme described in Section 5.3.4. This approach proves faster than the naive approach, but for a majority of the cells, we will be performing a large number of idle calculations, as most particles will be assigned to be a *dummy* particle.

We choose a hybrid approach. We set the number of particles per cell to be $N'_{pc} = 16$. However, particle indices that exceed the $N_{pc} = 8$ limit of particles per cell (termed *terminal* particles from now on) are treated differently in the force calculation. Particles that don't

fit in to this even larger cell limit (probability for which is $P \simeq 1.25 \cdot 10^{-7}$), are saved to a separate array of particles, designated for *out-of-cell* particles. (See Table 5.1)

We therefore treat three types of particles, based on the particle's index within the cell, i:

(a) In-cell particles: $i < N_{pc}$

(b) Terminal particles: $N_{pc} \leq i < N'_{pc}$

(c) Out-of-cell particles: $i \geq N'_{pc}$

Force calculation of terminal particles

We expect the population of cells (and super-cells) that contain terminal particles to be very low. We calculate the forces on terminal particles in a separate kernel (**nonBondedForce Terminal**). As with in-cell particles, each thread is assigned a specific particle (with index $i \geq N_{pc}$) in a specific cell. In contrary to in-cell particles, we exit the kernel if the assigned particle is a *dummy* particle. Exiting the kernel on a thread based if-statement creates branches which may slow down the kernel. Yet since this event (finding a terminal particle) is pretty rare. This branching is actually faster than following through with the entire function (see Figure 5.6 for timings).

In the event that a terminal particle is found, we calculate its interaction with other particles in neighboring cells and in the same cell. Calculating interactions of terminal particles in the same cell is done for only half the space (13 neighboring cells), to avoid repeating an interaction of a pair of particles, while for terminal particles in neighboring cells this is done over the entire space (26 neighboring cells), to cover all possible interactions.

Force calculation of out-of-cell particles

Since the probability of finding an out-of-cell particle is very low, we rarely encounter more than one out-of-cell particle at a time. Therefore, the limiting $\mathrm{O}\left(N^2\right)$ complexity is avoided.

We calculate the interaction of each out-of-cell particle with its surrounding particles by location the cell it should belong to in the cell list, and enumerate the full-space cells around it (26 neighboring cells, up to N'_{pc} particles each). If the number of out-of-cell particles exceeds 1, the interaction between these particles is explicitly calculated by a separate kernel (**nonBondedForceOOC**)

5.3.7 Bonded Forces

Bonded forces are also calculated in parallel on the GPU, even though their calculation is much faster than non-bonded forces (See Figure 5.6). This is to avoid copying particle information from the global device memory to the host memory and back. The kernel assigned for the calculation is called **bondedForce**.

We distribute the particles among different kernel blocks by molecule. Each bonded force calculating is limited to a specific molecule and therefore requires data (positions, for example) of particles only in the same molecule. By distributing the particles to blocks based on their molecular index, we ensure all position data will be available in shared memory, and all forces will be written to shared memory as well.

We allow for a maximum of 32 particles per molecule, which is sufficient for our needs. Once in the kernel, each of the 32 thread in a block is assigned a particle - either a true particle, or a *dummy* particle (if the molecule contains less than 32 particles). Each thread then loads the position, (x, y, z), of the particle into shared memory. The threads are then synchronized to make sure all the information has been uploaded before proceeding with the force calculation.

At this point, we change the designation of threads. First, each thread is assigned to one harmonic bond. The threads execute the calculation of the assigned bond based on the pre-loaded particle positions in shared memory. It adds the resulting force to the force array in the index corresponding to the bonded particles, in shared memory. Second, each thread is assigned to one angle bond, executing the angle force calculation based on the three particles constituting the assigned angle. Lastly, the same is done for dihedral angles.

The end result is the parallel calculation of all harmonic-bond, angle, and dihedral angle forces (based on Eq. 2.2). The forces are added into a shared memory array, to the index of all particles constituting the bond. The number of all bonds, angles and dihedral angles is each limited by the number of threads per block to 32. We have found that this limitation is sufficient in our simulations. Nonetheless, since this kernel is not the bottleneck of the simulation, it is possible to change the limit with no substantial changes to the simulation time.

The force on each particle in the molecule as a result of all bonded applied on it is added up through fast calculation in shared memory. It is then written back to global memory on a one thread per particle distribution.

5.3.8 Equations of Motion

We perform time integration based on the modified Velocity Verlet scheme described in Equation 2.7. We implement two kernels, each with number of threads corresponding to the number of particles.

One kernel, **moveParticles**, performs the first two stages of integration. It updates the position of each particle based on the velocity and force at the current time step. It then updates the estimated velocity, $\acute{\vec{v}}$, based on the current velocity and force. This velocity will be used in the non-bonded force calculation (Equation 2.3).

The second kernel, **updateVelocities**, is called after all the forces (bonded and non-bonded) on all particles have been calculated. It uses the average of the force at the previous time step along with the newly calculated force on the particle, to update the velocity on each particle.

These two kernels are truly parallelizable, as no communication is needed between any

two threads. This leads to substantial speedup in these kernels over the sequential CPU version.

5.3.9 Monte Carlo moves

We allow for several Monte Carlo (MC) moves in our simulation, including pressure moves and surface tension moves (see Section 2.4). All MC moves consist of changes to the box size, following the same procedure:

1. Calculate system's energy, including bonded and non-bonded energy contributions.
2. Propose a move in system box size: $(B'_x, B'_y, B'_z) = (R_x \times B_x, R_y \times B_y, R_z \times B_z)$
3. Rescale particles' coordinates according to the proposed move
4. Recalculate system's energy.
5. Accept or reject move, according to change in energy. Each move type has a different acceptance probability.
   (a) If the move is rejected: Rescale back particle's coordinates.
   (b) If the move is accepted: Save an continue

In practice we find that in a system with a lot of bonded molecules it is best to only rescale one bead per molecule [45]. All the other particles in the molecule are subsequently translated such that their relative position to the scale bead in the molecule remains the same. In that case, the bonded energy remains constant and only the non-bonded energy changes. The result is higher acceptance rates for all MC moves.

Calculating non-bonded energy (kernel **nonBondedEnergy**) is done analogously to non-bonded force calculation. Each GPU block corresponds to one super-cell, that calculates interactions with surrounding cells by loading their data into shared memory. Instead of saving the result directly into a per-particle array as done in the force calculation, a shared memory energy variable is set up. This variable will locally sum up the super-cell's contributions to the non-bonded energy. One thread will add this result to global memory at the end of the kernel.

Scaling of particle's coordinates is divided into two kernels. One kernel, **scaleNonMolecular**, scales the coordinates of all particles that are not bonded. In our simulation these include the water beads. Scaling of the coordinates is easily parallelizable where each thread treats each particle and sets the particle position to be $(x'_i, y'_i, z'_i) = (x_i \times R_x, y_i \times R_y, z_i \times R_z)$.

The second kernel, **scaleMolecular**, is distributed such that each block treats a different molecule. This scheme is analogous to the bonded force calculation. All the particles in a molecule require access to the position of the head bead in the molecule. Its position, before and after the scale, can be saved locally to shared memory. Each thread is then assigned to a

particle in the molecule. The position of particle i is then determined by its relative position to the head particle, 0. $(x'_i, y'_i, z'_i) = (x'_0, y'_0, z'_0) + (x_i - x_0, y_i - y_0, z_i - z_0)$.

One major obstacle in performing changes the box size, as proposed by MC moves, is that it might lead to changes in the super-cell list structure. As the size of each cell is limited by the force cutoff distance ($r_{cell} \geq R_c$), decreasing the box size might require recalculation of the entire super-cell list, as well as the bucket sort procedure (See Section 5.3.3). This recalculation is very expensive, especially as it might be needed frequently for a fluctuating box size.

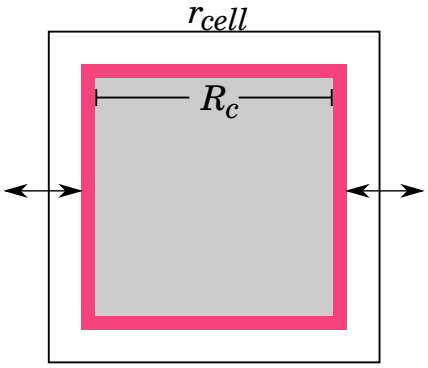


Figure 5.3: Cell size in a simulation with MC moves in system size. r_{cell} is set initially to a large value ($\varepsilon = 1.05$). The box size changes according to proposed MC moves, while the number of cells, super-cells remains constant. Each move, r_{cell} is recalculated and is assured to be larger than a minimum $R_c \cdot \varepsilon_{min}$, denoted by the magenta area around the cell. If the move proposes to decrease r_{cell} below the minimum value, the simulation is halted and restarted with a smaller number of super-cells.

We wish to avoid recalculating the super-cell list by maintaining a fixed number of cells and super-cells throughout the simulation. We can achieve that by initializing the cell size to be slightly larger the R_c, and constantly checking that it doesn't fall below a lower limit (the minimum cell boundary factor, ε_{min}). The lower limit is chosen such that only below that limit is it likely to propose a move that will change the cell size to be smaller than R_c.

We therefore initialize the cell size according to Equation 5.2, with $\varepsilon = 1.05$. The cell size is therefore ensured to exceed $r^{\alpha}_{cell} \geq R_c \cdot \varepsilon$ in all directions ($\alpha = \hat{x}, \hat{y}, \hat{z}$). We propose moves that change the box size via the MC procedure. The number of cells and super-cells does not change throughout the simulation. Each time a move is proposed, the cell size in each direction is recalculated ($r^{\alpha}_{cell} = \frac{B^{\alpha}}{N^{\alpha}_{cell}}$). If the cell size is lower than the minimum ($r^{\alpha}_{cell} < R_c \cdot \varepsilon_{min}$ for any $\alpha = \hat{x}, \hat{y}, \hat{z}$, $\varepsilon_{min} = 1.015$), we halt and restart the simulation with a smaller number of super-cells. A schematic representation of our cell size determination process is provided in Figure 5.3. We find that a restart of the simulation is needed very rarely, and only in cases where the initial configuration is very far from equilibrium.

We note that when the simulation is done under constant volume, with no MC moves, we do not use this procedure. In that case we set $\varepsilon = \varepsilon_{min} = 1.0$.

Kernel	Thread base	Threads per Block	No. of Blocks
movParticles	particle	128	$\lceil N/128 \rceil$
updateVelocities	particle	128	$\lceil N/128 \rceil$
updateSuperCell	super-cell	$N'_{pc} \times N_c = 128$	N_{SC}
moveSuperCell	particle	128	$\lceil N/128 \rceil$
nonBondedForce	super-cell	$N_{pc} \times N_c = 64$	$N_{rep} = N^x_{rep} \times N^y_{rep} \times N^z_{rep}$
nonBondedForceTerminal	super-cell	$N_{pc} \times N_c = 64$	$N_{rep} = N^x_{rep} \times N^y_{rep} \times N^z_{rep}$
nonBondedForceOOC	out-of-cell	N_{ooc}	N_{ooc}
nonBondedEnergy	super-cell	$N'_{pc} \times N_c = 128$	N_{SC}
bondedForce	molecule	32	$\lceil N_{mol}/32 \rceil$
scaleMolecular	molecule	32	$\lceil N_{mol}/32 \rceil$
scaleNonMolecular	particle	128	$\lceil N/128 \rceil$

Table 5.4: Overview of the different kernels in the GPU implementation of DPD. Each kernel is divided into blocks and threads to match either the total number of particles (N), the total number of molecules (N_{mol}), or the total number of super-cells (N_{SC}). $N_c = 8$ is the number of cells per super-cell. Details on the kernels can be found in Sections 5.3.1- 5.3.9.

5.4 Performance Results

To test the performance of our algorithm we compared the simulation times of our GPU implementation an equivalent CPU version. We used the GeForce 590 GTX GPU, which comes equipped with 1024 CUDA cores, 32 streaming multiprocessors (SMs) and 3GB DRAM. The card delivers performance of 2448.3 GFlops and 327.7 GB/s peak memory bandwidth. The PCIe 2.0 with 16 lanes is used to transfer data back and forth from the CPU to the GPU memory. The CUDA compiler driver NVCC along with gcc 4.1.2 with -O3 optimization flag is used for all GPU simulations.

We compared the performance tests to a serial CPU version of the DPD algorithm. The CPU version ran on 3 GHz Quad-Core Intel Xeon processor with 8GB DRAM. The intel ifort 10.1 compiler was used with -O3 optimization flag.

We simulated two types of systems. The simpler system includes a set of N 'water-like' beads. These are non-bonded particles, all of the same type. We performed these water-box simulations with varying number of particles ($192 \leq N \leq 8388608$). The system was simulated in either the NVT ensemble ($T = 0.7$; $V = {}^{N}/_{\rho} = {}^{N}/_{3.0}$), or the $NP\,T$ ensemble ($T = 0.7$; $P = 22.28$, the bulk pressure of water).

We additionally simulated various bilayer systems. We use a varying number of lipids ($128 \leq N_{lip} \leq 65536$) with the lipid model described in Section 2.2. The temperature is set to be well within the L_α phase of the bilayer, $T = 0.7$. We use an hydration layer of 30 water beads per lipid, reducing any correlation between periodic images of the system. The total number of particles in the bilayer system therefore amounts to $N = 30 \times N_{lip} + 13 \times N_{lip} = 43 \times N_{lip}$. We simulate this system in either the NVT ensemble, or the $NP\,\gamma\,T$ ensemble ($\gamma = 0.0$; $P = 22.28$). In the canonical ensemble we estimate the system's volume based on the typical area per lipid we observe in this model, $A_{pl} = 1.54$, as well as the typically observed membrane thickness, $d_{\mathrm{M}} = 5.6277$.

These four types of systems: waterbox / bilayer, NVT / non-NVT allow us to probe the performance of different segments in our code. Bilayer systems contain bonded interactions and display a larger diversity of non-bonded interactions with three bead types rather than one. Simulations performed in the NVT ensemble show the performance of only the DPD scheme, while non-NVT simulations show the interplay of the hybrid DPD-MC scheme. Both systems were simulated for 1000 cycles.

We note here that NVT simulations are expected to be longer in simulation time compared to non-NVT simulations. This is due to the fact that the number of DPD cycles, as well as DPD sub-steps, N_{sub} is different in both schemes. In the hybrid DPD-MC scheme, N_{sub} is randomly chosen from the uniform distribution ($N_{sub} \in \mathcal{U}(0, 50]$) to obey detailed balance. On average, $\langle N_{sub} \rangle = 25$ in the non-NVT schemes. In the DPD only scheme, however, N_{sub} is set to be fixed at 50.

