Pin the Protein on the Membrane: Using Numerical Simulations to Explore Membrane-Protein Systems

Kayla Sapp

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2016

Reading Committee:

Lutz Maibaum, Chair

David Masiello

Joshua Vaughan

Program Authorized to Offer Degree:
Department of Chemistry

University of Washington

Abstract

Pin the Protein on the Membrane:
Using Numerical Simulations to Explore Membrane-Protein Systems

Kayla Sapp

Chair of the Supervisory Committee:
Professor Lutz Maibaum
Department of Chemistry

The structure of the membrane gives rise to important biological phenomena. In this dissertation we focus on the shape of the membrane and its interactions through two different coupling schemes to membrane bound proteins. We develop a model based on the continuum description of the membrane that is used to investigate height-coupled and curvature-coupled membrane-protein systems. For these systems we use coupled Langevin equations to model the dynamics for the Fourier modes of the membrane and protein position in the xy-plane. We investigate similarities and differences in these two coupling schemes in how they alter membrane fluctuations, protein-protein interactions, and lateral protein diffusion. We further expand on previous dynamic membrane-protein models by deriving from the Fokker-Planck formalism a consistent Langevin equations that accurately models the dynamics of systems with protein diffusing along the membrane surface. We investigate how the model differs from previous in the discussion of later protein diffusion.

Table of Contents

List of Figures

Acknowledgments

I would like to thank Professor Lutz Maibaum for the opportunity to work on this research and for his input and guidance through out my time in his research group. I would also like to thank the other members of the Maibaum group for their support. Within the Chemistry Department I have had the support of the members of the Amphiphiliphiles and Theory Suite groups. I would like to thank the members of both for the opportunity to receive input from different audiences and for expanding my knowledge within their fields.

I am grateful for the support of my friends both near and far. I am lucky to have you all in my life. You have been there for me no matter, which has been amazing. Thank you Helen Litz for being my roommate and Seattle Family for five years.

I would like to thank my family for always loving and supporting me. Thanks to my brothers for pushing me to be better. Thank you Mom and Dad for being amazing people and always being there for me, especially in my pursuit of this degree.

Chapter 1

Introduction

1.1 Cell Membranes and Membrane-Protein Interactions

The plasma membrane is the barrier around the cell that that provides the cell's structure and protection. The membrane also mediates the many biological processes that occur within the cell. It is a highly dynamic and complex fluid system; comprised of many different phospholipid molecules, sterol molecules, and proteins. The spatial organization of the membrane plays an important role in the organization of cellular components. The membrane's complexity makes it difficult to study, for this reason it is common to study only its backbone lipid bilayer structure [1]. Lipid bilayers are fluid systems of phospholipids with a hydrophobic core.

Membrane shape deformations have been an area of interest as of late, due to the discovery of membrane sculpting proteins which alter membrane shape [2, 3, 4]. In addition to peripheral proteins causes of membrane deformations include composition heterogeneity, integral proteins, and membrane pinning. This work will focus on two mechanisms of membrane deformation that may lead to membrane-mediated interactions: geometrical constraints and curvature.

Actin binding is a prime example of geometric constraints. Upon binding actin constrains the membrane height which can give rise to the bundling of actin filaments. Suppressing membrane fluctuations is thought to induce weak long-range protein interactions leading to protein organization [5, 6, 7]. The effect of the spectrin network acting as pinning sites

has been investigated to observe the effect of locally suppressing membrane fluctuations [8]. This work will also discuss the effect of scaffolding proteins on the membrane. Membrane-sculpting proteins have been shown to induce local membrane curvature, which may in turn organize proteins on the surface [9, 10, 2, 11, 12, 13, 14].

1.2 Theoretical Methods

Spatial organization of the cell membrane can influence the organization of cellular components. This ordering occurs on length scales on the order of tens to hundred of nanometers, which can not be reached using atomistic or coarse-grained simulations. There have been many theoretical approaches to modeling lipid bilayers [15, 16, 17, 10, 18, 19, 20, 21, 22, 12, 23, 24], however, in order to reach the length and time scales necessary to observe the type of phenomena associated with these coupled membrane-protein systems these methods are not effective.

We use a continuum description of a membrane, coupled to a discrete particle model to reveal membrane-mediated protein-protein interactions. On large length scales, the membrane treated as an elastic sheet and a continuum description becomes valid [1]. Canham and Helfrich have shown that membrane shape fluctuations can be expressed by its geometric properties [25, 26]. With a continuum description of the membrane, we can model length scales that correspond to membrane fluctuations, as well as timescales long enough to achieve accurate statistical sampling. Further details ragarding the Canham-Helfirch model are included in Chapter 2. We also use a discrete particle model to describe the proteins. The details of combining a continuum membrane description with a discrete particle model are discussed in Chapters 3 and 4.

We will discuss using coupled Langevin equations to model the dynamics of the membrane shape and protein position in the xy-plane in Chapters 3 and 4. Chapter 5 discusses how the Langevin equations change when the proteins diffuse along the membrane surface

instead of the plane. Previous work has studied proteins diffusing on the surface having included the metric tensor in the Fokker-Planck equation for the proteins [27, 28]. To ensure detailed balance, however, it is necessary to go through the Fokker-Planck formalism while including the metric tensor for both the protein and the membrane. This technique is suggested in reference [29], however in that work do not continue with applications of the consistent coupled Langevin equations. We will determine these coupled Langevin equations and validate the system with the intention of applying the method to curvature-coupled membrane protein systems.

1.3 Specific Aims

The work in this thesis is divided into three main components. The first is a discussion of height-coupled membrane-protein systems with the proteins diffusing in the xy-plane. The second focuses on curvature-coupled membrane-protein systems with the proteins diffusing in the plane. The third component develops a consistent model for performing coupled membrane-protein dynamics if proteins are diffusing along the membrane surface. This method has been developed with the intention of performing research on curvature-coupled systems with the proteins diffusing on the membrane surface instead of in the plane.

Height-Coupled System

We developed a model with the aim of determining how quenching membrane fluctuations leads to membrane-mediated interactions and a reduction in lateral protein diffusion. The model combines the membrane's elastic behavior, the steric repulsion between proteins, and the pinning of the membrane height to the plane of the proteins. Figure 1.1 displays a schematic of this constraint on membrane fluctuations where the proteins act as a spring that does not allow the membrane to fluctuate away from a height of zero. We incorporate these energy contributions into coupled Langevin equations in order to model the dynamics

of these membrane-protein systems. We will show that a membrane-mediated interaction exists between proteins that is dependent on membrane rigidity. We will also show that the interaction between the membrane and the protein leads to a reduction in protein diffusion. This type of system will be discussed in-depth in Chapter 2.

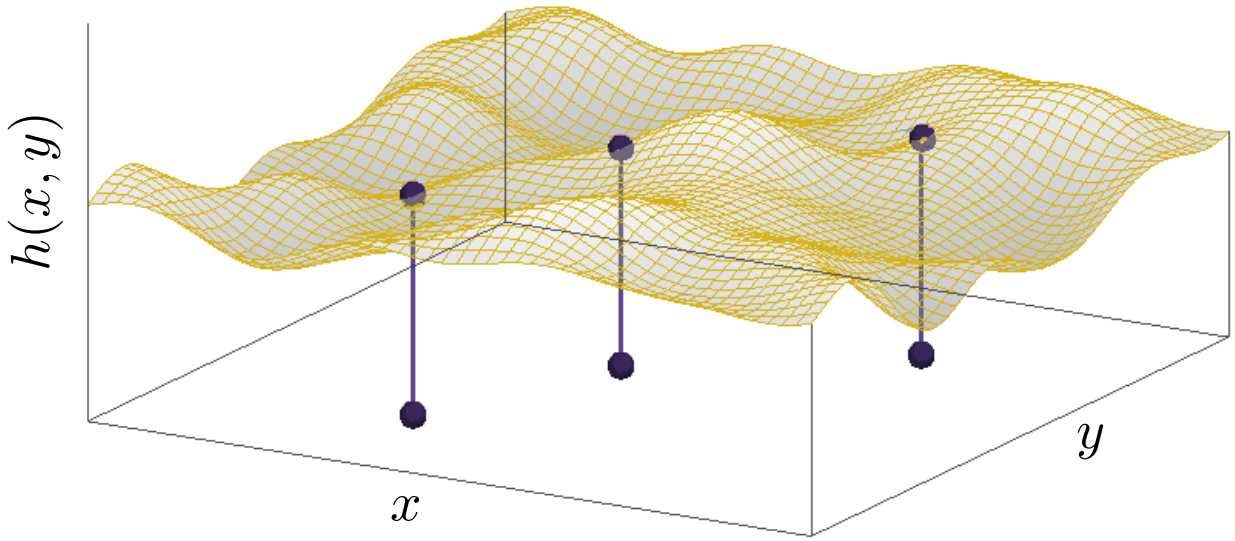

Figure 1.1: A schematic of a membrane in the Monge Gauge (explained in Chapter 2), with proteins represented as springs that are pinning the membrane to a height of zero on the xy-plane.

Curvature-Coupled System

We defined a model that investigates the effect of protein curvature on the dynamics and structure of a membrane-protein system. This model accounts for the elasticity of the membrane, the steric repulsion between proteins, and the local membrane deformation due to the curvature of the protein. Figure 1.2 shows a schematic of this curvature constraint on the membrane where the curvature of the protein is induced onto the membrane at the position of the protein. We use coupled Langevin equations to model the dynamics of these

membrane-protein systems. We will show that the membrane has little effect on the protein-protein interactions. We show that this interaction leads to a reduction in lateral protein diffusion. This interaction will be discussed in detail in Chapter 3.

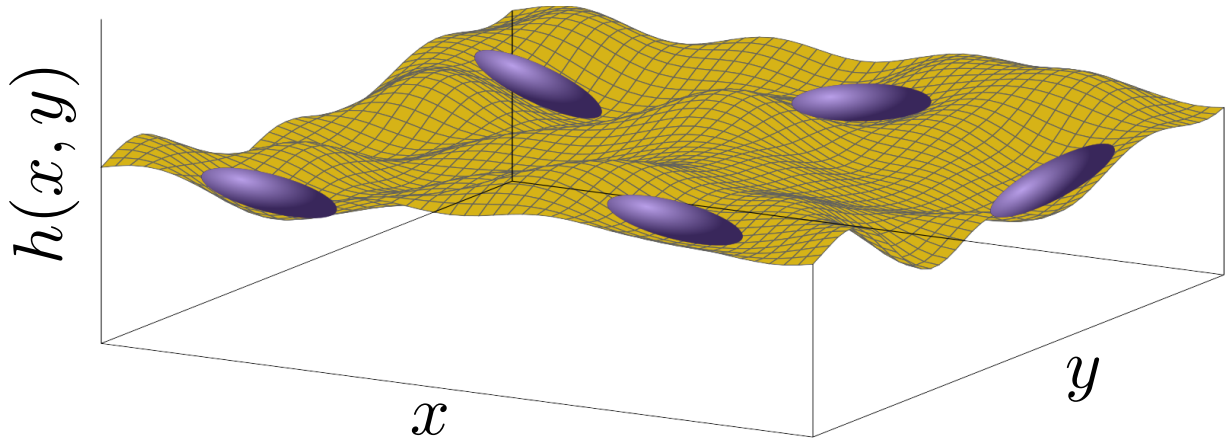

Figure 1.2: A schematic of a membrane in the Monge Gauge, with proteins represented as oblong spheroids inducing a curvature locally on the membrane.

Coupled Membrane-Protein Dynamcis

In order to effectively model protein motion on the surface instead of in the xy-plane we derived a model that builds on previous coupled dynamics models [27, 28]. Our model is consistent in how it handles the metric tensor in the Langevin equations for the membrane and the proteins. We will start from the two simple cases of a membrane with a static protein and a protein on a static surface. We will then formulate coupled dynamics using the Fokker-Planck equation. We will this model and compare it to previously studied methods [27, 28, 29] which do not account for the metric tensor of the membrane. We will also discuss some limitations of this method and how this method differs from previous models. The details of

the development and validation of this method will be discussed in Chapter 4.

Chapter 2

The Canham-Helfrich Model for Membrane Elasticity

In this chapter we will discuss the Canham-Helfrich energy functional that we will use throughout this dissertation to describe the membrane shape. We will also comment on some useful porperites of this model, for example different correlation functions, which can be helpful in understanding the different models we will present.

2.1 The Canham-Helfrich Model

We will start our discussion from the general Canham-Helfrich elastic energy description of the membrane, which defines the membrane as a fluid elastic sheet [25, 26, 1].

$$E_{\mathrm{m}} = \int \mathrm{d}S \left[\frac{\kappa}{2} \left(H - c_0 \right)^2 + \bar{\kappa} K + \sigma \right] \tag{2.1}$$

where H is the mean curvature of the membrane ($R_1^{-1}/2 + R_2^{-1}/2$), where R_1 and R_2 are the principal radii of curvature, and K is the Gaussian curvature($1/R_1R_2$). The three physical parameters κ, $\bar{\kappa}$, and c_0 are the bending rigidity, saddle-splay modulus, and spontaneous curvature of the membrane, respectively. The term containing σ accounts for the effect of surface tension on the membrane which defines surface tension as related to membrane surface area. By mapping equation (2.1) into the Monge Gauge, which represents the 3-D fluid as a 2-D fluid of height, $h(\boldsymbol{r})$, (see figure 2.1 for a description) we can simplify the function to

$$E_{\mathrm{m}} = \int \left[\frac{\sigma}{2} \left(\boldsymbol{\nabla} h(\boldsymbol{r}) \right)^2 + \frac{\kappa}{2} \left(\boldsymbol{\nabla}^2 h(\boldsymbol{r}) \right)^2 \right] \tag{2.2}$$

where $h(\boldsymbol{r})$ is the height of the membrane above the x, y plane. We have truncated equation (2.2) to quadratic order, which is valid if we assume that fluctuations in the height are small. The Gaussian curvature term is omitted due to the Gauss-Bonnet theorem, which states that this term is invariant with topology, and the fact the membrane topology can not change in this model.

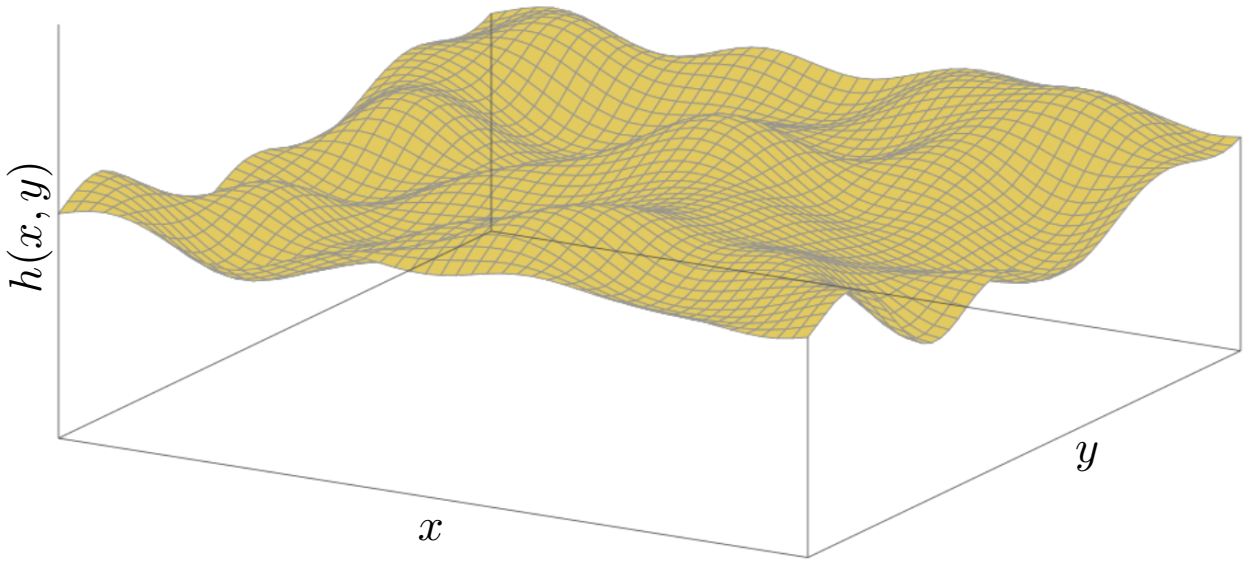

Figure 2.1: A schematic of a membrane defined by equation (2.2) in the Monge Gauge.

We simplify equation (2.2) by transforming it into Fourier space using,

$$h(\boldsymbol{r}) = L^{-2} \sum_{\boldsymbol{k}} \tilde{h}_{\boldsymbol{k}} \, \mathrm{e}^{\imath \boldsymbol{k} \cdot \boldsymbol{r}} \tag{2.3}$$

$$\tilde{h}_{\boldsymbol{k}} = \iint_{L^2} \mathrm{d}\boldsymbol{r} \, h(\boldsymbol{r}) \, \mathrm{e}^{-\imath \boldsymbol{k} \cdot \boldsymbol{r}} \tag{2.4}$$

with wavevectors $\boldsymbol{k} = (p\,2\pi/L, q\,2\pi/L)$ with $p, q \in \mathbb{Z}$. Therefore equation (2.2) becomes

$$E_{\mathrm{m}}\left(\left\{\tilde{h}_{\boldsymbol{k}}\right\}\right) = \frac{1}{L^2}\sum_{\boldsymbol{k}}\left(\frac{\sigma}{2}k^2 + \frac{\kappa}{2}k^4\right)|\tilde{h}_{\boldsymbol{k}}|^2. \tag{2.5}$$

Using the Fourier space representation, equation (2.5), periodic boundary conditions are implied and all Fourier modes become uncoupled. Through out the rest of this dissertation the Fourier coefficients will the dynamical variables of the membrane that evolve in time.

2.2 Useful properties of The Helfrich Model

Here we will discuss the equilibrium properties of the elastic membrane. We need to define the probability distribution of the Fourier components $\{\tilde{h}_{\boldsymbol{k}}\}$, which is the Boltzmann distribution

$$P(\{\tilde{h}_{\boldsymbol{k}}\}) = \frac{1}{Q}\,\mathrm{e}^{-\beta E_{\mathrm{m}}(\{\tilde{h}_{\boldsymbol{k}}\})}, \tag{2.6}$$

where Q is the partition function:

$$Q = \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\,\mathrm{e}^{-\beta E_{\mathrm{m}}(\{\tilde{h}_{\boldsymbol{k}}\})}. \tag{2.7}$$

The pre factor in the exponential, β, is $1/k_{\mathrm{B}}T$, where k_{B} is the Boltzmann constant. This integral is taken over all possible complex values for every Fourier coefficient $\tilde{h}_{\boldsymbol{k}}$. However, we realize that $\tilde{h}_{\boldsymbol{k}} = \tilde{h}^*_{-\boldsymbol{k}}$ due to Hermitian symmetry, which decreases the amount of modes to consider by a factor of two.

The fluctuation spectrum for an elastic membrane can be computed from the equipartition theorem

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle = \frac{k_{\mathrm{B}}TL^2}{\sigma k^2 + \kappa k^4}. \tag{2.8}$$

It is useful to note that

$$\left\langle \tilde{h}_{\boldsymbol{k}}\tilde{h}_{\boldsymbol{k}'}\right\rangle = \frac{k_{\mathrm{B}}TL^2}{\sigma k^2 + \kappa k^4}\delta_{\boldsymbol{k},-\boldsymbol{k}'} \tag{2.9}$$

is true as well.

In later chapters we will discuss how this changes due to different membrane-protein coupling schemes. We will show how the fluctuation spectrum can be calculated for stationary proteins numerically using matrix inversion. Within this chapter, however, we will assume a freely-fluctuating membrane. The rest of this section explores the different correlation functions that are consequences of this elastic membrane model.

2.2.1 Height-Height Correlation Function

The correlation between the membrane height at two different points on the membrane is expressed as

$$\langle h(\boldsymbol{r})\, h(\boldsymbol{r}')\rangle = \frac{1}{L^4}\sum_{\boldsymbol{k}}\sum_{\boldsymbol{k}'} \mathrm{e}^{\imath \boldsymbol{k}\cdot\boldsymbol{r}}\, \mathrm{e}^{\imath \boldsymbol{k}'\cdot\boldsymbol{r}'} \left\langle \tilde{h}_{\boldsymbol{k}}\tilde{h}_{\boldsymbol{k}'}\right\rangle . \tag{2.10}$$

Replacing the average on the right side of this equation with equation (2.9), equation (2.10) becomes:

$$\langle h(\boldsymbol{r})\, h(\boldsymbol{r}')\rangle = \frac{k_{\mathrm{B}}T}{L^2}\sum_{\boldsymbol{k}} \mathrm{e}^{\imath \boldsymbol{k}\,(\boldsymbol{r}-\boldsymbol{r}')}\frac{1}{\sigma k^2+\kappa k^4}. \tag{2.11}$$

To determine the distance dependence of the correlation it is necessary to replace the sum in equation (2.11) with an integral over $\boldsymbol{k}$

$$\langle h(\boldsymbol{0})\, h(\boldsymbol{r})\rangle = \frac{k_{\mathrm{B}}T}{4\pi^2}\int \mathrm{d}\boldsymbol{k}\frac{1}{\sigma k^2+\kappa k^4}\,\mathrm{e}^{-\imath \boldsymbol{k}\cdot\boldsymbol{r}} \tag{2.12}$$

$$= \frac{k_{\mathrm{B}}T}{2\pi}\int_0^\infty \mathrm{d}k\frac{J_0(kr)}{\sigma k+\kappa k^3}. \tag{2.13}$$

This integral clearly diverges at $k\to 0$ so we need to introduce cutoffs for k, $k_{\min}=2\pi/L$ and $k_{\max}=2\pi/a$, where a is a chosen length scale.

We now need to find this expression if $\boldsymbol{r} = \boldsymbol{r}'$

$$\langle h(\boldsymbol{r})\, h(\boldsymbol{r})\rangle = \left\langle h^2\right\rangle = \frac{k_{\mathrm{B}}T}{4\pi\sigma}\log\frac{4\pi^2\kappa + \sigma L^2}{4\pi^2\kappa + \sigma a^2} \tag{2.14}$$

To find this expression when $\boldsymbol{r} \neq \boldsymbol{r}'$ is much more complicated and therefore we solve it for the simplified case when $\sigma = 0$ and find

$$\frac{\langle h(\mathbf{0})\, h(\boldsymbol{r})\rangle}{\langle h^2\rangle} = 1 + \frac{\frac{\pi^4 r^4}{4}\left(\frac{{}_2F_3\left(1,1;2,3,3;-\pi^2r^2/a^2\right)}{a^2} - \frac{{}_3F_2\left(1,1;2,3,3;-\pi^2r^2/L^2\right)}{L^2}\right) - 2\pi^2r^2\log\frac{L}{a}}{L^2 - a^2}. \tag{2.15}$$

In this result ${}_2F_3$ is a generalized hypergeometric function and r is the distance. We then take the limit as $k_{max} \to \infty$

$$\frac{\langle h(\mathbf{0})\, h(\boldsymbol{r})\rangle}{\langle h^2\rangle} = 1 - \frac{\pi^4 r^4}{4L^4}{}_2F_3\left(1,1;2,3,3;-\pi^2r^2/L^2\right) + \frac{2\pi^2r^2}{L^2}\left[\log\frac{\pi r}{L} + \gamma - 1\right]. \tag{2.16}$$

In this expressionγ is Euler's constant. The dependance on distance in this correlation function appears as r/L, therefore the correlation length is related to system size.

2.2.2 Gradient-Gradient Correlation Function

Similar to equation (2.10), we can write down an expression for $\langle \boldsymbol{\nabla} h(\boldsymbol{r})\, \boldsymbol{\nabla} h(\boldsymbol{r}')\rangle$:

$$\langle \boldsymbol{\nabla} h(\boldsymbol{r})\, \boldsymbol{\nabla} h(\boldsymbol{r}')\rangle = \frac{1}{L^4}\sum_{\boldsymbol{k}}\sum_{\boldsymbol{k}'} \boldsymbol{k}\cdot\boldsymbol{k}'\, \mathrm{e}^{\imath\boldsymbol{k}\cdot\boldsymbol{r}}\, \mathrm{e}^{\imath\boldsymbol{k}'\cdot\boldsymbol{r}'}\left\langle \tilde{h}_{\boldsymbol{k}}\tilde{h}_{\boldsymbol{k}'}\right\rangle \tag{2.17}$$

$$= \frac{k_{\mathrm{B}}T}{L^2}\sum_{\boldsymbol{k}} \mathrm{e}^{\imath\boldsymbol{k}(\boldsymbol{r}-\boldsymbol{r}')}\frac{k^2}{\sigma k^2 + \kappa k^4}. \tag{2.18}$$

We once again rewrite this sum as an integral:

$$\langle \boldsymbol{\nabla} h\left(\mathbf{0}\right) \boldsymbol{\nabla} h\left(\boldsymbol{r}'\right)\rangle = \frac{k_{\mathrm{B}}T}{4\pi^2}\int \mathrm{d}\boldsymbol{k}\frac{k^2}{\sigma k^2+\kappa k^4}\,\mathrm{e}^{-\imath \boldsymbol{k}\,\cdot\,\boldsymbol{r}} \tag{2.19}$$

$$= \frac{k_{\mathrm{B}}T}{2\pi}\int_0^\infty \mathrm{d}k\frac{kJ_0\left(kr\right)}{\sigma+\kappa k^2}. \tag{2.20}$$

We now need to find this expression if $\boldsymbol{r}=\boldsymbol{r}'$

$$\left\langle \left(\boldsymbol{\nabla} h\right)^2\right\rangle = \frac{k_{\mathrm{B}}T}{4\pi\kappa}\log\left(\frac{L^2}{a^2}\frac{4\pi^2\kappa+\sigma a^2}{4\pi^2\kappa+\sigma L^2}\right). \tag{2.21}$$

As we did in the previous section, we want to determine this for when $\boldsymbol{r}\neq\boldsymbol{r}'$. When $\sigma=0$ we find

$$\frac{\langle \boldsymbol{\nabla} h\left(\mathbf{0}\right) \boldsymbol{\nabla} h\left(\boldsymbol{r}\right)\rangle}{\left\langle \left(\boldsymbol{\nabla} h\right)^2\right\rangle} = 1+\frac{\pi^2 r^2\left[\frac{{}_2F_3\left(1,1;2,2,2;-\pi^2r^2/L^2\right)}{L^2}-\frac{{}_2F_3\left(1,1;2,2,2;-\pi^2r^2/a^2\right)}{a^2}\right]}{2\log\frac{L}{a}}. \tag{2.22}$$

As before ${}_2F_3$ is a hypergeometric function. By taking the limit as $k_{max}\to\infty$ we find

$$\frac{\langle \boldsymbol{\nabla} h\left(\mathbf{0}\right) \boldsymbol{\nabla} h\left(\boldsymbol{r}\right)\rangle}{\left\langle \left(\boldsymbol{\nabla} h\right)^2\right\rangle} = \begin{cases} 1 & r=0 \\ 0 & r\neq 0. \end{cases} \tag{2.23}$$

This implies that the gradient is completely uncorrelated once the distance, r, is greater than zero.

