Physical principles of membrane reshaping

A theoretical study of self organization on membrane surfaces through membrane-mediated interactions

Dadunashvili, G.

DOI
10.4233/uuid:4971c301-0234-4762-8a67-74337f6acbbc

Publication date
2022

Document Version
Final published version

Citation (APA)
Dadunashvili, G. (2022). *Physical principles of membrane reshaping: A theoretical study of self organization on membrane surfaces through membrane-mediated interactions*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:4971c301-0234-4762-8a67-74337f6acbbc

Physical principles of membrane reshaping

A theoretical study of self organization on membrane surfaces through membrane-mediated interactions

PHYSICAL PRINCIPLES OF MEMBRANE RESHAPING

A THEORETICAL STUDY OF SELF ORGANIZATION ON MEMBRANE SURFACES THROUGH MEMBRANE-MEDIATED INTERACTIONS

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Wednesday 21 September 2022 at 10:00 o'clock

by

George DADUNASHVILI

Master of Science in Physics,
Ludwig Maximilians-Universiteit, München, Germany,
born in Tbilisi, Georgia.

This dissertation has been approved by the promotors

promotor: Dr. T. Idema
promotor: Prof. dr. A. M. Dogterom

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Dr. T. Idema,	Delft University of Technology, promotor
Prof. dr. A.M. Dogterom,	Delft University of Technology, copromotor
Independent members:	
Prof. dr. S.J. Marrink	University of Groningen
Prof. dr. G. Gompper	University of Cologne
Prof. dr. J.M. Thijssen	Delft University of Technology
Dr. M.S. Bauer	Delft University of Technology
Dr. L. Giomi	Leiden University
Prof.dr. C. Joo	Delft University of Technology, reserve member

Keywords: Membrane mediated interactions, Canham-Helfrich energy, Dynamically triangulated membrane Monte Carlo simulations

Front & Back: 'From guessing red blood cell shapes to triangulated membranes' by George Dadunashvili

Casimir PhD Series, Delft-Leiden 2022-25

ISBN 978-90-8593-535-3

An electronic version of this dissertation is available at
`https://repository.tudelft.nl/`.

Do not try the parallels in that way: I know that way all along. I have measured that bottomless night, and all the light and all the joy of my life went out there.

Farkas Bolyai writing to his son János

CONTENTS

SUMMARY

The research described in this doctoral thesis was part of the *BaSyC* consortium, which aims to build synthetic cells from bottom up. My small role in this large collaboration was to study the theoretical basis for cell division. This is a very complex process in living organisms, involves many components and can be accomplished in a variety of ways. To successfully replicate cell division in synthetic organisms it will be important to identify the simplest mechanisms and understand the basic components of the cell division machinery, that come together to reshape membranes and divide one compartment into two. The theoretical framework used to study these self-assembly processes, that lead to membrane deformations and division, is the framework of membrane mediated interactions. Let me use an example to explain this framework.

Imagine you sit down on a couch, while holding a saucer with a coffee cup on it. The coffee is too hot, so you set it down next to you ont the couch and while you wait you get lost in your thoughts...You feel like getting up and walking around in the room. While you are still lifting your body up from the sitting position, you hear the clinking of the cup against the saucer, and you see from the corner of your eye that the coffee cup is shaking. Without having touched the cup you managed to spill the coffee. The surface of the couch was deformed and pulled tight because you were sitting on it. When you got up the surface relaxed back to its original shape. This movement of the surface caused the cup to shake and spill its contents (see fig. 1 for an artist's impression). Membrane mediated interactions are just like that! When something deforms a membrane in one place, that deformation is felt at a distance by other objects that inhabit the membrane. Exactly how far and how strongly such signals are felt depends on how floppy or tense a membrane is. When we study membrane mediated interactions, our main goal is to quantify the effects membrane deforming objects can have on each other.

In **chapter 1** I give an overview of the biological context relevant to my work, I introduce the key elements of the membrane physics that help us to describe membrane-mediated interactions quantitatively, and finally I discuss several computational methods that are widely used in membrane physics.

In **chapter 2** I show how point like objects that generate forces on membranes, can affect each other and bundle and perform collective action on the surface of a membrane.

In **chapter 3** I look at a model system of membrane tubes, covered by proteins that can force the membrane to change its curvature. The research of protein decorated membrane tubes is a highly active research field, motivated by the ubiquity of tubular membrane shapes in biology.

Having met the limitations of purely analytical approaches in chapters 2 and 3, in **chapter 4** I introduce flippy, a simulation framework that I developed during my PhD. flippy is a c++ library that implements a very popular simulation method of dynamically triangulated membranes. flippy aims to streamline the process of setting up a simulation

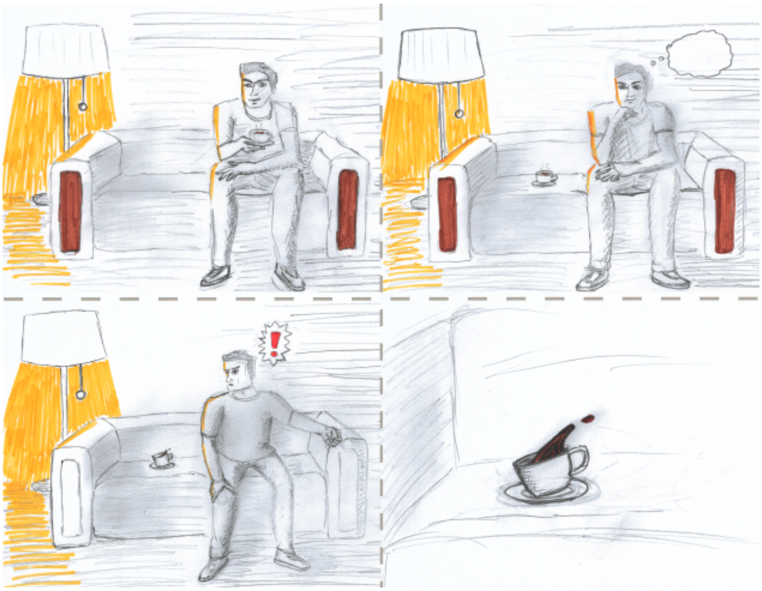

Figure 1: **An artist's impression of couch mediated interactions.** (top left) Sitting down on the couch with a cup of coffee. (top right) Setting down the coffee and getting lost in thoughts. (bottom left) Getting up from the couch. (bottom right) Distortions in the couch tip the coffee cup off its saucer and spill the coffee.
Een artistieke impressie van sofa-gefaciliteerde interacties. (links boven) men gaat op de sofa zitten met een kop koffie. (rechts boven) men zet de kop koffie neer en raakt verdwaalt in eigen gedachten. (links onder) men staat op van de sofa. (rechts onder) Verstoringen in de kussens van de sofa kieperen het kopje van het schoteltje en morst de koffie.

and helps physicists to skip the boring parts of reimplementing known algorithms, and get straight to their research questions.

Chapter 5 is a study of emergent order in systems of self-propelled communicating agents, like a swarm of ants or grasshoppers. This chapter is unrelated to the main topic of my PhD, and grew out of a bachelor thesis that was jointly supervised in Idema and Dubbeldam groups. In this work, my collaborators, and I managed to formulate a model of behaviour of swarms, based on complex network theory and then subsequently solve it analytically.

The final **chapter 6** contains a discussion of preliminary simulation results of membrane vesicle division, connecting back to the overarching goal of *BaSyC*, and ends with my concluding remarks.

Samenvatting

Het onderzoek in deze dissertatie is onderdeel van het BaSyC consortium, wat als doel heeft om synthetische cellen te bouwen uit alle benodigde bouwsteentjes ("bottom up"). Mijn bescheiden rol in dit grote samenwerkingsverband was het bestuderen van de onderliggende theorie van celdeling. Celdeling is een zeer complex proces in levende organismen, omdat er vele componenten van de cel voor nodig zijn en omdat een cel zich op verschillende manieren kan delen. Om de celdeling succesvol in synthetische organismen na te bootsen, is het dus belangrijk om de essentiële componenten van de celdeling te begrijpen en bovendien het eenvoudigste mechanisme voor celdeling te identificeren – de componenten die samen de cel en zijn membranen vormgeven en de cel in tweeën delen. Het theoretische framework van membraan-gefaciliteerde interacties wordt gebruikt voor studies naar deze zelfbouw processen die vervorming van membranen en celdeling tot gevolg hebben. Ik gebruik een voorbeeld om dit framework te illustreren.

Stel dat u op een bank gaat zitten terwijl u een schoteltje met een kop koffie vasthoudt. De koffie is nog te heet om te drinken, waardoor u het al wachtend naast u op de bank neerzet en in gedachten verzonken raakt... U voelt de behoefte om op te staan en door de kamer te lopen. Terwijl u opstaat uit de zittende houding, hoort u het kopje tegen het schoteltje kletteren, en ziet het kopje uit uw ooghoek bewegen. U heeft de koffie gemorst zonder het kopje aan te raken. Het oppervlak van de bank vervormde en werd strak getrokken terwijl u op de bank zat. Toen u opstond, ontspande het oppervlak naar zijn oorspronkelijke vorm. Deze beweging van het oppervlak zorgde voor de beweging van het kopje en het morsen van de koffie (fig. 1). Membraan-gefaciliteerde interacties werken precies hetzelfde! Als een object het membraan van vorm veranderd, dan wordt deze verandering op een afstand ook door andere objecten op het membraan gevoeld. Hoe ver en hoe sterk zulke signalen reiken hangt af van hoe los of strak het membraan is. Met het bestuderen van membraan-gefaciliteerde interacties is ons hoofddoel om de effecten te kwantificeren die objecten op elkaar hebben tijdens het vervormen van membranen.