Additionally, the number of DPD cycles is different. In the waterbox $NP\,T$ ensemble 0.2 of the cycles result in an MC move while 0.8 of the cycles result in a DPD move (see Algorithm 1). In the bilayer $NP\,\gamma\,T$ ensemble these ratios are 0.4 and 0.6, respectively. The total number of DPD sub-steps in the waterbox $NP\,T$ (bilayer $NP\,\gamma\,T$) simulation is

therefore 20000 (15000) on average, compared to a total of 50000 DPD sub-steps in the *NVT* simulation.

Simulation times of all four systems are shown in Figures 5.4 and 5.5. CPU simulation times are shown in blue symbols while GPU results are represented by red symbols. Results are shown in both linear and log-scale, for clarity. These results show an overwhelming speed-up in the DPD algorithm with GPU, reaching up to 30 times speed-up.

Figure 5.6 shows the breakdown of simulation time into kernels. As expected, non bonded force calculations are the most time consuming, taking up $\sim 83\%$ of simulation time in bilayer simulations. Updating the super-cell and performing MC moves constitutes $\sim 5\%$ of simulation time each. MC moves include a calculation of the non bonded energy twice. These calculations are expected to be at least as time consuming as the non bonded force calculations. However, since MC moves are performed less frequently than DPD moves, and contain only two energy calculations, compared to $N_{sub} \approx 25$ DPD sub-steps, the overall timing is much smaller.

5.5 Conclusions

In this chapter we present a novel algorithm for Dissipative Particle Dynamics. This GPU-based algorithm was constructed following the guidelines of minimum communication through global memory. Non-bonded force calculation, constituting the bottleneck in this algorithm, is calculated by a network of super-cells. Each super-cell is in charge of calculating interactions with its surrounding cells.

Parallelizing this algorithm is challenging due to the unique nature of the Dissipative Particle Dynamics algorithm. The DPD force does not depend only on the distance between the interacting beads, but also on a random number. Additionally, Newton's third law is maintained, such that each interaction must be calculated only once, and communicated to both interacting beads. Therefore, communication between different elements of the system is inherit in the technique.

We develop a method that decreases the communication to the necessary minimum. Using minimal number of loads to shared memory, we perform most of the calculations using the fast shared memory. We then communicate the results via global memory, doing so as little as computationally possible.

Our algorithm shows great speed-up of 30 times or higher over the CPU version. It now allows the simulation of much larger systems for much longer times.

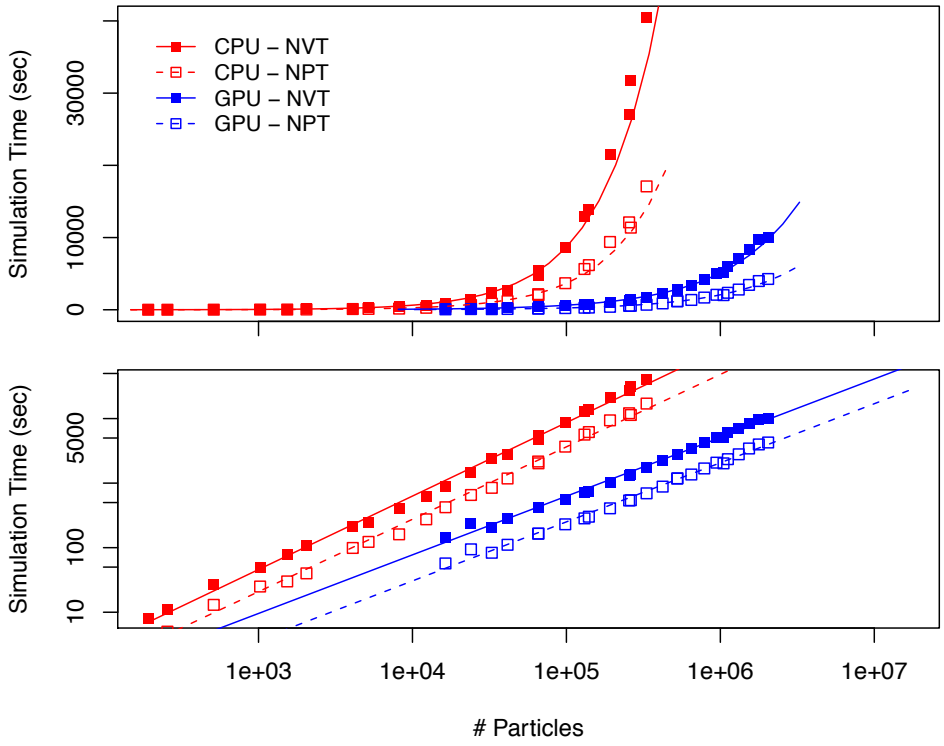


Figure 5.4: GPU performance as a function of system size for simulation of water particles. x-axis corresponds to the number of particles, N, in the system. Simulation times of $N_{step} = 1000$ cycles are presented in the y-axis, in both linear (top figure) and log (bottom figure) scale. Blue and red symbols correspond to GPU and CPU simulation times, respectively. Each system was simulated at constant volume (filled symbols), with only DPD moves, as well as in the NPT ensemble (empty symbols), with both DPD and MC moves (See Section 2.4.1). Fit lines were added to each plot for convenience.

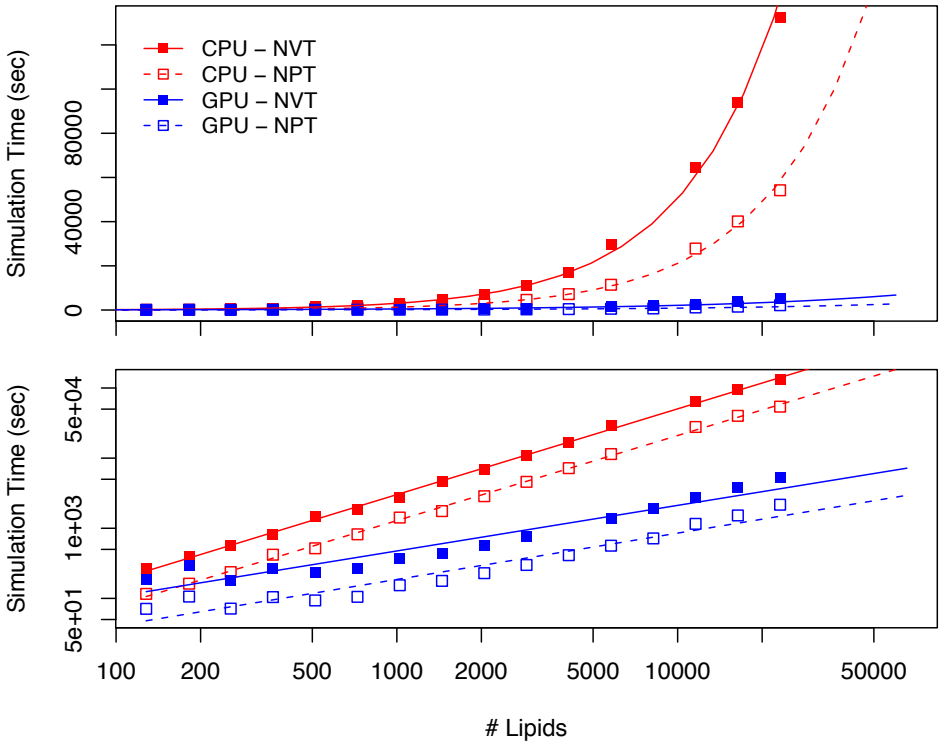


Figure 5.5: GPU performance as a function of system size for simulation of lipid bilayers. x-axis corresponds to the number of lipids in the bilayer, N_{lip}. Total number of particles in the system corresponds to $N = 43 \times N_{lip}$. Simulation times of $N_{step} = 1000$ cycles are presented in the y-axis, in both linear (top figure) and log (bottom figure) scale. Blue and red symbols correspond to GPU and CPU simulation times, respectively. Each system was simulated at constant volume (filled symbols), with only DPD moves, as well as in the $NP\gamma T$ ensemble (empty symbols), with both DPD and MC moves (See Section 2.4.1). Fit lines were added to each plot for convenience.

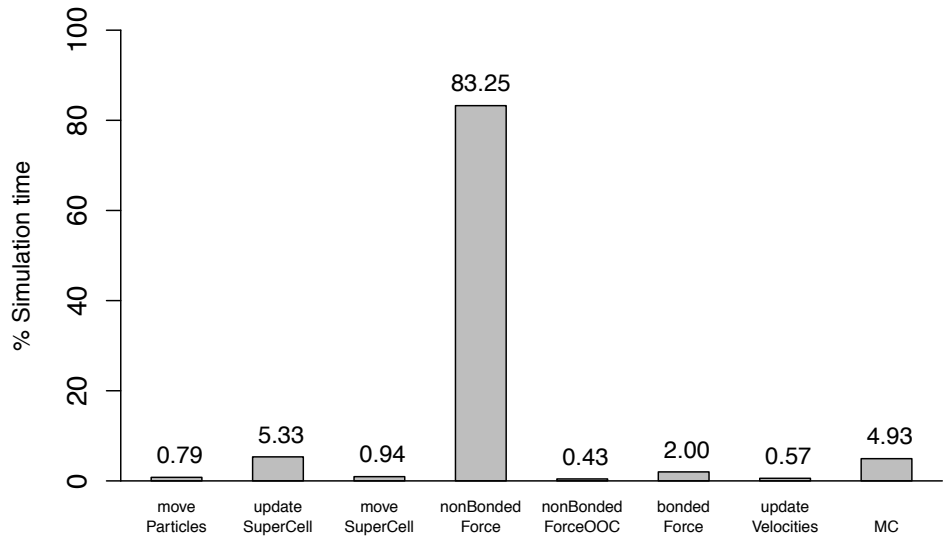


Figure 5.6: Breakdown of simulation time by kernels, for a test system containing 1024 lipids, simulated in the $NP\gamma T$ ensemble. Non bonded forces are the most time consuming, constituting 83.25% of simulation time.

5.6 Appendix

Table 5.5: Optimized distribution of non-local cells to shared memory loads.

Loaded cell name	Load Index	Local cell	Non-local super-cell position	Non-local cell index
$N_{0,0}$		C_0	$+\hat{\mathbf{x}}$	1
$N_{0,1}$		C_1	$+\hat{\mathbf{x}}$	3
$N_{0,2}$		C_2	$+\hat{\mathbf{x}}$	2
$N_{0,3}$		C_3	$-\hat{\mathbf{y}}$	6
$N_{0,4}$	0	C_4	$+\hat{\mathbf{x}}-\hat{\mathbf{y}}$	2
$N_{0,5}$		C_5	$+\hat{\mathbf{x}}$	0
$N_{0,6}$		C_6	$+\hat{\mathbf{y}}$	1
$N_{0,7}$		C_7	$+\hat{\mathbf{x}}+\hat{\mathbf{y}}$	0
$N_{1,0}$		C_0	$+\hat{\mathbf{y}}$	5
$N_{1,1}$		C_1	$+\hat{\mathbf{x}}+\hat{\mathbf{y}}$	1
$N_{1,2}$		C_2	$+\hat{\mathbf{x}}-\hat{\mathbf{y}}$	3
$N_{1,3}$		C_3	$-\hat{\mathbf{y}}$	7
$N_{1,4}$	1	C_4	$+\hat{\mathbf{z}}$	4
$N_{1,5}$		C_5	$+\hat{\mathbf{y}}$	4
$N_{1,6}$		C_6	$-\hat{\mathbf{z}}$	7
$N_{1,7}$		C_7	$+\hat{\mathbf{x}}+\hat{\mathbf{y}}+\hat{\mathbf{z}}$	0
$N_{2,0}$		C_0	$-\hat{\mathbf{z}}$	3
$N_{2,1}$		C_1	$+\hat{\mathbf{y}}$	0
$N_{2,2}$		C_2	$+\hat{\mathbf{z}}$	6
$N_{2,3}$		C_3	$+\hat{\mathbf{x}}$	3
$N_{2,4}$	2	C_4	$+\hat{\mathbf{x}}$	2
$N_{2,5}$		C_5	$+\hat{\mathbf{x}}+\hat{\mathbf{z}}$	2
$N_{2,6}$		C_6	$+\hat{\mathbf{x}}+\hat{\mathbf{y}}-\hat{\mathbf{z}}$	1
$N_{2,7}$		C_7	$+\hat{\mathbf{y}}-\hat{\mathbf{z}}$	1
$N_{3,0}$		C_0	$+\hat{\mathbf{y}}$	5
$N_{3,1}$		C_1	$+\hat{\mathbf{z}}$	2
$N_{3,2}$		C_2	$+\hat{\mathbf{y}}+\hat{\mathbf{z}}$	0
$N_{3,3}$		C_3	$+\hat{\mathbf{z}}$	6
$N_{3,4}$	3	C_4	$+\hat{\mathbf{y}}-\hat{\mathbf{z}}$	5
$N_{3,5}$		C_5	$+\hat{\mathbf{z}}$	0
$N_{3,6}$		C_6	$+\hat{\mathbf{x}}-\hat{\mathbf{y}}-\hat{\mathbf{z}}$	3
$N_{3,7}$		C_7	$-\hat{\mathbf{y}}-\hat{\mathbf{z}}$	7
$N_{4,0}$		C_0	$+\hat{\mathbf{x}}-\hat{\mathbf{z}}$	1
$N_{4,1}$		C_1	$+\hat{\mathbf{x}}-\hat{\mathbf{y}}+\hat{\mathbf{z}}$	2
$N_{4,2}$		C_2	$+\hat{\mathbf{x}}+\hat{\mathbf{z}}$	0
$N_{4,3}$		C_3	$-\hat{\mathbf{y}}+\hat{\mathbf{z}}$	6
$N_{4,4}$	4	C_4	$+\hat{\mathbf{y}}$	4
$N_{4,5}$		C_5	$+\hat{\mathbf{x}}-\hat{\mathbf{z}}$	3
$N_{4,6}$		C_6	$+\hat{\mathbf{y}}+\hat{\mathbf{z}}$	4
$N_{4,7}$		C_7	$-\hat{\mathbf{z}}$	5