2.2.3 Curvature-Curvature Correlation Function

The curvature of the membrane, to quadratic order, is:

$$H = \frac{1}{2}\nabla^2 h. \tag{2.24}$$

We can therefore calculate the curvature-curvature correlation function

$$\langle H(\boldsymbol{r})\, H(\boldsymbol{r}')\rangle = \frac{1}{4}\left\langle \nabla^2 h(\boldsymbol{r})\, \nabla^2 h(\boldsymbol{r}')\right\rangle \tag{2.25}$$

$$= \frac{1}{4L^4}\sum_{\boldsymbol{k}}\sum_{\boldsymbol{k}'} k^2 k'^2\, \mathrm{e}^{\imath \boldsymbol{k}\cdot\boldsymbol{r}}\, \mathrm{e}^{\imath \boldsymbol{k}'\cdot\boldsymbol{r}'}\left\langle \tilde{h}_{\boldsymbol{k}}\tilde{h}_{\boldsymbol{k}'}\right\rangle \tag{2.26}$$

$$= \frac{k_{\mathrm{B}}T}{4L^2}\sum_{\boldsymbol{k}} \mathrm{e}^{\imath \boldsymbol{k}(\boldsymbol{r}-\boldsymbol{r}')}\frac{k^4}{\sigma k^2 + \kappa k^4}. \tag{2.27}$$

By replacing the sum with an integral,

$$\langle H(\boldsymbol{r})\, H(\boldsymbol{r}')\rangle = \frac{k_{\mathrm{B}}T}{8\pi}\int_0^\infty \mathrm{d}k \frac{k^3 J_0(k|\boldsymbol{r}-\boldsymbol{r}'|)}{\sigma + \kappa k^2}. \tag{2.28}$$

We now need to find this expression if $\boldsymbol{r} = \boldsymbol{r}'$

$$\left\langle H^2\right\rangle = \frac{k_{\mathrm{B}}T}{16\pi\kappa^2}\left[\sigma \log\left(\frac{4\pi^2\kappa + \sigma L^2}{4\pi^2\kappa + \sigma a^2}\right) + \frac{4\pi^2\kappa\left(L^2 - a^2\right)}{a^2 L^2} - 2\sigma\log\left(\frac{L}{a}\right)\right]. \tag{2.29}$$

To determine the distance dependence of this correlation function we simplify the problem by setting $\sigma = 0$ and find

$$\frac{\langle H(\boldsymbol{0})\, H(\boldsymbol{r})\rangle}{\langle H^2\rangle} = \frac{aJ_1(2\pi r/L) - aLJ_1(2\pi r/a)}{\left(a^2 - L^2\right)\pi r}. \tag{2.30}$$

We then take the limit as $k_{max} \to \infty$

$$\frac{\langle H(\boldsymbol{0})\, H(\boldsymbol{r})\rangle}{\langle H^2\rangle} = \begin{cases} 1 & r = 0 \\ 0 & r \neq 0. \end{cases} \tag{2.31}$$

This reveals that the membrane curvature is completely uncorrelated with distance.

These are the most obvious correlation functions to determine. However, there may be more correlation functions that could reveal useful properties of this membrane model. To

do this, similar methods can be employed..

Chapter 3

Suppressing Membrane Height Fluctuations Leads to a Membrane–Mediated Interaction Among Proteins[1]

3.1 Introduction

Cellular membranes, such as the plasma membrane, play a critical role for the stability of a cell and the multitude of biochemical processes that occur therein. Their principal structure is a bilayer of phospholipid and sterol molecules that hosts a large variety of integral and peripheral membrane proteins. The spatial organization of these membranes has been a topic of great interest because it determines which proteins are in sufficiently close proximity to interact with one another, which in turn is a prerequisite for cellular processes such as signaling.

The structure of a membrane is determined by the interactions between proteins and the lipid bilayer. While the specifics of this interaction depend on molecular-level details of the involved molecules, the principal effects of a broad range of proteins on membrane organization can be understood by considering how binding to a membrane affects its shape and shape fluctuations. For example, earlier experimental work shows protein crowding can lead to membrane bending [31, 32]. It has also been shown both in experiment [2, 11, 33, 14] and theory [9, 19, 10, 12] that scaffolding proteins, which induce and/or sense local membrane curvature, can cause large-scale membrane deformations with concomitant lateral organization of the proteins. Similarly, actin filaments that exert a force on the membrane

---

[1]This chapter is reproduced from [30]; Sapp, K and Maibaum, L."Suppressing Membrane Height Fluctuations Leads to a Membrane–Mediated Interaction Among Proteins", *Physical Review E*, 94(5):052414.

due to polymerization can cause membrane deformations, which in turn induce a compressive force on the filaments that can lead to their bundling [34]. The dominant physical features of these illustrative examples are due to a coupling between protein degrees of freedom and the local membrane shape in the vicinity of the proteins.

The nature of this coupling depends on the type of protein under consideration. Important earlier work has focused on modeling transmembrane proteins as disk-shaped inclusions in elastic membranes, which impose geometric boundary conditions at the protein–membrane interface or that locally alter the membrane's elastic constants. These couplings result in an effective interaction between proteins that scales as $1/r^4$ where r is the distance between two such inclusions [5, 6, 7]. Others have considered adhesion sites that pin a membrane to a surface [35], and found a pair-wise interaction that depends logarithmically on the separation between adhesion sites. Typically these are many-body interactions, i.e., they cannot be decomposed into a sum of pairwise contributions [36].

In this work we consider proteins that impose local constraints on the position of the membrane. This scenario arises, for example, when the membrane is coupled to other components of the cellular environment. For example, actin filaments that are anchored in the cytosolic actin network and whose tips are in contact with the membrane effectively suppress shape fluctuations of the membrane. In related work, Speck and coworkers considered the effect of membrane pinning to a solid substrate, and showed that the quenching of membrane fluctuations gives rise to an attractive interaction between adhesion sites [37, 38].

Spatial organization of the combined membrane/protein system can occur over length scales of hundreds of nanometers to several micrometers. This makes modeling such processes at atomistic resolution unfeasible, and therefore many coarse-grained approaches have been suggested in the literature, see for example Refs. [15, 16, 17, 10, 18, 19, 20, 21, 22, 12, 23, 24]. Our approach combines a continuum description of the membrane and a discrete particle model for the proteins. The former is based on the classic description of membrane

elasticity due to Canham and Helfrich [25, 26], which we simulate using the Fourier Space Brownian Dynamics method developed by Brown and coworkers [39, 8, 1]. Proteins are treated as spherical objects with non-specific repulsive interactions. The coupling between both subsystems is a simple harmonic potential that centers the membrane position to a constant value at the position of the proteins. The model is described in detail in the following section. We simulate the dynamics of our model, which is governed by a set of coupled Langevin equations. In Section 3.3 we present our findings for membrane fluctuation spectra, protein-protein correlation functions, and protein diffusion constants. We show that our model predicts the emergence of a fluctuation-induced, membrane-mediated attraction between proteins that influences the spatial organization of the membrane–protein system.

3.2 The Model

We construct our model by considering the effects of membrane elasticity, protein–protein interactions, and membrane-protein coupling. The total energy E of the system for a given configuration is the sum of these three contributions.

3.2.1 Membrane Elasticity

We use the elastic model developed by Canham and Helfrich to capture the effect of membrane shape deformations [25, 26, 40]. This model views the membrane as a two-dimensional surface whose energetics are determined by its elastic material constants. We limit ourselves to membrane shapes that can be parametrized by a single scalar function $h(\boldsymbol{r})$ that represents the height deviation of the membrane from a flat reference plane. The membrane is located at the positions $(\boldsymbol{r}, h(\boldsymbol{r}))$, where $\boldsymbol{r} = (x, y) \in [0, L]^2$, and L is the side length of the simulation box. If membrane deformations are small one can expand the Canham-Helfrich

energy to quadratic order in h and its derivatives, and obtains

$$E_{\mathrm{m}}[h(\boldsymbol{r})] = \int_{L^2} \mathrm{d}\boldsymbol{r} \left[\frac{\sigma}{2} \left(\boldsymbol{\nabla} h(\boldsymbol{r}) \right)^2 + \frac{\kappa}{2} \left(\boldsymbol{\nabla}^2 h(\boldsymbol{r}) \right)^2 \right] \tag{3.1}$$

as the energy functional of a free membrane. The parameters describing the elastic properties are the surface tension σ and the bending rigidity κ. We assume that the membrane is symmetric so that there is no spontaneous curvature. We have omitted the contribution of Gaussian curvature to the energy, which is a constant offset if the membrane topology is closed and fixed, as is the case for our model.

Equation (3.1) can be simplified by expanding $h(\boldsymbol{r})$ in a plane wave basis, $h(\boldsymbol{r}) = L^{-2} \sum_{\boldsymbol{k}} \tilde{h}_{\boldsymbol{k}} \mathrm{e}^{i\boldsymbol{k}\cdot\boldsymbol{r}}$. The coefficients $\tilde{h}_{\boldsymbol{k}} = \iint_{L^2} \mathrm{d}\boldsymbol{r}\, h(\boldsymbol{r}) \mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r}}$ are the Fourier transform of $h(\boldsymbol{r})$, and the wavevector is of the form $\boldsymbol{k} = (p\,2\pi/L, q\,2\pi/L)$ with $p, q \in \mathbb{Z}$. With these definitions we can express the membrane energy as

$$E_{\mathrm{m}} \left(\left\{ \tilde{h}_{\boldsymbol{k}} \right\} \right) = \frac{1}{L^2} \sum_{\boldsymbol{k}} \left(\frac{\sigma}{2} k^2 + \frac{\kappa}{2} k^4 \right) |\tilde{h}_{\boldsymbol{k}}|^2. \tag{3.2}$$

The advantage of this representation is that different Fourier modes are uncoupled, which allows us to derive several useful results analytically.

In our simulations we treat the real and imaginary parts of the Fourier coefficients $\tilde{h}_{\boldsymbol{k}}$ as the dynamical variables that evolve in time. To do that we have to consider two modifications to this equation [41]. First, we must limit ourselves to a finite basis set, which we do by considering only wavevectors $\boldsymbol{k}$ with $|p| \leq P$ and $|q| \leq Q$. The numbers P and Q determine the magnitude of the largest wavevector (or the smallest lengthscale) that our model can resolve. Second, due to the symmetry relationship $\tilde{h}_{\boldsymbol{k}} = \tilde{h}^*_{-\boldsymbol{k}}$ only half of those Fourier coefficients are independent. Ignoring the $\boldsymbol{k} = (0, 0)$ mode, we choose the Fourier coefficients with wavevectors $\{(p2\pi/L, q2\pi/L) : (1 \leq p \leq P, q = 0) \,\mathrm{or}\, (-P \leq p \leq P, 1 \leq q \leq Q)\}$ as independent degrees of freedom, and refer to this set as $\boldsymbol{k} > 0$. Expressed in terms of these

modes, the membrane energy (3.2) becomes

$$E_{\mathrm{m}}\left(\left\{\tilde{h}_{\boldsymbol{k}}\right\}\right)=\frac{1}{L^2}\sum_{\boldsymbol{k}>0}\left(\sigma k^2+\kappa k^4\right)|\tilde{h}_{\boldsymbol{k}}|^2. \tag{3.3}$$

Unlike some previous work [39, 8, 27, 42], our approach does allow the imaginary part of the Fourier coefficients at the Nyquist modes to be non-zero, thereby preserving translational invariance even for those high-wavevector modes [41].

3.2.2 Protein-Protein Interactions

The interaction among multiple proteins is a complicated function that in principle depends on the type, molecular structure, orientation, and conformation of the biomolecules. Here we neglect nearly all this detail in order to construct a simple model that captures the basic physical feature that is independent of the specific nature of the proteins, which is the steric repulsion that prevents multiple proteins from occupying the same region in space. We consider a collection of N proteins, each of which is completely defined by its position $\boldsymbol{r}_i$ in the xy-plane ($1\leq i\leq N$). The potential energy E_{pp} of their interaction is pairwise additive,

$$E_{\mathrm{pp}}\left(\{\boldsymbol{r}_i\}\right)=\sum_{1\leq i<j\leq N}V\left(|\boldsymbol{r}_i-\boldsymbol{r}_j|\right), \tag{3.4}$$

and we choose the purely repulsive Weeks-Chandler-Andersen (WCA) potential [43]

$$V(r)=\begin{cases}4\varepsilon_{\mathrm{p}}\left[\left(\frac{\sigma_{\mathrm{p}}}{r}\right)^{12}-\left(\frac{\sigma_{\mathrm{p}}}{r}\right)^{6}\right]+\varepsilon_{\mathrm{p}} & \text{if } r<2^{1/6}\sigma_{\mathrm{p}}\\ 0 & \text{if } r\geq 2^{1/6}\sigma_{\mathrm{p}}\end{cases} \tag{3.5}$$

as the pair interaction. The parameters ε_{p} and σ_{p} denote the energy and length scales of this interaction, the latter being approximately the diameter of the proteins.

3.2.3 Membrane-Protein Interactions

There are multiple ways in which proteins can interact with cellular membranes. Here we consider a potential energy term that couples the protein position and the height field of the membrane:

$$E_{\mathrm{mp}}\left(\{\tilde{h}_{\boldsymbol{k}}\},\{\boldsymbol{r}_i\}\right)=\frac{\varepsilon}{2}\sum_i\left(h\left(\boldsymbol{r}_i\right)-l\right)^2. \tag{3.6}$$

The physical meaning of this interaction is that the position of the membrane above or below the proteins has to be close to the constant value l relative to the $z=0$ reference plane. The parameter ε described the strength of this constraint. If it is large, then the proteins effectively pin the membrane to the specific height l.

It is instructive to rewrite this interaction in terms of a potential energy surface in which the proteins diffuse: defining

$$\phi(\boldsymbol{r})=\frac{\varepsilon}{2}\left(h\left(\boldsymbol{r}\right)-l\right)^2, \tag{3.7}$$

equation (3.6) becomes

$$E_{\mathrm{mp}}\left(\{\boldsymbol{r}_i\}\right)=\sum_i\phi(\boldsymbol{r}_i), \tag{3.8}$$

where the potential ϕ depends on the membrane configuration and is therefore time-dependent.

3.2.4 Dynamics

Our goal is to study both dynamical and static equilibrium properties of the model. To this end we introduce coupled Langevin equations for both membrane and protein degrees of freedom that describe the time evolution of the system, and that generate an ensemble of configurations according to the canonical Boltzmann distribution.

The Langevin equation for the membrane Fourier modes $\tilde{h}_{\boldsymbol{k}}$ is based on the Fourier Space

Brownian Dynamics model [39, 8, 1]:

$$\frac{\mathrm{d}\tilde{h}_{\boldsymbol{k}}}{\mathrm{d}t} = -\tilde{\Lambda}_{\boldsymbol{k}} \left\{ \frac{\delta E}{\delta h(\boldsymbol{r})} \right\}_{\boldsymbol{k}} + \tilde{\xi}_{\boldsymbol{k}}(t) \tag{3.9}$$

$$= -\tilde{\Lambda}_{\boldsymbol{k}} \left[\left(\sigma k^2 + \kappa k^4 \right) \tilde{h}_{\boldsymbol{k}} + \varepsilon \sum_i \mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r}_i} \left(h(\boldsymbol{r}_i) - l \right) \right] + \tilde{\xi}_{\boldsymbol{k}}(t). \tag{3.10}$$

Here, $E = E_\mathrm{m} + E_\mathrm{pp} + E_\mathrm{mp}$ is the total energy of the system, $\delta E/\delta h(\boldsymbol{r})$ is its functional derivative with respect to a membrane height change at position $\boldsymbol{r}$, and $\{.\}_{\boldsymbol{k}}$ denotes the Fourier transform. The mobility factor $\tilde{\Lambda}_{\boldsymbol{k}}$ is the Oseen tensor [1],

$$\tilde{\Lambda}_{\boldsymbol{k}} = \frac{1}{4\eta k}, \tag{3.11}$$

which accounts for the hydrodynamic behavior of the aqueous solvent with viscosity η. The last part of (3.10) is a stochastic noise term. Its real and imaginary parts have mean zero and satisfy the Fluctuation-Dissipation relationships [41]

$$\left\langle \mathrm{Re}\left(\tilde{\xi}_{\boldsymbol{k}}(t)\right) \mathrm{Re}\left(\tilde{\xi}_{\boldsymbol{k}'}(t')\right) \right\rangle = \tilde{\Lambda}_{\boldsymbol{k}} k_\mathrm{B} T L^2 \delta(t-t') \left(\delta_{\boldsymbol{k},\boldsymbol{k}'} + \delta_{-\boldsymbol{k},\boldsymbol{k}'} \right) \tag{3.12}$$

$$\left\langle \mathrm{Im}\left(\tilde{\xi}_{\boldsymbol{k}}(t)\right) \mathrm{Im}\left(\tilde{\xi}_{\boldsymbol{k}'}(t')\right) \right\rangle = \tilde{\Lambda}_{\boldsymbol{k}} k_\mathrm{B} T L^2 \delta(t-t') \left(\delta_{\boldsymbol{k},\boldsymbol{k}'} - \delta_{-\boldsymbol{k},\boldsymbol{k}'} \right) \tag{3.13}$$

$$\left\langle \mathrm{Re}\left(\tilde{\xi}_{\boldsymbol{k}}(t)\right) \mathrm{Im}\left(\tilde{\xi}_{\boldsymbol{k}'}(t')\right) \right\rangle = 0 \tag{3.14}$$

which guarantee convergence to the correct equilibrium distribution.

The Langevin equation for the position of the i-th protein in the (x, y) plane, $\boldsymbol{r}_i$, is

$$\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} = -\gamma_\mathrm{p} \boldsymbol{\nabla}_i E + \boldsymbol{\zeta}_i \tag{3.15}$$

$$= -\gamma_\mathrm{p} \left[\varepsilon \sum_i \left(h(\boldsymbol{r}_i) - l \right) \boldsymbol{\nabla} h(\boldsymbol{r}_i) + \boldsymbol{\nabla}_i E_\mathrm{pp} \left(\{\boldsymbol{r}_i\} \right) \right] + \boldsymbol{\zeta}_i(t) \tag{3.16}$$

where $\boldsymbol{\nabla}_i$ is the gradient with respect to $\boldsymbol{r}_i$, γ_p is the mobility of the protein, and the stochastic term $\boldsymbol{\zeta}_i(t)$ has mean zero and variance

$$\left\langle \boldsymbol{\zeta}_{i,\alpha}(t) \, \boldsymbol{\zeta}_{j,\beta}(t') \right\rangle = 2\gamma_\mathrm{p} k_\mathrm{B} T \delta_{ij} \delta_{\alpha\beta} \delta(t - t') \tag{3.17}$$

where the subscripts α and β represent the Cartesian components of the two-dimensional vector $\boldsymbol{\zeta}_i$. As before, this choice of the stochastic term ensures proper equilibration of the system. For a free protein that does not interact with the membrane (i.e., $\varepsilon = 0$), equation (3.15) generates a two-dimensional Brownian motion with diffusion coefficient $D_0 = k_\mathrm{B} T \gamma_\mathrm{p}$.

3.2.5 Simulation Scheme

Solving the coupled Langevin equations for membrane and protein degrees of freedom allows us to study the equilibrium behavior of our model system. We use the Euler-Maruyama method to solve these equations numerically for small time increments Δt. In this discrete

representations, equations (3.10) and (3.16) become

$$\tilde{h}_{\boldsymbol{k}}\left(t+\Delta t\right)=\tilde{h}_{\boldsymbol{k}}\left(t\right)-\Delta t\,\tilde{\Lambda}_{\boldsymbol{k}}\left[\left(\sigma k^{2}+\kappa k^{4}\right)\tilde{h}_{\boldsymbol{k}}\left(t\right)+\varepsilon_{\mathrm{p}}\sum_{i}\left(h(\boldsymbol{r}_{i})-l\right)\mathrm{e}^{-i\boldsymbol{k}\cdot\boldsymbol{r}_{i}}\right]+\sqrt{2k_{\mathrm{B}}TL^{2}\Delta t\tilde{\Lambda}_{\boldsymbol{k}}}\;r \tag{3.18}$$

$$\boldsymbol{r}_{i}\left(t+\Delta t\right)=\boldsymbol{r}_{i}\left(t\right)-\Delta t\,\gamma_{\mathrm{p}}\left[\varepsilon_{\mathrm{p}}\sum_{i}\left(h(\boldsymbol{r}_{i})-l\right)\boldsymbol{\nabla}h(\boldsymbol{r}_{i})+\boldsymbol{\nabla}_{i}E_{\mathrm{pp}}\left(\{\boldsymbol{r}_{i}\}\right)\right]+\sqrt{2k_{\mathrm{B}}T\Delta t\gamma_{\mathrm{p}}}\;r \tag{3.19}$$

where r stands for independent, normally distributed random numbers with zero mean and unit variance for both real and imaginary parts of the membrane Fourier modes $\tilde{h}_{\boldsymbol{k}}$ for $\boldsymbol{k}>0$ (equation (3.18)) and for each Cartesian component of the position $\boldsymbol{r}_{i}$ of each protein (equation (3.19)).

The choice of the timestep Δt is determined by the requirement that it must be smaller than all relevant dynamical processes in the system so that they can be resolved by the numerical integration. Several such processes exist in our model. The first is the dynamics of the membrane. In absence of any proteins, each membrane mode relaxes on the wavevector-dependent timescale [27]

$$\tau_{\mathrm{m}}=\frac{1}{\tilde{\Lambda}_{\boldsymbol{k}}\left(\sigma k^{2}+\kappa k^{4}\right)} \tag{3.20}$$

which together with (3.11) implies that

$$\Delta t\ll\frac{4\eta}{\sigma k_{\max}+\kappa k_{\max}^{3}}. \tag{3.21}$$

where $k_{\max}=2\pi\sqrt{P^{2}+Q^{2}}/L$ is the largest wavevector in the system. Using a large number of wavevectors therefore necessitates a very small timestep if one wants to simulate the membrane dynamics accurately. The equilibrium properties, however, are independent of the choice of the mobility factor $\tilde{\Lambda}_{\boldsymbol{k}}$, as long as the stochastic noise terms satisfy the Fluctuation-

Dissipation relationships (3.12)–(3.14). For our equilibrium studies we choose

$$\tilde{\Lambda}_{\boldsymbol{k}} = \frac{\tau^{-1}}{\sigma k^2 + \kappa k^4} \tag{3.22}$$

which makes the membrane relaxation times (3.20) wavevector independent. The constant τ has dimensions of time and sets the timescale for membrane dynamics. Its value does not affect the equilibrium properties of the system, and we set it to one second for convenience. In simulations that aim to study dynamical properties, such as the diffusion of proteins discussed in Section 3.3.3, we use the physical expression (3.11) for the Oseen tensor.

The second constraint on Δt limits protein motions to distances less than their diameter per time-step, $\Delta t \ll \sigma_{\mathrm{p}}^2/4D_0$. This condition ensures proper sampling of protein-protein interactions. Finally, we impose that the typical protein displacement in a single time-step is small compared to the shortest wavelength of the membrane, $\Delta t \ll \pi^2/D_0 k_{\mathrm{max}}^2$.