In **hoofdstuk 1** geef ik een overzicht van de biologische achtergrond die relevant is voor mijn werk. Ik introduceer de essentiële elementen van membraanfysica waarmee de membraan-gefaciliteerde interacties kwantitatief beschreven kunnen worden. Dit hoofdstuk sluit af met een discussie van verschillende computationele methoden die veel gebruikt worden in de membraanfysica.

In **hoofdstuk 2** laat ik zien hoe puntige objecten die kracht uitoefenen op membranen elkaar kunnen beïnvloeden en een gezamenlijke werking op het oppervlak van een membraan kunnen hebben.

In **hoofdstuk 3** beschouw ik een modelsysteem van membraanbuisjes die bedekt zijn met eiwitten. Deze eiwitten kunnen het membraan dwingen om zijn kromming te veranderen. Er is zeer veel onderzoek naar membraanbuisjes bedekt met eiwitten, gemotiveerd door de alomtegenwoordigheid van buisvormige membranen in de biologie.

Hoofdstukken 2 en 3 leggen de limieten van de puur analytische methoden bloot. Met deze kennis, introduceer ik Flippy in **hoofdstuk 4**. Flippy is een framework voor simulaties dat ik heb ontwikkeld tijdens mijn PhD. Het is een c++ bibliotheek die dynamisch getrianguleerde membranen implementeert, een veelgebruikte methode voor simulaties. Door te helpen met het saaie werk, zoals het implementeren van bekende algoritmes en het ontwerpen van simulaties, helpt Flippy om simulaties te stroomlijnen en natuurkundigen hun onderzoek te laten doen.

Hoofdstuk 5 bevat een studie van orde die ontstaat in systemen van zelf aangedreven, communicerende agenten. Voorbeelden hiervan zijn een zwerm van sprinkhanen of mieren. Dit hoofdstuk is niet direct gerelateerd aan de hoofdlijnen van mijn PhD, en ontstond uit een bachelor thesis die gezamenlijk werd begeleid door de Idema en Dubbeldam groepen. In dit werk is het mij en mijn collega's gelukt om een model voor het gedrag van zwermen, gebaseerd op de theorie van complexe netwerken, te formuleren en dit analytisch op te lossen.

Tot slot keert **hoofdstuk 6** terug naar het doel van BaSyC met discussie van verkennende simulaties van deling van membraanblaasjes, eindigend met mijn slotopmerkingen

1

INTRODUCTION

Every science begins as philosophy and ends as art; it arises in hypothesis and flows into achievement.

Will Durant

Physics as the oldest among the sciences has long gone through it's first two phases of philosophy and science and has become art. We aspire to create a body of knowledge that did not exist before, like a painter filling an empty canvas with its creation. Like the painter we let ourselves be guided by the beauty of the nature and our intuition. The numbers and the equations are our tools and to say physics is about them is to say that painting is about brushes and canvases. Just as one may enhance the enjoyment of a piece of art by learning about the context of its creation, so can one enhance the enjoyment and understanding of science. The chapters of this thesis are the result of creative and scientific enterprise of my collaborators and me, and I strongly believe that they can stand on their own. However, I also believe that the introduction chapter provides an important context which will enhance the understanding and enjoyment of the subsequent chapters, by introducing the reader to the field of theoretical and computational membrane physics.

1.1. WHY STUDY CELL DIVISION

During my PhD I was often asked to explain what I was working on. When I would say that my thesis was related to studying cell division, I would always[1] get the same reaction: people would be surprised that there is still something left to study. When I first interviewed for this PhD position in late 2017 and became vaguely aware, that the research project was related to cell division, I also felt the same confusion. I remembered that I had learned in school how cells divide, so I assumed this was a solved problem. So how come they were still paying researchers to investigate it?

It is true we learn in school about cell division. We may learn which protein complexes participate in it, which role different proteins perform what they do to the cell membrane, how the material inside the cell redistributes, and we even learn how long all of this takes. Even though a large variation in types of cells exists, we have a pretty good idea how division machinery changes between them. However, when we reconstitute proteins that participate in cell division *in vivo* inside membrane vesicles, the division process does not take place, even though large scale membrane deformations are usually observed [1–3]. Examples of successful division experiments exist [4, 5], but they are neither self-sustaining, nor is it clear how one would make them self-sustaining [6]. So what are we overlooking? Is there some key biochemical mechanism that we have not identified yet, is there something about the mechanics of the process that we do not understand and fail to replicate? These types of questions need to be answered if we want to have a true bottom up understanding of cell division. This is the difference; we know how cell division works, but we cannot divide cells. The former is comparatively easy, it requires knowledge of the actors and the processes during cell division. The latter is harder because it requires knowledge of all the peripheral things as well, so we cannot only know the actors but also understand what makes these interactions so robust. Cell division happens in vastly different organisms and environments after all. The precise goal of my research is thus not to study cell division in general, but to study one of the actors in that process, the cell membrane. And to further the understanding of membrane mechanics during large scale reshaping, so that we better understand the processes that lead to cell division.

In this introduction I will first elaborate on what cell membranes, or more precisely phospholipid membrane bilayers are and what role they play in biology. Then I will present the theoretical foundation with which we approach the study of membranes in physics. Finally, I will give a broad overview of computational methods, that are crucial in the study of large scale membrane deformations.

1.2. MEMBRANES, WHAT ARE THEY GOOD FOR?!

Biological membranes are made of lipids. These molecules have hydrophilic heads that like to dissolve in water and hydrophobic tails that are not soluble. A schematic representation of such a molecule is shown fig. 1.1 (a). This means that the energetically most favourable situation for a lipid molecule is for its head to be in contact with water and its tail to be hidden from it. Depending on the temperature and concentration of the lipids, different configurations for large scale lipid assemblies are possible [7]. However,

[1] Apart from the cases when I would be talking to biologists.

in the biologically relevant region the lipids form two layers where the tail ends point to each other and are hidden from water by the heads as shown in fig. 1.1 (b). This bilayer of lipids forms a thin sheet (see fig. 1.1 (c)), which then closes into a spherical vesicle (see fig. 1.1 (d)) to avoid heaving edges where the hydrophobic tails would be exposed to water. Such vesicles that are enclosed by a single bilayer of a membrane are called unilamellar vesicles. When these vesicles are very large, i.e. have a radius in the order of tens of micrometers, compared to the membrane thickness of a few nanometers, they are called giant unilamellar vesicles (GUV). GUVs are important model systems in biophysics and will feature in this thesis prominently.

Since lipid membranes self assemble purely because their configurations are energetically more favoured, there are no bonds that holds individual lipids together, and they can freely diffuse in the bilayer. This means that if a shear stress is applied to it, the bilayer will simply flow and dissipate the stress. This is a rare material property and means that membranes are two-dimensional fluids. Meanwhile, a stress, applied normal to the membrane, will shift the lipids out of the bilayer which will expose lipid tails to the water. Since this is unfavourable, lipid membranes resist deformations in the normal direction. The precise way that membranes resist normal deformations is given by the membrane surface energy. Knowledge of this energy allows us to predict membrane shapes under different conditions and is the central object of study in membrane physics, from the theoretical point of view. I will discuss this subject in more detail in section 1.3.

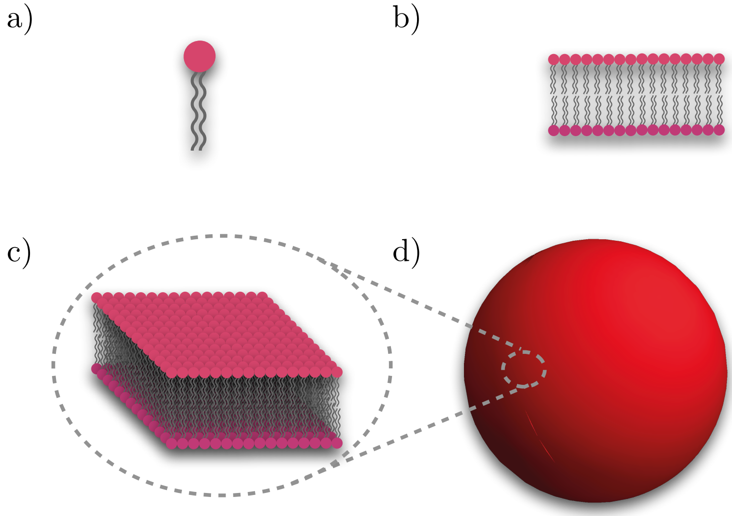

Figure 1.1: A schematic representation of a) a single lipid, b) a cross sectional view of a lipid bilayer, c) a sheet of a bilayer and d) unilamellar vesicle, enclosed by a lipid bilayer.