Table 5.5: For a given load index, each local cell loads exactly one non-local neighboring cell. The location of the neighboring cell is defined by the relative position of the neighboring super-cell (*Non-local super-cell position*) and the cell index within that super-cell (*Non-local cell index*). Loaded cell name is used for future reference

Table 5.6: Optimized scheme of calculating interactions with neighboring cells

Load Index	Iteration	Local cell	Calculate interaction with
0	0	0	$N_{0,3}$
		1	C_5
		2	$N_{0,6}$
		3	C_4
		4	$N_{0,2}$
		5	$N_{0,0}$
		6	$N_{0,5}$
		7	$N_{0,7}$
0	1	0	C_5
		1	C_3
		2	C_4
		3	C_7
		4	$N_{0,4}$
		5	$N_{0,5}$
		6	$N_{0,0}$
		7	$N_{0,1}$
0	2	0	C_6
		1	$N_{0,3}$
		2	C_7
		3	$N_{0,6}$
		4	$N_{0,0}$
		5	$N_{0,1}$
		6	$N_{0,7}$
		7	$N_{0,5}$
0	3	0	C_4
		1	C_7
		2	C_6
		3	C_5
		4	$N_{0,5}$
		5	$N_{0,4}$
		6	$N_{0,2}$
		7	$N_{0,0}$
1	0	0	C_7
		1	C_2
		2	$N_{1,0}$
		3	$N_{1,4}$
		4	$N_{1,6}$
		5	C_6
		6	$N_{1,1}$
		7	$N_{1,5}$
		0	$N_{1,6}$
		1	$N_{1,3}$

Table 5.6 (cont'd): Optimized scheme of calculating interactions with neighboring cells

Load Index	Iteration	Local cell	Calculate interaction with
1	1	2	C_5
		3	$N_{1,5}$
		4	$N_{1,2}$
		5	$N_{1,4}$
		6	C_7
		7	$N_{1,7}$
1	2	0	$N_{1,3}$
		1	$N_{1,4}$
		2	$N_{1,6}$
		3	$N_{1,0}$
		4	C_7
		5	$N_{1,2}$
		6	$N_{1,5}$
		7	$N_{1,1}$
2	0	0	$N_{2,0}$
		1	C_4
		2	$N_{2,1}$
		3	$N_{2,2}$
		4	C_6
		5	$N_{2,4}$
		6	$N_{2,3}$
		7	$N_{2,5}$
2	1	0	C_2
		1	$N_{2,2}$
		2	$N_{2,7}$
		3	$N_{2,1}$
		4	$N_{2,3}$
		5	$N_{2,5}$
		6	$N_{2,6}$
		7	$N_{2,4}$
3	0	0	$N_{3,7}$
		1	$N_{3,5}$
		2	C_3
		3	$N_{3,1}$
		4	C_5
		5	$N_{3,3}$
		6	$N_{3,4}$
		7	$N_{3,0}$
		0	C_3
		1	$N_{3,1}$
		2	$N_{3,4}$
		3	$N_{3,2}$

Table 5.6 (cont'd): Optimized scheme of calculating interactions with neighboring cells

Load Index	Iteration	Local cell	Calculate interaction with
3	1	4	$N_{3,6}$
		5	C_7
		6	$N_{3,0}$
		7	$N_{3,3}$
4	0	0	C_1
		1	$N_{4,3}$
		2	$N_{4,7}$
		3	C_6
		4	$N_{4,5}$
		5	$N_{4,2}$
		6	$N_{4,0}$
		7	$N_{4,6}$
4	1	0	$N_{4,7}$
		1	C_6
		2	$N_{4,4}$
		3	$N_{4,6}$
		4	$N_{4,0}$
		5	$N_{4,1}$
		6	$N_{4,5}$
		7	$N_{4,2}$

Note. — We preform 5 different loads to shared memory. Within each load, several iterations are performed. At each *Load Index* and each *Iteration*, each *Local Cell* calculates interactions with particles of a neighboring cell. The neighboring cell can either be a local cell, or a non-local cell. Its location in shared memory at the given load is readily determined by its name (as can be found under *Calculate interaction with*).
It can easily be confirmed that each neighboring cell is being accessed only once per iteration. We can therefore safely perform non-atomic writes to the neighboring cell variables.

Chapter 6

Bending Rigidity in Bilayers containing Cholesterol

Cholesterol is known to induce rigidity in lipid bilayers. We use our recently developed parallel algorithm to perform very large time and length scale explicit-solvent simulations of lipid bilayers at various Cholesterol concentrations. We show that the bending rigidity of membranes containing Cholesterol increases monotonically with Cholesterol concentration. We find that it displays a sharp increase at a concentration corresponding to a phase transition. While extracting the bending modulus of pure symmetric lipid bilayers is a common task, the method for extracting the bending modulus of asymmetric lipid bilayers containing a mixture of lipids is not as well known. We derive the Helfrich free energy for these asymmetric nearly flat bilayers and show that if this asymmetry is not taken into account one would observe an apparent non-zero surface tension in the undulation spectrum fit.

6.1 Introduction

Cholesterol is an essential component in living cell membranes. Typical Cholesterol concentration ranges from 20%-30%, and up to 50% in the plasmatic membrane [134]. Its functions in the cell vary from metabolism to being a precursor for vitamins and hormones.

In addition to its specific functions, Cholesterol is responsible for changes in the mechanical and structural properties of the cell membrane. Cholesterol plays an important role in increasing the membrane mechanical strength [135], in decreasing membrane permeability to water [136, 137] as well as in the condensation [138, 139], and in ordering [140] of the lipids in the bilayer.

The elastic properties of lipid membranes, including its bending rigidity, were also shown to be controlled by the presence of Cholesterol [141–143]. To obtain a molecular understanding of this effect, model systems containing one type of lipid and cholesterol are often investigated. Even in these simplified systems the effect of cholesterol is unclear, with large variability in results [144]. For example, in a DMPC bilayer containing 10% Cholesterol, reported bending

rigidities may range between $0.66 \cdot 10^{-19}$ J to $2.23 \cdot 10^{-19}$ J, even though reported errors are on the scale of $0.01 \cdot 10^{-19}$ J. All reported bending rigidities for DMPC/Cholesterol bilayers are shown for reference in Table 6.1.

Researchers have put forward different explanations for the increase in the bilayer bending rigidity in the presence of Cholesterol. The lipid raft model [145], in which Cholesterol induces the formation of a highly ordered lipid domains with the membrane plane is often proposed in that context. Pan *et al.* [141] suggest that the loose complexation of Cholesterol with neighboring lipids induces higher ordering in lipids. They argue that Cholesterol has a larger effect on ordering of lipids with more saturated chains since a larger number of lipids can be complexed with each Cholesterol molecule. The connection between lipid ordering and bending rigidity is not obvious, but Pan *et al.* observed that the ordering of the lipids and the bending rigidity show similar trends with increasing cholesterol concentration. In addition, Pan *et al.* also observed that the effect of cholesterol depends on the type of lipid. Namely, they note a large increase in bending rigidity for DMPC, a lipid with two saturated tails, and almost no increase in bending rigidity for DOPC, a lipid with two unsaturated tails, with increasing Cholesterol concentration.

Most of these molecular explanations for increasing bending rigidity with the addition of Cholesterol assume that changing the cholesterol concentration has little effect on the overall structure of the membrane. It is, however, known that cholesterol may induce phase transitions. For example, if we increase the cholestrol concentration in a DMPC/cholesterol system we observe a phase transition from the L_α liquid disordered phase to the L_o ordered phase. Such a phase transition may have a significant effect of the mechanical properties of the bilayer, including the bilayer bending rigidity.

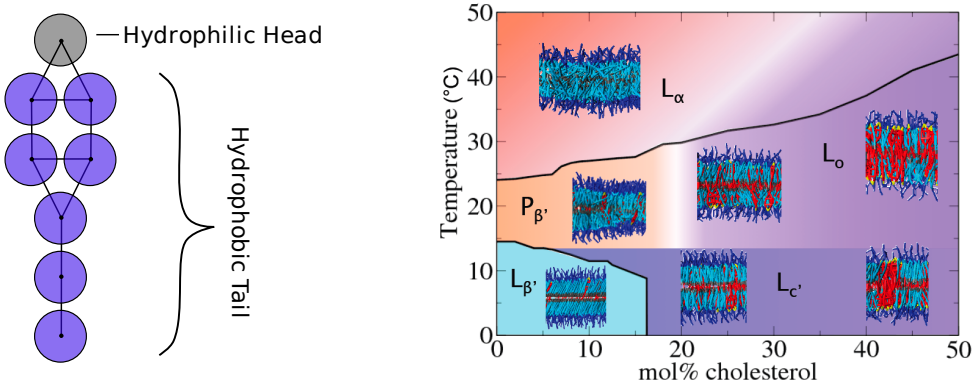
Molecular simulations can be utilized to explore the molecular level as well as collective behavior of lipids molecules simultaneously. Our CG lipid model, along with a CG model for Cholesterol has been used by de Meyer *et al.* to explore the temperature-composition structure diagram of a model DMPC/Cholesterol system [146]. Their simulations identify five distinct structural phases (Figure 6.1).

While these CG simulations allow one to perform simulations on a relatively large system for a long time, calculating the bending rigidity is a relatively difficult task. The bending rigidity is extracted from the elastic bending modes of a membrane. To ensure spatial independence and to access variable Cholesterol/DMPC domain compositions, the elastic deformations of a large membrane patch should be observed. The relaxation time of the largest bending modes, however, scales as $\tau \sim L^4 \sim N^2$, where L is the system size and N is the total number of molecules in the bilayer (lipids and Cholesterol) [152]. We therefore need to perform, even for a coarse-grained model, very long time and length scale simulations to accurately calculate the bending rigidity of a DMPC/Cholesterol membrane.

The novel GPU algorithm for Dissipative Particle Dynamics, described in Chapter 5, allows us to perform this otherwise unfeasible task. Taking advantage of the 30 times speed-up, we perform explicit-solvent simulations of a comparably large number of $N = 4096$ molecules (DMPC plus Cholesterol) at 10 different Cholesterol concentrations. In the following sections we discuss the details of the simulation and analysis, as well as the computed bending moduli.

Ref.	Temp. (°C)	% Chol.	κ (10^{-19} J)	Method
[147]	26°	0	$0.35 - 0.65$	Flickering of tubular vesicles
[148]	26°	0	2.4 ± 0.4	Flickering of tubular vesicles
[149]	27°	0	1.41 ± 0.13	Optical force / Confocal microscopy
[143]	27°	0	1.52 ± 0.06	Thermal fluctuations of GUVs
[150]	29°	0	0.56 ± 0.06	Micropipette aspiration
[141]	30°	0	0.58 ± 0.08	X-ray scattering
[142]	30°	0	1.15 ± 0.15	Fourier Analysis of GUV undulations
[143]	30°	0	1.30 ± 0.08	Thermal fluctuations of GUVs
[149]	30°	0	1.33 ± 0.12	Optical force / Confocal microscopy
[151]	37°	0	1.40 ± 0.03	Flickering
[147]	29°	6	$0.28 - 0.35$	Flickering of tubular vesicles
[141]	30°	10	0.66 ± 0.08	X-ray scattering
[143]	30°	10	2.0 ± 0.1	Thermal fluctuations of GUVs
[143]	27°	10	2.23 ± 0.07	Thermal fluctuations of GUVs
[141]	30°	20	1.32 ± 0.1	X-ray scattering
[142]	30°	20	2.1 ± 0.25	Fourier Analysis of GUV undulations
[141]	30°	30	2.74 ± 0.1	X-ray scattering
[142]	30°	30	4.0 ± 0.8	Fourier Analysis of GUV undulations
[143]	30°	30	4.1 ± 0.25	Thermal fluctuations of GUVs
[143]	30°	50	6.1 ± 0.3	Thermal fluctuations of GUVs

Table 6.1: Experimental results of bending rigidities in DMPC/Chol bilayers show large variability in numerical values. Description of experimental techniques can be found in each reference are reviewed by Bouvrais [144].



(a) (b)

Figure 6.1: (a) Coarse-Grained model of Cholesterol, as used by de Meyer *et al.* [146]. The stiff tetrameric cholesterol ring is represented by five beads while the Cholesterol tail is represented with two beads. The hydrophilic head bead is displayed in grey color and the hydrophobic tail beads in blue. Their non-bonded interaction parameters match that of the lipid model (See Section 2.2.1). (b) Temperature-composition structure diagram of DMPC-cholesterol bilayer shows five distinct structural phases (Figure adopted from de Meyer *et al.*)

6.2 Determining Bending Rigidity

To determine the bending rigidity of a lipid bilayer, we use the Helfrich model. On a large scale, a membrane can be considered a continuum surface. The Helfrich free energy [153] relates the membrane local principal curvatures, c_1 and c_2, as well as the membrane spontaneous curvature, c_0, to its elastic energy. For 2D surfaces with fixed topology the free energy takes the following form:

$$\mathcal{F} = \int_A dA \left[\sigma + 2\kappa(c_1 + c_2 - c_0)^2\right] \tag{6.1}$$

where A denotes the total area of the surface, σ is the intrinsic surface tension and κ is the bending modulus.

The Helfrich model has been successfully used in describing the thermal fluctuations around the equilibrium surface state [154, 155]. In the following sections we summarize the conventional method for extracting the intrinsic surface tension and bending modulus, and present the effect of symmetry breaking on this method.

6.2.1 Symmetric membranes

Simulations of lipid membranes may be performed in several different ensembles. In the thermodynamic limit, the ensemble in which the membrane is simulated should not affect

the undulations of the membrane surface. In finite-size membranes, however, the simulated ensemble influences the membrane undulations and dictates a different relation of the Helfrich free energy. Here we focus on two ensembles, constant area ($NP_\perp A_p T$) and constant frame tension ($NP_\perp \gamma T$), where the frame tension, γ, is conjugated to the projected area, A_p. The $NP_\perp A_p T$ ensemble is more useful to rigorously show how the elastic moduli can be extracted using the Helfrich theory. In practice, however, we prefer to simulate in the $NP_\perp \gamma T$ ensemble to better represent the physical tensionless system [42]. We therefore discuss and distinguish between both ensembles below.

We note here that the frame tension, γ, and the surface tension, σ, are different quantities. The former is conjugated to the projected area of the membrane (A_p) while the latter is conjugated to the total area of the membrane (A). The total area of the membrane is constructed of the projected area of the membrane as well as the local curvatures of the membrane. The total area is therefore always greater or equal to the projected area.