3.2.6 Simulation Parameters

Our model has a considerable number of parameters. Here we limit ourselves to systems where both the membrane surface tension σ and the preferred membrane height l are chosen to be zero. The strength ε of the protein-membrane interaction is set to $100\, k_{\mathrm{B}}T/\sigma_{\mathrm{P}}^2$, where k_{B} is Boltzmann's constant and T is the temperature. The energy scale ε_{p} of the protein–protein interactions is set equal to $k_{\mathrm{B}}T$. The linear dimension L of the simulation box is 50 times the diameter σ_{P} of the proteins. When studying the equilibrium behavior of our model we use a grid of $P = Q = 32$ Fourier modes together with expression (3.22) for the mobility factor $\tilde{\Lambda}_{\boldsymbol{k}}$. As discussed previously, studying the diffusive behavior of the proteins requires the use of (3.11) for the Oseen tensor, which leads to dramatically different relaxation timescales for membrane modes with different wavevector magnitudes. We therefore used a significantly reduced model system with $(P, Q) = (1, 0)$ or $(P, Q) = (1, 1)$ wavevectors for those simulations. The timestep Δt is set to $10^{-3}\pi^2/D_0 k_{\mathrm{max}}^2$ for simulations with $N = 2$

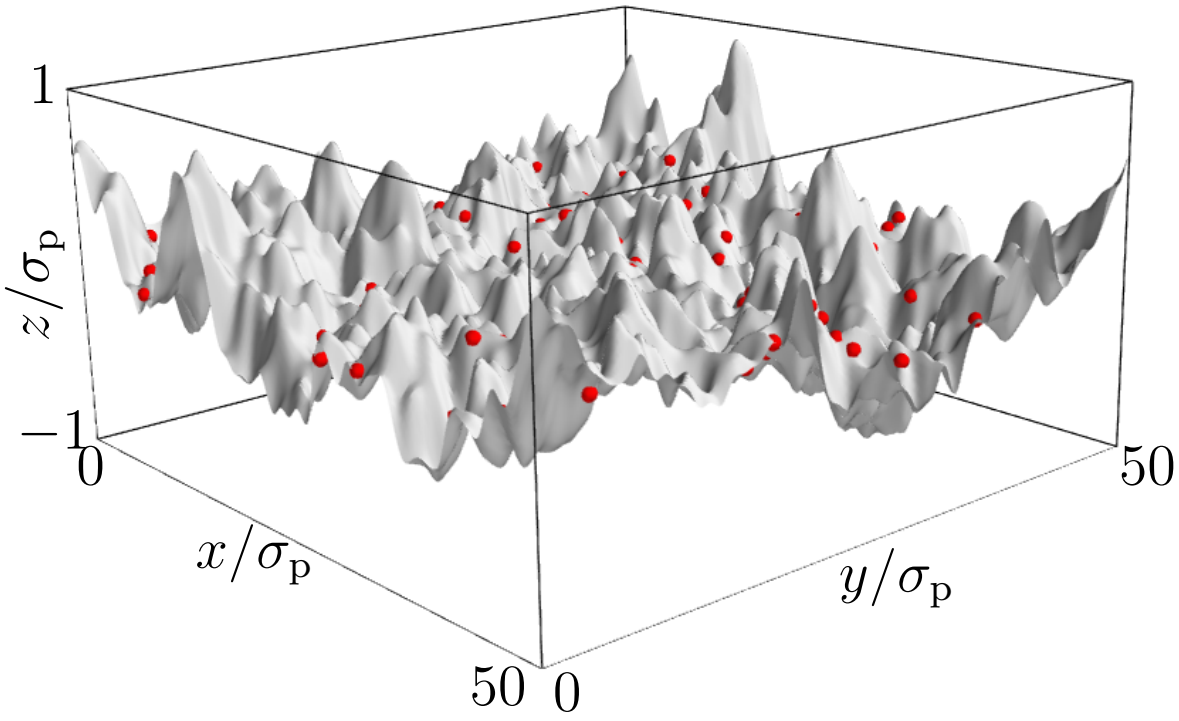

Figure 3.1: Snapshot of a simulation of the combined membrane/protein system. The gray surface shows the current membrane configuration; red dots illustrate protein positions. Simulation parameters are $P = Q = 32$, $\sigma = 0$, $\beta\kappa = 10$, $\varepsilon = 100 k_\mathrm{B}T/\sigma_\mathrm{p}^2$, $N = 100$.

proteins, as well as for $N = 100$ proteins with bending rigidities $\beta\kappa$ equal to 10 or 100 where $\beta = 1/k_\mathrm{B}T$. Simulations with $N = 100, \beta\kappa = 1$ and $N = 500, \beta\kappa \in \{10, 100\}$ needed a 5 times smaller timestep, while those with $N = 500, \beta\kappa = 1$ required another tenfold reduction in Δt to reach proper equilibrium behavior.

3.3 Results

A typical snapshot of our simulations is shown in Figure 3.1. The membrane exhibits undulations that are driven by thermal fluctuations. Proteins diffuse in the (x, y)-plane, effectively pinning the membrane to the height $l = 0$ at their positions.

Figure 3.2 shows the effective potential energy surface $\phi(\boldsymbol{r})$ in which the proteins diffuse. In this representation it becomes apparent that there are regions of high potential that are effectively inaccessible to the proteins. These regions are caused by fluctuations that move the membrane away from the protein's preferred height l. A protein entering such a region would require either a rare fluctuation to overcome this potential or waiting until the membrane has relaxed back to its ground state. This exclusion of proteins from parts of the membrane area leads to motion akin to diffusion in a dynamically crowded environment.

We will study protein diffusion in detail in Section 3.3.3. First we will focus on the equilibrium properties of the membrane and protein subsystems. To this end we define the probability distribution of a microstate, uniquely specified by the protein positions $\{\boldsymbol{r}_i\}$ and Fourier components $\{\tilde{h}_{\boldsymbol{k}}\}$, which is the Boltzmann distribution

$$P(\{\boldsymbol{r}_i\}, \{\tilde{h}_{\boldsymbol{k}}\}) = \frac{1}{Q}\, \mathrm{e}^{-\beta E(\{\boldsymbol{r}_i\}, \{\tilde{h}_{\boldsymbol{k}}\})} \tag{3.23}$$

where

$$Q = \int_{L^2} \mathrm{d}\{\boldsymbol{r}_i\} \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, \mathrm{e}^{-\beta E(\{\boldsymbol{r}_i\}, \{\tilde{h}_{\boldsymbol{k}}\})} \tag{3.24}$$

is the partition function. Here the first integral is taken over all possible positions of each protein in the square $[0, L]^2$. The second integral is over all possible complex values for every Fourier coefficient $\tilde{h}_{\boldsymbol{k}}$ with $\boldsymbol{k} > 0$.

3.3.1 Membrane Fluctuations

The interaction between protein and membrane degrees of freedom has a significant effect on the membrane fluctuation spectrum. In the absence of proteins ($N = 0$) or if the coupling strength ε is zero, the spectrum can be readily computed from the equipartition theorem to be

$$\langle |\tilde{h}_{\boldsymbol{k}}|^2 \rangle_0 = \frac{k_{\mathrm{B}} T L^2}{\sigma k^2 + \kappa k^4}, \tag{3.25}$$

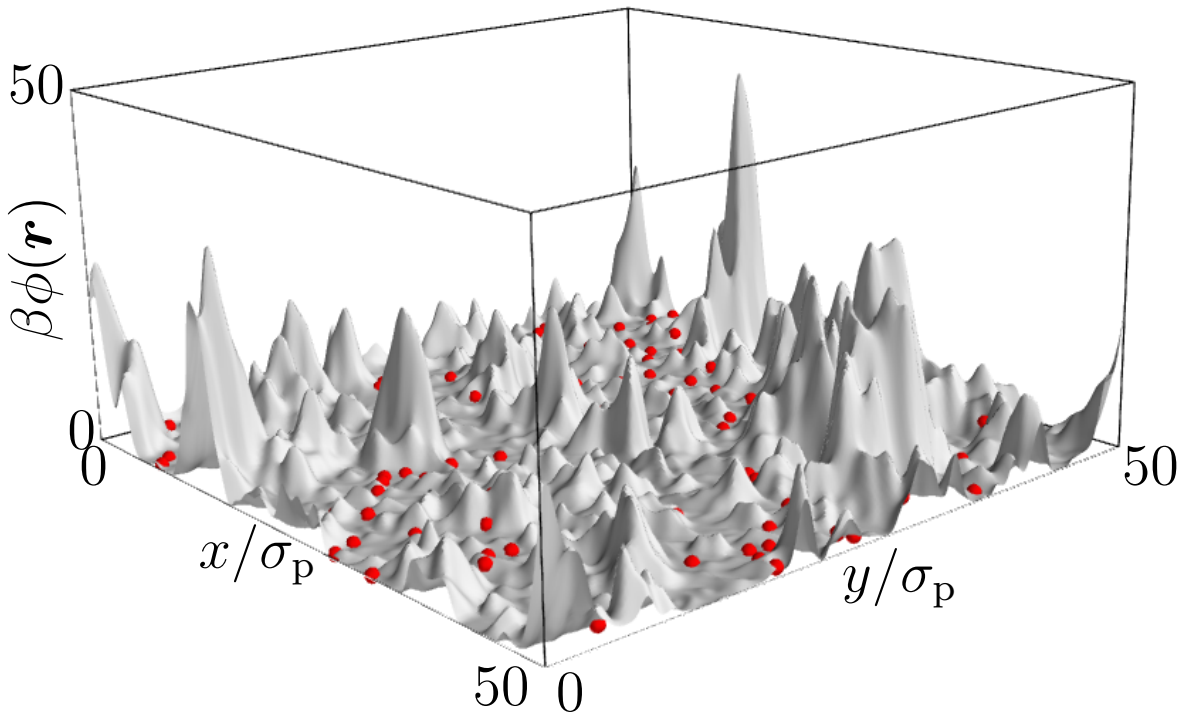

Figure 3.2: Same snapshot as in figure 3.1, but showing in gray the potential energy surface $\phi(\boldsymbol{r})$ instead of the membrane conformation. Proteins remain in the low-energy regions of this potential, whereas the membrane is free to fluctuate in protein-free regions.

If proteins at static positions ($\{\boldsymbol{r}_i\}$) interact with the membrane, the variance of the Fourier modes changes to

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle_{(\{\boldsymbol{r}_i\})} = \frac{\int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, |\tilde{h}_{\boldsymbol{k}}|^2\, \mathrm{e}^{-\beta\,[E_\mathrm{m}+E_\mathrm{mp}]}}{\int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, \mathrm{e}^{-\beta\,[E_\mathrm{m}+E_\mathrm{mp}]}} \tag{3.26}$$

Because the coupling term (3.6) is quadratic in the membrane height, this expression is the variance of a multivariate Gaussian probability distribution, which we compute numerically using matrix inversion (see Supplemental 3.5.1 for details).

Finally, if the proteins are mobile, the variance of a membrane Fourier amplitude $\tilde{h}_{\boldsymbol{k}}$ is

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle = \int_{L^2} \mathrm{d}\{\boldsymbol{r}_i\} \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, |\tilde{h}_{\boldsymbol{k}}|^2 P(\{\boldsymbol{r}_i\}, \{\tilde{h}_{\boldsymbol{k}}\}), \tag{3.27}$$

and we compute this quantity as the time-average of trajectories generated by the Langevin equations (3.18, 3.19).

Results for calculated membrane fluctuation spectra are shown in figure 3.3. Panel (a)shows the effect of protein density, specified by the dimensionless quantity $\rho^* = N\sigma_{\mathrm{p}}^2/L^2$. As expected, both the numerical calculation (3.26) and the simulation result yield the analytical free membrane result (3.25) if no proteins are present ($\rho^* = 0$), which serves as a useful test to validate the respective computer codes. As the protein density increases, the variance of membrane Fourier modes with small wavevectors, corresponding to large length scales, is significantly suppressed, while modes with high wavevectors are not affected by the interaction with the proteins. Membrane spectra computed with static proteins at randomly chosen positions are indistinguishable from those obtained from simulations in which proteins were freely diffusing.

Panel (b) of figure 3.3 illustrates how bending rigidity affects membrane spectra at fixed protein density. The relative reduction in variance of Fourier amplitude at small wavevectors is largest for flexible membranes with low bending rigidities. A natural interpretation of this result is that membranes with high bending rigidities already have smaller fluctuations, and are therefore less affected by the additional pinning to a specific height imposed by the proteins.

3.3.2 Membrane-Induced Protein Interactions

From the proteins' perspective the membrane acts like a time-dependent external potential $\phi(\boldsymbol{r}, t)$. Because all proteins are embedded in the same membrane, its elastic behavior mediates an effective interaction among the proteins. To quantify this coupling, we compute

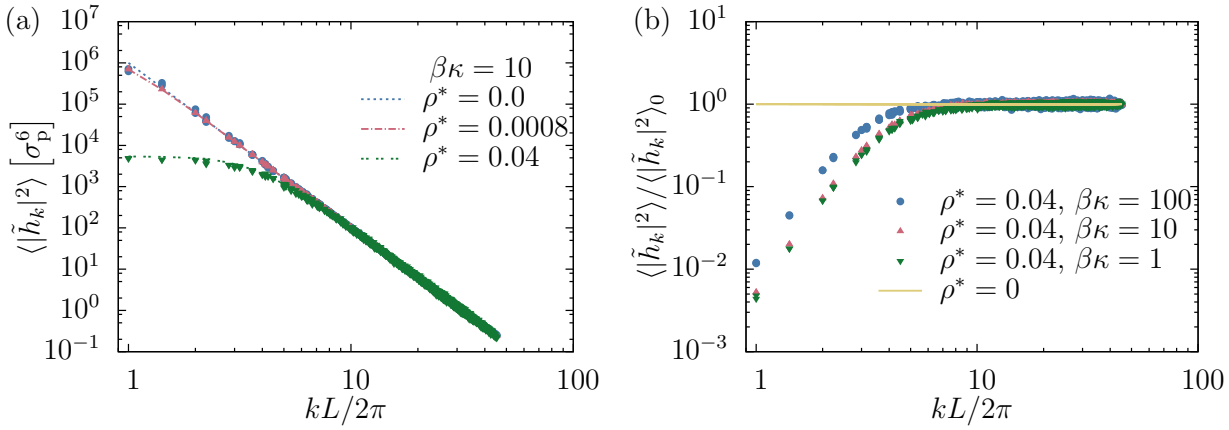

Figure 3.3: (a) Dependence of membrane fluctuations on protein density. If no proteins are present the spectra are given by the free membrane result (3.25). As the protein density increases, the variance of membrane modes with small wavevectors is significantly reduced. Lines represent numerical solutions of (3.26); points were obtained from simulations. (b) Dependence of membrane fluctuations on bending rigidity. The reduction of membrane fluctuations (relative to the free membrane result) is largest for flexible membranes that have low bending rigidity.

the free energy for a configuration with proteins at positions $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N$ by performing a partial trace over membrane degrees of freedom:

$$F(\{\boldsymbol{r}_i\}) = -k_\mathrm{B}T \log \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, \mathrm{e}^{-\beta E(\{\boldsymbol{r}_i\},\{\tilde{h}_{\boldsymbol{k}}\})} \tag{3.28}$$

$$= E_\mathrm{pp}(\{\boldsymbol{r}_i\}) + F_\mathrm{mm}(\{\boldsymbol{r}_i\}) \tag{3.29}$$

where

$$F_\mathrm{mm}(\{\boldsymbol{r}_i\}) = -k_\mathrm{B}T \log \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, \mathrm{e}^{-\beta\,(E_\mathrm{m} + E_\mathrm{mp})} \tag{3.30}$$

is the membrane-mediated interaction between the proteins. It is in general a complicated function of the protein positions that is not pairwise additive. However, due to the Gaussian nature of the protein-membrane coupling it can be easily computed numerically (see

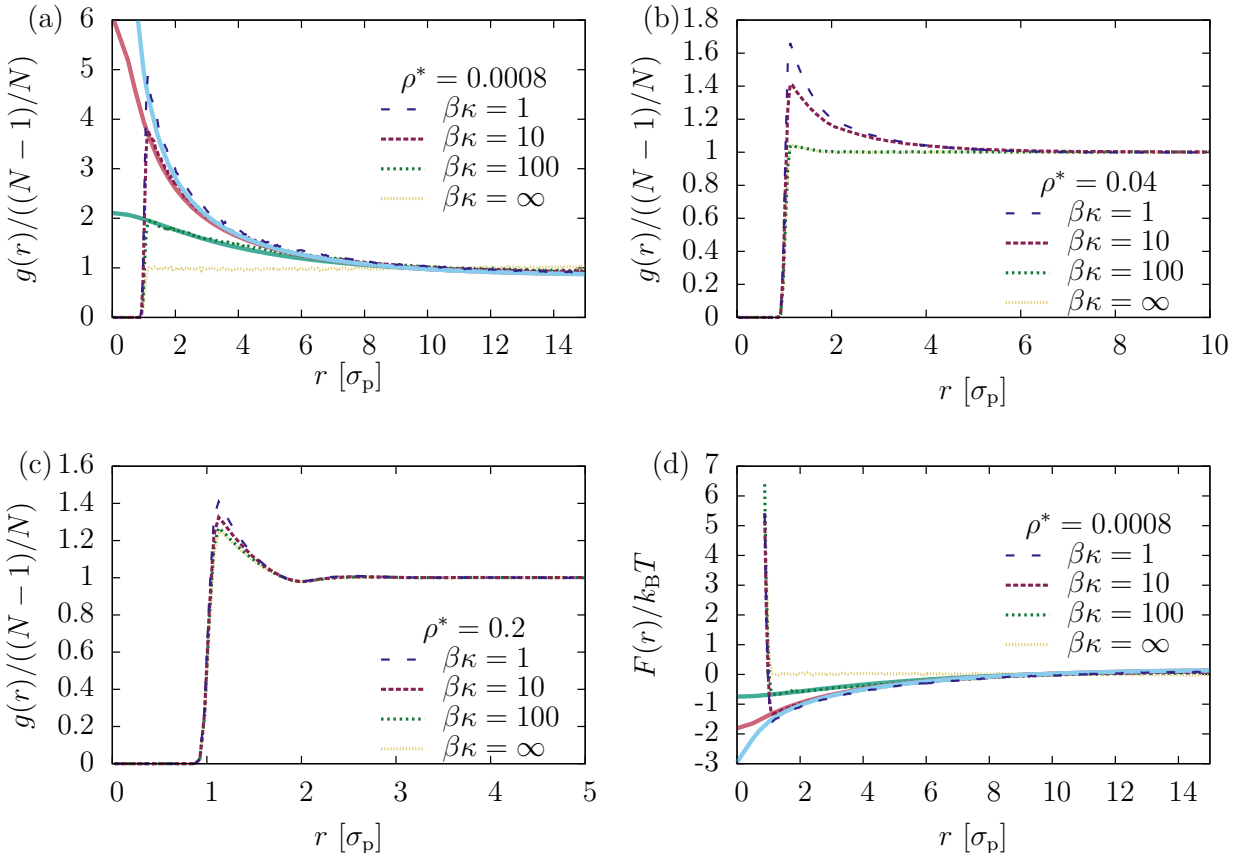

Figure 3.4: Effect of membrane rigidity on protein pair correlation functions at different densities. (a) At low densities ($N = 2$) the RDF is essentially flat outside the protein diameter if the membrane is rigid, and develops a significant peak as the bending rigidity is reduced. (b) At intermediate densities ($N = 100$) this effect is less pronounces. (c) At high densities ($N = 500$) there is a peak in the pair correlation function, but it is due to packing and nearly independent of the membrane bending rigidity. (d) Free energy of the protein–protein interaction, equation (3.32), and membrane-mediated interaction, equation (3.30). Dashed lines are obtained from simulations; solid lines in panels (a) and (d) are obtained numerically as described in Supplemental 3.5.2.

Supplemental 3.5.2 for details).

From our simulations we obtain the radial distribution function (RDF) [44]

$$g\left(\boldsymbol{r}\right)=\frac{L^2}{N^2}\sum_{i\neq j}\langle\delta\left(\boldsymbol{r}-\left(\boldsymbol{r}_j-\boldsymbol{r}_i\right)\right)\rangle. \tag{3.31}$$

This function is a measure for density correlations between two points in space that are separated by the distance vector $\boldsymbol{r}$.

If there are only two proteins in the system, then the RDF is related to the effective free energy of the protein subsystem as

$$g(\boldsymbol{r})\propto \mathrm{e}^{-\beta F(\boldsymbol{r}_1,\boldsymbol{r}_2)} \tag{3.32}$$

where $\boldsymbol{r}=\boldsymbol{r}_2-\boldsymbol{r}_1$ is the separation of the proteins. Even though our model is not fully isotropic due to the square shape of the reference domain, we typically consider rotationally averaged quantities $g(r)$ and $F(r)$.

Protein density correlation functions obtained from simulations are shown in figure 3.4 for different values of membrane bending rigidities and protein densities. If the membrane is completely rigid ($\kappa=\infty$) proteins behave like a two-dimensional WCA fluid: the pair correlation function is equal to one at distances larger than the protein diameter if the density is low (figure 3.4a). Softening the membrane (i.e., reducing the value of κ) has a large effect on the RDF in this case: a significant peak forms right at the constact distance σ_p with a broad shoulder that extends to distances about five times the protein diameter (figure 3.4a). This emergence of protein density correlations is a consequence of their coupling to the membrane, which mediates an attractive interaction. The magnitude of this effect decreases with increasing protein density (figure 3.4b,c). In this case we observe a peak in $g(r)$ even for rigid membranes, which is due to the steric repulsion between proteins and their resulting packing structure. Decreasing κ enhances this peak to a lesser extent than in the case of low

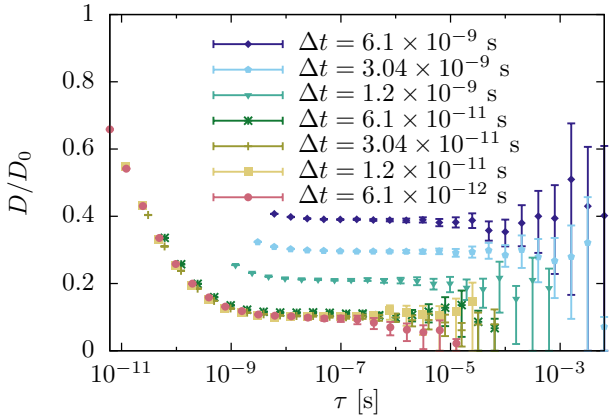

Figure 3.5: Determination of the diffusion constant from mean squared displacements. The timestep Δt must be chosen sufficiently small so that observed MSDs do not depend on its value. The lag time τ must be large enough so that the dynamics reaches the diffusive regime, indicated by a constant value of D as computed by (3.33). Uncertainties increase significantly with τ if the total length of the simulation data is fixed. Simulation parameters: $L = 250\,\mathrm{nm}$,$P = 1$, $Q = 0$, $N = 1$, $D_0 = 5 \times 10^9$ nm^2/s, $\beta\kappa = 10$, $\sigma = 0$, $\eta = 10^{-29}$ Js/nm^3, $\varepsilon = 100\, k_\mathrm{B}T/\sigma_\mathrm{P}^2$.

protein density.

In the low-density limit the RDF is directly related to the free energy of the protein subsystem, equation (3.32). In figure 3.4d we show the free energy surface $F(r)$ as obtained from the simulations of $N = 2$ proteins, together with numerical calculations of the membrane-mediated interaction $F_\mathrm{mm}(r)$. We find that for physically relevant membrane bending energies ($\beta\kappa \approx 10$) the membrane-mediated attraction contributes approximately $1.3\, k_\mathrm{B}T$, or $3\,\mathrm{kJ/mol}$, to the protein-protein interaction at contact.

3.3.3 Protein Diffusion

The diffusion constant D of proteins moving in the (x, y)-plane is a quantity of great interest [45, 29, 46, 27, 47, 48]. Due to the coupling between protein and membrane degrees

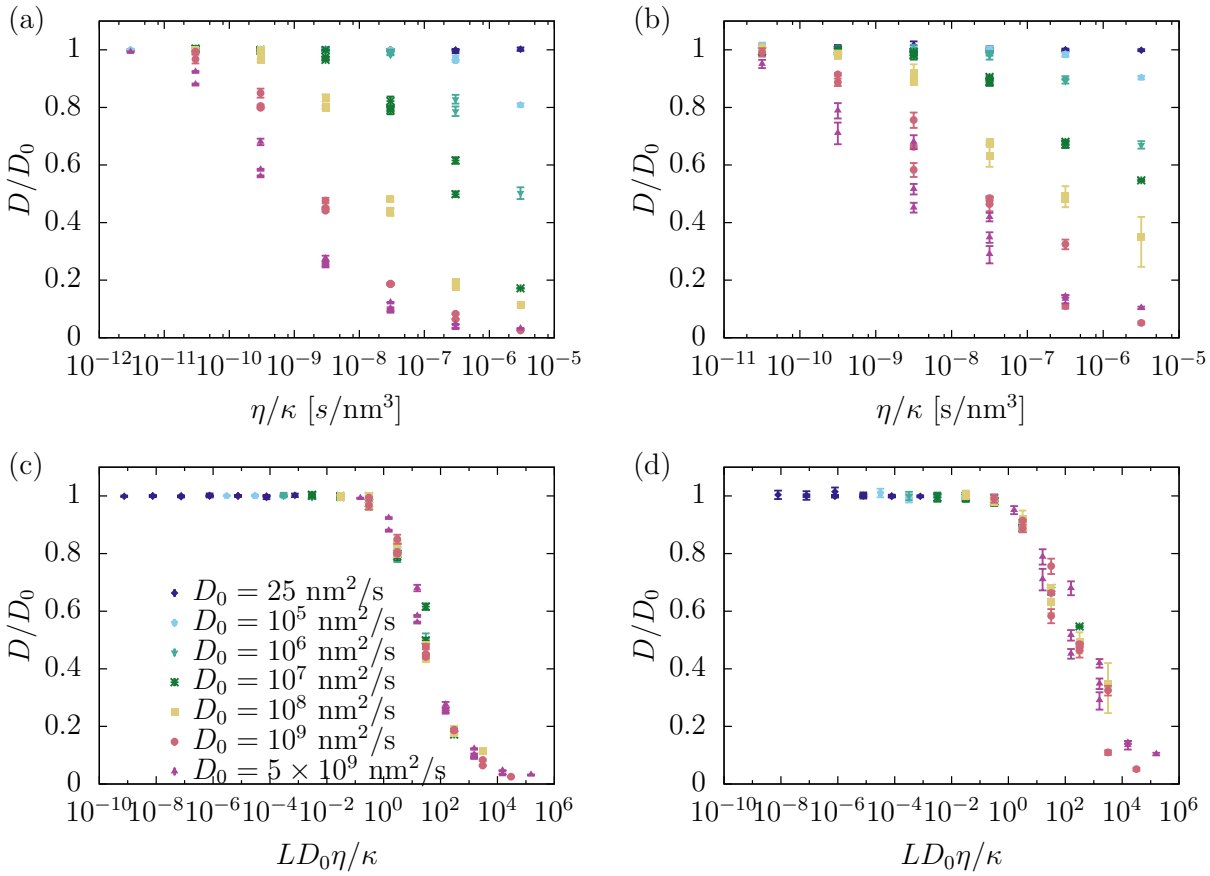

Figure 3.6: Renormalization of protein diffusion as a function of membrane bending rigidity κ and solvent viscosity η. The ratio of actual to bare diffusion constant, D/D_0, varies strongly with system parameters both in the one-dimensional case ($P = 1, Q = 0$, panel (a)) and the most basic two-dimensional case ($P = 1, Q = 1$, panel (b)). When plotted as a function of $D_0\eta/\kappa$ the data points collapse onto single curve in both cases (panels (c) and (d), respectively).

of freedom this constant can differ from the bare diffusion constant D_0 that the Langevin equation (3.16) would generate if no such coupling existed, or if the membrane was infinitely rigid.