Before we dive into the technical details of vesicle shape prediction, it will be insightful to discuss the various biological systems where membranes play a key role. Conceptually the easiest job that membranes perform in biology is that of a barrier. They separate inside from outside, and thus allow to form small compartments that regulate their internal composition locally. The most famous example of this is of course a cell. However, cell membranes are much more complex than simple lipid bilayers as described above. They consist only to roughly 50% of lipids (by weight), the rest are proteins [8, Ch.10]. The lipid part of the membrane is not made of a single type of lipids but typically consists of a mixture of a large variety of them [9]. The proteins are dispersed in the lipid bilayer as depicted in fig. 1.2. This description of lipid membranes is called the fluid mosaic model and applies to most membranes that one encounters in biology [10]. This diverse composition is what allows membranes to perform the diverse functions that are required of them in an animal cell. The cell membrane itself needs to be more than just a barrier to the outside, it also has to actively participate in the regulation of the processes in the interior. This can be done by selectively regulating the flow of the material from the inside to the outside, trough membrane associated proteins. Or through adopting different shapes that further compartmentalize the interior space. Furthermore, during cell division the membrane has to undergo dramatic deformations while keeping structural integrity and maintaining its other functions. Animal cells also contain many small subunits called organelles that perform different functions, some of which are depicted in fig. 1.3. We see that the shapes that these organelles take can be very complex. Even the spherical looking nucleus from fig. 1.3 a) is a double-layered sphere with multiple holes and a connection of the outer layer with the endoplasmic reticulum (ER), a very important organelle that is among other things responsible for production of proteins and lipids. The ER which can be seen in fig. 1.3 b) has two subunits: the rough ER which also consists of folded double membrane and the smooth ER which consists of a network

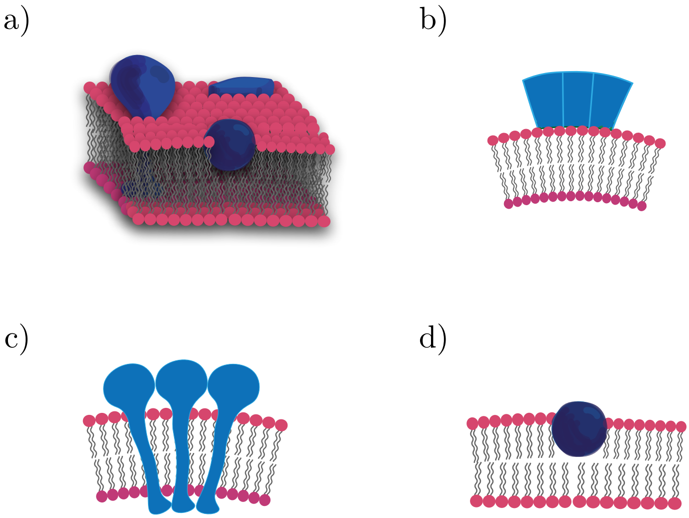

Figure 1.2: A schematic representation of the fluid mosaic model. a) Membrane layer with different types of proteins. b)-d) a cross sectional view of a lipid bilayer, with membrane adhered, trans membrane and membrane intercalated proteins in b) c) and d) respectively.

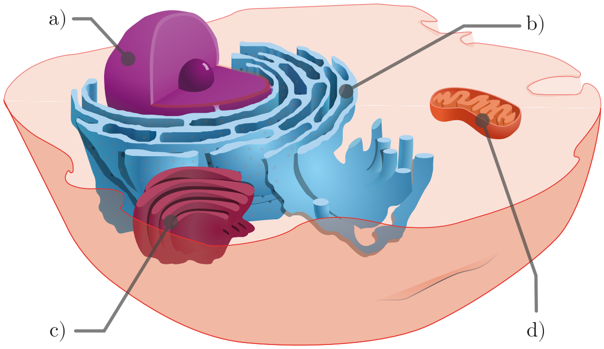

Figure 1.3: A schematic representation of an animal cell. In addition of being enveloped by membranes, cells also contain organelles that perform specific functions and are themselves encapsulated by membranes. a) Cell nucleus contains the DNA. b) Endoplasmic reticulum (ER) is responsible for the assembly of proteins and then their folding into their functional shapes. c) Golgi apparatus. Its job is to package proteins in small membrane envelopes for the transport to them where they are needed. d) Mitochondria, are responsible for production of ATP, which is a molecule that stores energy. Figure adapted from a public domain picture from wikimedia [11].

of tubes. The rough ER contains ribosomes (seen as the dark dots in fig. 1.3 c)), responsible for translating mRNA into amino acids. The ribosomes embed in the membrane after they encounter an mRNA and start translation. The function of smooth ER can vary in different cells, and its size and branching varies with it. The proteins that get assembled in the ER then move on to the Golgi apparatus (the organelle that has a stacked membrane structure seen in fig. 1.3 c)),which dispatches the proteins to their destinations [8, Ch.12]. Another very important organelle the mitochondrion also consist of a double membrane bilayer (seen in fig. 1.3 d)), where the inner membrane is folded several times to increase its surface area. This aids the mitochondria to increase its production capacity for ATP. The main function of mitochondria is the production of ATP molecules which are used in the cell as an energy source for protein activity. The proteins that catalyze the ATP synthesis inside the mitochondria are embedded on the inner side of the inner membrane, thus increasing the membrane area increases the production capacity [8, Ch.14].

We see in these examples that the form of the membrane can be directly influenced by the functional requirements of the organelle. This coupling of form to function is why it is so insightful to understand mechanical properties of membranes. By understanding what shapes membranes can adopt in different circumstances we can infer new information about their function [12].

1.3. MEMBRANE PHYSICS

The theory that we use to describe the response of the cell membranes to environmental stress, was first developed in the 1970s by Canham [13] and Helfrich [14]. The resulting membrane energy functional and its variants are called Canham-Helfrich energy. Though this energy functional can be arrived at from a microscopic theory, it can also be formulated by considering symmetry requirements on the system and understanding the energetics of different modes of deformation. In fig. 1.4 we can see different ways which a flat patch of membrane can be deformed. Translations and rotations correspond to global changes in the coordinate system, which do not contribute to the energy. While it definitely costs energy to shear a membrane, the flow of the lipids dissipates this energy. Stretching[2] of a membrane is an elastic deformation, that is hard to excite, and the stress sufficient to stretch a membrane can also rapture it [15, 16]. Bending is the only elastic mode of the membrane deformation, that changes the energy stored in the surface shape. Thus, we are looking for the local energy density, expressed in terms of the local membrane curvature. We can then integrate this energy over the whole surface to get the total bending energy. The local curvature is characterized by a two-dimensional tensor $C_{\alpha\beta}$ called the curvature matrix or the second fundamental form. Since the local energy density must be a scalar, independent of the coordinate transformations, it has to be expressed as a combination of elements of the curvature matrix and their covariant derivatives. An infinite number of such invariant combinations can be constructed, but since the elements of the curvature tensor have a dimension 1/length, these invariants, can be ordered by their dimension. Up to the order 1/length^2, we only have two invariants: the local mean curvature H and the gaussian curvature C_G [17]. Both invariants can be constructed from the eigenvalues of the curvature matrix $c_1 = 1/R_1$ and $c_2 = 1/R_2$, which themselves give the values for principal curvatures (see fig. 1.5 b)), representing the largest and the smallest curvature at any given point of the membrane. The eigenvectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ meanwhile point in the direction tangential to the paths of highest and

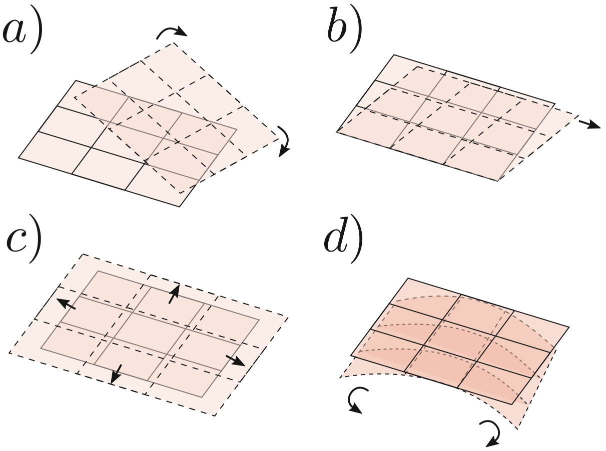

Figure 1.4: Different types of membrane transformation. a) Translation and rotation, b) shear, c) stretching, d) bending.

[2] Here I mean an actual increase of membrane area per lipid molecules when I say stretching. There is another effect of projected area stretching that enters versions of Canham-Helfrich energy and is also referred to as stretching.