In the $NP_\perp A_p T$ ensemble the size of the simulation box along the $\hat{\mathbf{x}}-\hat{\mathbf{y}}$ plane (the projected simulation box area, A_p) is fixed. The total number of lipids is also fixed, and therefore fluctuations in the surface area A can only arise from forming local surface curvatures. The volume of the simulation box may change to provide constant normal pressure ($P_\perp$), by changing only the box size in the $\hat{\mathbf{z}}$ direction.

The statistical mechanics of a simulated membrane with fixed projected area has been rigorously presented in previous publications [156–158]. These works show the equivalence between the thermodynamic definition of the surface tension, σ, and bending modulus, κ, and their extraction from the surface undulations.

In short, one may relate the Helfrich free energy to the surface height, $h(x,y)$, at the 2D coordinate (x,y). Assuming small curvature ($^{\partial h}/_{\partial x} \ll 1$; $^{\partial h}/_{\partial y} \ll 1$), one may rewrite the Helfrich free energy (Eq. 6.1) in terms of the derivatives of $h(x,y)$:

$$\begin{aligned} dA &\approx dxdy\left[1+\frac{1}{2}\left|\nabla h\right|^2\right] \\ c_1+c_2 &\approx \frac{1}{2}\nabla^2 h \end{aligned} \tag{6.2}$$

$$\Rightarrow F \approx \int_{A_p} dxdy\left[1+\frac{1}{2}\left|\nabla h\right|^2\right]\left[\sigma+2\kappa\left(\frac{1}{2}\nabla^2 h-c_0\right)^2\right] \tag{6.3}$$

For symmetric membranes the spontaneous curvature is zero ($c_0=0$). Omitting terms of order $\mathcal{O}(h^3)$ or higher, we obtain the following equation for the free energy of the surface:

$$F_{symmetric} \approx \sigma A_p + \frac{\sigma}{2}\int_{A_p} dxdy\left|\nabla h\right|^2 + \frac{\kappa}{2}\int_{A_p} dxdy\left|\nabla^2 h\right|^2 \tag{6.4}$$

The Helfrich free energy can now be rewritten with a discrete Fourier space basis set. Let $\{\vec{r}_g=(x_g,y_g)\}$ be a set of M^2 grid points. The discrete Fourier transform of $h(\vec{r}_g)$ is defined

by:

$$h_{\vec{q}} = \frac{1}{L_p} \sum_{\vec{r}_g} h(\vec{r}_g) e^{-i\vec{q}\cdot\vec{r}_g} \tag{6.5}$$

where $L_p = \sqrt{A_p}$ is the projected linear size of the membrane and $\vec{q} = (q_x, q_y)$ satisfies:

$$\begin{aligned} &\{(q_x, q_y) = {}^{2\pi}/_{L_p} \cdot (n_x, n_y)\} \\ &\{n_x, n_y = -{}^{M}/_{2}, \ldots, {}^{M}/_{2} - 1\} \end{aligned} \tag{6.6}$$

In Fourier space the free energy of a symmetric membrane becomes, up to a constant:

$$F_{symmetric} \approx \sigma A_p + \frac{l^2}{2} \sum_{\vec{q}} \left(\sigma q^2 + \kappa q^4\right) |h_{\vec{q}}|^2 \tag{6.7}$$

where $l = {}^{L_p}/_{M}$ is the grid size in both $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ directions and $q = |\vec{q}|$. In the $NP_\perp A_p T$ ensemble, we can then obtain the following relationship between the fluctuations about a flat surface and the thermodynamic moduli σ and κ [156, 158–161]:

$$\left\langle |h_{\vec{q}}|^2 \right\rangle_{NP_\perp A_p T} = \frac{\mathrm{k_B T}}{l^2 \left[\sigma q^2 + \kappa q^4 + \mathcal{O}(q^6)\right]} \tag{6.8}$$

Equation 6.8 provides a useful route for obtaining σ and κ. By averaging the Fourier transform of the membrane height at predefined grid points and fitting to Eq. 6.8 one may extract those thermodynamic moduli in the $NP_\perp A_p T$ ensemble.

Lipid bilayers, however, are typically not simulated in the $NP_\perp A_p T$ ensemble. Constant frame tension, $\gamma = 0$, is typically used to mimic the tensionless state of a biological membrane [42]. In this ensemble, the projected area (A_p), conjugate to the frame tension (γ), is allowed to fluctuate. In that case, it is more convenient to obtain the fluctuation of each undulation mode, $h_{\vec{n}}$; $n = |(n_x, n_y)|$, rather than each wavelength [152].

$$\left\langle |h_{\vec{n}}|^2 \right\rangle_{NP_{\perp}\gamma T} = \left\langle \frac{\mathrm{k_B T} L_p^2}{l^4 \left[\sigma \left({}^{2\pi n}/_{L_p}\right)^2 + \kappa \left({}^{2\pi n}/_{L_p}\right)^4\right]} \right\rangle_{NP_{\perp}\gamma T} \tag{6.9}$$

$$\left\langle \left(\frac{|h_{\vec{n}}|}{L_p}\right)^2 \right\rangle_{NP_{\perp}\gamma T} = M^4 \cdot \left\langle \frac{\mathrm{k_B T}}{\sigma(2\pi n)^2 {L_p}^2 + \kappa(2\pi n)^4} \right\rangle_{NP_{\perp}\gamma T}$$

$$\approx M^4 \cdot \frac{\mathrm{k_B T}}{\sigma(2\pi n)^2 \left\langle L_p \right\rangle^2_{NP_{\perp}\gamma T} + \kappa(2\pi n)^4}$$

If we now define a normalized height function $z_{\vec{n}} = \frac{|h_{\vec{n}}|}{L_p}$ and rename the variables $\kappa' = \kappa(2\pi n)^4$, $\sigma' = \sigma(2\pi)^2 \left\langle L_p \right\rangle^2_{NP_{\perp}\gamma T}$ we obtain the following equation relating the thermodynamic moduli κ, σ to the normalized height fluctuations $z_{\vec{n}}$:

$$\left\langle |z_{\vec{n}}|^2 \right\rangle_{NP_{\perp}\gamma T} \approx M^4 \cdot \frac{\mathrm{k_B T}}{\sigma' n^2 + \kappa' n^4} \tag{6.10}$$

Fitting the normalized membrane height fluctuation as a function of n yields the elastic properties of the membrane. This procedure has been extensively used in constant surface tension simulations of bilayer surfaces [152, 158, 160, 162, 163].

6.2.2 Asymmetric membranes

Bilayers may not be symmetric. In such cases the spontaneous curvature (c_0) may have a finite, non-zero, value. The derivation of Equation 6.8 would not hold under these conditions.

Here we develop the spectral analysis equation for asymmetric bilayers, equivalent to Equation 6.8. We proceed from Equation 6.3, which describes the Helfrich free energy of a nearly flat surface. We note here that the following derivation would therefore only hold for bilayers who are nearly flat as well as displaying non-zero spontaneous curvature, limiting our discussion to membrane displaying small spontaneous curvature.

By omitting terms of order $\mathcal{O}(h^3)$ or higher in Equation 6.3, but *not* assuming zero spontaneous curvature ($c_0 \neq 0$), we obtain the following equation for the free energy of the surface:

$$F_{asymmetric} \approx \sigma A_p + \frac{\sigma}{2}\int_{A_p} dxdy \left|\nabla h\right|^2 + \frac{\kappa}{2}\int_{A_p} dxdy \left|\nabla^2 h\right|^2 \tag{6.11}$$

$$+ 2\kappa {c_0}^2 A_p - 2\kappa c_0 \int_{A_p} dxdy(\nabla^2 h) + \kappa {c_0}^2 \int_{A_p} dxdy \left|\nabla h\right|^2$$

which converges to $F_{symmetric}$ in the limit $c_0 \to 0$, as expected. Using Laplace transform we can again rewrite this equation in Fourier space. The free energy now reads:

$$F_{asymmetric} \approx (\sigma + 2\kappa {c_0}^2)A_p + \frac{l^2}{2}\sum_{\vec{q}} \left[\left(\sigma + 2\kappa {c_0}^2\right) q^2 + \kappa q^4\right] |h_{\vec{q}}|^2 - 2\kappa c_0 h_{\vec{q}} \tag{6.12}$$

The leading term in front of $|h_{\vec{q}}|^2$ now reads $[(\sigma + 2\kappa {c_0}^2)\, q^2 + \kappa q^4]$, compared to $[\sigma q^2 + \kappa q^4]$ in the symmetric case. This means that if we use the spectral analysis described in Section 6.2.1 for asymmetric bilayers, we obtain the same bending modulus κ but a different surface tension:

$$\begin{aligned} \left\langle |h_{\vec{q}}|^2 \right\rangle_{NP_\perp A_p T} &= \frac{1}{l^2} \cdot \frac{\mathrm{k_B T}}{(\sigma + 2\kappa {c_0}^2)\, q^2 + \kappa q^4} \\ &= \frac{1}{l^2} \cdot \frac{\mathrm{k_B T}}{\sigma^* q^2 + \kappa q^4} \end{aligned} \tag{6.13}$$

$$S(n) \equiv \left\langle |z_{\vec{n}}|^2 \right\rangle_{NP_\perp \gamma T} \approx M^4 \cdot \frac{\mathrm{k_B T}}{\left(\sigma' + 2\kappa' {c_0'}^2\right) n^2 + \kappa' n^4} \tag{6.14}$$

where $c_0' = c_0 \cdot {\langle L_p \rangle_{NP_\perp \gamma T}}/{2\pi}$. In practice, the spectrum of undulations will have the same form as in the symmetric bilayer case. When fitting the intensity of height fluctuations as a function of q, one would obtain $\sigma^* = (\sigma + 2\kappa {c_0}^2)$ as the effective surface tension, rather than σ. As $\kappa > 0$ this would always yield an effective surface tension *higher* than the actual surface tension: $\sigma^* \geq \sigma$.

We note that asymmetry in bilayers may arise from having different number of lipids on each leaflet, as well as from having different lipid molecules on each leaflet. Bilayers with a mixture of lipid types may have local patches where the transient composition of lipids in the upper and lower leaflets is different, thereby causing local asymmetric bilayer.

6.3 Methods

In this work we simulated bilayers with different DMPC/Cholesterol ratios and obtained the elastic properties of these membranes. We simulate these systems at their tensionless state [42] in the $NP\gamma T$ ensemble. The temperature is set to $\mathrm{T}^* = 0.35$ $(30°C)$ throughout all simulations.

6.3.1 Size and Time Dependence

The common practice is to simulate a relatively small membrane patch with periodic boundary conditions. Most bilayer properties do not show significant finite-size effects when sampling a system whose linear size is a few lipids long. The long-wavelength undulations of a membrane, however, are constrained by the periodic boundary conditions [164]. To correctly

sample the elastic properties of a membrane, the system size should be large enough to include those long-wavelength undulations that contribute to the bending coeficients. In this work we chose a relatively large membrane patch of $N = 4096$ lipids (DMPC or Cholesterol).

Using a large system size necessitates the use of a long simulation time. Not only because the system is larger but more importantly because the relaxation time of each undulation mode scales with the linear size of the simulated membrane to the fourth power [152]: $\tau_n \propto (L_p/n)^4$. Therefore, to maintain a fair sampling of all undulation modes, we simulate each system for a very long 10^6 cycles. Additionally, we simulate 10 independent samples of each system (10 copies for each of the 10 cholesterol concentration systems). Each DMPC/Cholesterol composition is therefore simulated for what would correspond in real time to a total of 400 μs. These simulation sizes and lengths were made possible because of our novel GPU based algorithm (Chapter 5).

We obtain a snapshot of the membrane at time frame intervals of 2500 DPD sub-steps (see details in Section 6.3.2). To ensure the independence of each sample, we calculate the time correlation of each undulation mode from these snapshots of the membrane. Typical time correlation curves of the lowest undulation modes are presented in Figure 6.2(a). One can clearly see that the relaxation time of the lowest undulation modes ($\vec{n} = (0,1);\ (1,0)$) is indeed the largest. We extract the typical relaxation time by fitting these curves to an exponential decay function. 10% of all simulations displayed a relaxation time of 20 frames or slower. We therefore chose to skip 20 frames (equal to 50000 DPD sub-steps $\simeq 130\ ns$) between every sampled membrane configuration. Figure 6.2(b) shows the time correlation curve of the lowest undulation modes when 20 frames are being skipped. No significant correlation between samples is seen. Simulations that displayed a relaxation time larger than 20 frames were omitted from further analysis.

6.3.2 Determining Membrane Surface

The elastic theory described in Section 6.2 is appropriate for an infinitely thin surface. Lipid bilayers, on the other hand, have finite thickness. A bilayer can perform undulations of both leaflets at fixed thickness (undulatory motion), as well as fluctuations in the thickness itself (peristaltic motion). There will be energetic cost associated with each of these modes of fluctuation, separated by the principle of equipartition [160].

Undulatory motions are defined by the average position of the two monolayer surfaces. We therefore setup a 32x32 grid for each monolayer at each time frame and determine the height of the monolayer at each grid point. The average undulatory surface is then defined as the average of the two monolayer heights at each grid point.