In this section we report simulation results that were obtained using a single protein. This allows us to isolate the effect of membrane fluctuations on the diffusion constant from that due to protein crowding. For the reasons stated in Section 3.2.6 we chose an artificially small number of wavevectors, $(P, Q) = (1, 0)$ or $(P, Q) = (1, 1)$. In the former case we only study the protein motion along the x-axis ($d = 1$), while in the latter we consider both the x- and the y-direction ($d = 2$).

The diffusion constant can be obtained from the mean-squared displacement (MSD) at sufficiently long times through the relationship

$$D = \frac{\langle (\Delta \boldsymbol{r}(\tau))^2 \rangle}{2d\tau} \tag{3.33}$$

where the numerator is the variance of the protein displacement over a time period τ.

If there are no interactions, the discretized Langevin equation (3.19) generates a stochastic process that has the diffusion constant D_0 independent of the chosen values for the lag time τ or the timestep Δt. This is no longer the case if the diffusing protein interacts with the membrane, as illustrated in figure 3.5. First, we find that one must use a very small timestep to obtain MSDs that are independent of Δt, and that therefore correspond to the continuum limit $\Delta t \to 0$. Second, even with a properly chosen timestep the apparent diffusion constant depends on the lag time if the latter is not sufficiently large. The likely cause for this behavior is the finite relaxation time of the membrane whose dynamics affects the motion of the proteins. This problem can be avoided by making τ larger than the timescale of all other dynamical processes in the system. This approach, however, has the disadvantage that for a fixed trajectory length one has fewer independent data points to estimate the MSD if τ is large. This leads to large uncertainties in the estimate of $\langle (\Delta \boldsymbol{r}(\tau))^2 \rangle$, and therefore

of D. To quantify the uncertainty in our estimate of the MSD we use the fact that the variance estimator for an underlying Gaussian process follows the chi-squared distribution, and compute the confidence interval

$$\left[\frac{(n-1)\,S^2}{\chi^2_{\alpha/2}}, \frac{(n-1)\,S^2}{\chi^2_{1-\alpha/2}}\right]. \tag{3.34}$$

Here S^2 is the sample variance, which is an unbiased estimate for the population variance $\langle(\Delta \boldsymbol{r}(\tau))^2\rangle$, n is the number of independent observations of displacements with lag time τ, and the parameter α is set to 0.05, which corresponds to a 95% confidence level. This interval is typically not symmetric around the estimated MSD.

Figure 3.5 illustrates how we calculate the diffusion constant for a typical choice of system parameters. First, we successively decrease the timestep Δt in our simulations until the observed MSD, and therefore the apparent diffusion constant (3.33), becomes independent of the timestep. We then identify the lag time τ at which the estimate for D is within the confidence intervals of all subsequent estimates using longer lag times. This estimate of the diffusion constant, together with its uncertainties, is then used in further analysis.

The ratio of the actual diffusion constant D and the bare diffusion constant D_0 in principle depends on all parameters in the model, in particular the solvent viscosity η, the membrane bending rigidity κ, and D_0 itself. This behavior can be seen in Figure 3.6, which shows results for D/D_0 obtained from simulations using $P = 1, Q = 0$ (a) and $P = 1, Q = 1$ (b) Fourier modes and a wide range of parameter values. We find that in general the relative diffusion constant decreases with increasing solvent viscosity, decreasing membrane rigidity, or increasing D_0. In no case did we observe an increase of the diffusion constant above the bare value D_0.

The interaction between the proteins and the membrane couples the diffusive dynamics of the former to the relaxation dynamics of the latter. Protein diffusion throughout the system

occurs on a time scale $\tau_{\text{diff}} \propto L^2/D_0$. According to (3.21) the membrane relaxation time is $\tau_{\text{m}} \propto \eta L^3/\kappa$ for the longest wavevector mode if the surface tension vanishes. Plotting the effective diffusion constant D/D_0 as a function of the ratio $\tau_{\text{m}}/\tau_{\text{diff}}$ of these two timescales, we find that all data points collapse onto a single curve (Figures 3.6c,d). This fact illustrates that the competition between planar diffusion and membrane relaxation determines the actual diffusion constant of the proteins. For biologically realistic values of the parameters involved ($D_0 = 10^{-13}\,\text{m}^2/\text{s}$, $\eta = 10^{-3}\,\text{Js}/\text{m}^3$, $\kappa = 10\,k_{\text{B}}T$, $T = 310\,\text{K}$) the ratio $LD_0\eta/\kappa$ is approximately 6×10^{-4}, at which $D/D_0 \approx 1$. In this case the coupling between the membrane and the proteins therefore does not have a significant effect on the effective protein diffusion constant.

3.4 Discussion

Our model is based on a continuum description of a cellular membrane and a simple particle representation of peripheral proteins. Both components interact through a harmonic interaction that suppresses membrane height fluctuations at the protein locations. Our results show that this interaction gives rise to multiple observable changes in the behavior of both subsystems. First, membrane fluctuations over long length scales are depressed; the extent of which increases with protein density and decreasing membrane rigidity. Second, the coupling to the membrane induces an effective attraction between proteins. Third, the apparent diffusion constant of the proteins decreases if the bending rigidity is low. This effect, however, is likely negligible for typical biological membranes.

The emergence of a membrane-mediated interaction between proteins has been previously established for different types of couplings between the two subsystems. Early work in this field established an effective interaction that scales with $1/r^4$ for membrane inclusion [5, 6, 7]. The same dependence has been calculated for the interaction between rods absorbed to membranes [49, 50]. Theoretical studies of the distance-dependent free energy of membrane

adhesion sites have different functional forms [37, 38, 35]. In all cases the interaction is induced by the thermal fluctuations of the membrane, and therefore entropic in origin.

We find that the induced attraction contributes on the order of 3 kJ/mol to the interaction energy between two proteins for biologically relevant values of the involved parameters. This contribution decreases at higher protein densities. In particular, in no case did we observe a condensation of proteins into a high-density droplet, as one observes in systems with sufficiently attractive pairwise interactions that exhibit a liquid-vapor transition.

In this work we focused on a specific coupling between the membrane and the proteins, namely a quadratic interaction the depends on the membrane height at the positions of the proteins. This model is appropriate for proteins that suppress membrane height fluctuations. There are other classes of proteins that couple differently to the membrane. For example, some membrane scaffolding proteins sense and induce local curvature upon membrane binding [51, 3, 4]. To study these systems our model can be modified by replacing the coupling to the membrane height $h(\boldsymbol{r}_i)$ in equation (3.6) by one to the local mean curvature $\nabla^2 h(\boldsymbol{r}_i)/2$. Related models have been proposed in the literature [19, 48, 28, 52, 47, 29]. Higher resolution models that take into account the non-isotropic shape of specific proteins have shown that the combination of membrane-mediated and direct protein–protein interactions can give rise to orientationally ordered aggregates [53, 54, 55, 12]. Together, these studies highlight the variety and complexity of structures that can emerge from seemingly simple membrane–protein interactions.

Acknowledgments

K. S. was supported in part by a training grant from the National Institute of General Medical Sciences of the National Institutes of Health, award number T32GM008268.

3.5 Supplemental

3.5.1 Calculation of Membrane Fluctuation Spectra

In Section 3.3.1 we derived equation (3.26) for the membrane fluctuation spectrum in the presence of N static proteins at positions $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_N$. This expression is the variance of a multivariate Gaussian probability distribution. Here we explain in detail how we computed this variance numerically.

We begin by enumerating the independent real-valued degrees of freedom [41]. We do so by choosing the following ordering of the real and imaginary parts of the membrane Fourier modes: $\mathrm{Re}\,\tilde{h}_{1,0},\ \mathrm{Im}\,\tilde{h}_{1,0},\ \ldots,\ \mathrm{Re}\,\tilde{h}_{P,0},\ \mathrm{Im}\,\tilde{h}_{P,0},\ \mathrm{Re}\,\tilde{h}_{-P,1},\ \mathrm{Im}\,\tilde{h}_{-P,1},\ \ldots,\ \mathrm{Re}\,\tilde{h}_{P,1},\ \mathrm{Im}\,\tilde{h}_{P,1},\ \ldots,$ $\mathrm{Re}\,\tilde{h}_{-P,Q}, \mathrm{Im}\,\tilde{h}_{-P,Q}, \ldots, \mathrm{Re}\,\tilde{h}_{P,Q}, \mathrm{Im}\,\tilde{h}_{P,Q}$. The total number of real-valued degrees of freedom is $M = 2P + 2Q + 4PQ$. For notational convenience we relabel this sequence as $d_1, \ldots, d_M$ such that

$$\mathrm{Re}\,\tilde{h}_{p,q} = d_{2(p+q(2P+1))} \tag{3.35}$$

$$\mathrm{Im}\,\tilde{h}_{p,q} = d_{2(p+q(2P+1)+1)} \tag{3.36}$$

This allows us to rewrite the exponent in (3.26) as

$$-\beta\left[E_{\mathrm{m}}(\{\boldsymbol{r}_i\}) + E_{\mathrm{mp}}(\{\boldsymbol{r}_i\})\right] = -\frac{1}{2}\boldsymbol{d}^{\intercal}\boldsymbol{X}\boldsymbol{d}, \tag{3.37}$$

where $\boldsymbol{d}$ is the M-dimensional column vector containing the variables d_i, $\boldsymbol{d}^{\intercal}$ is its transpose, and the coupling matrix $\boldsymbol{X}$ is of the form

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{C}_{1,1} & \cdots & \boldsymbol{C}_{1,M/2} \\ \vdots & \ddots & \vdots \\ \boldsymbol{C}_{M/2,1} & \cdots & \boldsymbol{C}_{M/2,M/2} \end{bmatrix}. \tag{3.38}$$

Here $\boldsymbol{C}_{n,m}$ is a 2×2 submatrix that couples the real and imaginary part of a single Fourier mode $\tilde{h}_{\boldsymbol{k}}$. Its components are

$$(\boldsymbol{C}_{n,m})_{0,0} = \frac{2\beta}{L^2}\left(\sigma \boldsymbol{k}_n^2 + \kappa \boldsymbol{k}_n^4\right)\delta_{n,m} + \frac{4\varepsilon\beta}{L^4}\sum_{i=1}^{N}\cos\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\cos\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{3.39}$$

$$(\boldsymbol{C}_{n,m})_{1,0} = \frac{4\varepsilon\beta}{L^4}\sum_{i=1}^{N}\cos\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\sin\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{3.40}$$

$$(\boldsymbol{C}_{n,m})_{0,1} = -\frac{4\varepsilon\beta}{L^4}\sum_{i=1}^{N}\sin\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right)\cos\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right) \tag{3.41}$$

$$(\boldsymbol{C}_{n,m})_{1,1} = \frac{2\beta}{L^2}\left(\sigma \boldsymbol{k}_n^2 + \kappa \boldsymbol{k}_n^4\right)\delta_{n,m} + \frac{4\varepsilon\beta}{L^4}\sum_{i=1}^{N}\sin\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\sin\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{3.42}$$

The coefficients (p, q) of the wavevector $\boldsymbol{k}_n$ are $p = (n - P - 1) \bmod (2P + 1) - P$, $q = \lfloor (n - P - 1)/(2P + 1) \rfloor + 1$.

With the exponent written in the form (3.37) we can compute the covariances as elements of the inverse of $\boldsymbol{X}$,

$$\langle d_j d_{j'} \rangle = \left[\mathbf{X}^{-1}\right]_{jj'}, \tag{3.43}$$

with $1 \leq j, j' \leq M$.

The variance of a single membrane mode can then be computed as the sum of the variances of its real and imaginary parts,

$$\langle |\tilde{h}_{p,q}|^2 \rangle = \langle (\operatorname{Re} \tilde{h}_{pq})^2 \rangle + \langle (\operatorname{Im} \tilde{h}_{pq})^2 \rangle \tag{3.44}$$

$$= \langle d^2_{2(p+q(2P+1))} \rangle + \langle d^2_{2(p+q(2P+1)+1)} \rangle. \tag{3.45}$$

3.5.2 Calculation of Protein Density Correlation Functions

To calculate the membrane-mediated interaction F_{mm} for a static collection of proteins at positions ($\{\boldsymbol{r}_i\}$) we proceed as in Supplemental 3.5.1 and rewrite the Boltzmann factor in the form (3.37). The integration in equation (3.30) can then be performed analytically and yields

$$F_{\mathrm{mm}}(\{\boldsymbol{r}_i\}) = \frac{1}{2} k_{\mathrm{B}} T \log \det \boldsymbol{X}. \tag{3.46}$$

Here we have suppressed constants independent of the protein positions that cause an overall shift in the free energy. Instead, we normalize the free energy such that it is zero at the largest possible distance within our simulation box.

In the special case of only $N = 2$ proteins we can then directly calculate the pair correlation function from this energy using equation (3.32).

Chapter 4

Suppression of Membrane Curvature

4.1 Introduction

The spatial organization of membranes has been an area of interest because of its ability to induce interactions between membrane proteins on the surface of the membrane. In the previous chapter, we discussed one possible interaction: suppressing membrane height. Another such interaction is used by membrane-sculpting proteins: affecting local membrane curvature [9, 19, 10, 2, 11, 12, 33, 14]. While the molecular details of these interaction play an important role, it is possible to discuss these interactions in terms of the shape deformations of the membrane due to the binding of the proteins.

In this chapter, we focus on systems that impose a curvature on the membrane. It is important to note that earlier work investigated proteins as disk inclusions, which impose geometric boundary conditions at the protein–membrane interface or that locally alter the membrane's elastic constants. These couplings result in an effective interaction between proteins that scales as $1/r^4$ where r is the distance between two such inclusions [5, 6, 7]. Systems of adhesion sites pinning the membrane have been studied to find a logarithmic separation between adhesion sites [35]. Similar systems have been previously observed to investigate the effect on the membrane on protein diffusion.; it is thought that protein diffusion is decreased [28, 29]. It is clear that the coupling between protein position and membrane shape provide the basis for the the long range ordering of both.

Due to the length and time scale necessary to observe the desired interactions using atomistic simulations is currently infeasible. We use a hybrid model that combines a continuum description for the membrane with a discrete particle model for the proteins. Similarly to

Chapter 3 we will use the elastic description of the membrane developed by Canham and Helfrich [25, 26] with Fourier Space Brownian Dynamics [39, 8, 1]. We model proteins as point-wise particles defined by their curvature and a general repulsive potential. The coupling between the membrane and proteins is a potential that alters the membrane curvature at the protein position with a specific strength. We will derive coupled Langevin equations to describe the dynamics of the system for proteins diffusing in the x, y plane. Details about this model are described in the next section. We show in section 4.3 how the membrane-protein interactions alters membrane fluctuations, protein diffusion, and protein-protein interactions.

4.2 The Model

We will follow a similar formalism to Section 3.2 for describing the energy and dynamics for the types of systems described in this Chapter. Equation (3.2) will be used to describe the elasticity of the membrane

$$E_{\mathrm{m}}\left(\left\{\tilde{h}_{\boldsymbol{k}}\right\}\right) = \frac{1}{L^2}\sum_{\boldsymbol{k}}\left(\frac{\sigma}{2}k^2 + \frac{\kappa}{2}k^4\right)|\tilde{h}_{\boldsymbol{k}}|^2.$$

We will also continue to use equation (3.4) with equation (3.5) to describe the steric repulsion between proteins

$$E_{\mathrm{pp}}\left(\{\boldsymbol{r}_i\}\right) = \sum_{1\le i<j\le N} V\left(|\boldsymbol{r}_i - \boldsymbol{r}_j|\right)$$

$$V(r) = \begin{cases} 4\varepsilon_{\mathrm{p}}\left[\left(\frac{\sigma_{\mathrm{p}}}{r}\right)^{12} - \left(\frac{\sigma_{\mathrm{p}}}{r}\right)^{6}\right] + \varepsilon_{\mathrm{p}} & \text{if } r < 2^{1/6}\sigma_{\mathrm{p}} \\ 0 & \text{if } r \ge 2^{1/6}\sigma_{\mathrm{p}}. \end{cases}$$

However we will not follow Section 3.2.3 for the membrane-protein coupling scheme. In this chapter we use a potential energy that instead couples the protein position and membrane

curvature:

$$E_{\mathrm{mp}}\left(\{\tilde{h}_{\boldsymbol{k}}\},\{\boldsymbol{r}_i\}\right)=\frac{\varepsilon}{2}\sum_i\left(\boldsymbol{\nabla}^2h\left(\boldsymbol{r}_i\right)-2C_{\mathrm{p}}\right)^2. \tag{4.1}$$

This is a harmonic interaction that ensures the membrane curvature at the protein position is close to C_p, the preferred protein curvature. The parameter ε describes the strength of this constraint.

As we did in Section 3.2.3, we can rewrite this interaction as a potential energy surface:

$$\phi(\boldsymbol{r})=\frac{\varepsilon}{2}\left(\boldsymbol{\nabla}^2h\left(\boldsymbol{r}\right)-2C_{\mathrm{p}}\right)^2, \tag{4.2}$$

equation (4.1) becomes

$$E_{\mathrm{mp}}\left(\{\boldsymbol{r}_i\}\right)=\sum_i\phi(\boldsymbol{r}_i), \tag{4.3}$$

where the potential ϕ is time-dependent. We now define the total energy of the combined membrane-protein system, E, as the sum of these three contributions.

We introduce coupled Langevin equations for both the membrane and protein degrees of freedom to probe the dynamical and equilibrium properties of this model.

The Langevin equation for the membrane Fourier modes $\tilde{h}_{\boldsymbol{k}}$ is

$$\begin{aligned}\frac{\mathrm{d}\tilde{h}_{\boldsymbol{k}}}{\mathrm{d}t}&=-\tilde{\Lambda}_{\boldsymbol{k}}\left\{\frac{\delta E}{\delta h(\boldsymbol{r})}\right\}_{\boldsymbol{k}}+\tilde{\xi}_{\boldsymbol{k}}(t) &(4.4)\\ &=-\tilde{\Lambda}_{\boldsymbol{k}}\left[\left(\sigma k^2+\kappa k^4\right)\tilde{h}_{\boldsymbol{k}}-\varepsilon\sum_i\left(\boldsymbol{\nabla}^2h\left(\boldsymbol{r}_i\right)-C_p\right)k^2\,\mathrm{e}^{-\imath\boldsymbol{k}\cdot\boldsymbol{r}_i}\right]\\ &\quad+\tilde{\xi}_{\boldsymbol{k}}(t). &(4.5)\end{aligned}$$

The energy used here for E is the total energy of the syste,: $E=E_{\mathrm{m}}+E_{\mathrm{pp}}+E_{\mathrm{mp}}$, and $\left\{\frac{\delta E}{\delta h(\boldsymbol{r})}\right\}_{\boldsymbol{k}}$ is the Fourier transform of the functional derivative of the energy. The mobility of

the membrane, $\tilde{\Lambda}_{\boldsymbol{k}}$, is once again the Oseen tensor [1],

$$\tilde{\Lambda}_{\boldsymbol{k}} = \frac{1}{4\eta k}$$

and the stochastic noise term, $\tilde{\xi}_{\boldsymbol{k}}$, has mean zero and satisfies the Fluctuation-Dissipation relationships [41]

$$\begin{aligned}
\left\langle \mathrm{Re}\left(\tilde{\xi}_{\boldsymbol{k}}(t)\right) \mathrm{Re}\left(\tilde{\xi}_{\boldsymbol{k}'}(t')\right)\right\rangle &= \tilde{\Lambda}_{\boldsymbol{k}} k_{\mathrm{B}} T L^2 \delta(t-t') \left(\delta_{\boldsymbol{k},\boldsymbol{k}'} + \delta_{-\boldsymbol{k},\boldsymbol{k}'}\right) \\
\left\langle \mathrm{Im}\left(\tilde{\xi}_{\boldsymbol{k}}(t)\right) \mathrm{Im}\left(\tilde{\xi}_{\boldsymbol{k}'}(t')\right)\right\rangle &= \tilde{\Lambda}_{\boldsymbol{k}} k_{\mathrm{B}} T L^2 \delta(t-t') \left(\delta_{\boldsymbol{k},\boldsymbol{k}'} - \delta_{-\boldsymbol{k},\boldsymbol{k}'}\right) \\
\left\langle \mathrm{Re}\left(\tilde{\xi}_{\boldsymbol{k}}(t)\right) \mathrm{Im}\left(\tilde{\xi}_{\boldsymbol{k}'}(t')\right)\right\rangle &= 0.
\end{aligned}$$

The Langevin equation for the position of the i-th protein in the (x, y) plane, $\boldsymbol{r}_i$, is

$$\begin{aligned}
\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} &= -\gamma_{\mathrm{p}} \boldsymbol{\nabla}_i E + \boldsymbol{\zeta}_i && (4.6) \\
&= -\gamma_{\mathrm{p}} \left[\varepsilon \sum_i \left(\boldsymbol{\nabla}^2 h\left(\boldsymbol{r}_i\right) - C_P\right) \boldsymbol{\nabla}_i \left(\boldsymbol{\nabla}^2 h\left(\boldsymbol{r}_i\right)\right) + \boldsymbol{\nabla}_i E_{\mathrm{pp}}\left(\{\boldsymbol{r}_i\}\right)\right] \\
&\quad + \boldsymbol{\zeta}_i(t). && (4.7)
\end{aligned}$$

The gradient is taken with respect to the position of the i-th protein, γ_{p} is the mobility of the proteins, and $\boldsymbol{\zeta}_i(t)$ follows the fluctuation dissipation theorem

$$\left\langle \boldsymbol{\zeta}_{i,\alpha}\left(t\right) \boldsymbol{\zeta}_{j,\beta}\left(t'\right)\right\rangle = 2\gamma_{\mathrm{p}} k_{\mathrm{B}} T \delta_{ij} \delta_{\alpha\beta} \delta\left(t-t'\right).$$

We follow the same method as in Section 3.2.5 to solve these equations numerically.

Equations (4.5) & (4.7) become

$$\tilde{h}_{\boldsymbol{k}}\left(t+\Delta t\right)=\tilde{h}_{\boldsymbol{k}}\left(t\right)-\Delta t\tilde{\Lambda}_{\boldsymbol{k}}\left[\left(\sigma k^{2}+\kappa k^{4}\right)\tilde{h}_{\boldsymbol{k}}\left(t\right)-\varepsilon\sum_{i}\left(\boldsymbol{\nabla}^{2}h(\boldsymbol{r}_{i})-C_{p}\right)k^{2}\,\mathrm{e}^{-\imath\boldsymbol{k}\,\cdot\,\boldsymbol{r}_{i}}\right]$$
$$+\sqrt{2k_{\mathrm{B}}TL^{2}\Delta t\tilde{\Lambda}_{\boldsymbol{k}}}\;r \tag{4.8}$$

$$\boldsymbol{r}_{i}\left(t+\Delta t\right)=\boldsymbol{r}_{i}\left(t\right)-\Delta t\,\gamma_{\mathrm{p}}\left[\varepsilon\sum_{i}\left(\boldsymbol{\nabla}^{2}h(\boldsymbol{r}_{i})-C_{p}\right)\imath\boldsymbol{k}\boldsymbol{\nabla}^{2}h(\boldsymbol{r}_{i})+\boldsymbol{\nabla}_{i}E_{\mathrm{pp}}\left(\{\boldsymbol{r}_{i}\}\right)\right]$$
$$+\sqrt{2k_{\mathrm{B}}T\Delta t\gamma_{\mathrm{p}}}\;r. \tag{4.9}$$

The timestep, Δt, must be smaller than the dynamical process in the system in order to resolve these processes with numerical integration. This leads to the same three constraints on the size of the timestep as we discussed in Section 3.2.5: the relaxation of membrane modes, protein motion must be smaller than their diameter and protein displacements are small compared to the shortest wavelength of the membrane. In Section 3.2.5; we changed the Mobility factor ($\tilde{\Lambda}_{\boldsymbol{k}}$) for equilibrium processes in order to make the membrane relaxation time-independent of the number of wave-vectors. In this section we will do the same and therefore we use the definition $\tilde{\Lambda}_{\boldsymbol{k}}=\frac{\tau^{-1}}{\sigma k^{2}+\kappa k^{4}}$ for the mobility factor.

This model relies on many parameters. We have limited ourselves to systems where the membrane surface tension, σ, is zero. We will use many parameters from Chapter 3: $\varepsilon=100\,k_{\mathrm{B}}T\sigma_{\mathrm{P}}^{2}$, $\varepsilon_{\mathrm{p}}=k_{\mathrm{B}}T$, $L=50\sigma_{\mathrm{P}}$, and $P=Q=32$. We will use equation (3.22)

$$\tilde{\Lambda}_{\boldsymbol{k}}=\frac{\tau^{-1}}{\sigma k^{2}+\kappa k^{4}}$$

for the results showed in Sections 4.3.1 & 4.3.2. The timestep Δt is set to $10^{-3}\pi^{2}/D_{0}k_{\mathrm{max}}^{2}$. For studying diffusive behavior (the results shown in Section 4.3.3), we use equation (3.11) for the Oseen tensor as is required. This leads us to using a reduced system, with $(P,Q)=(1,1)$ wavevectors for those simulations because using equation (3.11) leads to a much longer

membrane relaxation time.