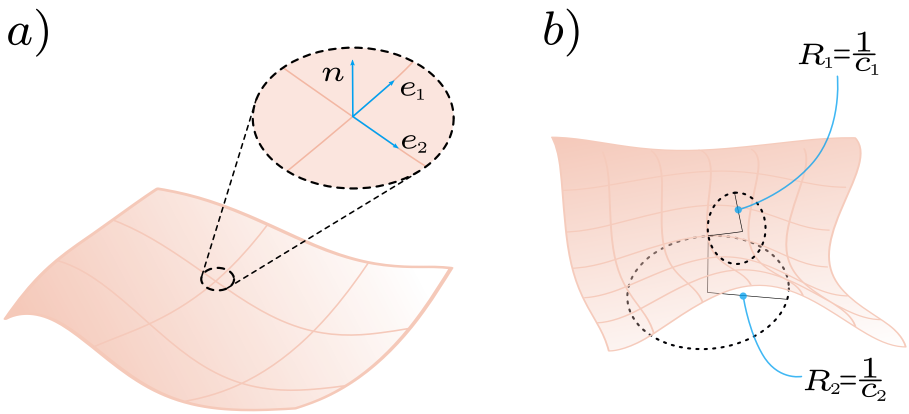

Figure 1.5: Parametrization of a 2D manifold. a) A curved manifold with a local normal vector $\boldsymbol{n}$ and tangential vectors $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$. Thee tangential vectors are chosen such that they point in the direction of the principal curvatures. b) Principal curvatures at a point of a manifold are the largest and the smallest curvatures at that point.

lowest curvature at any given point (see fig. 1.5 a)).

$$e(x_1, x_2) = \frac{\kappa}{2}(c_1 + c_2 - H_0)^2 + \bar{\kappa} c_1 c_2, \tag{1.1}$$

where κ, $\bar{\kappa}$ and H_0 are parameters of the membrane, called bending modulus, saddle splay modulus and spontaneous curvature respectively. The integral of this energy density over the surface leads to the total bending energy

$$E_\text{b} = \int \mathrm{d}A \left[\frac{\kappa}{2}(2H - H_0)^2 + \bar{\kappa} C_G\right], \tag{1.2}$$

which is referred to as the Canham-Helfrich energy. We used common notation to denote the local mean curvature

$$H = \frac{1}{2}(c_1 + c_2), \tag{1.3}$$

and Gaussian curvature

$$C_G = c_1 c_2. \tag{1.4}$$

The area element of the curved surface is given by

$$\mathrm{d}A = \sqrt{g}\mathrm{d}x_1 \mathrm{d}x_2 \tag{1.5}$$

with $g = \det(\boldsymbol{g})$ denoting the determinant of the metric tensor $\boldsymbol{g}$ which is defined as

$$g_{ij} = \boldsymbol{e}_i \cdot \boldsymbol{e}_j. \tag{1.6}$$

Inspecting eq. (1.2), we can interpret the physical meanings of the constants. The bending modulus κ represents the energy cost for the membrane to have a local total curvature $2H$ deviate from the preferred spontaneous curvature H_0. A large bending rigidity means that the membranes are stiffer. The usual values of κ for lipid membranes at biologically relevant temperatures are at the order of 10 $k_{\mathrm{B}}T$. Spontaneous curvature H_0 is usually zero for membranes with symmetric bilayers [16]. Membranes that have asymmetry in the lipid content between their bilayers, or are covered by proteins on one side, will bend spontaneously to accommodate this asymmetry and are modeled with a nonzero spontaneous curvature. Lastly, we have the saddle-splay modulus $\bar{\kappa}$, which is very hard to measure experimentally but is usually estimated at $\bar{\kappa} = -\kappa$. Since the sign of the gaussian curvature can only be negative if different principal curvatures have different signs, i.e. if the membrane has a saddle point, higher magnitudes of saddle-splay modulus indicate a higher cost of creating saddle shapes. However, the Gauss-Bonnet theorem guarantees that the surface integral of the Gaussian curvature is constant, unless the boundary or the topology of the membrane surface changes. This means that for a great number of the model systems, the last term in eq. (1.2) can be ignored, since it only contributes a constant to the total energy.

The bending energy alone rarely describes the membrane shape in equilibrium. Usually one also has two additional constraints that need to be considered, the area and the volume conservation. These constraints are incorporated via Lagrange multipliers into the energy expression, which leads to the total surface energy

$$E_{\mathrm{surf}} = \int \mathrm{d}A \left[\frac{\kappa}{2}(2H - H_0)^2 + \bar{\kappa} C_G\right] + \sigma \int \mathrm{d}A + \Delta P \int \mathrm{d}V, \tag{1.7}$$

where σ and ΔP are Lagrange multipliers that fix the total area of the membrane and the total volume enclosed by it. Typically, σ is interpreted as the tension of the membrane and ΔP is interpreted as the Laplace pressure, i.e., the pressure difference between the outside and the inside of the membrane $\Delta P = P_{\mathrm{out}} - P_{\mathrm{in}}$. The constraint of the volume conservation is not always used since the experimental justification for it does not always exist. It is only necessary, when considering closed membrane vesicles at time scales below the timescale of the membrane permeability. However, the constraint of area conservation is almost always enforced even in cases when the area is not necessarily constant, but the membrane still can be considered to have a constant tension. This is because σ can have several different meanings and its correct interpretation can change from system to system. This is the most confounding issue when working with eq. (1.7), a recent theoretical discussion of different interpretations of membrane tension and their relations with each other can be found in the reference [18] and a review of experimental research related to calculating membrane tension in different regimes can be found in the reference [16, Sec.4]. For now let us discuss few analytic ways that one would use eq. (1.7) to obtain membrane shapes in equilibrium.

Before we obtain membrane shapes we need to obtain shape equations from eq. (1.7). The most general way to do this is to assume that some explicit function $\boldsymbol{r}(x_1, x_2)$ describes the equilibrium shape of the membrane and then perform a variation around that shape

$$\boldsymbol{r}'(x_1, x_2) = \boldsymbol{r}(x_1, x_2) + h(x_1, x_2)\hat{\boldsymbol{n}}(x_1, x_2), \tag{1.8}$$

where $\boldsymbol{r}'(x_1, x_2)$ is the new perturbed shape, $h(x_1, x_2)$ is a small height ($|h| \ll \|r\|$) away from the equilibrium shape and $\hat{\boldsymbol{n}}(x_1, x_2)$ is the normalized normal vector. If we express every quantity in eq. (1.7) in terms of the $\boldsymbol{r}(x_1, x_2)$ and then perform the variation from eq. (1.8), we will arrive at the general shape equation

$$\Delta P - 2\sigma H + \kappa\left(2H + H_0\right)\left(2H^2 - 2C_G - H_0 H\right) + 2\kappa\nabla^2 H = 0, \tag{1.9}$$

where ∇^2 is the Laplace operator in curvilinear coordinates

$$\nabla^2 = \frac{1}{g}\partial_{x^i}(g^{ij}\sqrt{g}\partial_{x^j}). \tag{1.10}$$

The general shape equation shown in eq. (1.9) was first derived by Ou-Yang and Helfrich [19] and later an alternative method was introduced by Guven [20]. Given this explicit function $\boldsymbol{r}(x_1, x_2)$, that provides a map from its local coordinates x_1 and x_2 to the 3-dimensional embedding space, we can calculate the total curvature, area and enclosed volume of the manifold and obtain the total energy. A concrete example of such an embedding for a sphere with a fixed radius R is the explicit parametrization of $\boldsymbol{r}$ can be given in the spherical coordinates as $\boldsymbol{r}(\phi,\theta) = R\boldsymbol{r}_{S(1)}$.
Where $\boldsymbol{r}_{S(1)} = (\cos(\phi)\sin(\theta), \sin(\phi)\sin(\theta), \cos(\theta))$ is the parametrization of a unit sphere. If such an explicit parametrization of the initial geometry is provided, the variation becomes significantly simpler than in the general case. Mathematically this is done by first explicitly parametrizing the manifold, then obtaining a function that describes its shape and then performing functional variation to take the functional derivative of the energy. In the case of our sphere example, a surface close to it could be represented as $\boldsymbol{r}(\phi,\theta) = \boldsymbol{r}_{S(1)}(R + h(\phi,\theta))$. This approach is called Monge gauge, it allows us to use known geometries and express shapes as deviations from them. It has to be noted however, that a height function restricts us to shapes that have no overhangs with respect to the reference frame. This is however a reasonable restriction for a lot of relevant cases, and this rarely becomes an issue. Another problem is that the Monge gauge does not fully get us to the finish line. We have just replaced one unknown function $\boldsymbol{r}$ by another unknown function h. To make progress from here onward, one has to resort to a series expansion in the height field. For our sphere example this would mean we have to assume that $\epsilon = |h|/R \ll 1$ and expand in powers of ϵ. This expansion is usually truncated at the second order in h and its derivatives. Varying that energy expression then leads to a linear shape equation. A concrete example of such a derivation, from a general energy expression to a concrete shape equation can be found in the supplement of chapter 2, in particular section S.5.1. The resulting shape equation is still a two-dimensional PDE and cannot always be solved. The necessity of a series expansion in small deformations, limits the utility of the linearized Monge gauge approach, which is most useful for a linear stability analysis of a known minimal shape [21–24]
An alternative way to make progress with eq. (1.7) analytically, is a mean field approach. In this case one assumes that the shape of interest is a stable minimum of the surface energy, like a spherical vesicle or a tethered membrane tube. The deformations that are considered are global, e.g., a spherical vesicle that could shrink or expand. This approach can be very useful if the system under consideration is reacting to global stresses, and we make use of it in chapter 3.