The height of a monolayer at a given grid point is not a well-defined property. One may average the z-position of a specific bead (*e.g.* the first lipid head bead) to determine the local height [161, 165] of the monolayer. Our system, however, contains molecules of two types: DMPC and Cholesterol. Determining the surface of a monolayer based on the bead position of only one of these molecules would result in misrepresentation of the monolayer surface. Instead, we use a method developed for liquid vapor interface analysis [166]. By

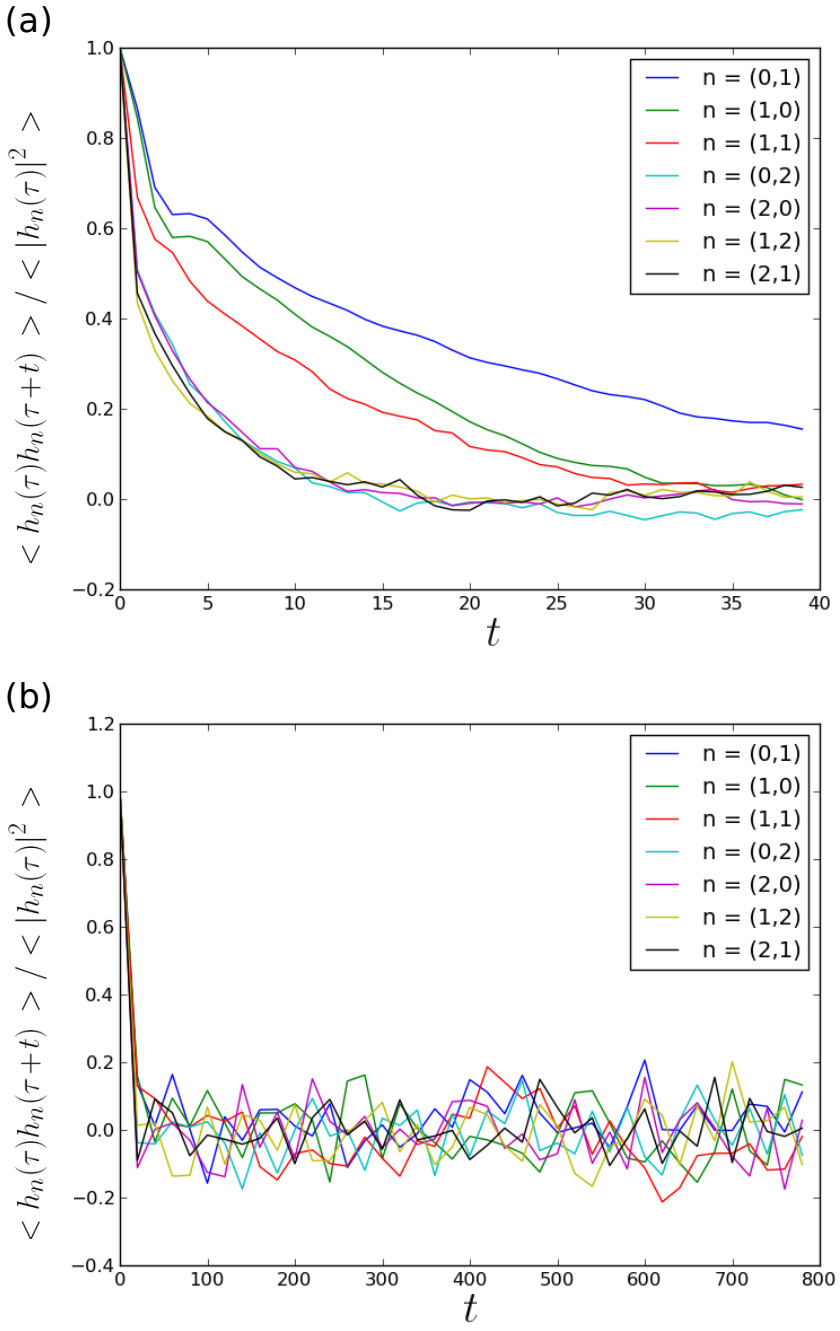


Figure 6.2: Typical correlations in lowest undulation modes of a membrane containing 5% Cholesterol (T=0.35). (a) Shows the time correlation of 7 lowest undulation modes when all time frames are included. (b) Shows the time correlation of 7 lowest undulation modes when skipping 20 time frames. A single time frame here refers to 2500 DPD steps. In both subfigures each color represents a different undulation mode $\vec{n} = (n_x, n_y)$, as described in the legend.

constructing a coarse-grained density field, this method enables one to define a surface that is continuous in space as well as in time. In practice, we search for the surface that corresponds to where the density of hydrophobic beads drops to half its initial value. Since both DMPC and Cholesterol models are defined by the same hydrophobic and hydrophilic bead types, this type of definition incorporates both molecule types into the surface definition. Additionally, the surface is not defined solely by the position of one bead but rather by the density profile of all hydrophobic beads in the molecule.

To determine the surface height of each monolayer at each time frame, we set a gaussian of width $\xi = 1\,\mathrm{d}_0$ around the center of each hydrophobic bead. For each grid point we sum the contribution of all gaussians to the hydrophobic density at $K = 100$ discrete z levels ($z_{i=0\ldots K} = \frac{L_z \cdot i}{K}$). For each grid point $\vec{r}_g$ we then interpolate $\bar{z}(\vec{r}_g)$ as the height at which the hydrophobic density drops to half its maximal value. The set of $\bar{z}(\vec{r}_g)$ for all grid points defines the surface of the monolayer at time frame t. A typical surface construction for both monolayers can be seen in Figure 6.3.

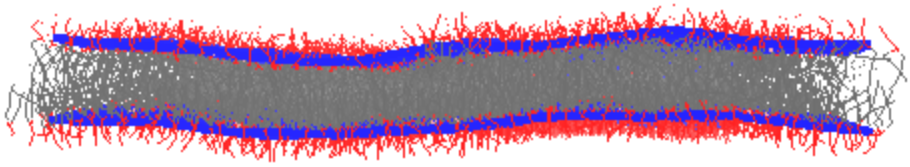


Figure 6.3: Hydrophobic surface construction of a typical configuration of a lipid bilayer containing 5% Cholesterol. Hydrophilic and hydrophobic beads are represented by red and grey colors, respectively. The hydrophobic surface grid for each monolayer, constructed according to the method described by Willard *et al.* [166], is represented in blue. The undulation surface is then determined by the mean of the two monolayer surfaces.

6.3.3 Spectral Analysis

To compute the bending modulus of each membrane, we analyze the fluctuation of the surface and fit the average height fluctuation to Equations 6.10, 6.14. The method described in the previous section provides us with a discretization of the surface height on a $M \times M$ grid. The first step in the analysis is computing the discrete fourier transform of this grid, at each time frame (t), to obtain $|h_{\vec{q}}(t)|^2$. The normalized height function $|z_{\vec{n}}(t)|^2$ is obtained by dividing this quantity by the projected area at each time frame. We average the value of $|z_{\vec{n}}|^2$ for each mode n throughout the simulation. To reduce biases caused by the grid discretization, we divide each mode by a damping factor, following Cooke and Deserno [163].

Plotting the average value of the normalized height function ($S(n) = \left\langle |z_{\vec{n}}|^2 \right\rangle$) as a function of n should then yield the fitting parameters σ', κ', c_0'. We emphasize here that this relation would only hold for the lowest undulation modes. When the wavelength of undulation is on

the order of the bilayer thickness, the elastic theory breaks down as small bilayer protrusions become a dominant factor [160]. Shiba *et al.* [167] have shown that to extract the bending modulus accurately one must fit an inverse power-spectrum, rewriting Equations 6.10, 6.14 to the form:

$$\frac{M^4 \cdot \mathrm{k_B T}}{n^2 S(n)} = \left(\sigma' + 2\kappa' {c_0'}^2\right) + \kappa' n^2 \tag{6.15}$$

where $c_0' = c_0 = 0$ for symmetric membranes. This results in a linear fit, where the intercept is $\sigma' + 2\kappa' {c_0'}^2$ and the slope corresponds to κ'.

6.4 Simulation Results

6.4.1 DMPC-only bilayers

Bilayers containing 0% Cholesterol (DMPC-only bilayers), with the same number and type of lipids in each leaflet and under isotropic stress conditions are expected to be symmetric. We therefore expect the spontaneous curvature to be zero. Additionally, Farago and Pincus [156] have shown that the surface tension extracted from spectral analysis of membrane fluctuations, σ, should be equal to the applied frame tension, γ. Since we simulate these systems with an applied zero frame tension ($\gamma = 0$), we expect $S(n)$ to fit to $S(n) \propto {}^1/_{\kappa' n^4}$.

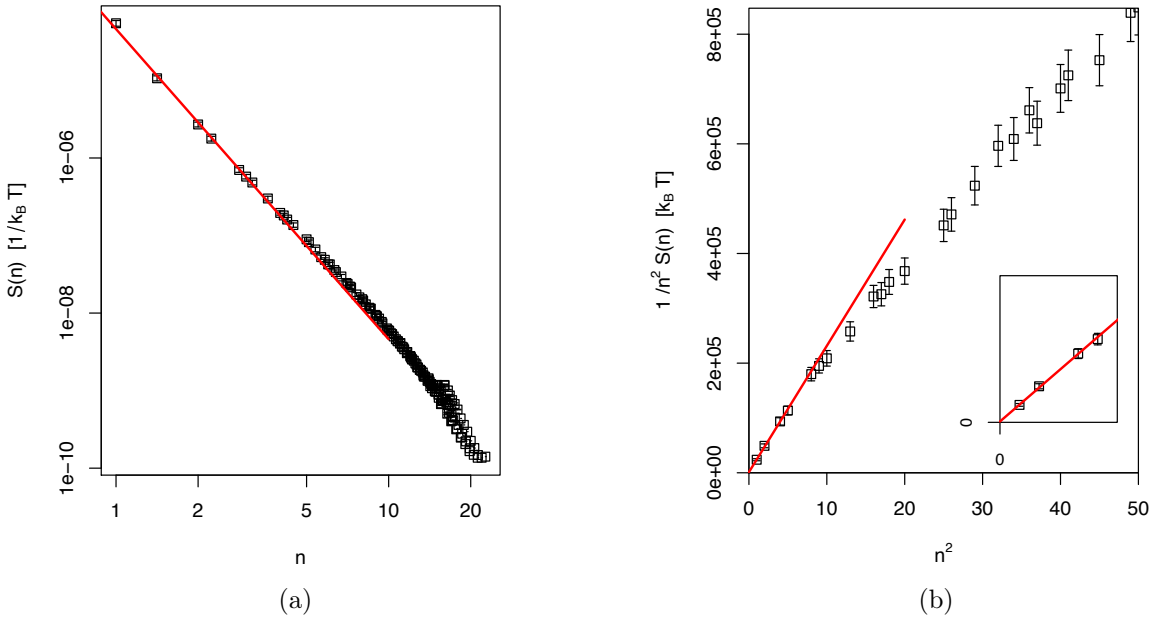
Figure 6.4 shows the average spectrum of undulation for a system of $N = 4096$ DMPC lipids, at constant temperature ($\mathrm{T}^* = 0.35$) and frame tension ($\gamma = 0$). The normal pressure was also kept constant at a value matching the pressure of bulk water. The curve of $S(n)$ as a function of n is displayed in Figure 6.4(a) on a log scale plot. The plot clearly shows the asymptotic fit to $S(n) \propto n^{-4}$ at the lowest undulation modes.

To extract the bending modulus we fit the undulation spectrum to Equation 6.15. This inverse power-spectrum allows one to extract the bending modulus (κ') and the surface tension (σ') as the slope and intercept of a linear fit at the limit of $n \to 0$, respectively. The red line in Figure 6.4(b) shows the best linear fit to the inverse spectrum. This yields, for Cholesterol free membrane, $\kappa = (0.62 \pm 0.02) \cdot 10^{-19}\ J$; $\sigma = 0.00011 \pm 0.00012\ N/m$.

6.4.2 DMPC/Cholesterol bilayers

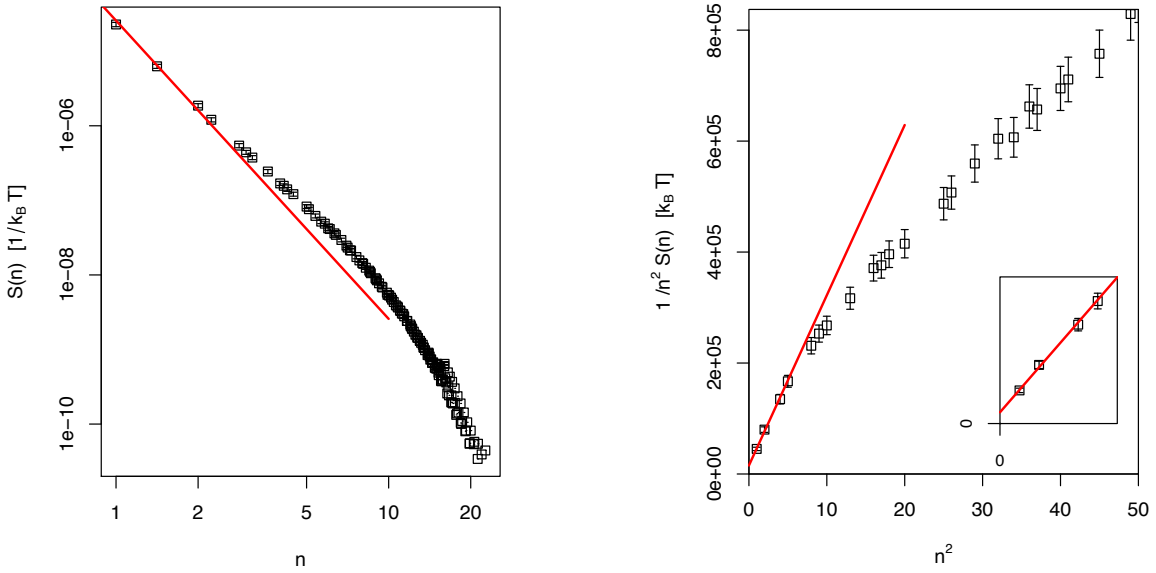
In bilayers containing Cholesterol, there is no guarantee that the bilayer will have zero spontaneous curvature, even in the case that we intitiate our system with an equal number of lipids and cholesterol molecules in both leaflets. In fact, spontaneous curvature is known to be induced in a mixture of lipids with different shapes [168] as a result of a mismatch between the size of the head and tail groups. In our simulations, local transient aggregations of Cholesterol molecules might form local non-zero spontaneous curvature.

Simulations with periodic boundary condition suppress the overall spontaneous curvature of a membrane. The mean membrane height change is constrained to be zero by construction



(a) (b)

Figure 6.4: Spectral analysis of Cholesterol-free membrane (a) Shows a log-log plot of $S(n)$ as a function of n. The red curve represent a n^{-4} slope, which is the theoretical prediction for the asymptotic behavior of $S(n)$. (b) Shows a linear fit (red line) to the inverse power-spectrum $1/n^2 S(n)$ as a function of n^2. According to Eq. 6.15 this should be a linear fit, with a slope of κ' and an intercept of σ'. The fitted surface tension is $\sigma' \approx 0$ as expected.



(a) (b)

Figure 6.5: Spectral analysis of a membrane containing 30% Cholesterol. (a) Shows a log-log plot of $S(n)$ as a function of n. The red curve represent a n^{-4} slope, which is the theoretical prediction for the asymptotic behavior of $S(n)$ in the case of symmetric bilayers. In this asymmetric bilayer there is clear deviation from that slope. (b) Shows a linear fit (red line) to the inverse power-spectrum $1/n^2 S(n)$ as a function of n^2. According to Eq. 6.15 this should be a linear fit, with a slope of κ' and an intercept of $\sigma' + 2\kappa' {c'_0}^2$, where $\sigma' \approx 0$. For asymmetric bilayers with $c'_0 \neq 0$ the intercept is expected to be greater than zero. The fitted intercept in this asymmetric bilayer is indeed non-zero, measuring $0.0017 \pm 0.0004\ N/m$

($\langle h(x,y)\rangle = 0$) in such simulation. However, the heterogeneous nature of the bilayer containing a mixture of lipids may result in local patches where $c_0 \neq 0$ [169]. In those cases we expect to find the bilayer follows the asymmetric free energy described in Section 6.2.2.