4.3 Results

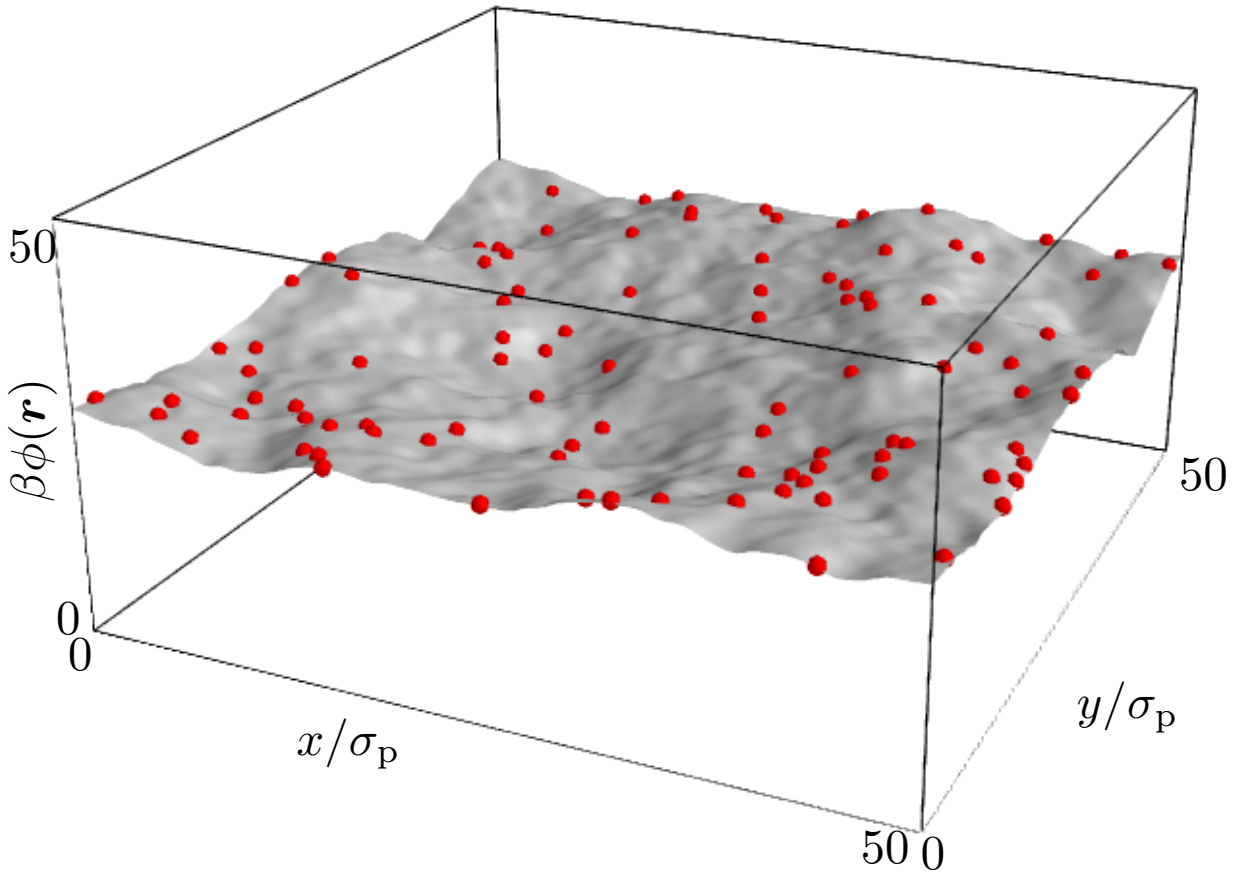

Figure 4.1: Snapshot of a simulation of the combined membrane/protein system. The gray surface shows the current membrane configuration; red dots illustrate protein positions. Simulation parameters are $P = Q = 32$, $\sigma = 0$, $\beta\kappa = 10$, $\varepsilon = 100 k_{\mathrm{B}} T \sigma_{\mathrm{p}}^2$, $C_{\mathrm{p}} = 0.\sigma_{\mathrm{p}}^{-1}$,$N = 100$.

A typical snapshot of our simulations is shown in Figure 4.1. The membrane exhibits undulations that are driven by thermal fluctuations. Proteins diffuse in the xy-plane and

induce a curvature of $C_\mathrm{p} = 0.\sigma_\mathrm{p}^{-1}$ at their positions on the membrane.

Figure 4.2 shows the effective potential energy surface $\phi(\boldsymbol{r})$ in which the proteins diffuse. In this representation it becomes apparent that there are regions of high potential that are effectively inaccessible to the proteins. These regions are caused by fluctuations that move the membrane away from the protein's preferred curvature C_p. A protein entering such a region would require either a rare fluctuation to overcome this potential or for enough time to pass until the membrane has relaxed back to its ground state. This exclusion of proteins from parts of the membrane area leads to motion akin to diffusion in a dynamically crowded environment.

We will study protein diffusion in detail in Section 4.3.3. First we will focus on the equilibrium properties of the membrane and protein subsystems. Therefore we can define the probability distribution using equations (3.23) & (3.24) from Section 3, except with $E = E_\mathrm{m} + E_\mathrm{pp} + E_\mathrm{mp}$.

4.3.1 Membrane Fluctuations

We know from Chapter 3 that for a membrane without proteins the membrane fluctuation spectrum is given by equation (3.25)

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle_0 = \frac{k_\mathrm{B}TL^2}{\sigma k^2 + \kappa k^4}.$$

However, if static proteins interact with the membrane through E_mp that changes to equation (3.26)

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle_{(\{\boldsymbol{r}_i\})} = \frac{\int_\mathbb{C} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, |\tilde{h}_{\boldsymbol{k}}|^2\, \mathrm{e}^{-\beta\,[E_\mathrm{m} + E_\mathrm{mp}]}}{\int_\mathbb{C} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, \mathrm{e}^{-\beta\,[E_\mathrm{m} + E_\mathrm{mp}]}},$$

with E_mp given by equation (4.1). This coupling term is still quadratic and we can therefore compute this numerically as we did in Section 3.3.1 (for details about the coupling matrix and mutivariate Gaussion see Supplemental 4.5.1 & 3.5.1). If the proteins are mobile, the

variance of a membrane Fourier amplitude is given by equation (3.27)

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle = \int_{L^2} \mathrm{d}\{\boldsymbol{r}_i\} \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, |\tilde{h}_{\boldsymbol{k}}|^2 P(\{\boldsymbol{r}_i\}, \{\tilde{h}_{\boldsymbol{k}}\}),$$

which is computed from the Langevin dynamics simulation trajectories.

Results for the membrane fluctuation spectra are shown in Fig 4.3. It shows the effect of protein density. The dots represent the simulation result, which matches the dashed line taken from the numerical matrix inversion method. All the lines and points in this plot lie on top of each other, this means that there is little to no effect on membrane fluctuations due to the membrane-protein interaction presented in this chapter. This is because the protein does not alter membrane height or fluctuations, its presence only alters membrane shape.

4.3.2 Membrane-Induced Protein Interactions

The proteins can be considered as diffusing in a time-dependent external potential, $\phi(\boldsymbol{r}, t)$, provided by the membrane. Therefore, as we did in Section 3.3.2, we can determine the free energy associated with membrane-mediate interactions

$$F_{\mathrm{mm}}(\{\boldsymbol{r}_i\}) = -k_{\mathrm{B}}T \log \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, \mathrm{e}^{-\beta\,(E_{\mathrm{m}} + E_{\mathrm{mp}})}.$$

We can compute this numerically since it is Gaussian (see Supplemental 3.5.2 for details).

The correlations among proteins were quantified from simulation using the radial distribution function (RDF), equation (3.31)

$$g\,(\boldsymbol{r}) = \frac{L^2}{N^2} \sum_{i \neq j} \langle \delta\,(\boldsymbol{r} - (\boldsymbol{r}_j - \boldsymbol{r}_i))\rangle.$$

The RDF describes the variation in density with distance from a reference particle and is used to determine protein-protein interactions. If the membrane is completely rigid ($\kappa = \infty$)

proteins behave like a two-dimensional WCA fluid: the pair correlation function is equal to one at distances larger than the protein diameter and zero at distances less than the protein diameter. If the density is low (figure 4.4a) the pair correlation function remains flat, indicating that there is no membrane-mediated interaction. From figure 4.4d this is made clearer; the interaction peak determined by the method described in Supplemental 3.5.2 in the free energy relaxes to zero before the WCA radius cut-off and therefore does not play a role. The effect of protein density is observed in figure 4.4b,c; it shows that by increasing the density a membrane-mediated interaction is still not induced.

4.3.3 Protein Diffusion

Here describe how the membrane-protein interaction alters protein diffusion in the (x, y)-plane in order to see if protein diffusion slows as it did in the previous section. Due to the coupling between protein and membrane degrees of freedom this constant can differ from the bare diffusion constant, D_0, to that the Langevin equation (4.7) would generate if no such coupling existed or if the membrane was infinitely rigid. We obtain the diffusion constant from the MSD using equation (3.33)

$$D = \frac{\langle (\Delta \boldsymbol{r}(\tau))^2 \rangle}{2d\tau}.$$

We report simulation results that were obtained using a single protein diffusing in the xy-plane. As we did in Section 3.3.3, we used a reduced system of $P = Q = 1$. Figure 4.5 shows how the ratio of D to the bare diffusion D_0 changes with respect to the ratio of solvent viscosity, η, and membrane rigidity, κ. We see for the parameters chosen there is little difference from the bare diffusion constant. In chapter 3, we saw that for large bare diffusion constants there was a drastic change in the lateral diffusion, however, there was little change for smaller bare diffusion constants. In this chapter we see that for the same small bare diffusion constants, which are more biologically relevant, there is little change in

the projected diffusion constant found from simulation..

4.4 Discussion

Our model is based on a continuum description of a cellular membrane and a simple particle representation of peripheral proteins. Both components interact through a harmonic interaction that couples protein curvature to local membrane curvature. Our results show that this interaction changes multiple observables of both subsystems. Membrane fluctuations are unaltered by the presence of the proteins interacting through the curvature-coupled model. The radial distribution function shows that unlike in Chapter 3 there is little to no membrane-mediated interaction among the proteins. The lateral diffusion constant of the proteins varies little at the parameters observed in this chapter. We calculated the ratio of D/D_0 for a bare diffusion constant of $D_0 = 10^{-11}\,\mathrm{m}^2/\mathrm{s}$ and saw that there was no effect on protein diffusion due to the membrane-protein coupling. We will need to investigate further parameter sets in order to determine if a similar trend to that in Chapter 3 can be observed.

In this work, we focus on a specific coupling between the membrane and the proteins, namely a quadratic interaction the depends on the membrane curvature at the positions of the proteins. This model is appropriate for membrane-sculpting proteins which induce local membrane curvature. In this chapter, we discuss this interaction for proteins in the xy-plane, however some such proteins diffuse along the membrane surface instead which has a great effect on their dynamics. To study these systems, our model needs to account for the current state of the system in the stochastic dynamics; we will discuss the development of such a model in Chapter 5.

Acknowledgments

K. S. was supported in part by a training grant from the National Institute of General Medical Sciences of the National Institutes of Health, award number T32GM008268. This

work was facilitated though the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system at the University of Washington

4.5 Supplemental

4.5.1 Calculation of Membrane Fluctuation Spectra

Here we explain how the variance of a multivariate Gaussian probability distribution can be computed numerically. We follow that same formalism as in Supplemental 3.5.1 except with a coupling matrix given by

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{C}_{1,1} & \cdots & \boldsymbol{C}_{1,M/2} \\ \vdots & \ddots & \vdots \\ \boldsymbol{C}_{M/2,1} & \cdots & \boldsymbol{C}_{M/2,M/2} \end{bmatrix}. \tag{4.10}$$

Here $\boldsymbol{C}_{n,m}$ is a 2×2 submatrix that couples the real and imaginary part of a single Fourier mode $\tilde{h}_{\boldsymbol{k}}$. Its components are

$$(\boldsymbol{C}_{n,m})_{0,0} = \frac{2\beta}{L^2}\left(\sigma \boldsymbol{k}_n^2 + \kappa \boldsymbol{k}_n^4\right)\delta_{n,m} + \frac{4\varepsilon\beta}{L^4} k_n^2 k_m^2 \sum_{i=1}^{N} \cos\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\cos\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{4.11}$$

$$(\boldsymbol{C}_{n,m})_{1,0} = -\frac{4\varepsilon\beta}{L^4} k_n^2 k_m^2 \sum_{i=1}^{N} \cos\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\sin\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{4.12}$$

$$(\boldsymbol{C}_{n,m})_{0,1} = -\frac{4\varepsilon\beta}{L^4} k_n^2 k_m^2 \sum_{i=1}^{N} \sin\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right)\cos\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right) \tag{4.13}$$

$$(\boldsymbol{C}_{n,m})_{1,1} = \frac{2\beta}{L^2}\left(\sigma \boldsymbol{k}_n^2 + \kappa \boldsymbol{k}_n^4\right)\delta_{n,m} + \frac{4\varepsilon\beta}{L^4} k_n^2 k_m^2 \sum_{i=1}^{N} \sin\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\sin\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right). \tag{4.14}$$

The coefficients (p, q) of the wavevector $\boldsymbol{k}_n$ are $p = (n - P - 1) \bmod (2P + 1) - P$, $q = \lfloor (n - P - 1)/(2P + 1) \rfloor + 1$.

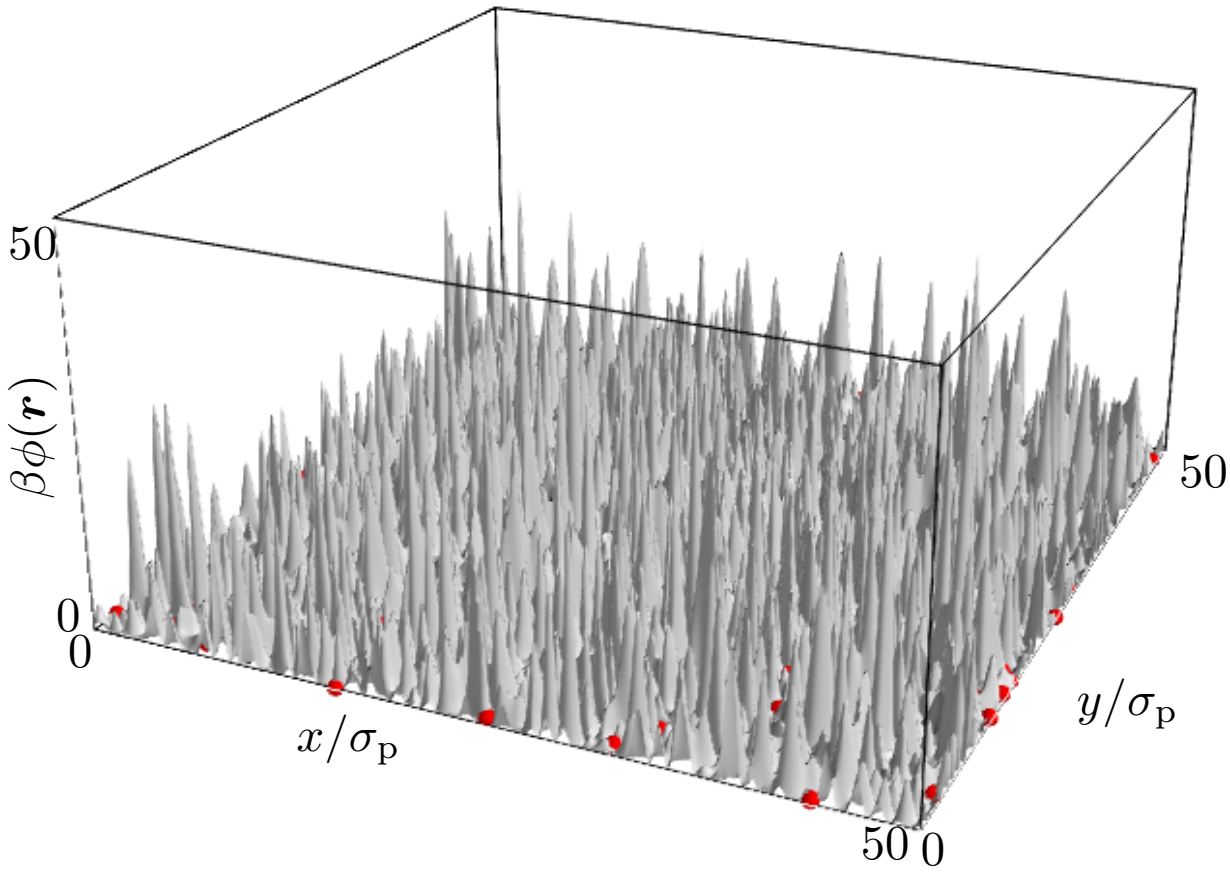

Figure 4.2: Same snapshot as in Fig. 4.1, but showing in gray the potential energy surface $\phi(\boldsymbol{r})$ instead of the membrane conformation. Proteins remain in the low-energy regions of this potential, whereas the membrane is free to fluctuate in protein-free regions.

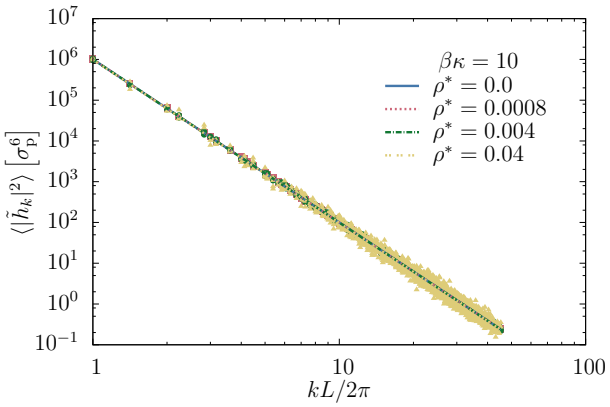

Figure 4.3: Dependence of membrane fluctuations on protein density. If no proteins are present the spectra are given by the free membrane result (3.25). As the protein density increases, the variance of membrane modes is unchanged. Lines represent numerical solutions of (3.26); points were obtained from simulations.

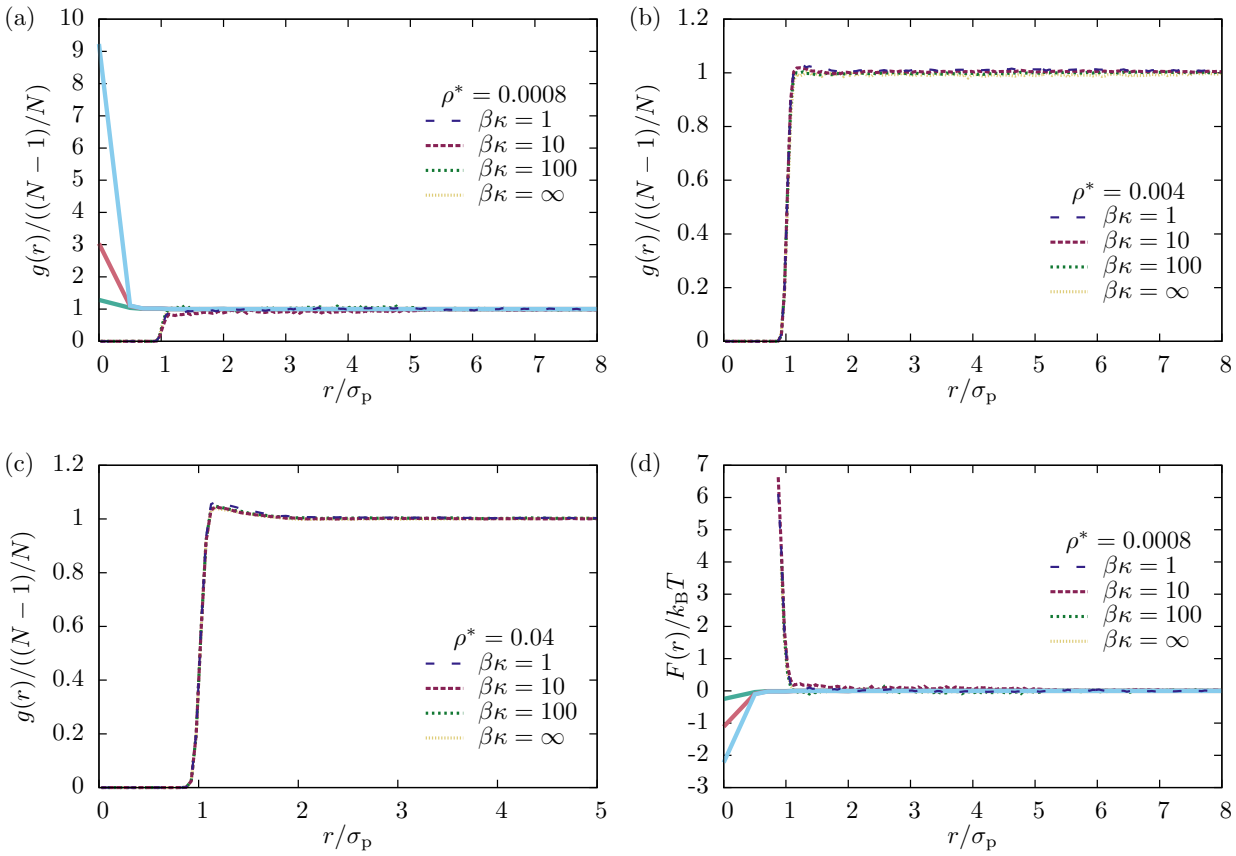

Figure 4.4: Effect of membrane rigidity on protein pair correlation functions at different densities. (a) At low densities ($N = 2$) the RDF is essentially flat outside the protein diameter if the membrane is rigid, and remains flat as the bending rigidity is reduced. (b) At intermediate densities ($N = 10$) the RDF is essentially flat outside the protein diameter if the membrane is rigid, and remains flat as the bending rigidity is reduced.(c)The RDF matches the rigid membrane case for at an intermediate density ($N = 100$). (d) Free energy of the protein–protein interaction, (3.32), and membrane-mediated interaction, (3.30). Dashed lines are obtained from simulations; solid lines are obtained numerically as described in Supplemental 3.5.2.

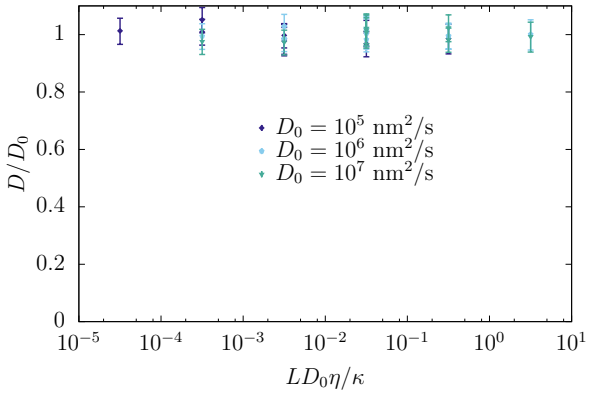

Figure 4.5: The ratio of actual diffusion to bare diffusion plotted as a function of $D_0\eta/\kappa$ shows that there is no change in the diffusion at these parameters. Simulation parameters: $P = Q = 1$, $L = 50$

Chapter 5

Langevin Equations for Coupled Membrane-Protein Dynamics

5.1 Introduction

In the previous chapters, we described the proteins as diffusing in the (x, y)-plane, however, peripheral proteins can diffuse along the membrane surface instead. By implementing this additional geometry we can determine the effect of a fluctuating surface on properties of protein organization and diffusion. This more complex system can also be more revealing when investigating the curvature-coupled model (Chapter 4), where the proteins diffuse on the membrane instead of in the plane.

When investigating a system where proteins diffuse along the membrane surface, we treat the proteins as diffusing within a time dependent surface. This means that the motion of the protein depends on the current membrane curvature. The curvature of the membrane also increases the density of states of the protein. This increase in the density of states comes from the fact that the area of the curved surface is larger than that of the projection. The conversion factor that accounts for this excess area is the metric tensor, more specifically the determinant of the metric tensor. The metric tensor, inverse metric tensor, and the determinant of the metric tensor are

$$g_{ij} = \delta_{ij} + \partial_i h \partial_j h \tag{5.1}$$

$$\left(g^{-1}\right)_{ij} = \delta_{ij} - \frac{\partial_i h \partial_j h}{|g|} \tag{5.2}$$

$$|g| = 1 + (\nabla h)^2 \tag{5.3}$$

where the indices $i, j = x, y$ and ∂_i is the partial derivative in the x-direction. In order to account for the membrane curvature the metric tensor needs to be taken into account [40, 29, 27, 28].

Previous works have described similar systems to investigate how a membrane alters protein diffusion[27, 28]. In these works, however, the metric tensor is not considered in the membrane dynamics, which is inconsistent with detailed balance. Work by Sigurdsson et. al.[29] comments on the lack of drift term for the membrane in such systems, however it does not report on the outcome of using such a term; they instead focus on in-plane dynamics.

In this chapter, we will derive coupled Langevin equations for the dynamics of the Fourier amplitudes of the membrane height and the positions of the proteins in the membrane. This is consistent with detailed balance and considers the Fokker-Planck equation for both the membrane and protein together. We will start by discussing the simple cases of a protein diffusing on a static surface and of a membrane fluctuating with a static protein on its surface. We will then continue to derive the coupled system, where both species are dynamic. Lastly we will comment on how this method can be expanded to investigate curvature-coupled systems.

5.2 Determination of Langevin Equations for Coupled Membrane-Protein Dynamics

We will first discuss the two cases where one of the two subsystems are static and then the combined case in order to break down the complexity of the problem.