For the class of all shapes that have a rotational symmetry around some axis one can simplify the description of the membrane from a two-dimensional surface to a one-dimensional contour. This results into a change of shape equation from a PDE to a set of ODE's. Since these ODEs are 4-th order and highly nonlinear, their solution is analytically impossible apart from few trivial geometries, but they can be solved numerically, and we will discuss this later in section 1.4.1.

1.4. COMPUTATIONAL MEMBRANE PHYSICS

In this section I will give a short overview of numeric methods and simulation types widely used in the membrane physics, e.g. to predict vesicle shapes. To select the methods that I present here, I simply chose ones that I had familiarity with. Either having used them myself or having read about them in the literature, so while I hope that the list is extensive it is not exhaustive. In the following subsections I make a distinction between two types of computational approaches. Numeric methods typically solve differential equations that describe a physical system, with the use of computers, because their solutions are not obtainable by hand. Simulations, use equations that describe individual agents of the simulation and interactions between these agents. Thus, the simulation performs a coarse graining and brings the local agent level interactions to the global level. This simulation of emergent behavior is what, for the purposes of this introduction, distinguishes simulations from numerics. In the latter case the coarse graining is performed in theory and the resulting equation is numerically solved. This means that simulations make less assumptions about the system and thus are often able to model a wider variety of systems, while numeric solutions are usually significantly faster.

1.4.1. NUMERIC METHODS

NUMERIC INTEGRATION OF AXIS SYMMETRIC SHAPE EQUATIONS

Exploiting rotational axis symmetry is a powerful tool for calculating vesicle shapes, since it effectively reduces the spatial dimension of the problem. This transforms the shape equation from a nonlinear PDE to a nonlinear ODE and thus significantly simplifies the computational complexity. In fact the original 1970 paper by Canham [13] that introduced bending energy of the membrane as the determining factor of its shape, exploited the rotational symmetry of red blood cells (RBC) to explain their biconcavity. Although Canham did not actually solve a shape equation but used a modified Cassini Oval to approximate the shape of the RBC and then integrated the square of the curvature of the oval to calculate its bending energy, the use of rotational symmetry was crutial for his approach to work. The 1973 paper by Helfrich [14] devotes a whole section to problems with rotational symmetry and derives shape equations for specialized shapes. The first catalogue of axis symmetric vesicle shapes was presented by Dueling and Helfrich already in 1976 [25]. In 1991 Seifert et al. [26] used axis-symmetric shape equations to obtain vesicle shape, phase diagrams, considering a large variety of membrane parameters.

To obtain an axis symmetric shape equation, we need to describe the shape of the contour, rotation of which around the symmetry axis will create the surface of the membrane. The contour of the revolution is usually parametrized by its distance $r(s)$ from the axis of revolution, its projection $z(s)$ on the axis of revolution and the slope $\psi(s)$ of the tangent vector at any given position s along the contour (see fig. 1.6). Not all three parameters are independent however and using this method we get a set of three dependent ordinary differential equations(3rd order in ψ which translates to a 4th order equation in r)

$$\begin{aligned}
\dddot{\psi} = &-\frac{2\cos(\psi)}{r}\ddot{\psi} - \frac{1}{2}\dot{\psi}^2 - \frac{3\sin(\psi)}{2r}\dot{\psi}^2 + \frac{3\cos^2(\psi)-1}{2r^2}\dot{\psi} \\
&+ \frac{\sigma}{\kappa}\dot{\psi} - \frac{\cos^2(\psi)+1}{2r^3}\sin(\psi) + \frac{\sigma}{\kappa}\frac{\sin(\psi)}{r} + \frac{\Delta P}{\kappa}, \qquad (1.11) \\
\dot{r} = &\cos(\psi), \qquad (1.12) \\
\dot{z} = &-\sin(\psi), \qquad (1.13)
\end{aligned}$$

where the dot stands for a derivative with respect to the arc length s i.e., $\dot{f} = \frac{\mathrm{d}f}{\mathrm{d}s}$ (for a derivation see ref [27]). Many software packages exist that can perform numerical integration of one dimensional equations, so this is usually technically not very challenging. However, issues with numerical stability often arise, due to the non-linearity of the equations, especially when complex boundary conditions are involved. But this is a tried and tested method that usually reliably delivers results for an axis symmetric systems. When such a symmetry is however not present, one needs to go back to numerically solving PDEs.

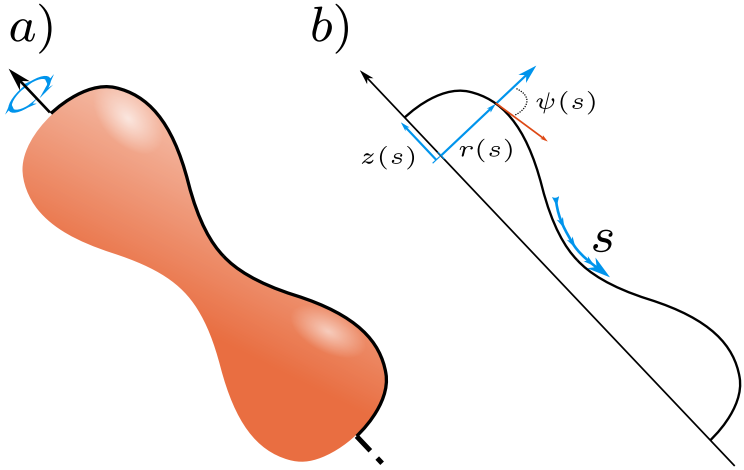

Figure 1.6: A schematic representation of an axis symmetric shape. a) Axis symmetric shape that can be obtained through the revolution of the contour (black solid line) around the exes of symmetry. b) Parametrization of the contour by the contour length s. $r(s)$ is the distance from the axes of symmetry to the contour at the contour point s and $z(s)$ is the projection of the contour along the z-axis. The field $\psi(s)$ that parametrizes the contour shape is given by the angle between $r(s)$ and the tangent of the curve at s.

Finite difference method

General PDEs describing membrane shapes are two-dimensional equations, one needs to discretize two-dimensional surfaces, which can be done in different ways and the choice of this discretization influences the types of methods that we can use. Conceptually the simplest numeric method to solve two-dimensional shape equations is the finite difference method. This idea is as old as calculus itself and its development is attributed to Brook Taylor's 1715 work *Methodus incrementorum directa et inversa* [28]. As the name implies, the core idea of the method is to use the differences between function values, to approximate the derivatives of the function [29]. To this end we need to discretize the space into a lattice and define any function, that we want to examine, on that lattice. In 1d a function $f(x)$ then transforms into a vector of values $f(x_i)$ (see fig. 1.7 a)). Where x_i is the i-th lattice point of the discretized x-axis, with an equal spacing Δx on the lattice. Our goal to express any n-th order derivative $f^{(n)}$ as a difference of function values on

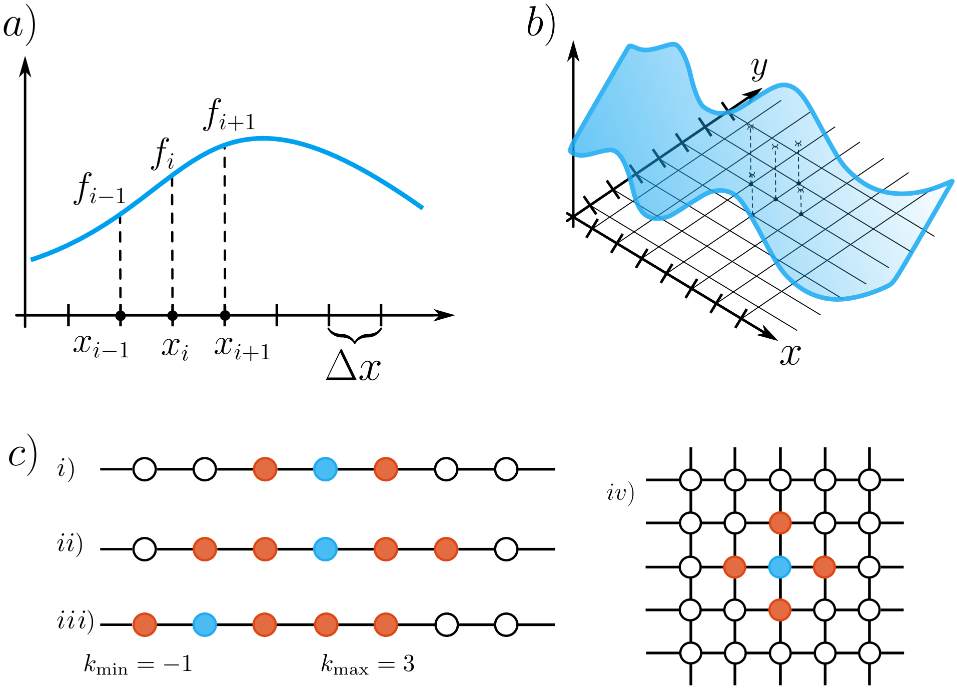

Figure 1.7: Finite difference methods. a) Discretization of a one dimensional function ($f(x) \to f_i$) on a evenly spaced lattice x_i with lattice spacing Δx. b) A two dimensional function discretized on a square lattice. c) A stencil can vary in size and be symmetric or asymmetric. Larger stencils increase the accuracy of the discretized differential operators but using them requires more computation. Asymmetric stencils are used near boundaries if there are not the same number of points available on both sides of the point of differentiation; or in the case of initial value problems where function values are not available on one side of the point of differentiation.