We simulated bilayers containing DMPC/Chol mixtures (with 5 - 45 % Chol), with ten copies of each system (see Section 6.3.1 for details). We calculated the normalized height spectrum for each Cholesterol concentration. Figure 6.5 shows these results for systems containing 30% Cholesterol. Even though these systems were simulated at zero frame tension, their characteristic $S(n)$ does not follow the asymptotic n^{-4} slope (Fig. 6.5(a)). This is an indication of a non-zero coefficient of n^2 in Equation 6.14. We claim that this non-zero n^2 coefficient is a result of a non-zero spontaneous curvature term.

We additionally fit the normalized height spectrum to Equation 6.15 and display the fitted line in Figure 6.5(b). The plot clearly shows non-zero intercept in the linear fit, supporting our hypothesis of non-zero spontaneous curvature term. This fit allows one to extract both the bending modulus, κ, as well as the intercept. Since all simulation were performed similarly with $\gamma = 0$, we may assume $\sigma \approx 0$ [156] and extract the spontaneous curvature, c_0, from the intercept.

6.4.3 Bending Modulus and Spontaneous Curvature

Results for extracted bending modulus as a function of Cholesterol concentration are displayed in Figure 6.6. The bending modulus monotonically increases with Cholesterol concentration, as seen experimentally [141–143]. Additionally, we note an inflection point in the bending modulus at %Chol ≈ 30. This inflection point matches the Cholesterol concentration for which we detect a structural phase transition at $T = 30°C$ (See phase diagram in Figure 6.1(b)). Such an increase in bending modulus has also been seen experimentally [141], though comparison with the experimental data shows that this increase is larger in the experiments. (See detailed values for ref. [141] in Table 6.1).

Pan *et al.* [141] hypothesize that the large increase in bending modulus of DMPC bilayers with Cholesterol concentration is related to complexation of lipids and cholesterol. Pan *et al.* assign different complexation rates for different lipid species thereby explaining why the increase in bending modulus with Cholesterol concentration is different for each lipid molecule. They suggest that saturated lipids, such as DMPC, have a higher complexation rate than lipids with unsaturated tails, such as DOPC, which leads to higher ordering of lipids and higher bending rigidity with increasing Cholesterol concentration.

We offer a different explanation, related to the thermodynamic properties of the bilayer. The large change in the bending modulus for Cholesterol concentrations larger than 30% is a result of a change in the structural phase of the system. Namely, a change from the disordered liquid phase, L_α to the ordered L_o phase. Other lipid species, such as DOPC, will display a different phase diagram and will therefore show a different change in the bending modulus for similar Cholesterol concentrations.

Figure 6.7(a) shows the fitted intercept values. We note clear distinction between the bilayer containing no Cholesterol and the bilayers with Cholesterol. The former displays

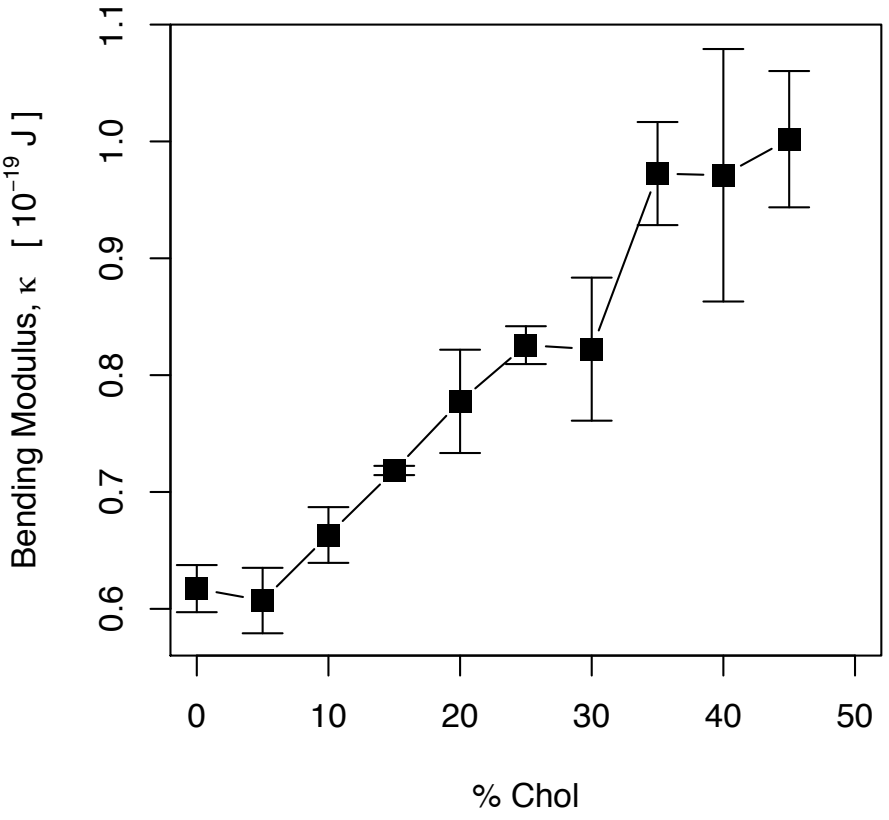
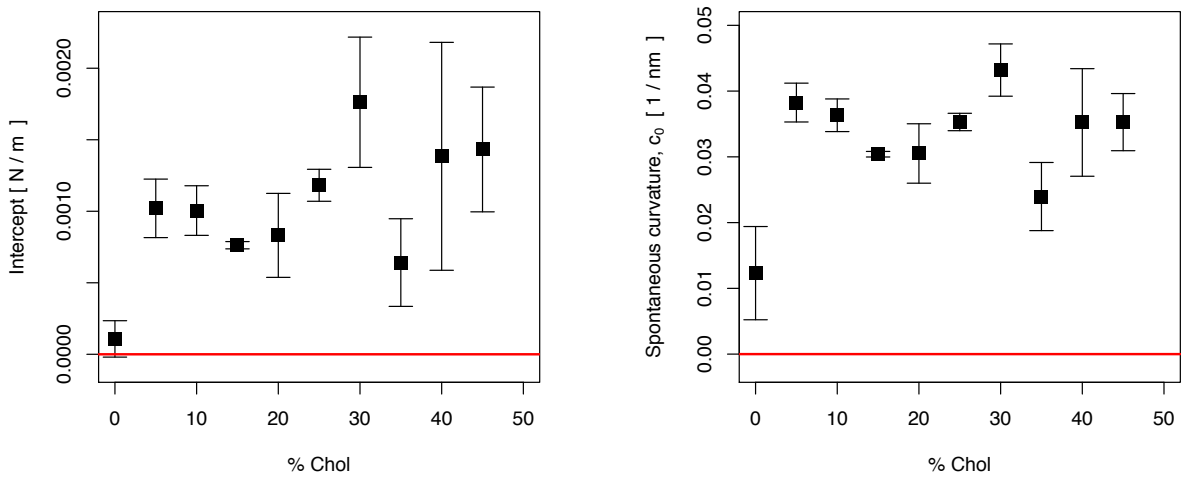


Figure 6.6: Bending modulus as function of Cholesterol concentration at T^*=0.35 (30°C) shows monotonic increase with a larger jump at 30% Cholesterol concentration.



(a) (b)

Figure 6.7: Fitting parameters for Eq. 6.15 for different Cholesterol concentrations. (a) Shows the linear fit intercept. There is a clear distinction between symmetric bilayers containing no Cholesterol, whose intercept is approximately zero and asymmetric bilayers containing Cholesterol whose fitted intercept is greater than 0. (b) Shows the extraction of spontaneous curvature from the fitted intercept, assuming $\sigma = 0$.

a fitted intercept of approximately zero, while the latter show clear non-zero intercept. Extracting the spontaneous curvature, c_0 from the intercept (Figure 6.7(b)) shows the same trend. The Cholesterol-free bilayer is fitted a significantly smaller value than bilayers containing Cholesterol.

It is interesting to note that the extracted c_0 does not seem to monotonically vary with Cholesterol concentration. Wang *et al.* [169] show that if two lipid types are distributed unevenly between both bilayer leaflets, the spontaneous curvature depends linearly on the lipid concentration ratio. Here, however, the number of Cholesterol molecule in both leaflets is identical. The spontaneous curvature created here is not a result of non-matching lipid volume between leaflets, but rather a result of local patches of varying lipid concentration. The number of patches as well as the change in local spontaneous curvature it creates does not necessarily depend linearly on the Cholesterol concentration. We hypothesize that for this reason we see no clear trend in the dependence of c_0 on the Cholesterol concentration.

6.5 Discussion

In this chapter we have calculated the bending modulus of DMPC/Chol bilayers at various cholesterol concentrations. To capture the full spectrum of bilayer undulations, this calculation requires very large systems that are simulated for very long time. Without our GPU algorithm (Chapter 5) such simulations would require a month each. We therefore utilize our newly developed GPU parallel algorithm to perform these simulations in feasible time.

Our results show that bilayers with no Cholesterol fit the classical theory of the Helfrich model. The spectrum of membrane undulations follows an asymptotic $S(n) \propto 1/\kappa n^4$ at $n \to 0$, and the bending modulus can be extracted from fitting this equation. For membranes containing Cholesterol we see a different behavior. Fitting the undulation spectrum of these membranes to the classical model would yield a non-zero surface tension, σ. We extended the Helfrich model for asymmetric membranes and show that this apparent surface tension is a result of a non-zero spontaneous curvature term in the Helfrich model of asymmetric membranes.

The bending modulus extracted from our simulations shows an increase with Cholesterol concentration, as seen in experimental results [141]. Specifically, we see a sharp increase in the value of the bending modulus at 30% Cholesterol concentration. This Cholesterol concentration also corresponds to where a change in the structural properties of the bilayer is seen [146]. We therefore conclude that the sharp increase in bending modulus is associated with a thermodynamic phase transition.

Bibliography

[1] Jacob N Israelachvili, D John Mitchell, and Barry W Ninham. Theory of self-assembly of lipid bilayers and vesicles. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 470 (2):185–201, 1977.

[2] Lipid composition atlas. URL http://opm.phar.umich.edu/atlas.php?membrane=Eukaryotic%20plasma%20membrane.

[3] Demetrios Papahadjopoulos and N Miller. Phospholipid model membranes. i. structural characteristics of hydrated liquid crystals. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 135(4):624–638, 1967.

[4] Michael Edidin. The state of lipid rafts: from model membranes to cells. *Annual review of biophysics and biomolecular structure*, 32(1):257–283, 2003.

[5] Cell membrane. URL http://www.goldiesroom.org/Note%20Packets/03%20Cytology/00%20Cytology--WHOLE.htm.

[6] Almén Markus, Nordström Karl, Fredriksson Robert, and Schiöth Helgi. Mapping the human membrane proteome: a majority of the human membrane proteins can be classified according to function and evolutionary origin. *BMC Biology*, 7, 2009.

[7] Gerald Karp. *Cell and Molecular Biology: Concepts and Experiments 4th Edition with Plus Set*. John Wiley & Son, 2006.

[8] Milton H Saier, Can V Tran, and Ravi D Barabote. Tcdb: the transporter classification database for membrane transport protein analyses and information. *Nucleic acids research*, 34(suppl 1):D181–D186, 2006.

[9] Bertil Hille. *Ion channels of excitable membranes*. Sinauer, Sunderland, MA, 3rd edition, 2001.

[10] Richard O Hynes et al. Integrins: a family of cell surface receptors. *Cell*, 48(4):549, 1987.

[11] A. Holt and J. A. Killian. Orientation and dynamics of transmembrane peptides: the power of simple models. *European Biophysics Journal with Biophysics Letters*, 39(4): 609–621, 2010.

[12] Florian Cymer, Anbazhagan Veerappan, and Dirk Schneider. Transmembrane helix-helix interactions are modulated by the sequence context and by lipid bilayer properties. *Biochimica Et Biophysica Acta-Biomembranes*, 1818(4):963–973, 2012.

[13] O. S. Andersen and R. E. Koeppe. *Bilayer thickness and membrane protein function: An energetic perspective*, volume 36 of *Annual Review of Biophysics*, pages 107–130. Annual Reviews, Palo Alto, 2007.

[14] Frederick Jean-Marie de Meyer, Maddalena Venturoli, and Berend Smit. Molecular simulations of lipid-mediated protein-protein interactions. *Biophysical Journal*, 95(4): 1851–1865, 2008.

[15] Martin Karplus and J Andrew McCammon. Molecular dynamics simulations of biomolecules. *Nature Structural & Molecular Biology*, 9(9):646–652, 2002.

[16] Tomas Hansson, Chris Oostenbrink, and WilfredF van Gunsteren. Molecular dynamics simulations. *Current opinion in structural biology*, 12(2):190–196, 2002.

[17] Erik Lindahl, Berk Hess, and David Van Der Spoel. Gromacs 3.0: a package for molecular simulation and trajectory analysis. *Molecular modeling annual*, 7(8):306–317, 2001.

[18] James C Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D Skeel, Laxmikant Kale, and Klaus Schulten. Scalable molecular dynamics with namd. *Journal of computational chemistry*, 26(16):1781–1802, 2005.

[19] David A Case, Thomas E Cheatham, Tom Darden, Holger Gohlke, Ray Luo, Kenneth M Merz, Alexey Onufriev, Carlos Simmerling, Bing Wang, and Robert J Woods. The amber biomolecular simulation programs. *Journal of computational chemistry*, 26(16): 1668–1688, 2005.

[20] Bernard R Brooks, Charles L Brooks, Alexander D MacKerell, Lennart Nilsson, RJ Petrella, Benoît Roux, Youngdo Won, Georgios Archontis, Christoph Bartels, Stefan Boresch, et al. Charmm: the biomolecular simulation program. *Journal of computational chemistry*, 30(10):1545–1614, 2009.

[21] Steve O Nielsen, Carlos F Lopez, Goundla Srinivas, and Michael L Klein. Coarse grain models and the computer simulation of soft materials. *Journal of Physics: Condensed Matter*, 16(15):R481, 2004.