5.2.1 One Dynamic Protein on a Static Membrane

For diffusion on a static curved surface the Laplace operator in the Fokker-Planck equation needs to be replaced by the Laplace-Beltrami operator which includes effects of the metric

tensor. This leads to a Fokker-Planck equation for this subsystem of

$$\frac{\partial \rho(x,t)}{\partial t} = -\boldsymbol{\nabla}\left(\boldsymbol{\nu} - \gamma\left(g^{-1}\right)_{ij}\boldsymbol{\nabla}E\right)\rho(x,t) + \boldsymbol{\nabla}^2 D_0\left(g^{-1}\right)_{ij}\rho(x,t). \tag{5.4}$$

This equation uses the Laplace operator and wraps the effects of the metric tensor into an additional drift in the first term of the equation, $\boldsymbol{\nu}$. The Fokker-Planck equation is also the divergence of the flux

$$\frac{\partial \rho(x,t)}{\partial t} = -\boldsymbol{\nabla}\cdot\boldsymbol{J}. \tag{5.5}$$

Comparing equation (5.4) and equation (5.5) the expression for the flux is

$$J = \left(\boldsymbol{\nu} - \frac{1}{2}k_{\mathrm{B}}T\gamma_{\mathrm{p}}\boldsymbol{\nabla}\left(g^{-1}\right)_{ij}\right)\rho - \frac{1}{2}k_{\mathrm{B}}T\gamma_{\mathrm{p}}\left(g^{-1}\right)_{ij}\boldsymbol{\nabla}\rho. \tag{5.6}$$

The desired probability distribution for a diffusing protein on a static membrane is

$$\rho\left(\{\boldsymbol{r}_i\}\right) = \frac{1}{Z}\sqrt{|g_i|}\,\mathrm{e}^{-E/k_{\mathrm{B}}T}, \tag{5.7}$$

where $|g_i|$ is the determinant of the metric tensor at the position of the protein. Using this probability distribution and solving for $\boldsymbol{j} = 0$ the added drift term, $\boldsymbol{\nu}$, is

$$\boldsymbol{\nu} = -D_0\left[\left(g^{-1}\right)_{jk}\partial_j\partial_k h\left(\boldsymbol{r}_i\right)\right]\frac{\partial_i h\left(\boldsymbol{r}_i\right)}{|g_i|} + \frac{1}{2}k_{\mathrm{B}}T\gamma_{\mathrm{p}}\boldsymbol{\nabla}\cdot\left(g^{-1}\right)_{ij}. \tag{5.8}$$

This ensures that equation (5.7) is the stationary distribution and invariant under detailed balance.

From equation (5.4) and equation (5.8) the Langevin equation can be determined

$$\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} = -\gamma_{\mathrm{p}}\left(g^{-1}\right)_{jk}\left[k_{\mathrm{B}}T\partial_j\partial_k h\left(\boldsymbol{r}\right)\frac{\partial_i h\left(\boldsymbol{r}\right)}{|g_i|} + \nabla E\right] + \sqrt{2D_0\left(g^{-1}\right)_{jk}}\boldsymbol{r}_j(t), \tag{5.9}$$

where $\boldsymbol{r}_j(t)$ is a random force with zero mean and variance one.

5.2.2 Dynamic Membrane with a Static Protein

Similar to the previous section the Fokker-Planck equation for a fluctuating membrane with a static protein is

$$\frac{\partial \rho}{\partial t} = -\frac{\partial}{\partial \tilde{h}_{\boldsymbol{k}}} \left[-\tilde{\Lambda}_{\boldsymbol{k}} \left\{ \frac{\delta E}{\delta h(\boldsymbol{r})} \right\}_{\boldsymbol{k}} + \boldsymbol{\nu} \right] \rho + \frac{1}{2} \frac{\partial^2}{\partial^2 \tilde{h}_{\boldsymbol{k}}} \left[\tilde{\Lambda}_{\boldsymbol{k}} \rho \right]. \tag{5.10}$$

Following the process described above the added drift term in this Fokker-Planck equation is

$$\boldsymbol{\nu} = k_{\mathrm{B}} T \tilde{\Lambda}_{\boldsymbol{k}} \left\{ \frac{\delta \ln \sqrt{|g_i|}}{\delta h(\boldsymbol{r})} \right\}_{\boldsymbol{k}} + \frac{1}{2} \left\{ \frac{\delta \Lambda(\boldsymbol{r})}{\delta h(\boldsymbol{r})} \right\}_{\boldsymbol{k}}. \tag{5.11}$$

This ensures that

$$\rho \left(\{ \tilde{h}_{\boldsymbol{k}} \} \right) = \frac{1}{Z} \sqrt{|g_i|}\, \mathrm{e}^{-E/k_{\mathrm{B}}T} \tag{5.12}$$

is the stationary distribution, where g_i is the metric tensor at the protein position.

The drift term, equation (5.11), can be viewed as generated by a potential $-\frac{1}{2} k_{\mathrm{B}} T \ln |g_i|$, which leads to an effective energy

$$E_{\mathrm{eff}} = E_{\mathrm{m}} - \frac{1}{2} k_{\mathrm{B}} T \ln |g_i|. \tag{5.13}$$

The Fokker Planck equation can then be written as

$$\frac{\partial \rho}{\partial t} = -\boldsymbol{\nabla} \left[-\tilde{\Lambda}_{\boldsymbol{k}} \boldsymbol{\nabla} E_{\mathrm{eff}} \left(\{ \tilde{h}_{\boldsymbol{k}} \} \right) \right] \rho + \frac{1}{2} \boldsymbol{\nabla}^2 \left[\tilde{\Lambda}_{\boldsymbol{k}} \rho \right], \tag{5.14}$$

which gives the stationary solution. We therefore use this energy(equation (5.13)) in the Langevin equation for a membrane with a static protein

$$\frac{\mathrm{d}\tilde{h}_{\boldsymbol{k}}}{\mathrm{d}t} = -\tilde{\Lambda}_{\boldsymbol{k}} \left[\left\{ \frac{\delta E_{\mathrm{eff}}}{\delta h(\boldsymbol{r})} \right\}_{\boldsymbol{k}} \right] + \tilde{\xi}_{\boldsymbol{k}}, \tag{5.15}$$

where $\tilde{\xi}_{\boldsymbol{k}}$ is the same random force as described in the previous chapters.

5.2.3 Combined Membrane-Protein Dynamics

We now combine the two scenarios in Sections 5.2.1 and 5.2.2 to determine the coupled membrane-protein dynamics. The probability distribution for coupled membrane-protein systems, with the proteins diffusing in the membrane is given by:

$$\rho\left(\{\tilde{h}_{\boldsymbol{k}}\},\{\boldsymbol{r}_i\}\right)=\frac{1}{Z}\sqrt{\prod_i|g_i|}\,\mathrm{e}^{-E/k_{\mathrm{B}}T}, \tag{5.16}$$

where Z is the partition function for the system . The drift terms for the membrane and protein are

$$\boldsymbol{\nu}_{\mathrm{m}}=k_{\mathrm{B}}T\tilde{\Lambda}_{\boldsymbol{k}}\left\{\frac{\delta\ln\sqrt{\prod_i|g_i|}}{\delta h(\boldsymbol{r})}\right\}_{\boldsymbol{k}}+\frac{1}{2}\left\{\frac{\delta\tilde{\Lambda(\boldsymbol{r})}}{\delta h(\boldsymbol{r})}\right\}_{\boldsymbol{k}} \tag{5.17}$$

$$\boldsymbol{\nu}_{\mathrm{p}}=-D_0\left[\left(g^{-1}\right)_{jk}\partial_j\partial_k h\left(\boldsymbol{r}_i\right)\right]\frac{\partial_i h\left(\boldsymbol{r}_i\right)}{|g_i|}+\frac{1}{2}k_{\mathrm{B}}T\gamma_{\mathrm{p}}\boldsymbol{\nabla}\cdot\left(g^{-1}\right)_{ij}, \tag{5.18}$$

respectively, because they correctly produce the stationary distribution. Combining equations (5.17) and (5.18) with the Fokker-Planck equations gives the stationary distribution We therefore write the two Langevin equations

$$\frac{\mathrm{d}\tilde{h}_{\boldsymbol{k}}}{\mathrm{d}t}=--\tilde{\Lambda}_{\boldsymbol{k}}\left[\left\{\frac{\delta E}{\delta h(\boldsymbol{r})}\right\}_{\boldsymbol{k}}-\sum_i 2k_{\mathrm{B}}T\boldsymbol{\nabla}h\left(\boldsymbol{r}_i\right)\imath\boldsymbol{k}\,\mathrm{e}^{\imath\boldsymbol{k}\cdot\boldsymbol{r}_i}\right]+\tilde{\xi}_{\boldsymbol{k}} \tag{5.19}$$

$$\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t}=-\gamma_{\mathrm{p}}\left(g^{-1}\right)_{jk}\left[k_{\mathrm{B}}T\partial_j\partial_k h\left(\boldsymbol{r}\right)\frac{\partial_i h\left(\boldsymbol{r}\right)}{|g_i|}+\nabla E\right]+\sqrt{2D_0\left(g^{-1}\right)_{jk}}\boldsymbol{r}_j(t) \tag{5.20}$$

for the membrane and protein, respectively. Limitations of this model are discussed in Supplemental 5.5..

5.2.4 Simulation Scheme

Since we use the quadratic approximation for the elastic membrane energy term, we will apply the quadratic approximation to the effective energy that is used in the Langevin equation (5.19)

$$E_{\text{eff}} = \frac{1}{2} k_{\text{B}} T \sum_i \left(\boldsymbol{\nabla} h(\boldsymbol{r}_i) \right)^2 . \tag{5.21}$$

We follow the same method as in Section 3.2.5 to solve the Langevin equations numerically. Equations (5.19) & (5.20) become

$$\begin{aligned}
\tilde{h}_{\boldsymbol{k}}(t+\Delta t) = \tilde{h}_{\boldsymbol{k}}(t) - \Delta t \tilde{\Lambda}_{\boldsymbol{k}} \left[\left(\sigma k^2 + \kappa k^4 \right) \tilde{h}_{\boldsymbol{k}}(t) - 2 k_{\text{B}} T \sum_i \boldsymbol{\nabla} h(\boldsymbol{r}_i) \imath \boldsymbol{k} \, \mathrm{e}^{-\imath \boldsymbol{k} \cdot \boldsymbol{r}_i} \right] \\
+ \sqrt{2 k_{\text{B}} T L^2 \Delta t \tilde{\Lambda}_{\boldsymbol{k}}} \; r \qquad (5.22) \\
x_i(t+\Delta t) = -\frac{D_0}{g} \left[\partial_{xx} h \partial_x h - \frac{\partial_x h^2}{g} \partial_x \partial_x h \partial_x h - \frac{2 \partial_x h \partial_y h}{g} \partial_{xy} h \partial_x h + \partial_{yy} h \partial_x h - \frac{\partial_y h^2}{g} \partial_{yy} h \partial_x h \right] \\
+ \sqrt{2 D_0} \left[\left(1 - \frac{\partial_x h^2}{g + \sqrt{g}} \right) r_x - \frac{\partial_x h \partial_y h}{g + \sqrt{g}} r_y \right] \qquad (5.23) \\
y_i(t+\Delta t) = -\frac{D_0}{g} \left[\partial_{xx} h \partial_y h - \frac{\partial_x h^2}{g} \partial_{xx} h \partial_y h - \frac{2 \partial_x h \partial_y h}{g} \partial_{xy} h \partial_y h + \partial_{yy} h \partial_y h - \frac{\partial_y h^2}{g} \partial_{yy} h \partial_y h \right] \\
+ \sqrt{2 D_0} \left[\left(1 - \frac{\partial_y h^2}{g + \sqrt{g}} \right) r_y - \frac{\partial_x h \partial_y h}{g + \sqrt{g}} r_x \right] . \qquad (5.24)
\end{aligned}$$

The timestep, Δt, must be smaller than the dynamical process in the system in order to resolve these processes with numerical integration. This leads to the same three constraints on the size of the timestep as we discussed in Section 3.2.5: the relaxtion of membrane modes, protein motion must be smaller that their diameter and protein displacements are small compared to the shortest wavelength of the membrane. In Section 3.2.5 we changed the mobility factor ($\tilde{\Lambda}_{\boldsymbol{k}}$) for equilibrium processes in order to make the membrane relaxation time independent of the number of wave-vectors. In this section, we will do the same, and therefore we use the definition $\tilde{\Lambda}_{\boldsymbol{k}} = \frac{\tau^{-1}}{\sigma k^2 + \kappa k^4}$ for the mobility factor.

This model relies on many parameters. We will use many parameters from Chapter 3: $\varepsilon_{\mathrm{p}} = k_{\mathrm{B}}T$, $L = 50\sigma_{\mathrm{P}}$, and $P = Q = 32$. We will also use equation (3.22)

$$\tilde{\Lambda}_{\boldsymbol{k}} = \frac{\tau^{-1}}{\sigma k^2 + \kappa k^4}$$

for the results showed in Sections 5.2.5. The timestep Δt is set to $10^{-3}\pi^2/D_0 k_{\mathrm{max}}^2$.

For studying diffusive behavior the use of equation (3.11) for the Oseen tensor is required. In previous chapters we used a reduced system for studying the diffusive behavior of proteins, however here we will use the parameters discussed in reference [27], i.e. $L = 100\sigma_{\mathrm{P}}$ and $P = Q = 25$.

5.2.5 Results

Membrane Fluctuations

We know from Chapter 3 that for a membrane without proteins, the membrane fluctuation spectrum is given by equation (3.25)

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle_0 = \frac{k_{\mathrm{B}}TL^2}{\sigma k^2 + \kappa k^4}.$$

However, if the drift terms are included in the Langevin equations, this no longer holds and we must continue as we did in Section 3.3.1 (for details about the coupling matrix and mutivariate Gaussion see Supplimental 3.5.1 & 5.5.1). If the proteins are mobile, the variance of a membrane Fourier amplitude is given by equation (3.27)

$$\langle|\tilde{h}_{\boldsymbol{k}}|^2\rangle = \int_{L^2} \mathrm{d}\{\boldsymbol{r}_i\} \int_{\mathbb{C}} \mathrm{d}\{\tilde{h}_{\boldsymbol{k}}\}\, |\tilde{h}_{\boldsymbol{k}}|^2 P(\{\boldsymbol{r}_i\}, \{\tilde{h}_{\boldsymbol{k}}\}),$$

which is computed from the Langevin dynamics simulation trajectories.

Figure 5.1 shows the fluctuation spectrum for the membrane-protein system described

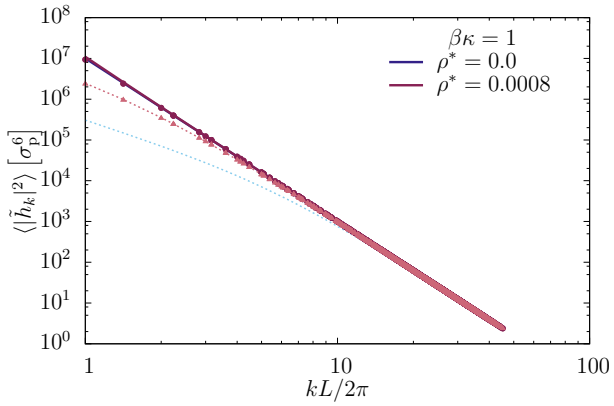

Figure 5.1: Dependence of membrane fluctuations on protein density. If no proteins are present the spectra are given by the free membrane result (3.25). As the protein density increases, the variance of membrane modes is altered. Lines represent numerical solutions of (3.26) for zero surface tension, dashed lines represent numerical solutions for $\sigma = 0.05 k_B T/\sigma_p^2$, and points were obtained from simulations.

above. The solid lines show the spectra determined from the numerical matrix inversion for different densities with zero surface tension. The dashed lines show the spectra determined from the numerical matrix inversion for different densities with surface tension non-zero ($\sigma = 0.05 k_B T/\sigma_p^2$). The points represent data determined from simulation using equation (3.27). We find that for systems with non-zero surface tension thatthe Fourier modes with small wave-vectors are increased slightly. For systems with zero surface tension long length scale fluctuations are increased more drastically, because the surface tension term is what counteracts the effect of the drift term.

Protein-Protein Interactions

The correlations among proteins were quantified from simulation using the RDF. Figure 5.2 shows the radial distribution function for two different membrane surface tensions and vary membrane rigidity. Figure 5.2a shows the RDF for zero surface tension; outside the cut-off

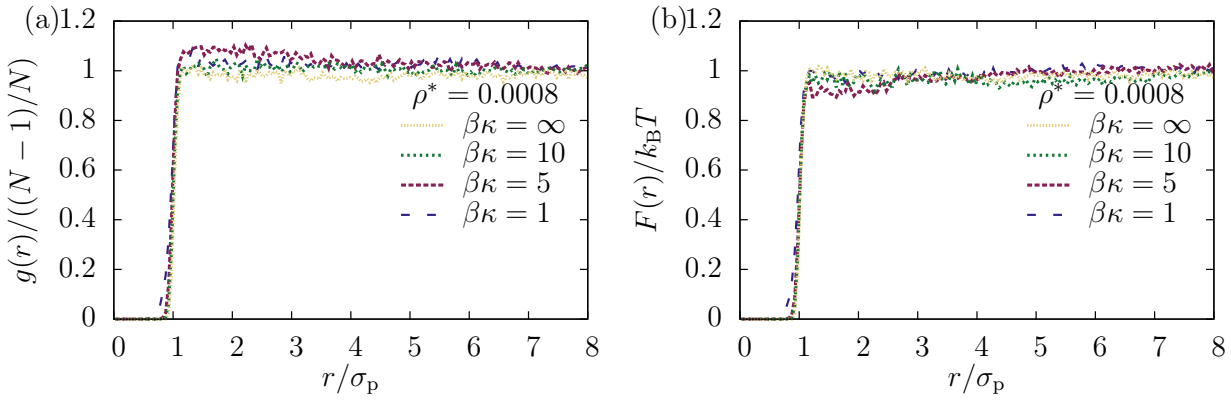

Figure 5.2: Effect of membrane rigidity and surface tension on protein pair correlation functions at low densities($N = 2$). (a) For zero surface tension the RDF remains mostly flat at all distances. (b) For non-zero surface tension the RDF remains flat past the protein diameter.

of the repulsive potential the RDF remains flat indicating that there is no protein-protein interaction Systems of non-zero surface tension are shown figure 5.2b; the addition of the drift term does not induce any sort of protein-protein interaction.

Protein Diffusion

In order to compare this model to previous models such as reference [28, 27] we will investigate the lateral diffusion constant of the proteins. We obtain the diffusion constant from the MSD using equation (3.33)

$$D = \frac{\langle(\Delta \boldsymbol{r}(\tau))^2\rangle}{2d\tau}.$$

Figure 5.3 shows the diffusion constants from simulation using the method laid out in ref [27] and using the method described above. Reister-Gottfried et al. saw a decrease in protein diffusion for all the parameters they investigated. We were able to observe a similar trend using their method. Using the method in this chapter we were unable to produce the

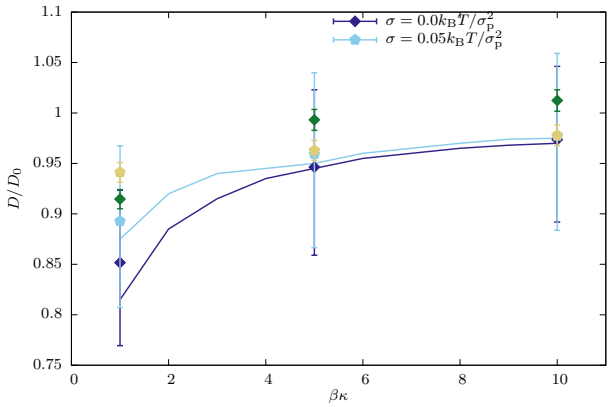

Figure 5.3: The dependance of the ration of actual diffusion to bare diffusion on bending rigidity. The purple points represents diffusion constants from simulations using the method described in ref. [27] for zero surface tension. The blue points represents diffusion constants from simulations using the method described in ref. [27] for non-zero surface tension. The lines are data taken from ref. [27]. The yellow points and grea points represent data from simulations using the method described in this chapter for zero and non-zero surface tension, respectively. We see a variation among the two methods, using the present method slightly increase the ration in comparison to the previous method. Simulation parameters: $L = 100\sigma_p$, $P = Q = 25$.

same results though. We found that the diffusion constant decreases by less than what is seen for the previous methods.

5.3 The Langevin Equations for Coupled Membrane-Protein Dynamics with Coupling Energy

In this section, we will expand the model described in Section 5.2 to membrane-protein curvature-coupled systems. In general, it is a similar process to determine the coupled Langevin equations for the membrane and proteins.

The probability distribution for coupled membrane-protein systems where the membrane and protein are coupled through curvature is

$$\rho\left(\{\tilde{h}_{\boldsymbol{k}}\},\{\boldsymbol{r}_i\}\right)=\frac{1}{Z}\sqrt{\prod_i|g_i|}\,\mathrm{e}^{-E\left(\{\tilde{h}_{\boldsymbol{k}}\},\{\boldsymbol{r}_i\}\right)/k_{\mathrm{B}}T}, \tag{5.25}$$

where Z is the partition function for the system. The energy used here is the total energy from Chapter 4 which includes equation (4.1) as the coupling energy. This distribution has to be invariant under detailed balance. The Fokker-Planck equation is

$$\begin{aligned}\frac{\partial\rho}{\partial t}&=-\boldsymbol{\nabla}\left[-\boldsymbol{\Gamma}\boldsymbol{\nabla}E\left(\{\tilde{h}_{\boldsymbol{k}}\}\right)+\boldsymbol{\nu}\right]\rho+\frac{1}{2}\boldsymbol{\nabla}^2\left[\boldsymbol{\Gamma}\rho\right] \qquad (5.26)\\ &=-\boldsymbol{\nabla}\left[\left(-\boldsymbol{\Gamma}\boldsymbol{\nabla}E\left(\{\tilde{h}_{\boldsymbol{k}}\}\right)+\boldsymbol{\nu}-\frac{1}{2}\boldsymbol{\nabla}\boldsymbol{\Gamma}\right)\rho-\frac{1}{2}\boldsymbol{\Gamma}\boldsymbol{\nabla}\rho\right]\\ &=-\boldsymbol{\nabla}\cdot J\\ J&=\left(-\boldsymbol{\Gamma}\boldsymbol{\nabla}E\left(\{\tilde{h}_{\boldsymbol{k}}\}\right)+\boldsymbol{\nu}-\frac{1}{2}\boldsymbol{\nabla}\boldsymbol{\Gamma}\right)\rho-\frac{1}{2}\boldsymbol{\Gamma}\boldsymbol{\nabla}\rho \qquad (5.27)\end{aligned}$$

where the derivatives are taken over both $\{\tilde{h}_{\boldsymbol{k}}\}$ and $\{\boldsymbol{r}_i\}$. Solving for $J=0$ will give the same drift terms from Section 5.2.3(equations (5.17) and (5.18)). This leads to the Langevin

equations

$$\frac{\partial \tilde{h}_{\boldsymbol{k}}}{\partial t} = -\tilde{\Lambda}_{\boldsymbol{k}} \left[\left\{ \frac{\delta(E_{\mathrm{m}} + E_{\mathrm{mp}})}{\delta h(\boldsymbol{r})} \right\}_{\boldsymbol{k}} - 2 \sum_i k_{\mathrm{B}} T \boldsymbol{\nabla} h\left(\boldsymbol{r}_i\right) \imath \boldsymbol{k}\, \mathrm{e}^{\imath \boldsymbol{k} \cdot \boldsymbol{r}_i} \right] + \tilde{\xi}_{\boldsymbol{k}} \tag{5.28}$$

$$\frac{\mathrm{d}\boldsymbol{r}_i}{\mathrm{d}t} = -\gamma_{\mathrm{p}} \left[\left(g^{-1}\right)_{ij} \boldsymbol{\nabla}_j (E_{\mathrm{pp}} + E_{\mathrm{mp}}) + k_{\mathrm{B}} T \left(g^{-1}\right)_{jk} \partial_j \partial_k h\left(\boldsymbol{r}\right) \frac{\partial_i h\left(\boldsymbol{r}\right)}{g} \right] + \sqrt{2 D_0 \left(g^{-1}\right)_{ij}} \boldsymbol{\zeta}_j(t) \tag{5.29}$$

for the membrane and protein, respectively. This differs from the Langevin equations in Section 5.2.3. Here in equation (5.29) the force term also contains information about the metric tensor; the remaining terms and all the terms in equation (5.28) are identical to the previously derived equations of motion.

5.4 Discussion

We derived a model that explicitly couples the dynamics of a cellular membrane described as a continuum and peripheral proteins described as point-wise particles which diffuse along the membrane surface. Previous work has investigated similar models [27, 28], however detailed balance is not conserved in these models. They consider the effect of the varying density of states on the protein dynamics, but fail to do so for the membrane. In this chapter we derive a model that is consistent in detailed balance: drift terms (due to the metric tensor) are present for both the membrane and the proteins.

The model presented here is consistent in its treatment of membrane and protein dynamics and we will now comment how it alters observables compared to the previous models. The membrane fluctuations are increased for the short wave vector modes. This differs from the models that do not consider the metric tensor for the membrane, in those models the membrane does not have any interaction with the protein and is essentially just a freely fluctuating membrane. This model shows that the inclusion of the drift term does not reveal a protein-protein interaction. We compared the model presented here to that described by

Reister-Gottfried et al. by investigating the protein diffusion constant. We find that we show less of a decrease in protein diffusion than previously seen. Since we use the quadratic approximation for the membrane drift term, we observe that with a finite number of Fourier modes the model breaks down when temperature gets too high. This is due to the fact that surface tension alone can not keep the counteract the entropy increase from the increase in the density of states.

We have expanded the method described in Section 5.2 to the curvature-coupled systems described in Chapter 4. It is our hope that with this method that one can accurately investigate the protein diffusion, membrane-fluctuations, and membrane-mediated interactions as we did in Chapter 4, but instead with the proteins existing on the membrane surface.