the lattice can be formalized as

$$f^{(n)}(x_i) \approx \sum_{k=k_{\text{min}}}^{k_{\text{max}}} C_k f(x_i + k\Delta x) = \sum_{k=k_{\text{min}}}^{k_{\text{max}}} C_k f_{i+k}, \tag{1.14}$$

where $k_{\text{min}} \leq 0$ and $k_{\text{max}} \geq 0$ set the number of the lattice points around point i that will be used in the calculation. The collection of these points is called the stencil and the stencil size is then $S = k_{\text{max}} - k_{\text{min}} + 1$ (see fig. 1.7 c)). Comparing eq. (1.14) to the Taylor expansion of the function

$$f(x_i + k\Delta x) = \sum_{n=0}^{\infty} \frac{k^n \Delta^n}{n!} f^{(n)}(x_i), \tag{1.15}$$

we get a system of S equations to determine the weights C_k

$$\frac{\Delta x^m}{m!} \sum_{k=k_{\min}}^{k_{\max}} k^m C_k = \delta_{mn}, \tag{1.16}$$

where δ_{mn} denotes the Kronecker delta and $0 \le m \le S$. The accuracy of the approximation in eq. (1.14) depends on the size of the stencil and goes as $\mathcal{O}\left(\Delta x^S\right)$ [30]. The finite difference method discussed here only works on a square lattice, which is a severe limitation. This makes the method for example unusable for solving differential equations in spherical geometry. This limitation can be overcome by extending the method. For irregularly spaced lattice points one can use the divided differences [29, §9] and even non-Cartesian lattices can be accommodated [31, Sec. 7.9], however these extensions overly complicate the finite difference method, which really shines in simple geometries. Usually when we want to solve differential equations in more complex geometries or incorporate boundary conditions that do not work on Cartesian grids, it is far smarter to avoid the finite difference method altogether and use the finite element method (FEM), which is an umbrella term for a variety of methods that are well suited for solving PDEs in complex geometries and with complex boundary conditions [31, Sec. 7.12]. In contrast to the FDMs I have not made use of this method myself, thus I am not able to elaborate on them in detail[3]. However, a discussion of numerical methods would not be complete without mentioning FEMs, since they seem to be more popular than the finite difference methods, in the literature (Especially the thin shell FEM family [33]). And seem to be quite capable to describe membrane reshaping in complex geometries [34, 35], as well as being a necessary step in the phase field methods (discussed later in section 1.4.2).

1.4.2. SIMULATION METHODS

Biological membranes exhibit a large variety of complex behaviour. Depending on the biological agents that the membranes are interacting with the reshaping of the membrane can happen through many pathways and the span several orders of magnitude [36]. Accordingly, there is a large variety of models to choose from, that capture membrane behaviour at different length scales. A somewhat old but excellent review of different simulation methods is provided by ref. [37]. And a newer review with a narrower focus on membrane protein interactions is provided by ref. [38]. Starting from the smallest length scale to the largest, we have all atom membrane simulations [39, 40], coarse grained lipid simulations [41–43], self assembled [44, 45] and triangulated membrane [46] simulations and finally phase field methods[4] [47, 48] and cellular Potts models [49].

All atom simulations are very close to actual experiments and use first principle interactions between molecules, however this is an expensive approach and only small

[3] For general information on FEMs the reference [31] and a more recent textbook [32] can be recommended.

[4] Phase field methods are actually a numeric method and not a simulation according to my earlier classification, but I included them here since they better fit in the context of discussing different length scales.

length scale		models
< 1nm	atomic scale	all atom simulation, coarse grained molecular models
$\sim 1 - 100$nm	membrane thickness, small vesicles	coarse grained lipid models
$\gg 1$nm	membrane compartment, organelles to cells, large to giant vesicles	self assembled membranes, triangulated membranes
$\gg 1\mu$m	several membrane compartments; multicellular and tissue scale	cellular potts model, phase field methods

Table 1.1: Overview of computational membrane models organized by length scale.

patches of the membrane over a small duration of time can be analyzed. Coarse grained lipid simulations allow to investigate the behaviour of larger membrane patches while still retaining individual lipids as elements in the model. These simulations differ in the level of coarse graining and explicit [50, 51] or implicit [43] inclusion of the solvent. Implicit solvent simulations with coarse grained lipids make assumptions about the solvent-lipid interactions and have a simplified model of how individual lipids interact with each other. This coarse graining of interactions looses information on the sub-molecular level but makes it possible to look at larger membrane patches or even at small vesicles [52, 53]. Coarse grained lipid models are a valuable asset if we need to perform simulations of small vesicles and need to take the leaflets of the bilayer individually into account, like when analyzing the effects of lipid flip-flop on vesicle equilibrium shapes [54]. If truly large scale simulations are required then we have to give up the lipid level resolution and coarse grain the membrane further. Meshless coarse grained and triangulated membrane simulations have been used effectively to describe large scale membrane reshaping of large vesicles[5]and membrane patches [18, 55–57], as well as to describe pattern formation of membrane deforming objects, like curvature inducing proteins [57, 58] and colloids [59–62]. The membrane particles in these simulations usually represent a small patch of membrane instead of individual lipids, thus the cost of doing simulations on the large scale is the lost ability to track membrane layers and individual lipids separately. Finally, rounding off this list are the cellular Potts model and the phase field methods, that can be used to model single cell migration or interactions between cells and tissue level dynamics [63–67]. As we see each level of description has its limitations and strengths. With current level of technology it is impossible to perform an all-atom or even coarse grained lipid simulation on the scale of a cell, however multiscale simulations as a goal are actively being pursued [68–70].

In the following section I will go into more details of one particular method, the dynamically triangulated membrane monte carlo method DTMMCs. As already discussed, triangulated methods are a coarse grained descriptions of the membrane and allow the modeling of global reshaping of cell sized membrane compartments. This was the pri-

[5] Vesicles with radius larger than a μm.

mary reason for choosing this method for simulating membranes during my PhD.

DYNAMICALLY TRIANGULATED MEMBRANE MONTE CARLO (DTMMC) SIMULATIONS

In dynamically triangulated membrane methods, the modeled surface is approximated by a triangulation and then the elastic energy of this triangulated surface is calculated. This is accomplished by keeping track of the areas, volumes and normal vectors associated to each node of the triangulation as depicted in fig. 1.9. The expressions of the local curvature volume and area are presented in section 2.2 of chapter 4. Once we have a triangulated surface we can update the coordinates of the triangulation nodes randomly (see fig. 1.8 a)) and keep track of the overall elastic energy. To make the triangulated surface fluid in its tangential direction, we have to allow for free movement of the nodes of the triangulation, and allow the nodes to exchange edges with each other (see fig. 1.8 b)). This is called bond flipping and the reason the method is called dynamic (because the triangulation is changing during the simulation). However, these bond flips needs to be done carefully such that they dont accidentally puncture the surface, this makes the bond flips computationally expensive compared to node moves. In each update step we can either move a bead or flip a bond, calculate the energy difference $E_{old} - E_{new}$ and always accept the new state if the energy has been lowered, i.e. if $E_{old} > E_{new}$. If the energy has been increased we only accept the move according to the probability $\exp\left(\frac{E_{old}-E_{new}}{k_B T}\right)$. This is the Metropolis algorithm [71] and using this update rule we can reach a shape that is fluctuating around a minimum energy configuration. To obtain the proper minimum energy shape the temperature can be lowered to approach a fluctuation free zero temperature shape that is the minimum of the free energy. A caveat here is that the minimum one finds may not be a global minimum, this means that the minimization procedure is usually more complex than just equilibrate the energy and then lower the temperature and usually involves some kind of temperature cycling to explore the energy landscape. This problem however exist for all Monte Carlo simulations and a large body of literature exists, that deals with strategies to overcome the problems with complicated free energy landscapes[72, 73]. We discuss a specific example of such a procedure to obtain the equilibrium shapes in section 3 of chapter 4. Even with this caveat DTMMCs can be very powerful due to their versatility and speed. An implementation that is economical

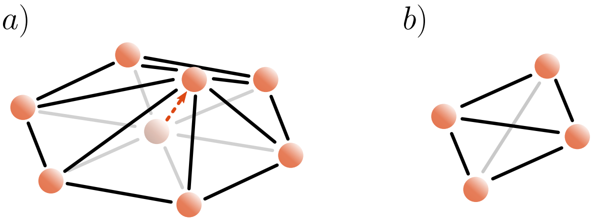

Figure 1.8: Visual representation of a dynamic triangulation. Grayed out beads and edges represent the old state. a) Visualization of a membrane bead move. Each bead can be moved individually and upon the change of the bead position the edges and triangles connected to the bead also change, which translates into the local geometry change and leads to the change of surface energy. b) Bond flipping. In a dimond configuration like depicted here a bond can flip between two pairs of the beads. This flip leads to the destruction of two old triangles and the formation of two new ones, which also leads to changes in local geometry and surface energy.