[22] Siewert J Marrink, H Jelger Risselada, Serge Yefimov, D Peter Tieleman, and Alex H de Vries. The martini force field: coarse grained model for biomolecular simulations. *The Journal of Physical Chemistry B*, 111(27):7812–7824, 2007.

[23] P. J. Bond, J. Holyoake, A. Ivetac, S. Khalid, and M. S. P. Sansom. Coarse-grained molecular dynamics simulations of membrane proteins and peptides. *Journal of Structural Biology*, 157(3):593–605, 2007.

[24] Valentina Tozzini. Coarse-grained models for proteins. *Current opinion in structural biology*, 15(2):144–150, 2005.

[25] M. Venturoli, B. Smit, and M. M. Sperotto. Simulation studies of protein-induced bilayer deformations, and lipid-induced protein tilting, on a mesoscopic model for lipid bilayers with embedded proteins. *Biophysical Journal*, 88(3):1778–1798, 2005.

[26] Ayelet Benjamini and Berend Smit. Lipid mediated packing of transmembrane helices–a dissipative particle dynamics study. *Soft Matter*, 9(9):2673–2683, 2013.

[27] Ayelet Benjamini and Berend Smit. Robust driving forces for transmembrane helix packing. *Biophysical Journal*, 103:1227–1235, 2012.

[28] Kurt Binder and Dieter W Heermann. *Monte Carlo simulation in statistical physics: an introduction*. Springer, 2010.

[29] David P Landau and Kurt Binder. *A guide to Monte Carlo simulations in statistical physics*. Cambridge university press, 2005.

[30] Dennis C Rapaport. *The art of molecular dynamics simulation*. Cambridge university press, 2004.

[31] Hans C Andersen. Rattle: A âĂIJvelocityâĂİ version of the shake algorithm for molecular dynamics calculations. *Journal of Computational Physics*, 52(1):24–34, 1983.

[32] Gang Zou and Robert D Skeel. Robust biased brownian dynamics for rate constant calculation. *Biophysical journal*, 85(4):2147–2157, 2003.

[33] R. D. Groot and P. B. Warren. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulation. *Journal of Chemical Physics*, 107(11):4423–4435, 1997.

[34] L. Monticelli, S. K. Kandasamy, X. Periole, R. G. Larson, D. P. Tieleman, and S. J. Marrink. The martini coarse-grained force field: Extension to proteins. *Journal of Chemical Theory and Computation*, 4(5):819–834, 2008.

[35] M. Kranenburg, M. Venturoli, and B. Smit. Phase behavior and induced interdigitation in bilayers studied with dissipative particle dynamics. *Journal of Physical Chemistry B*, 107(41):11491–11501, 2003.

[36] I. T. Todorov and W. Smith. *The DLPOLY 4 User Manual*. STFC Daresbury Laboratory, Daresbury, Warrington WA4 4AD, Cheshire, UK, 4.01.0 edition, October 2010.

[37] M. Kranenburg, C. Laforge, and B. Smit. Mesoscopic simulations of phase transitions in lipid bilayers. *Physical Chemistry Chemical Physics*, 6(19):4531–4534, 2004.

[38] PJ Hoogerbrugge and JMVA Koelman. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhysics Letters*, 19:155, 1992.

[39] P Español and P Warren. Statistical mechanics of dissipative particle dynamics. *EPL (Europhysics Letters)*, 30:191–196, 1995.

[40] William C Swope, Hans C Andersen, Peter H Berens, and Kent R Wilson. A computer simulation method for the calculation of equilibrium constants for the formation of physical clusters of molecules: Application to small water clusters. *The Journal of Chemical Physics*, 76:637, 1982.

[41] Jocelyn M Rodgers and Berend Smit. On the equivalence of schemes for simulating bilayers at constant surface tension. *Journal of Chemical Theory and Computation*, 8 (2):404–417, 2012.

[42] YH Zhang, Scott E Feller, Bernard R Brooks, and Richard W Pastor. Computer-simulation of liquid/liquid interfaces. 1. theory and application to octane/water. *Journal of Chemical Physics*, 103(23):10252–10266, 1995.

[43] H Jelger Risselada and Siewert J Marrink. The molecular face of lipid rafts in model membranes. *Proceedings of the National Academy of Sciences*, 105(45):17367–17372, 2008.

[44] Jocelyn M Rodgers, Michael Webb, and Berend Smit. Alcohol solubility in a lipid bilayer: Efficient grand-canonical simulation of an interfacially active molecule. *The Journal of chemical physics*, 132:064107, 2010.

[45] Daan Frenkel and Berend Smit. *Understanding molecular simulation: from algorithms to applications*, volume 1. Academic press, 2001.

[46] Bernd A Berg and Alain Billoire. *Markov Chain Monte Carlo Simulations*. Wiley Online Library, 2004.

[47] RD Groot and KL Rabone. Mesoscopic simulation of cell membrane damage, morphology change and rupture by nonionic surfactants. *Biophysical Journal*, 81(2):725–736, 2001.

[48] Lewis Whitehead, Colin M Edge, and Jonathan W Essex. Molecular dynamics simulation of the hydrocarbon region of a biomembrane using a reduced representation model. *Journal of Computational Chemistry*, 22(14):1622–1633, 2001.

[49] W. N. Green and N. S. Millar. Ion-channel assembly. *Trends in Neurosciences*, 18(6): 280–287, 1995.

[50] Martin J. Lohse. Dimerization in gpcr mobility and signaling. *Current Opinion in Pharmacology*, 10(1):53–58, 2010.

[51] Dermot Cox, Marian Brennan, and Niamh Moran. Integrins as therapeutic targets: lessons and opportunities. *Nature Reviews Drug Discovery*, 9(10):804–820, 2010.

[52] E. Psachoulia, D. P. Marshall, and M. S. P. Sansom. Molecular dynamics simulations of the dimerization of transmembrane alpha-helices. *Accounts of Chemical Research*, 43 (3):388–396, 2010.

[53] Florian Cymer and Dirk Schneider. Transmembrane helix-helix interactions involved in erbb receptor signaling. *Cell Adhesion & Migration*, 4(2):299–312, 2010.

[54] Edwin Li and Kalina Hristova. Receptor tyrosine kinase transmembrane domains function, dimer structure and dimerization energetics. *Cell Adhesion & Migration*, 4 (2):249–254, 2010.

[55] J. Liang. Experimental and computational studies of determinants of membrane-protein folding. *Current Opinion in Chemical Biology*, 6(6):878–884, 2002.

[56] K. G. Fleming, A. L. Ackerman, and D. M. Engelman. The effect of point mutations on the free energy of transmembrane alpha-helix dimerization. *Journal of Molecular Biology*, 272(2):266–275, 1997.

[57] K. R. MacKenzie, J. H. Prestegard, and D. M. Engelman. A transmembrane helix dimer: Structure and implications. *Science*, 276(5309):131–133, 1997.

[58] S. Kim, T. J. Jeon, A. Oberai, D. Yang, J. J. Schmidt, and J. U. Bowie. Transmembrane glycine zippers: Physiological and pathological roles in membrane proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 102(40):14278–14283, 2005.

[59] Neta Sal-Man, Doron Gerber, Itai Bloch, and Yechiel Shai. Specificity in transmembrane helix-helix interactions mediated by aromatic residues. *Journal of Biological Chemistry*, 282(27):19753–19761, 2007.

[60] R. F. S. Walters and W. F. DeGrado. Helix-packing motifs in membrane proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 103 (37):13658–13663, 2006.

[61] S. H. Park and S. J. Opella. Tilt angle of a trans-membrane helix is determined by hydrophobic mismatch. *Journal of Molecular Biology*, 350(2):310–318, 2005.

[62] S. K. Kandasamy and R. G. Larson. Molecular dynamics simulations of model trans-membrane peptides in lipid bilayers: A systematic investigation of hydrophobic mismatch. *Biophysical Journal*, 90(7):2326–2343, 2006.

[63] J. Lee and W. Im. Role of hydrogen bonding and helix-lipid interactions in transmembrane helix association. *Journal of the American Chemical Society*, 130(20):6456–6462, 2008.

[64] Taehoon Kim and Wonpil Im. Revisiting hydrophobic mismatch with free energy simulation studies of transmembrane helix tilt and rotation. *Biophysical Journal*, 99(1): 175–183, 2010.

[65] Erik Strandberg, Santi Esteban-Martin, Anne S. Ulrich, and Jesus Salgado. Hydrophobic mismatch of mobile transmembrane helices: Merging theory and experiments. *Biochimica Et Biophysica Acta-Biomembranes*, 1818(5):1242–1249, 2012.

[66] J. A. Killian, I. Salemink, M. R. R. dePlanque, G. Lindblom, R. E. Koeppe, and D. V. Greathouse. Induction of nonbilayer structures in diacylphosphatidylcholine model membranes by transmembrane alpha-helical peptides: Importance of hydrophobic mismatch and proposed role of tryptophans. *Biochemistry*, 35(3):1037–1045, 1996.

[67] J. H. Ren, S. Lew, J. Y. Wang, and E. London. Control of the transmembrane orientation and interhelical interactions within membranes by hydrophobic helix length. *Biochemistry*, 38(18):5905–5912, 1999.

[68] E. Sparr, W. L. Ash, P. V. Nazarov, D. T. S. Rijkers, M. A. Hemminga, D. P. Tieleman, and J. A. Killian. Self-association of transmembrane alpha-helices in model membranes - importance of helix orientation and role of hydrophobic mismatch. *Journal of Biological Chemistry*, 280(47):39324–39331, 2005.

[69] Lars V. Schaefer, Djurre H. de Jong, Andrea Holt, Andrzej J. Rzepiela, Alex H. de Vries, Bert Poolman, J. Antoinette Killian, and Siewert J. Marrink. Lipid packing drives the segregation of transmembrane helices into disordered lipid domains in model membranes. *Proceedings of the National Academy of Sciences of the United States of America*, 108 (4):1343–1348, 2011.

[70] Daniel L. Parton, Jochen W. Klingelhoefer, and Mark S. P. Sansom. Aggregation of model membrane proteins, modulated by hydrophobic mismatch, membrane curvature, and protein class. *Biophysical Journal*, 101(3):691–699, 2011.

[71] Sarah Yohannan, Salem Faham, Duan Yang, Julian P Whitelegge, and James U Bowie. The evolution of transmembrane helix kinks and the structural diversity of g protein-coupled receptors. *Proceedings of the National Academy of Sciences of the United States of America*, 101(4):959–963, 2004.

[72] David Chandler. Interfaces and the driving force of hydrophobic assembly. *Nature*, 437 (7059):640–647, 2005.

[73] C. O. Pabo and R. T. Sauer. Protein-dna recognition. *Annual Review of Biochemistry*, 53:293–321, 1984.

[74] Carl Branden and John Tooze. *Introduction to protein structure*. Garland Pub., New York, 1991.

[75] F. Jahnig. What is the surface tension of a lipid bilayer membrane? *Biophysical Journal*, 71(3):1348–1349, 1996.

[76] Eric W. Weisstein. Point-line distance–3-dimensional. MathWorld–A Wolfram Web Resource. URL http://mathworld.wolfram.com/Point-LineDistance3-Dimensional.html.

[77] S Chandrasekhar. Liquid crystals. *Press, Cambridge*, 168, 1992.

[78] Dan Sunday. Distance between lines ad segments with cpa. URL http://www.softsurfer.com/Archive/algorithm_0106/algorithm_0106.htm.

[79] Norbert Kucerka, John F. Nagle, Jonathan N. Sachs, Scott E. Feller, Jeremy Pencer, Andrew Jackson, and John Katsaras. Lipid bilayer structure determined by the simultaneous analysis of neutron and x-ray scattering data. *Biophysical Journal*, 95(5): 2356–2367, 2008.

[80] Research collaboratory for structural bioinformatics (rcsb) protein data bank. URL http://www.rcsb.org.

[81] Sunhwan Jo, Taehoon Kim, Vidyashankara G. Iyer, and Wonpil Im. Software news and updates - charnim-gui: A web-based grraphical user interface for charmm. *Journal of Computational Chemistry*, 29(11):1859 – 1865, 2008.

[82] Y SAKAMOTO, M ISHIGURO, and G KITAGAWA. *Akaike Information Criterion Statistics*. D. Reidel., 1986.

[83] John M. Chambers and Trevor Hastie. *Statistical models in S*. Wadsworth & Brooks/Cole computer science series. Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, Calif., 1992.

[84] G. M. Torrie and J. P. Valleau. Non-physical sampling distributions in monte-carlo free-energy estimation - umbrella sampling. *Journal of Computational Physics*, 23(2): 187–199, 1977.

[85] S Kumar, D Bouzida, RH Swendsen, PA Kollman, and JM Rosenberg. The weighted histogram analysis method for free-energy calculations on biomolecules .1. the method. *Journal of Computational Chemistry*, 13(8):1011–1021, 1992.

[86] K. Bohinc, V. Kralj-Iglic, and S. May. Interaction between two cylindrical inclusions in a symmetric lipid bilayer. *Journal of Chemical Physics*, 119(14):7435–7444, 2003.

[87] S May and A Ben-Shaul. Molecular theory of lipid-protein interaction and the l-alpha-h-ii transition. *Biophysical Journal*, 76(2):751–767, 1999.

[88] M. M. Sperotto and O. G. Mouritsen. Mean-field and monte-carlo simulation studies of the lateral distribution of proteins in membranes. *European Biophysics Journal*, 19 (4):157–168, 1991.

[89] W. G. J. Hol. The role of the alpha-helix dipole in protein function and structure. *Progress in Biophysics & Molecular Biology*, 45(3):149–195, 1985.

[90] R. D. Groot. Electrostatic interactions in dissipative particle dynamics-simulation of polyelectrolytes and anionic surfactants. *Journal of Chemical Physics*, 118(24): 11265–11277, 2003.

[91] A Wada. The alpha-helix as an electric macro-dipole. *Advances in Biophysics*, 9:1 – 63, 1976.

[92] D. Sengupta, R. N. Behera, J. C. Smith, and G. M. Ullmann. The alpha helix dipole: Screened out? *Structure*, 13(6):849–855, 2005.

[93] J. L. Popot and D. M. Engelman. Membrane-protein folding and oligomerization - the 2-stage model. *Biochemistry*, 29(17):4031–4037, 1990.