Acknowledgments

K. S. was supported in part by a training grant from the National Institute of General Medical Sciences of the National Institutes of Health, award number T32GM008268. This work was facilitated though the use of advanced computational, storage, and networking infrastructure provided by the Hyak supercomputer system at the University of Washington

5.5 Supplemental

5.5.1 Variance of Gaussian Integral Determination of Limits

We use the numerical matrix inversion method discussed in Chapter 3 to determine the limitations of this model for a membrane with static proteins. This uses equation (3.26), with the energy being replaced by E_{eff}(equation (5.13)). This coupling term is still quadratic, and we can therefore use equation (3.37) with a coupling matrix, $\boldsymbol{X}$, that reflects the current coupling scheme. For the system of interest the components of this coupling matrix are:

$$(\boldsymbol{C}_{n,m})_{0,0} = \frac{2\beta}{L^2}\left(\sigma \boldsymbol{k}_n^2 + \kappa \boldsymbol{k}_n^4\right)\delta_{n,m} - \frac{4\beta}{L^4}\sum_{i=1}^{N} k_n k_m \sin\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\sin\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{5.30}$$

$$(\boldsymbol{C}_{n,m})_{1,0} = -\frac{4\beta}{L^4}\sum_{i=1}^{N} k_n k_m \sin\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\cos\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{5.31}$$

$$(\boldsymbol{C}_{n,m})_{0,1} = -\frac{4\beta}{L^4}\sum_{i=1}^{N} k_n k_m \cos\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right)\sin\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right) \tag{5.32}$$

$$(\boldsymbol{C}_{n,m})_{1,1} = \frac{2\beta}{L^2}\left(\sigma \boldsymbol{k}_n^2 + \kappa \boldsymbol{k}_n^4\right)\delta_{n,m} - \frac{4\beta}{L^4}\sum_{i=1}^{N} k_n k_m \cos\left(\boldsymbol{k}_n \cdot \boldsymbol{r}_i\right)\cos\left(\boldsymbol{k}_m \cdot \boldsymbol{r}_i\right) \tag{5.33}$$

Solving the matrix inversion using one Fourier mode leads to the development conditions to ensure the model does not break down. This is seen by looking at the eigenvalues of the matrix, increasing the temperature past a certain cutoff leads to negative eigenvalues. The appearance of these negative eigenvalues predict when the model breaks down. The limit at which the model breaks down decreases as you increase the number of Fourier modes in one dimension, this is seen in figure 5.4.

For one Fourier mode the point of breakdown is reproduced below in another method.

5.5.2 Analytical Determination of Limits

We can also comment on the limitations of this model by looking at an analog to a two-dimensional ideal gas. First we will look at the average area of a membrane without proteins. In this case, the energy is just the Canham-Helfrich energy (using one Fourier mode)

$$E = \frac{1}{L^2}\left(\sigma k^2 + \kappa k^4\right)\tilde{a}_{\boldsymbol{k}}^2, \tag{5.34}$$

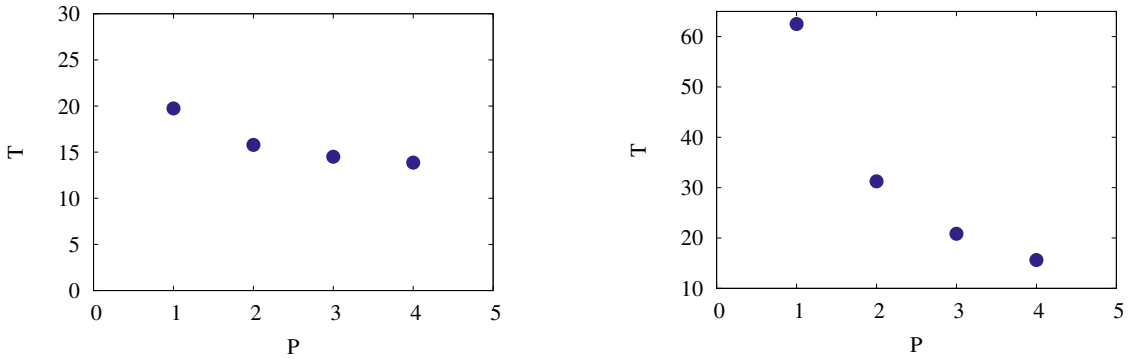

Figure 5.4: Left: Dependence of the breakdown of the model on the number of wave vectors in one dimension for zero surface tension with $\beta\kappa = 1$. Right: Dependence of the breakdown of the model on the number of wave vectors in one dimension for zero bending rigidity with $\sigma = 0.05 k_{\mathrm{B}} T/\sigma_{\mathrm{p}}^2$. The points represent at what temperature the model will no longer work.

where $\tilde{a}_{\boldsymbol{k}}$ is the real part of the Fourier mode, and the partition function is

$$Z = \int \mathrm{d}\tilde{a}_{\boldsymbol{k}} \mathrm{d}x \, \mathrm{e}^{-E/k_{\mathrm{B}}T}. \tag{5.35}$$

The average area of the membrane is defined by

$$\langle A \rangle = \frac{1}{Z} \int \mathrm{d}\tilde{a}_{\boldsymbol{k}} \mathrm{d}x \, A \mathrm{e}^{-E/k_{\mathrm{B}}T}, \tag{5.36}$$

where

$$A = L \int \mathrm{d}x \, \left[1 + \frac{1}{2} (\boldsymbol{\nabla} h(x))^2 \right]. \tag{5.37}$$

If we define the membrane shape as

$$h\left(\boldsymbol{r}_i\right) = \frac{2}{L^2} \tilde{a}_{\boldsymbol{k}} \cos\left(\boldsymbol{k} \cdot \boldsymbol{r}_i\right)' \tag{5.38}$$

the average area is

$$\langle A \rangle = L^2 + \frac{2\pi^2 k_{\mathrm{B}} T}{k^2 L^2 (k^2 \kappa + \sigma)}. \tag{5.39}$$

This gives a linear relation between the average area and the temperature of the system, show in figure 5.5. We want to investigate the same relationship for a membrane with a

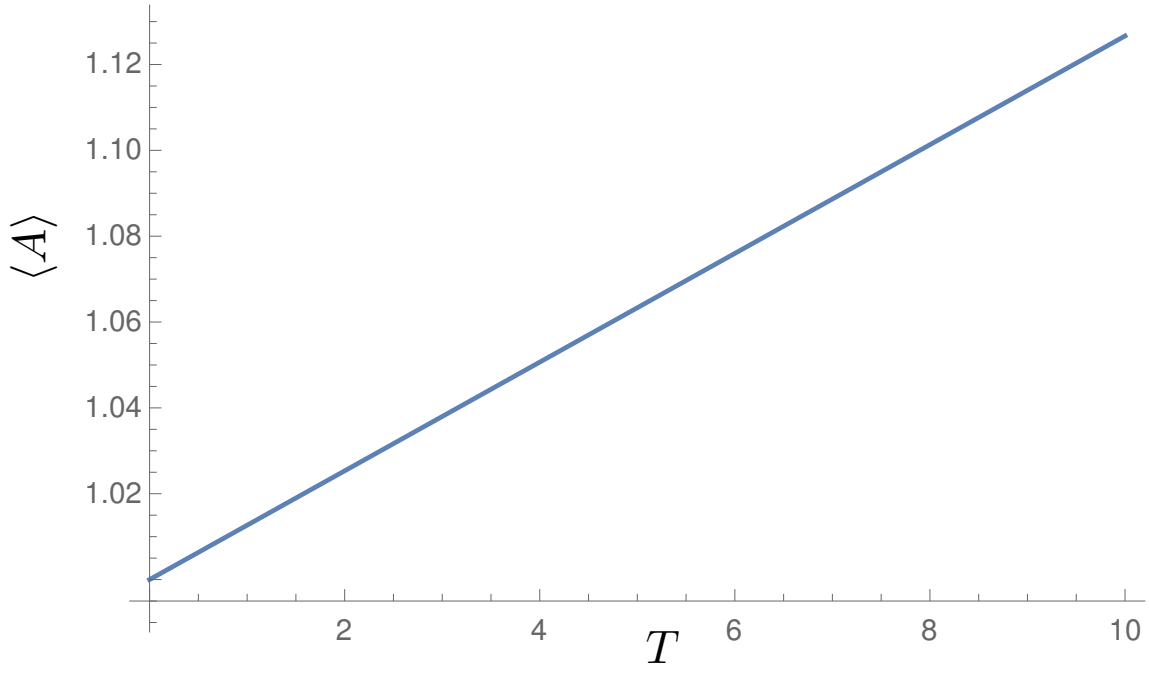

Figure 5.5: The average membrane area as a function of temperature for an elastic membrane. The average area depends linearly on the temperature and does not diverge for a membrane without proteins.

static protein, to determine at what parameters this model breaks down. In this scenario the energy is given by

$$E = \frac{1}{L^2} \left(\sigma k^2 + \kappa k^4 \right) \tilde{a}_{\boldsymbol{k}}^2 - \frac{k_{\mathrm{B}} T}{2} \left(\partial_x h \left(\boldsymbol{r}_i \right) \right)^2 \tag{5.40}$$

and the partition function is

$$Z = \int \mathrm{d}\tilde{a}_{\boldsymbol{k}} \mathrm{d}x \, \mathrm{e}^{-E/k_{\mathrm{B}}T}. \tag{5.41}$$

Using equations (5.36) and (5.38) we can determine how the average are of the membrane is altered due to the protein. This is a complex integral which diverges. This divergence occurs at

$$-2k_{\mathrm{B}}T + L^2(k^2\kappa + \sigma) = 0 \tag{5.42}$$

and is shown in figure 5.7.

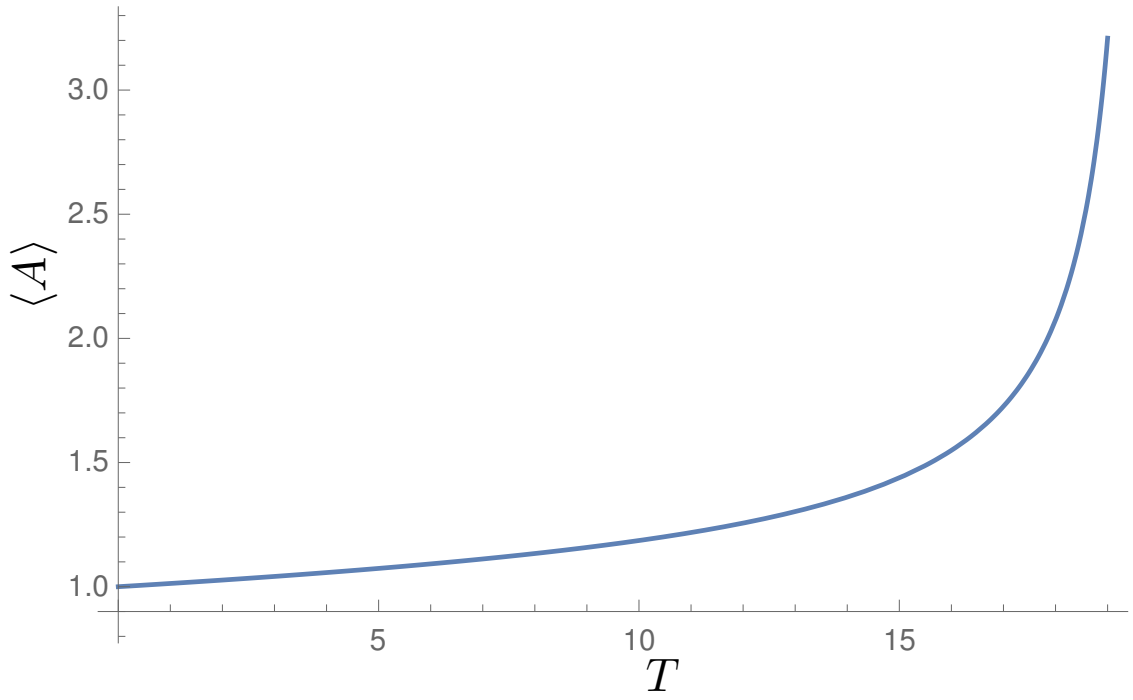

Figure 5.6: The average membrane area as a function of temperature for an elastic membrane with static proteins. The average area diverges when $\sigma = 2k_{\mathrm{B}}T/L^2$.

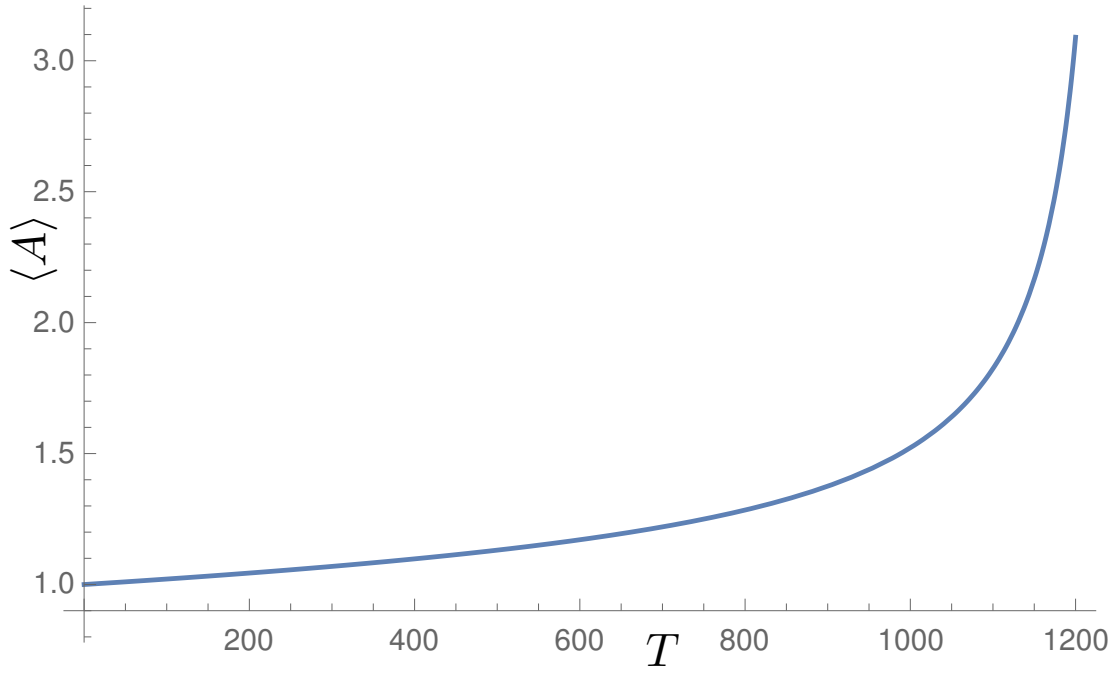

Figure 5.7: The average membrane area as a function of temperature for an elastic membrane with static proteins. The average area diverges when $\kappa = 2k_{\mathrm{B}}T/L^2k^2$.

From this we have the condition that if $\kappa = 0$, the model breaks down when $\sigma = 2k_{\mathrm{B}}T/L^2$ when $\kappa = 0$, $\kappa = 2k_{\mathrm{B}}T/L^2k^2$ when $\sigma = 0$, and only one Fourier mode is used. This provides a sufficient, but not necessary, condition of the model. For systems of a finite number of Fourier modes greater than one this is not necessarily still a condition.

Chapter 6

Summary and Future Directions

We have presented a model that successfully couples a continuum description of a membrane with a discrete particle model of proteins used to investigate membrane-mediated protein interactions for multiple types of coupling schemes. In Chapter 3, our model describes the membrane-protein interaction as an harmonic potential, which constrains the membrane height to a reference plane and suppresses membrane fluctuations. We found that due to the quenching of membrane fluctuations, the membrane induces a long-range attractive protein-protein interaction that could potentially lead to protein bundling or clustering. We also determined that protein diffusion depends on $D_0\eta/\kappa$, where D_0 is the bare protein diffusion and η/κ is the ratio of solvent viscosity to membrane bending rigidity. For faster bare diffusion, D_0, this ratio relates to the membrane relaxation time, which slows protein diffusion. This implies a competition between the two present timescales; that of the protein versus that of the membrane.

In Chapter 4, we described a model that constrains membrane shape at the positions of the proteins. We found the membrane-protein interaction has no effect on membrane fluctuations or the protein pair correlation function. For a subset of the parameters used in Chapter 3 to determine the lateral protein diffusion, we did not see much of an effect on the protein diffusion. If we investigate further parameters sets we may be able to find a similar trend as for the height-coupling systems. If we were to investigate a non-zero protein curvature the results seen here may vary. We could expand on this work by including more degrees of freedom, i.e. rotational. The addition of rotational degrees of freedom is especially important for proteins with a non-symmetric shape.

In Chapter 5, we derived coupled Langevin equations for the Fourier amplitudes of the membrane and the protein positions that take into account the shape of the membrane, which is time-dependent. The model we derived is consistent with detailed balance, unlike previous models that simulate similar systems [27, 28]. We find that membrane fluctuations are altered due to the effective energy (equation (5.13)). This varies from previous models, where the membrane behaves as a freely fluctuating membrane. We compared the lateral protein diffusion constant obtained from our presented method with that of previous methods [27]. We found that protein diffusion in the model presented in Chapter 5 does not see as strong of a decrease in protein diffusion. We also found that the described model leads to an increase in membrane fluctuations for the short wave-vector modes.

We presented a method by which this model can be expanded to a curvature-coupled membrane-protein system, similar to that investigated in Chapter 4. In the future it would be beneficial to further investigate such systems using this model, in order to determine how the membrane mediates interactions, and how the membrane shape and fluctuations are altered due to the presence of the proteins. One could also choose non-symmetric proteins and include rotational degrees of freedom for the proteins as previously discussed.

Overall, we were able to determine that while the molecular detail is important in membrane-protein interactions, a lot can be learned about the effect of binding on membrane shape and fluctuations from coarse-grained models. For the two types of membrane-protein interactions we discussed, we were able to investigate membrane fluctuations, protein-protein interactions, and protein diffusion in order to see how the different coupling schemes alter these properties.

Bibliography

[1] F. L. H. Brown. Elastic modeling of biomembranes and lipid bilayers. *Ann. Rev. Phys. Chem.*, 59:685–712, 2008.

[2] A. Frost, R. Perera, A. Roux, K. Spasov, O. Destaing, E. H. Egelman, P. de Camilli, and V. M. Unger. Structural basis of membrane invagination by F-BAR domains. *Cell*, 132(5):807–817, 2008.

[3] A. Frost, V. M. Unger, and P. de Camilli. Boomerangs, bananas and blimps: Structure and function of FBAR domains in the context of the BAR domain superfamily. In P. Aspenström, editor, *The Pombe Cdc15 Homology Proteins*. Landes Bioscience, 2009.

[4] A. Frost, V. M. Unger, and P. de Camilli. The BAR domain superfamily: Membrane-molding macromolecules. *Cell*, 137(2):191–196, 2009.

[5] M. Goulian, R. Bruinsma, and P. Pincus. Long-range forces in heterogeneous fluid membranes. *Europhys. Lett.*, 22:145, 1993.

[6] J.-M. Park and T. C. Lubensky. Interactions between membrane inclusions on fluctuating membranes. *J. Phys. I France*, 6(9):1217–1235, 1996.

[7] K. S. Kim, J. Neu, and G. Oster. Curvature-mediated interactions between membrane proteins. *Biophys. J.*, 75(5):2274–2291, 1998.

[8] L. C.-L. Lin and F. L. H. Brown. Dynamic simulations of membranes with cytoskeletal interactions. *Phys. Rev. E*, 72:011910, 2005.

[9] P. D. Blood and G. A. Voth. Direct observation of bin/amphiphysin/rvs (bar) domain-induced membrane curvature by means of molecular dynamics simulations. *Proc. Natl. Acad. Sci. USA*, 103(41):15068–15072, 2006.

[10] B. J. Reynwar, G. Illya, V. A. Harmandaris, M. M. Müller, K. Kremer, and M. Deserno. Aggregation and vesiculation of membrane proteins by curvature-mediated interactions. *Nature*, 447:461–464, 2007.

[11] M. C. Heinrich, B. R. Capraro, A. Tian, J. M. Isas, R. Langen, and T. Baumgart. Quantifying membrane curvature generation of drosophila amphiphysin N-BAR domains. *J. Phys. Chem. Lett.*, 1(23):3401–3406, 2010.

[12] M. Simunovic, A. Srivastava, and G. A. Voth. Linear aggregation of proteins on the membrane as a prelude to membrane remodeling. *Proc. Natl. Acad. Sci. USA*, 110(51):20396–20401, 2013.

[13] Z. Shi and T. Baumgart. Membrane tension and peripheral protein density mediate membrane shape transitions. *Nat. Comm.*, 6:5974, 2014.

[14] Z. Chen, Z. Shi, and T. Baumgart. Regulation of membrane-shape transitions induced by I-BAR domains. *Biophys. J.*, 109(2):298–307, 2015.

[15] M. Laradji and P. B. Sunil Kumar. Dynamics of domain growth in self-assembled fluid vesicles. *Phys. Rev. Lett.*, 93:198105, 2004.

[16] I. R. Cooke, K. Kremer, and M. Deserno. Tunable generic model for fluid bilayer membranes. *Phys. Rev. E*, 72:011506, 2005.

[17] M. Venturoli, M. M. Sperotto, M. Kranenburg, and B. Smit. Mesoscopic models of biological membranes. *Phys. Rep.*, 437(1–2):1–54, 2006.

[18] S. J. Marrink, H. J. Risselada, S. Yefimov, D. P. Tieleman, and A. H. de Vries. The MARTINI force field: Coarse grained model for biomolecular simulations. *J. Phys. Chem. B*, 111(27):7812–7824, 2007.

[19] G. S. Ayton, P. D. Blood, and G. A. Voth. Membrane remodeling from n-bar domain interactions: Insights from multi-scale simulation. *Biophys. J.*, 92(10):3595–3602, 2007.

[20] A. Pasqua, L. Maibaum, G. Oster, D. A. Fletcher, and P. L. Geissler. Large-scale simulations of fluctuating biological membranes. *J. Chem. Phys.*, 132:154107, 2010.

[21] L. Chen, G. Lianghui, F. Weihai, and L. Golubovic. How the antimicrobial peptides kill bacteria: Computational physics insights. *Commun. Comput. Phys*, 11(3):709–725, March 2012.

[22] L. Chen, N. Jia, G. Lianghui, F. Weihai, and L. Golubovic. Effects of antimicrobial peptide revealed by simulations: Translocation, pore formation, membrane corrugation and Euler buckling. *Int. J. Mol. Sci.*, 14(4):7932–7958, April 2013.

[23] A. Srivastava and G. A. Voth. Hybrid approach for highly coarse-grained lipid bilayer models. *J. Chem. Theo. Comp.*, 9(1):750–765, 2013.

[24] M. Simunovic, E. Evergren, I. Golushko, C. Prevost, H. F. Renard, L. Johannes, McMahon H. T., V. Lorman, G. Voth, and P. Bassereau. How curvature-generating proteins build scaffolds on membrane nanotubes. *Proc. Natl. Acad. Sci.*, 113(40):11226–11231, 2016.

[25] P. B. Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the red blood cell. *J. Theor. Biol.*, 26:61–81, 1970.

[26] W. Helfrich. Elastic properties of lipid bilayers - Theory and possible experiments. *Z. Naturforsch.*, C 28:693–703, 1973.

[27] E. Reister-Gottfried, S. M. Leitenberger, and U. Seifert. Hybrid simulations of lateral diffusion in fluctuating membranes. *Phys. Rev. E*, 75(1):011908, 2007.

[28] A. Naji, P. J. Atzberger, and F. L. H. Brown. Hybrid elastic and discrete-particle approach to biomembrane dynamics with application to the mobility of curved integral membrane proteins. *Phys. Rev. Lett.*, 102:138102, 2009.

[29] J. K. Sigurdsson, F. L. H. Brown, and P. J. Atzberger. Hybrid continuum-particle method for fluctuating lipid bilayer membranes with diffusing protein inclusions. *J. Comp. Phys.*, 252:65–85, 2013.

[30] K. Sapp and L. Maibaum. Suppressing membrane height fluctuations leads to a membrane-mediated interaction among proteins. *Phys. Rev. E*, 94(5):052414, Nov 2016.

[31] J. C. Stachowiak, C. C. Hayden, and D.Y. Sasaki. Steric confinement of proteins on lipid membranes can drive curvature and tubulation. *Proc. Natl. Acad. Sci.*, 107(17):7781–7786, 2010.

[32] J. C. Stachowiak, E. M. Schmid, C. J. Ryan, H. S. Ann, D. Y. Sasaki, M. B. Sherman, P. L. Geissler, D. A. Fletcher, and C. C. Hayden. Membrane bending by proteinprotein crowding. *Nat. Cell Biol.*, 14(9):944–949, 2012.

[33] Z. Shi and T. Baumgart. Membrane tension and peripheral protein density mediate membrane shape transitions. *Nat. Commun.*, 6(May 2014):5974, 2015.

[34] A. P. Liu, D. L. Richmond, L. Maibaum, S. Pronk, P. L. Geissler, and D. A. Fletcher. Membrane-induced bundling of actin filaments. *Nat. Phys.*, 4:789–793, 2008.

[35] O. Farago. Statistical thermodynamics of adhesion points in supported membranes. *Advances in Planar Lipid Bilayers and Liposomes*, 14:129–155, 2011.

[36] C. Yolcu, R. Haussman, and M. Deserno. The effective field theory approach towards membrane-mediated interactions between particles. *Adv. Colloid Interface Sci.*, 208:89–109, 2014.

[37] T. Speck, E. Reister, and U. Seifert. Specific adhesion of membranes: Mapping to an effective bond lattice gas. *Phys. Rev. E*, 82:021923, 2010.

[38] T. Speck. Effective free energy for pinned membranes. *Phys. Rev. E*, 83:050901(R), 2011.

[39] L. C.-L. Lin and F. L. H. Brown. Brownian dynamics in Fourier space: Membrane simulations over long length and time scales. *Phys. Rev. Lett.*, 93:256001, 2004.