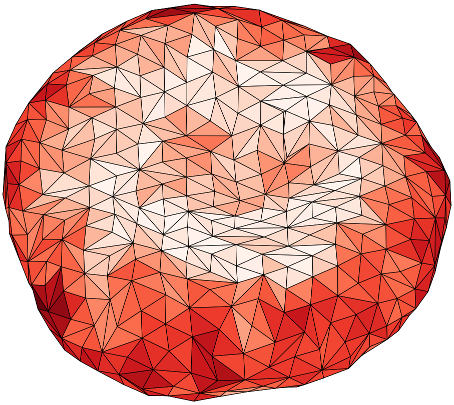

Figure 1.9: Example of a triangulated membrane. Color indicated local mean curvature. Regions of zero mean curvature are white and the intensity of the red color increases with curvature.

with calculation of local quantities can perform tens of millions of moves per second (see flippy chapter), which allows us to investigate large scale membrane reshaping phenomenon effectively.

BIBLIOGRAPHY

[1] E. Godino, J. N. López, D. Foschepoth, C. Cleij, A. Doerr, C. F. Castellà, and C. Danelon, "De novo synthesized min proteins drive oscillatory liposome deformation and regulate FtsA-FtsZ cytoskeletal patterns", Nature Communications **10**, 4969 (2019).

[2] D. A. Ramirez-Diaz, A. Merino-Salomón, F. Meyer, M. Heymann, G. Rivas, M. Bramkamp, and P. Schwille, "Ftsz induces membrane deformations via torsional stress upon gtp hydrolysis", Nature communications **12** (2021).

[3] T. Litschel, B. Ramm, R. Maas, M. Heymann, and P. Schwille, "Beating vesicles: encapsulated protein oscillations cause dynamic membrane deformations", Angewandte Chemie International Edition **57**, 16286 (2018).

[4] S. Deshpande, W. K. Spoelstra, M. Van Doorn, J. Kerssemakers, and C. Dekker, "Mechanical division of cell-sized liposomes", ACS nano **12**, 2560 (2018).

[5] J. Steinkühler, R. L. Knorr, Z. Zhao, T. Bhatia, S. M. Bartelt, S. Wegner, R. Dimova, and R. Lipowsky, "Controlled division of cell-sized vesicles by low densities of membrane-bound proteins", Nature communications **11** (2020).

[6] P. Schwille, "Division in synthetic cells", Emerging Topics in Life Sciences **3**, 551 (2019).

[7] H. Qiu and M. Caffrey, "The phase diagram of the monoolein/water system: metastability and equilibrium aspects", Biomaterials **21**, 223 (2000).

[8] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, "Molecular biology of the cell", 5th ed.(Garland science, Taylor & Francis Group) (2008).

[9] D. Casares, P. V. Escribá, and C. A. Rosselló, "Membrane lipid composition: effect on membrane and organelle structure, function and compartmentalization and therapeutic avenues", International journal of molecular sciences **20**, 2167 (2019).

[10] S. J. Singer and G. L. Nicolson, "The fluid mosaic model of the structure of cell membranes: cell membranes are viewed as two-dimensional solutions of oriented globular proteins and lipids.", Science **175**, 720 (1972).

[11] Wikimedia Commons, *File:Animal Cell Unannotated.svg — Wikimedia Commons, the free media repository,* in public domain., 2021.

[12] F. Frey and T. Idema, "More than just a barrier: using physical models to couple membrane shape to cell function", Soft Matter **17**, 3533 (2021).

[13] P. B. Canham, "The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell", Journal of theoretical biology **26**, 61 (1970).

[14] W. Helfrich, "Elastic properties of lipid bilayers: theory and possible experiments", Zeitschrift für Naturforschung C **28**, 693 (1973).

[15] K. Olbrich, W. Rawicz, D. Needham, and E. Evans, "Water permeability and mechanical strength of polyunsaturated lipid bilayers", Biophysical journal **79**, 321 (2000).

[16] R. Phillips, "Membranes by the numbers", in *Physics of biological membranes*, edited by P. Bassereau and P. Sens (Springer International Publishing, Cham, 2018), pp. 73–105.

[17] R. Capovilla, J. Guven, and J. Santiago, "Deformations of the geometry of lipid vesicles", Journal of Physics A: Mathematical and General **36**, 6281 (2003).

[18] G. Gueguen, N. Destainville, and M. Manghi, "Fluctuation tension and shape transition of vesicles: renormalisation calculations and monte carlo simulations", Soft Matter **13**, 6100 (2017).

[19] O.-Y. Zhong-Can and W. Helfrich, "Bending energy of vesicle membranes: general expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders", Physical Review A **39**, 5280 (1989).

[20] J. Guven, "Membrane geometry with auxiliary variables and quadratic constraints", Journal of Physics A: Mathematical and General **37**, L313 (2004).

[21] S. Leibler, "Curvature instability in membranes", Journal de Physique **47**, 507 (1986).

[22] J.-B. Fournier and P. Galatola, "Critical fluctuations of tense fluid membrane tubules", Physical review letters **98**, 018103 (2007).

[23] S. Monnier, S. B. Rochal, A. Parmeggiani, and V. L. Lorman, "Long-range protein coupling mediated by critical low-energy modes of tubular lipid membranes", Physical review letters **105**, 028102 (2010).

[24] J. Agudo-Canalejo and R. Golestanian, "Pattern formation by curvature-inducing proteins on spherical membranes", New Journal of Physics **19**, 125013 (2017).

[25] H. Deuling and W. Helfrich, "The curvature elasticity of fluid membranes: a catalogue of vesicle shapes", Journal de Physique **37**, 1335 (1976).

[26] U. Seifert, K. Berndl, and R. Lipowsky, "Shape transformations of vesicles: phase diagram for spontaneous-curvature and bilayer-coupling models", Physical review A **44**, 1182 (1991).

[27] F. Jülicher and U. Seifert, "Shape equations for axisymmetric vesicles: a clarification", Physical Review E **49**, 4728 (1994).

[28] L. Feigenbaum, "Brook taylor and the method of increments", Archive for history of exact sciences **34**, 1 (1985).

[29] C. Jordan and K. Jordán, *Calculus of finite differences*, Vol. 33 (American Mathematical Soc., 1965).

[30] B. Fornberg, "Generation of finite difference formulas on arbitrarily spaced grids", Mathematics of computation **51**, 699 (1988).

[31] C. Gerald and P. Wheatley, “Applied numerical analysis”, 4th ed. Addison-Wesley Publ. Co., Boston, MA. (1989).

[32] J. van Kan, A. Segal, and F. J. Vermolen, *Numerical methods in scientific computing* (VSSD, 2005).

[33] F. Feng and W. S. Klug, “Finite element modeling of lipid bilayer membranes”, Journal of Computational Physics **220**, 394 (2006).

[34] L. Ma and W. S. Klug, “Viscous regularization and r-adaptive remeshing for finite element analysis of lipid membrane mechanics”, Journal of Computational Physics **227**, 5816 (2008).

[35] P. W. Miller, N. Stoop, and J. Dunkel, “Geometry of wave propagation on active deformable surfaces”, Physical Review Letters **120**, 268001 (2018).

[36] P. Bassereau, R. Jin, T. Baumgart, M. Deserno, R. Dimova, V. A. Frolov, P. V. Bashkirov, H. Grubmüller, R. Jahn, H. J. Risselada, et al., “The 2018 biomembrane curvature and remodeling roadmap”, Journal of physics D: Applied physics **51**, 343001 (2018).

[37] H. Noguchi, “Membrane simulation models from nanometer to micrometer scale”, Journal of the Physical Society of Japan **78**, 041007 (2009).

[38] N. Ramakrishnan, R. P. Bradley, R. W. Tourdot, and R. Radhakrishnan, “Biophysics of membrane curvature remodeling at molecular and mesoscopic lengthscales”, Journal of Physics: Condensed Matter **30**, 273001 (2018).

[39] D. J. Tobias, K. Tu, and M. L. Klein, “Atomic-scale molecular dynamics simulations of lipid membranes”, Current opinion in colloid & interface science **2**, 15 (1997).

[40] A. P. Lyubartsev and A. L. Rabinovich, “Recent development in computer simulations of lipid bilayers”, Soft Matter **7**, 25 (2011).

[41] W. Shinoda, R. DeVane, and M. L. Klein, “Coarse-grained molecular modeling of non-ionic surfactant self-assembly”, Soft Matter **4**, 2454 (2008).

[42] 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”, The journal of physical chemistry B **111**, 7812 (2007).

[43] I. R. Cooke, K. Kremer, and M. Deserno, “Tunable generic model for fluid bilayer membranes”, Physical Review E **72**, 011506 (2005).