[94] S. H. White and G. von Heijne. The machinery of membrane protein assembly. *Current Opinion in Structural Biology*, 14(4):397–404, 2004.

[95] F. H. C. Crick. The packing of alpha-helices - simple coiled-coils. *Acta Crystallographica*, 6(8-9):689–697, 1953.

[96] D. Langosch and J. Heringa. Interaction of transmembrane helices by a knobs-into-holes packing characteristic of soluble coiled coils. *Proteins-Structure Function and Genetics*, 31(2):150–159, 1998.

[97] F. J. Barrantes. Structural basis for lipid modulation of nicotinic acetylcholine receptor function. *Brain Research Reviews*, 47(1-3):71–95, 2004.

[98] L. E. Cybulski, D. Albanesi, M. C. Mansilla, S. Altabe, P. S. Aguilar, and D. de Mendoza. Mechanism of membrane fluidityoptimization: isothermal control of the bacillus subtilis acyl-lipid desaturase. *Molecular Microbiology*, 45(5):1379–1388, 2002.

[99] P. S. Aguilar, A. M. Hernandez-Arriaga, L. E. Cybulski, A. C. Erazo, and D. de Mendoza. Molecular basis of thermosensing: a two-component signal transduction thermometer in bacillus subtilis. *Embo Journal*, 20(7):1681–1691, 2001.

[100] F. Dumas, J. F. Tocanne, G. Leblanc, and M. C. Lebrun. Consequences of hydrophobic mismatch between lipids and melibiose permease on melibiose transport. *Biochemistry*, 39(16):4846–4854, 2000.

[101] M. A. Lomize, A. L. Lomize, I. D. Pogozheva, and H. I. Mosberg. Opm: Orientations of proteins in membranes database. *Bioinformatics*, 22(5):623–625, 2006.

[102] Juergen Koepke, Eva-Maria Krammer, Astrid R Klingen, Pierre Sebban, G Matthias Ullmann, and Günter Fritzsch. ph modulates the quinone position in the photosynthetic reaction center from rhodobacter sphaeroides in the neutral and charge separated states. *Journal of molecular biology*, 371(2):396–409, 2007.

[103] W. Kabsch and C. Sander. Dictionary of protein secondary structure - pattern-recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

[104] M. Bansal, S. Kumar, and R. Velavan. Helanal: A program to characterize helix geometry in proteins. *Journal of Biomolecular Structure & Dynamics*, 17(5):811–819, 2000.

[105] Naveen Michaud-Agrawal, Elizabeth J. Denning, Thomas B. Woolf, and Oliver Beckstein. Software news and updates mdanalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry*, 32(10):2319–2327, 2011.

[106] M. Gimpelev, L. R. Forrest, D. Murray, and B. Honig. Helical packing patterns in membrane and soluble proteins. *Biophysical Journal*, 87(6):4075–4086, 2004.

[107] C. Chothia, M. Levitt, and D. Richardson. Helix to helix packing in proteins. *Journal of Molecular Biology*, 145(1):215–250, 1981.

[108] A. J. Li and R. Nussinov. A set of van der waals and coulombic radii of protein atoms for molecular and solvent-accessible surface calculation, packing evaluation, and docking. *Proteins-Structure Function and Genetics*, 32(1):111–127, 1998.

[109] Andrei L. Lomize, Irina D. Pogozheva, Mikhail A. Lomize, and Henry I. Mosberg. Positioning of proteins in membranes: A computational approach. *Protein Science*, 15 (6):1318–1333, 2006.

[110] Andrei L. Lomize, Irina D. Pogozheva, and Henry I. Mosberg. Anisotropic solvent model of the lipid bilayer. 2. energetics of insertion of small molecules, peptides, and proteins in membranes. *Journal of Chemical Information and Modeling*, 51(4):930–946, 2011.

[111] J. Kyte and R. F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, 1982.

[112] C. Chothia, M. Levitt, and D. Richardson. Structure of proteins - packing of alpha-helices and pleated sheets. *Proceedings of the National Academy of Sciences of the United States of America*, 74(10):4130–4134, 1977.

[113] J. U. Bowie. Helix packing angle preferences. *Nature Structural Biology*, 4(11):915–917, 1997.

[114] J. U. Bowie. Helix packing in membrane proteins. *Journal of Molecular Biology*, 272 (5):780–789, 1997.

[115] B. Efron and R. Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical Science*, 1(1):54–75, 1986.

[116] David Kirk. Nvidia cuda software and gpu parallel computing architecture. In *International Symposium on Memory Management: Proceedings of the 6 th international symposium on Memory management*, pages 103–104, 2007.

[117] John D Owens, Mike Houston, David Luebke, Simon Green, John E Stone, and James C Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899, 2008.

[118] Paulius Micikevicius. 3d finite difference computation on gpus using cuda. In *Proceedings of 2nd Workshop on General Purpose Processing on Graphics Processing Units*, pages 79–84. ACM, 2009.

[119] Pawan Harish and P Narayanan. Accelerating large graph algorithms on the gpu using cuda. *High Performance Computing–HiPC 2007*, pages 197–208, 2007.

[120] Jens Krüger and Rüdiger Westermann. Linear algebra operators for gpu implementation of numerical algorithms. In *ACM Transactions on Graphics (TOG)*, pages 908–916. ACM, 2003.

[121] Joshua A Anderson, Chris D Lorenz, and Alex Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics*, 227(10):5342–5359, 2008.

[122] Weiguo Liu, Bertil Schmidt, Gerrit Voss, and Wolfgang Müller-Wittig. Accelerating molecular dynamics simulations using graphics processing units with cuda. *Computer Physics Communications*, 179(9):634–641, 2008.

[123] Mark S Friedrichs, Peter Eastman, Vishal Vaidyanathan, Mike Houston, Scott Legrand, Adam L Beberg, Daniel L Ensign, Christopher M Bruns, and Vijay S Pande. Accelerating molecular dynamic simulation on graphics processing units. *Journal of computational chemistry*, 30(6):864–872, 2009.

[124] Carolyn L Phillips, Joshua A Anderson, and Sharon C Glotzer. Pseudo-random number generation for brownian dynamics and dissipative particle dynamics simulations on gpu devices. *Journal of Computational Physics*, 230(19):7191–7201, 2011.

[125] Hao Wu, Junbo Xu, Shengfei Zhang, and Hao Wen. Gpu accelerated dissipative particle dynamics with parallel cell-list updating. *IEIT Journal of Adaptive & Dynamic Computing*, pages 26–32, 2011.

[126] T Rożen, K Boryczko, and W Alda. Gpu bucket sort algorithm with applications to nearest-neighbour search. *Journal of WSCG*, 2008.

[127] Alexander Heinecke. *A short introduction to nvidia's CUDA*. Ferienakademie, 2007.

[128] Nvidia cuda programming guide, version 2.5.0, 2012. URL http://developer.download.nvidia.com/GPU_Programming_Guide/GPU_Programming_Guide.pdf.

[129] MP Allen and DJ Tildesley. Computer simulation of liquids. *New York: Oxford*, 385, 1987.

[130] Loup Verlet. Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. *Physical Review*, 159(1):98, 1967.

[131] Dieter W Heermann and Anthony N Burkitt. Parallelization of the ising model and its performance evaluation. *Parallel Computing*, 13(3):345–357, 1990.

[132] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2001.

[133] CUDA NVIDIA. Curand library, 2010.

[134] P Emmelot. The organization of the plasma membrane of mammalian cells: structure in relation to function. *Mammalian cell membranes*, 2:1–54, 1977.

[135] Myer Bloom, Evan Evans, and Ole G Mouritsen. Physical properties of the fluid lipid-bilayer component of cell membranes: a perspective. *Quarterly reviews of biophysics*, 24(03):293–397, 1991.

[136] Alan Finkelstein. Effect of cholesterol on the water permeability of thin lipid membranes. *Nature*, 216:717–718, 1967.

[137] T Parasassi, M Di Stefano, M Loiero, G Ravagnan, and E Gratton. Cholesterol modifies water concentration and dynamics in phospholipid bilayers: a fluorescence study using laurdan probe. *Biophysical journal*, 66(3):763–768, 1994.

[138] Alexander I Greenwood, Stephanie Tristram-Nagle, and John F Nagle. Partial molecular volumes of lipids and cholesterol. *Chemistry and physics of lipids*, 143(1):1–10, 2006.

[139] Wei-Chin Hung, Ming-Tao Lee, Fang-Yu Chen, and Huey W Huang. The condensing effect of cholesterol in lipid bilayers. *Biophysical journal*, 92(11):3960–3967, 2007.

[140] Julio A Urbina, Sara Pekerar, Hong-biao Le, Jessica Patterson, Bernard Montez, and Eric Oldfield. Molecular order and dynamics of phosphatidylcholine bilayer membranes in the presence of cholesterol, ergosterol and lanosterol: a comparative study using ^{2}H,^{13}C and ^{31}P nmr spectroscopy. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 1238(2):163–176, 1995.

[141] Jianjun Pan, Thalia T Mills, Stephanie Tristram-Nagle, and John F Nagle. Cholesterol perturbs lipid bilayers nonuniversally. *Physical review letters*, 100(19):198103, 2008.

[142] HP Duwe, J Kaes, and E Sackmann. Bending elastic moduli of lipid bilayers: modulation by solutes. *Journal de Physique*, 51(10):945–961, 1990.

[143] Philippe Meleard, Claire Gerbeaud, Tanja Pott, Laurent Fernandez-Puente, Isak Bivas, Marin D Mitov, Jean Dufourcq, and Pierre Bothorel. Bending elasticities of model membranes: influences of temperature and sterol content. *Biophysical journal*, 72(6): 2616–2629, 1997.

[144] Hélène Bouvrais. Bending rigidities of lipid bilayers: their determination and main inputs in biophysical studies. *Advances in Planar Lipid Bilayers and Liposomes*, 15:1, 2012.

[145] DA Brown and E London. Functions of lipid rafts in biological membranes. *Annual review of cell and developmental biology*, 14(1):111–136, 1998.

[146] Frederick J. M. de Meyer, Ayelet Benjamini, Jocelyn M. Rodgers, Yannick Misteli, and Berend Smit. Molecular simulation of the dmpc-cholesterol phase diagram. *Journal of Physical Chemistry B*, 114(32):10451–10461, 2010.

[147] H Engelhardt, HP Duwe, and E Sackmann. Bilayer bending elasticity measured by fourier analysis of thermally excited surface undulations of flaccid vesicles. *Journal de Physique Lettres*, 46(8):395–400, 1985.

[148] G Beblik, R-M Servuss, and W Helfrich. Bilayer bending rigidity of some synthetic lecithins. *Journal de Physique*, 46(10):1773–1778, 1985.

[149] Chau-Hwang Lee, Wan-Chen Lin, and Jyhpyng Wang. All-optical measurements of the bending rigidity of lipid-vesicle membranes across structural phase transitions. *Physical Review E*, 64(2):020901, 2001.

[150] E Evans and W Rawicz. Entropy-driven tension and bending elasticity in condensed-fluid membranes. *Physical Review Letters*, 64(17):2094–2097, 1990.

[151] Amy C Rowat, Per Lyngs Hansen, and John Hjort Ipsen. Experimental evidence of the electrostatic contribution to membrane bending rigidity. *EPL (Europhysics Letters)*, 67 (1):144, 2004.

[152] Oded Farago. Mode excitation monte carlo simulations of mesoscopically large membranes. *The Journal of chemical physics*, 128:184105, 2008.

[153] Wolfgang Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift für Naturforschung. Teil C: Biochemie, Biophysik, Biologie, Virologie*, 28 (11):693, 1973.

[154] WSRM Helfrich and R-M Servuss. Undulations, steric interaction and cohesion of fluid membranes. *Il Nuovo Cimento D*, 3(1):137–151, 1984.

[155] Scott T Milner and SA Safran. Dynamical fluctuations of droplet microemulsions and vesicles. *Physical Review A*, 36(9):4371, 1987.

[156] Oded Farago and Philip Pincus. Statistical mechanics of bilayer membrane with a fixed projected area. *The Journal of chemical physics*, 120:2934, 2004.

[157] W Cai, TC Lubensky, P Nelson, and T Powers. Measure factors, tension, and correlations of fluid membranes. *Journal de Physique II*, 4(6):931–949, 1994.

[158] Zun-Jing Wang and Daan Frenkel. Modeling flexible amphiphilic bilayers: a solvent-free off-lattice monte carlo study. *The Journal of chemical physics*, 122:234711, 2005.

[159] Samuel A Safran. *Statistical thermodynamics of surfaces, interfaces, and membranes*. Addison-Wesley Reading, MA, 1994.

[160] Erik Lindahl and Olle Edholm. Mesoscopic undulations and thickness fluctuations in lipid bilayers from molecular dynamics simulations. *Biophysical Journal*, 79(1):426–433, 2000.

[161] Live Rekvig, Bjørn Hafskjold, and Berend Smit. Simulating the effect of surfactant structure on bending moduli of monolayers. *The Journal of chemical physics*, 120:4897, 2004.

[162] SJ Marrink and AE Mark. Effect of undulations on surface tension in simulated bilayers. *The Journal of Physical Chemistry B*, 105(26):6122–6127, 2001.

[163] Ira R Cooke and Markus Deserno. Solvent-free model for self-assembling fluid bilayer membranes: Stabilization of the fluid phase based on broad attractive tail potentials. *The Journal of chemical physics*, 123:224710, 2005.

[164] Scott E Feller and Richard W Pastor. On simulating lipid bilayers with an applied surface tension: periodic boundary conditions and undulations. *Biophysical journal*, 71 (3):1350–1355, 1996.

[165] Live Rekvig, Bjørn Hafskjold, and Berend Smit. Chain length dependencies of the bending modulus of surfactant monolayers. *Physical review letters*, 92(11):116101, 2004.

[166] Adam P Willard and David Chandler. Instantaneous liquid interfaces. *The Journal of Physical Chemistry B*, 114(5):1954–1958, 2010.

[167] Hayato Shiba and Hiroshi Noguchi. Estimation of the bending rigidity and spontaneous curvature of fluid membranes in simulations. *Physical Review E*, 84(3):031926, 2011.

[168] Ira R Cooke and Markus Deserno. Coupling between lipid shape and membrane curvature. *Biophysical journal*, 91(2):487–495, 2006.

[169] H. Wang, D. Hu, and P. Zhang. Measuring the spontaneous curvature of bilayer membranes by molecular dynamics simulations. *Communications in Computational Physics*, 13(4):1093–1106, 2013.