[40] S. A. Safran. *Statistical Thermodynamics of Surfaces, Interfaces and Membranes.* Addison–Wesley Publishing, 1994.

[41] K. Sapp, R. Shlomovitz, and L. Maibaum. Seeing the forest in lieu of the trees: Continuum simulations of cell membranes at large length scales. In Ralph A. Wheeler, editor, *Annual Reports in Computational Chemistry*, volume 10, pages 47–76, Amsterdam, 2014. Elsevier.

[42] B. A. Camley and F. L. H. Brown. Dynamic simulations of multicomponent lipid membranes over long length and time scales. *Phys. Rev. Lett.*, 105:148102, 2010.

[43] J. D. Weeks, D. Chandler, and H. C. Andersen. Role of repulsive forces in determining the equilibrium structure of simple liquids. *J. Chem. Phys.*, 54(12):5237–5247, 1971.

[44] J.-P. Hansen and I. R. McDonald. *Theory of Simple Liquids.* Academic Press, 4th edition, 2013.

[45] F. L. Brown, D. M. Leitner, J. A. McCammon, and K. R. Wilson. Lateral diffusion of membrane proteins in the presence of static and dynamic corrals: suggestions for appropriate observables. *Biophys. J.*, 78(5):2257–2269, may 2000.

[46] E. Reister and U. Seifert. Lateral diffusion of a protein on a fluctuating membrane. 71(September):6, 2005.

[47] E. Reister-Gottfried, S. M. Leitenberger, and U. Seifert. Diffusing proteins on a fluctuating membrane: Analytical theory and simulations. *Phys. Rev. E*, 81(3):031903, 2010.

[48] S. M. Leitenberger, E. Reister-Gottfried, and U. Seifert. Curvature coupling dependence of membrane protein diffusion coefficients. *Langmuir*, 24(4):1254–1261, 2008.

[49] R. Golestanian, M. Goulian, and M. Kardar. Fluctuation-induced interactions between rods on membranes and interfaces. *Europhys. Lett.*, 33(3):241–245, 1996.

[50] R. Golestanian, M. Goulian, and M. Kardar. Fluctuation-induced interactions between rods on a membrane. *Phys. Rev. E*, 54:6725, 1996.

[51] B. Antonny. Membrane deformation by protein coats. *Cur. Op. Cell Biol.*, 18(4):386–394, 2006.

[52] G. S. Ayton, E. Lyman, V. Krishna, R. D. Swenson, C. Mim, V. M. Unger, and G. A. Voth. New insights into BAR domain-induced membrane remodeling. *Biophys. J.*, 97(6):1616–1625, 2009.

[53] A. Arkhipov, Y. Yin, and K. Schulten. Four-scale description of membrane sculpting by bar domains. *Biophys. J.*, 95(6):2806–2821, 2008.

[54] Y. Yin, A. Arkhipov, and K. Schulten. Simulations of membrane tubulation by lattices of amphiphysin N-BAR domains. *Structure*, 17(6):882–892, 2009.

[55] H. Cui, C. Mim, F. X. Vazquez, E. Lyman, V. M. Unger, and G. A. Voth. Understanding the role of amphipathic helices in N-BAR domain driven membrane remodeling. *Biophys. J.*, 104(2):404–411, 2013.

Appendix A

Python scripts for Analytical Calculations

In this appendix are Python scripts that perform the different analytical calculations presented in this thesis.

A.1 Example Parameter File

This is an example parameters file that can be sued with the following scripts as well as with scripts that perform the numerical simulations.

```
import numpy as np

# required input for system definition
 # Membrane dimensions
Lx = 50.
Ly = 50.
A = Lx * Ly

 # Number of wavevectors in each direction
P = 32
Q = 32

 # Unit constants
T = 1. # Temperature
sig = 1. # sigma in the WCA potential
rc = 2.**(1./6.)*sig # WCA cut off distance

# Proteins' constants
N = 100 # Number of Proteins
D0 = 1.0
gamma = D0/T # friction coeff
eps = 100. # interaction strength
```

```
l = sig/5. # interaction length
Cp = 0.75
Eps = 1. # WCA energy

# Membrane constants
sigma = 0. # surface tension
kappa = 1. # bending rigidity
eta = 0.01  # viscosity
lamb = 4.*eta # hydrodynamic kernel
Gamma = 1./lamb # friction coeff for the membrane


# derived quantities
dkx = 2*np.pi/Lx  # spacing of Fourier modes in x direction
dky = 2*np.pi/Ly  # spacing of Fourier modes in y direction

kmax = np.sqrt((P*dkx)**2 + (Q*dky)**2)                          # magnitude of largest
    wavevector
tau_min = 4. * eta / (kappa * kmax**3 + sigma * kmax)    # relaxation timescale of fastest
    membrane mode
kmin = np.min((dkx,dky))                                         # magnitude of smallest
    wavevector
tau_max = 4. * eta / (kappa * kmin**3 + sigma * kmin)    # relaxation timescale of slowest
    membrane mode

dt = ((2*np.pi/kmax)**2/(4*T*gamma))/1000 # timestep
tmax = 2E4 # total simulation time
```

A.2 Variance of Gaussian Integral

This is an example Python script that calculates the numerical matrix inversion laid out in 3.5.1. This example only calculates the analytical spectrum for the types of systems discussed in Chapter 3. Similar scripts can be written for the systems described in Chapters 4 and 5.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

out_filename = "an_N100_p32q32_kapppa1.dat"
```

```
6
7  from parameters100 import *
8
9  # protein positions
10 X = Lx*np.random.rand(N)
11 Y = Ly*np.random.rand(N)
12 Z = np.zeros(N)
13 rp = column_stacl((X,Y,Z))
14 # check protein positions
15 i = 0
16 while i < N-1:
17     rij = rp[i+1,:]-rp[i}
18     rij[:,0] -= Lx*np.round(rij[:,0]/Lx)
19     rij[:,1] -= Ly*np.round(rij[:,1]/Ly)
20     r = np.sqrt(rij[:,0]**2+rij[:,1]**2+rij[:,2]**2)
21     for j in xrange(r.shape[0]):
22         if r[j] > rc:
23             i += 1
24         else:
25             rp[i+1] = np.column_stack((Lx*np.random.rand(1),Ly*np.random.rand(1), np.zeros
       (1))
26
27 nx = np.array(range(P+1)+range(-P,0))
28 ny = np.array(range(Q+1))
29
30 k = np.array([[dkx*m,0] for m in range(1,P+1)]).astype(np.float)
31 k = np.append(k,[[dkx*m,dky*n] for m in range(-P,P+1) for n in range(1,Q+1)], axis = 0)
32
33 k2 = np.zeros(k.shape[0], dtype = np.float)
34 for i in xrange(k.shape[0]):
35     k2[i] = np.sqrt(k[i,0]**2*dkx**2+k[i,1]**2*dky**2)
36
37 k3 = np.zeros((k.shape[0],N), dtype = np.float)
38 for i in xrange(k3.shape[0]):
39     for x in xrange(rp.shape[0]):
40         k3[i,x] = k[i,0]*dkx*rp[x,0]+k[i,1]*dky*rp[x,1]
41
42 C = np.zeros((2*k2.shape[0], 2*k2.shape[0]), dtype = np.float)
43 Cr = np.zeros((k2.shape[0], 2*k2.shape[0]), dtype = np.float)
44 Ci = np.zeros((k2.shape[0], 2*k2.shape[0]), dtype = np.float)
```

```
for i in xrange(k2.shape[0]):
    for j in xrange(2*k2.shape[0]):
        if j % 2 == 0:
            Cr[i,j] = (4*eps/(T*A**2))*np.sum(np.cos(k3[i])*np.cos(k3[int(j/2)]))
            Ci[i,j] = (-4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.cos(k3[int(j/2)]))
        else:
            Cr[i,j] = (-4*eps/(T*A**2))*np.sum(np.cos(k3[i])*np.sin(k3[int(j/2)]))
            Ci[i,j] = (4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.sin(k3[int(j/2)]))

for i in xrange(Cr.shape[0]):
    Cr[i,2*i] = (2/(T*A))*(sigma*k2[i]**2+kappa*k2[i]**4)+(4*eps/(T*A**2))*np.sum((np.cos(k3
    [i]))**2)
    Cr[i,2*i+1] = (-4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.cos(k3[i]))
    Ci[i,2*i+1] = (2/(T*A))*(sigma*k2[i]**2+kappa*k2[i]**4)+(4*eps/(T*A**2))*np.sum((np.sin(
    k3[i]))**2)
    Ci[i,2*i] = (-4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.cos(k3[i]))

for i in xrange(C.shape[0]):
    if i % 2 == 0:
        C[i] = Cr[int(i/2)]
    else:
        C[i] = Ci[int(i/2)]


Cinv = np.linalg.inv(np.matrix(C))

fluc = np.zeros(k2.shape[0], dtype = np.float)
for i in xrange(C.shape[0]):
    if i % 2 ==0:
        fluc[int(i/2)] = Cinv[i,i]+Cinv[i+1, i+1]
outputfile = open(out_filename, "w")
for i in xrange(k2.shape[0]):
    print>> outputfile, k2[i], fluc[i]
```

A.3 Membrane-Mediated Free Energy Calculation

This is an example Python script that calculates the numerical matrix determinant laid out in 3.5.2. This example only calculates the free energy for the types of systems discussed in

Chapter 3. Similar scripts can be written for the systems described in Chapters 4 and 5.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

out_filename = "gr_p32q32_kappa1_an.dat"
out_filename2 = "gr_p32q32_kappa1_an_2.dat"

from parameters100 import *

rp = np.array([[0.,0.], [0.,0.]])
dx = 0.5
dy = 0.5
Nstep = Lx

nx = np.array(range(P+1)+range(-P,0))
ny = np.array(range(Q+1))

k = np.array([[dkx*m,0] for m in range(1,P+1)]).astype(np.float)
k = np.append(k,[[dkx*m,dky*n] for m in range(-P,P+1) for n in range(1,Q+1)], axis = 0)

k2 = np.zeros(k.shape[0], dtype = np.float)
for i in xrange(k.shape[0]):
    k2[i] = np.sqrt(k[i,0]**2*dkx**2+k[i,1]**2*dky**2)

k3 = np.zeros((k.shape[0],N), dtype = np.float)
for i in xrange(k3.shape[0]):
    for x in xrange(rp.shape[0]):
        k3[i,x] = k[i,0]*dkx*rp[x,0]+k[i,1]*dky*rp[x,1]

C = np.zeros((2*k2.shape[0], 2*k2.shape[0]), dtype = np.float)
Cr = np.zeros((k2.shape[0], 2*k2.shape[0]), dtype = np.float)
Ci = np.zeros((k2.shape[0], 2*k2.shape[0]), dtype = np.float)

def V(r):
    if r >= rc:
        return 0.0
    else:
        return 4.0 * Eps * (sig**12*r**-12 - sig**6*r**-6) + Eps

```

```
40 def compute_wca():
41     for i in xrange(0,N-1):
42         for j in xrange(i+1,N):
43             rij = rp[j]-rp[i]
44             r = np.sqrt(np.dot(rij,rij))
45             wca = V(r)
46     return wca
47 
48 
49 def radial_avg(f):
50     P = (f.shape[0]-1)/2
51     nx = np.array(range(P+1)+range(-P,0))
52     Q = (f.shape[1]-1)/2
53     ny = np.array(range(Q+1)+range(-Q,0))
54     erg = []
55     indices = (nx**2)[:,np.newaxis]+(ny**2)[np.newaxis,:]
56     for ind in np.unique(indices):
57         erg.append(np.array((np.sqrt(float(ind)),np.mean(f[indices==ind]))))
58     erg = np.array(erg)
59     return erg[:,0], erg[:,1]
60 
61 F = np.zeros((Nstep/dx+1,Nstep/dy+1))
62 
63 countx = 0
64 while rp[1,0] <Nstep+dx:
65     rp[1,1] = 0
66     county=0
67     while rp[1,1] <Nstep+dy:
68         k3 = np.zeros((k.shape[0],N), dtype = np.float)
69         for i in xrange(k3.shape[0]):
70             for j in xrange(rp.shape[0]):
71                 k3[i,j] = k[i,0]*dkx*rp[j,0]+k[i,1]*dky*rp[j,1]
72 
73         for i in xrange(k2.shape[0]):
74             for j in xrange(2*k2.shape[0]):
75     if j % 2 == 0:
76                     Cr[i,j] = (4*eps/(T*A**2))*np.sum(np.cos(k3[i])*np.cos(k3[int(j/2)]))
77                     Ci[i,j] = (-4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.cos(k3[int(j/2)]))
78                 else:
79                     Cr[i,j] = (-4*eps/(T*A**2))*np.sum(np.cos(k3[i])*np.sin(k3[int(j/2)]))
```

```
80                      Ci[i,j] = (4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.sin(k3[int(j/2)]))
81
82          for i in xrange(Cr.shape[0]):
83              Cr[i,2*i] = (2/(T*A))*(sigma*k2[i]**2+kappa*k2[i]**4)+(4*eps/(T*A**2))*np.sum((
    np.cos(k3[i]))**2)
84              Cr[i,2*i+1] = (-4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.cos(k3[i]))
85              Ci[i,2*i+1] = (2/(T*A))*(sigma*k2[i]**2+kappa*k2[i]**4)+(4*eps/(T*A**2))*np.sum
    ((np.sin(k3[i]))**2)
86              Ci[i,2*i] = (-4*eps/(T*A**2))*np.sum(np.sin(k3[i])*np.cos(k3[i]))
87          for i in xrange(C.shape[0]):
88              if i % 2 == 0:
89                  C[i] = Cr[int(i/2)]
90              else:
91                  C[i] = Ci[int(i/2)]
92
93
94          (sign,logdet) = np.linalg.slogdet(np.matrix(C))
95          F[countx,county] = (1./2.)*sign*logdet - (P/2.)*np.log(2*np.pi)
96          rp[1,1]+=dx
97          county+=1
98      rp[1,0]+=dy
99      countx+=1
100
101 x,y = radial_avg(F)
102 x/=2.
103 y -= y[-1]
104
105 output = open(out_filename,"w")
106 for i in xrange(x.shape[0]):
107     print >> output, x[i], np.exp(-y[i])
108 output.close()
109 output2 = open(out_filename2, "w")
110 for i in xrange(x.shape[0]):
111     print >> output, x[i], y[i]
112 output2.close()
```

Appendix B

Scripts for Analysis Functions

In this appendix are Python scripts that perform different analysis functions on trajectories produced by the simulations.

B.1 Radial Distribution Functions

This is an example Python script that calculates the radial distribution function.

```
import matplotlib.pyplot as plt
import numpy as np
import xtc
import os

# parameters
Lx = 50.  # length in x direction
Ly = 50.  # length in y direction

nhis = 500 # number of bins
delg = Lx/(2.*nhis) # bin size
g = np.zeros(nhis)
a = 0

for root, dirs, files in os.walk('/home/kayla/Documents/pinnedmem/kappa/N2/kappa_inf/'):
    for f in files:
        a+=1
        gr = np.zeros(nhis)

        reader = xtc.XTCReader(os.path.join(root,f), firstframe=1, stepframe=1)
        framecount = 0   # number of frames that have been averaged over

        N = None

```

```
25          try:
26              for c in reader:
27
28                  if not N:
29                      # if this is the first frame that has been read, store the number of
    particles
30                      N = c.coords.shape[0]
31                  else:
32                      # if this is not the first frame, make sure that the number of particles
     did not change
33                      assert c.coords.shape[0] == N
34
35                  for i in xrange(0,N-1):
36                      for j in xrange(i+1,N):
37                          rij = c.coords[j]-c.coords[i]
38                          rij[0] -= Lx * np.round(rij[0]/Lx)
39                          rij[1] -= Ly * np.round(rij[1]/Ly)
40                          r = np.sqrt(rij[0]**2+rij[1]**2)
41                          if r < Lx/2.:
42                              ig = int(np.floor(r/delg))
43                              gr[ig] += 2
44
45                  framecount += 1
46          except IOError:
47              pass
48          rho = (N)/(Lx*Ly)
49          r = np.zeros(nhis, dtype= float)
50          for i in xrange(nhis):
51              r[i] = delg*(i+0.5)
52              ab = ((i+1)**2-i**2)*delg**2
53              nid = np.pi*ab*rho
54              gr[i] = gr[i]/(framecount*(N)*nid)
55          g+=gr
```

B.2 Diffusion Constants

This is an example Python script that calculates the diffusion constant. There are three methods within the file, however, in this thesis we used the first method, which calculates

the diffusion constant from the mean squared displacement.

```
import xtc
import copy
import os
import numpy as np
import scipy
from scipy import integrate
from scipy import stats
import matplotlib.pyplot as plt

### ALL METHODS ARE CURRENTLY CALCULATED USING SWITCHES, SO SET SWITCH = TRUE BEFORE EACH CALCULATION YOU WISH TO PERFORM ###

        ### Note that currently this calculates D/D0, so if you don't want the ratio to bare diffusion that will need to be changed ###

#Define system parameters (most of these are unnecessary are just included for completeness)
Lx = Ly = 50 # System in size in sigma
T = 1.0 #1. # thermal energy in k_BT
eta = 3.17E-9 # viscosity of water in kBTs/sigma^3
sigma = 0.0 # surface tension in kBT/sigma^2
kappa = 0.1*T #1.0 # bending rigidity in kBT

P = 1
Q = 1
dkx = 2*np.pi/Lx  # spacing of Fourier modes in x direction
dky = 2*np.pi/Ly  # spacing of Fourier modes in y direction
kmax = np.sqrt((P*dkx)**2 + (Q*dky)**2)
kmin = np.min((dkx,dky))

D0 = 2.0E8
gamma_x = T/D0 #1.0 # friction of prtein in membrane in kBTs/sigma^2
eps = 100.*T # pinning energy in kBT/sigma^2
l = 0

tau_min = 4. * eta*kmax / (kappa * kmax**4 + sigma * kmax**2)
tau_max = 4. * eta*kmin / (kappa * kmin**4 + sigma * kmin**2)

# Calculation of D using MSD method

```

```
38 def disp(x,y, i, sliding = False):
39  # function to find the displacements at each step, can be done using the sliding method of
       by using individual intervals
40     if sliding:
41         dx = x[i::]-x[:-i:]
42         dy = y[i::]-y[:-i:]
43         dr = np.sqrt(dx**2+dy**2)
44         return dr
45     else:
46         dx = x[i::i]-x[:-i:i]
47         dy = y[i::i]-y[:-i:i]
48         dr = np.sqrt(dx**2+dy**2)
49         return dr
50 
51 def diff(x, alpha):
52  # This calculates the diffusion constant and error bars
53     D = np.mean(x**2)
54     df = len(x)
55     s = np.sqrt(np.mean(x**2))
56     chi_left = scipy.stats.chi2.ppf(alpha/2, df)
57     chi_right = scipy.stats.chi2.ppf(1-alpha/2, df)
58     error_left = df*s**2/chi_left
59     error_right = df*s**2/chi_right
60     return D, error_left, error_right
61 
62 Switch = True #SET TO TRUE TO CALCULATE D USING MSD
63 if Switch:
64     f1 = open('msd.dat', "w")
65     f2 = open('D.dat', "a")
66     reader = xtc.XTCReader('trajectory.xtc')
67     Nstep = 0
68     N = None
69     x= []
70     y = []
71     try:
72         for c in reader:
73             if not N:
74                 N = c.coords.shape[0]
75             else:
76                 assert c.coords.shape[0] == N
```

```
77              x.append(copy.deepcopy(c.coords[:,0]))
78              y.append(copy.deepcopy(c.coords[:,1]))
79              Nstep+=1
80      except IOError:
81          pass
82      dt = (c.time/c.step)*100.
83      x = np.array(x)
84      y = np.array(y)
85      msd1=np.array([np.var(disp(x,y,t)) for t in range(1,Nstep-1)])
86      msd2=np.array([np.mean(disp(x,y,t)**2) for t in range(1,Nstep-1)])
87      t = np.arange(1,Nstep-1)*dt
88
89  # for which ever definition of msd you prefer print the data to file 1, this section also
        plots the msd on a log scale
90      for i in xrange(msd1.shape[0]):
91          print >> f1, t[i], msd1[i], msd2[i]
92      f1.close()
93
94      fig = plt.figure()
95      ax = fig.add_subplot(111)
96      ax.set_xlabel("Time t")
97      ax.set_ylabel("MSD")
98      ax.plot(dt*times, msd1, '*-', label="<x^2>-<x>^2")
99      ax.plot(dt*times, msd2, '*-', label="<x^2>")
100     ax.set_xscale('log')
101     ax.set_yscale('log')
102     ax.legend()
103
104 # calc D using msd/4t = D and the function defined above
105     times=2**np.arange(np.floor(np.log2(Nstep/5))+1,dtype=int)
106     D = np.zeros(times.shape[0])
107     D_error = np.zeros((times.shape[0],2))
108     for i in range(times.shape[0]):
109         D[i], D_error[i,0], D_error[i,1] = diff(disp(x,y,times[i]), 0.05)
110     D/=(4*dt*times)
111     D_error[:,0]/=(4*dt*times)
112     D_error[:,1]/=(4*dt*times)
113     error=[(D-D_error[:,0])/D0,(D_error[:,1]-D)/D0]
114     D/=D0 #comment this out if you aren't normalizing by D0
115
```

```
116  # comment this out if you do not want to see the plot of D/D0 vs t
117      fig = plt.figure()
118      ax = fig.add_subplot(111)
119      ax.set_xlabel("Time t")
120      ax.set_ylabel("D/D0")
121      ax.set_xscale("log")
122      ax.set_ylim(-2.,2.)
123      plt.errorbar(dt*times, D, yerr=error ,fmt='o')
124      plt.show()
125
126  # print the diffusion constants found to a file
127      for i in range(times.shape[0]):
128          print >>f2, dt*times[i], D[i], error[0][i], error[1][i]
129      print >>f2, "\n"
130      f2.close()
131
132 # Calculation for the force-force correlation method
133 def corr(x, tmax = None):
134  #calculates <F(0)F(t)>
135      if tmax is None or tmax > len(x)-1:
136          tmax = len(x)-1
137      corr = np.zeros(tmax+1)
138      for dt in xrange(tmax+1):
139          if dt == 0:
140              corr[dt] = np.mean(x**2)
141          else:
142              corr[dt] = np.mean(x[:-dt]*x[dt:])
143      return corr
144
145 Switch = False #SET TO TRUE TO CALCULATE D USING <F(0)F(t)>
146 if Switch:
147      f3 = open('force_corr.dat', "w")
148      f4 = open('D.dat', "a")
149      F = np.loadtxt('forces.dat')[:,3]
150      Fc = corr(F, tmax = 10000)
151      D = 0.5 * dt * (scipy.integrate.trapz(Fc))+1
152      t = np.arange(Fc.shape[0])*dt
153
154  # comment this section out if you do not want to plot the correlation function
155      fig = plt.figure()
```

```
156     ax = fig.add_subplot(111)
157     ax.sex_xlim(0,len(Fc)*dt)
158     ax.set_xlabel("Time t")
159     ax.set_ylabel("<F(0)F(t)>")
160     ax.plot(dt*np.arange(len(Fc)),Fc, label="<F(0)F(t)>")
161     plt.show()
162  # save to file
163     for i in xrange(t.shape[0]):
164         print >>f3, t[i], Fc[i]
165     f3.close()
166     print >> f4, D
167 
168 # Calculation of D using the probability distribution method
169 def func(x, D):
170     return (x/(2*D*interval))*np.exp(-x**2/(4*D*interval))
171 def prob(x, y, step):
172     dx = x[step::step]-x[:-step:step]
173     dy = y[step::step]-y[:-step:step]
174     dr = np.sqrt(dx**2+dy**2)
175     return dr
176 def prob2(d):
177     h, xedges = np.histogram(d, bins = P)
178     midpoints = (xedges[1:]+xedges[:-1])/2
179     binsize = xedges[1]-xedges[0]
180     C = 1/(binsize*sum(h))
181     return h, midpoints, C
182 
183 Switch = False #SET TO TRUE TO CALCULATE D USING PROBABILITY DISTRIBUTION
184 if Switch:
185     f5 = open('prob.dat', "w")
186     f6 = open('D.dat', "a")
187     Lx = 50. # length of membrane(box) in x-direction
188     Ly = 50. # length of membrane(box) in y-direction
189     dx = 5./10   # target bin size in x direction (might change)
190     dy = 5./10   # target bin size in y direction (might change)
191     P = int(np.round(((Lx/dx)-1)/2.))  # find number of bins
192     Q = int(np.round(((Ly/dy)-1)/2.))
193     dx = Lx/(2*P+1)      # find bin size
194     dy = Ly/(2*Q+1)
195     step = 10
```

```
196     reader = xtc.XTCReader('trajectory.xtc')
197     Nstep = 0
198     N = None
199     x= []
200     y = []
201     try:
202         for c in reader:
203             if not N:
204                 N = c.coords.shape[0]
205             else:
206                 assert c.coords.shape[0] == N
207             x.append(copy.deepcopy(c.coords[:,0]))
208             y.append(copy.deepcopy(c.coords[:,1]))
209             Nstep+=1
210     except IOError:
211         pass
212     dt = (c.time/c.step)*100.
213     x = np.array(x)
214     y = np.array(y)
215     t = np.arange(Nstep)*dt
216     interval = step*dt
217     dr = prob(x,y,step)
218     h, midpoints, C = prob2(dr)
219     popt, pcov = scipy.optimize.curve_fit(func, midpoints, h*C)
220 
221 # comment this section out if you do not want to plot
222     fig = plt.figure()
223     ax1 = fig.add_subplot(211)
224     ax2 = fig.add_subplot(212)
225     ax1.plot(midpoints, C*h)
226     ax2.plot(midpoints,func(midpoints,*popt))
227     plt.draw()
228 
229 # Save data to file
230     D = popt
231     error = pcov
232     print >> f5, midpoints, C*h
233     print >> f6, interval, D, error
```