[44] J.-M. Drouffe, A. Maggs, and S. Leibler, “Computer simulations of self-assembled membranes”, Science **254**, 1353 (1991).

[45] H. Yuan, C. Huang, J. Li, G. Lykotrafitis, and S. Zhang, “One-particle-thick, solvent-free, coarse-grained model for biological and biomimetic fluid membranes”, Physical review E **82**, 011905 (2010).

[46] G. Gompper and D. M. Kroll, “Network models of fluid, hexatic and polymerized membranes”, Journal of Physics: Condensed Matter **9**, 8795 (1997).

[47] Q. Du, C. Liu, and X. Wang, “Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions”, Journal of computational physics **212**, 757 (2006).

[48] J. S. Lowengrub, A. Rätz, and A. Voigt, "Phase-field modeling of the dynamics of multicomponent vesicles: spinodal decomposition, coarsening, budding, and fission", Physical Review E **79**, 031926 (2009).

[49] F. Graner and J. A. Glazier, "Simulation of biological cell sorting using a two-dimensional extended Potts model", Physical review letters **69**, 2013 (1992).

[50] R. Goetz and R. Lipowsky, "Computer simulations of bilayer membranes: self-assembly and interfacial tension", The Journal of chemical physics **108**, 7397 (1998).

[51] M. Laradji and P. S. Kumar, "Dynamics of domain growth in self-assembled fluid vesicles", Physical review letters **93**, 198105 (2004).

[52] T. Chen, Y. Zhang, X. Li, C. Li, T. Lu, S. Xiao, and H. Liang, "Curvature-mediated pair interactions of soft nanoparticles adhered to a cell membrane", Journal of Chemical Theory and Computation **17**, 7850 (2021).

[53] N. Urakami, T. Jimbo, Y. Sakuma, and M. Imai, "Molecular mechanism of vesicle division induced by coupling between lipid geometry and membrane curvatures", Soft Matter **14**, 3018 (2018).

[54] N. Urakami, Y. Sakuma, T. Chiba, and M. Imai, "Vesicle deformation and division induced by flip-flops of lipid molecules", Soft Matter **17**, 8434 (2021).

[55] H. Noguchi and G. Gompper, "Fluid vesicles with viscous membranes in shear flow", Physical review letters **93**, 258102 (2004).

[56] H. Noguchi and G. Gompper, "Dynamics of vesicle self-assembly and dissolution", The Journal of chemical physics **125**, 164908 (2006).

[57] L. Harker-Kirschneck, A. E. Hafner, T. Yao, C. Vanhille-Campos, X. Jiang, A. Pulschen, F. Hurtig, D. Hryniuk, S. Culley, R. Henriques, et al., "Physical mechanisms of escrt-iii–driven cell division", Proceedings of the National Academy of Sciences **119** (2022).

[58] S. C. J. Helle, Q. Feng, M. J. Aebersold, L. Hirt, R. R. Grüter, A. Vahid, A. Sirianni, S. Mostowy, J. G. Snedeker, A. Šarić, et al., "Mechanical force induces mitochondrial fission", Elife **6**, e30292 (2017).

[59] A. Šarić and A. Cacciuto, "Mechanism of membrane tube formation induced by adhesive nanocomponents", Physical review letters **109**, 188101 (2012).

[60] A. H. Bahrami, R. Lipowsky, and T. R. Weikl, "Tubulation and aggregation of spherical nanoparticles adsorbed on vesicles", Physical review letters **109**, 188102 (2012).

[61] A. Šarić and A. Cacciuto, "Fluid membranes can drive linear aggregation of adsorbed spherical nanoparticles", Physical Review Letters **108**, 118101 (2012).

[62] C. Van Der Wel, A. Vahid, A. Šarić, T. Idema, D. Heinrich, and D. J. Kraft, "Lipid membrane-mediated attraction between curvature inducing objects", Scientific reports **6** (2016).

[63] D. Shao, W.-J. Rappel, and H. Levine, "Computational model for cell morphodynamics", Physical review letters **105**, 108104 (2010).

[64] J. Löber, F. Ziebert, and I. S. Aranson, "Collisions of deformable cells lead to collective migration", Scientific reports **5** (2015).

[65] A. Goychuk, D. B. Brückner, A. W. Holle, J. P. Spatz, C. P. Broedersz, and E. Frey, "Morphology and motility of cells on soft substrates", arXiv preprint arXiv:1808.00314 (2018).

[66] L. A. Hoffmann, L. N. Carenza, J. Eckert, and L. Giomi, "Defect-mediated morphogenesis", arXiv preprint arXiv:2105.15200 (2021).

[67] F. Thüroff, A. Goychuk, M. Reiter, and E. Frey, "Bridging the gap between single-cell migration and collective dynamics", Elife **8**, e46842 (2019).

[68] A. Davtyan, M. Simunovic, and G. A. Voth, "Multiscale simulations of protein-facilitated membrane remodeling", Journal of structural biology **196**, 57 (2016).

[69] W. Pezeshkian, M. König, S. J. Marrink, and J. H. Ipsen, "A multi-scale approach to membrane remodeling processes", Frontiers in Molecular Biosciences **6**, 59 (2019).

[70] W. Pezeshkian, M. König, T. A. Wassenaar, and S. J. Marrink, "Backmapping triangulated surfaces to coarse-grained membrane models", Nature communications **11** (2020).

[71] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines", The journal of chemical physics **21**, 1087 (1953).

[72] C. Tsallis and D. A. Stariolo, "Generalized simulated annealing", Physica A: Statistical Mechanics and its Applications **233**, 395 (1996).

[73] Z. Li and H. A. Scheraga, "Monte carlo-minimization approach to the multiple-minima problem in protein folding.", Proceedings of the National Academy of Sciences **84**, Publisher: Proceedings of the National Academy of Sciences, 6611 (1987).

2

Membrane-mediated interactions induce spontaneous filament bundling

The plasma membrane and cytoskeleton of living cells are closely coupled dynamical systems. Internal cytoskeletal elements such as actin filaments and microtubules continually exert forces on the membrane, resulting in the formation of membrane protrusions. In this paper we investigate the interplay between the shape of a cell distorted by pushing and pulling forces generated by microtubules and the resulting rearrangement of the microtubule network. From analytical calculations, we find that two microtubules that deform the vesicle can both attract or repel each other, depending on their angular separations and the direction of the imposed forces. We also show how the existence of attractive interactions between multiple microtubules can be deduced analytically, and further explore general interactions through Monte Carlo simulations. Our results suggest that the commonly reported parallel structures of microtubules in both biological and artificial systems can be a natural consequence of membrane mediated interactions.

This chapter is based on: A.Vahid, G.Dadunashvili, T.Idema, Membrane-mediated interactions induce spontaneous filament bundling (in preparation)

2.1. Introduction

Cells are enveloped by a plasma membrane which serves as a selective soft physical barrier and is home to many functional proteins. The stability and shape of cellular membranes are determined not only by inherent properties of the membrane, but also by interactions with the cell's cytoskeleton [1]. The highly dynamic cytoskeletal network is vital for numerous biological processes, including cell motility, cell migration, and cell signaling [2, 3]. A typical feature occurring in such processes is the formation of membrane protrusions. Protrusions commonly emerge in the form of microvilli, filopodia or lamellipodia [4, 5]. These leading-edge protrusions, the existence of which is vital for responding to external cues, can be driven, controlled and elongated by a complicated crosstalk between the membrane and underlying filaments.

The spatial arrangement of cytoskeletal filaments, force generation mechanisms, and cytoskeletal networks coupling to the shape of cells have been investigated extensively, both theoretically and experimentally [6–10]. For example, when growing encapsulated microtubules inside an artificial spherical membrane, it has been shown that the vesicle exhibits a diverse range of morphologies, from a simple elongated shape to dumbbell-like geometries [7]. The diversity in the shape of such vesicles results from both the elongation dynamics of the filaments inside them and the material properties of the membrane. Such spatial rearrangement of filaments stems from the conditions imposed on them from various elements, one of which is the cell shape.

In this paper, we investigate the interplay between the shape of vesicles that are deformed by internal force generating filaments like microtubules, and the rearrangement of those filaments. In a biological cell, microtubules undergo treadmilling and dynamic instabilities (catastrophes) which are controlled by associated proteins [11]. Only a few of the microtubules that grow inside a cell can reach the cell membrane [12]. The pushing and pulling forces generated by those few microtubules can be harnessed for creating protrusions of the membrane [13]. Membrane mediated interactions between microtubule-induced protrusions may influence the arrangement of other functional filaments in addition to microtubules themselves [14, 15]. Therefore, we study how the presence of a lipid bilayer membrane, which has both elastic and fluid properties, alters the interaction between microtubules. This interaction could both drive processes like the formation of filament bundles or inhibit microtubule aggregation.

We use a modified version of the theoretical framework that has been developed for investigating membrane mediated interactions between proteins embedded in or bounded to a fluid membrane [16, 17]. We first explain the model in detail. We then study the effects of all the possible elements on the interaction between microtubules. We reveal that the relative orientation of the deforming forces (generated by the microtubules) determines the nature of their interactions. In particular, we demonstrate that bundling is a stable effect observed for any number of interacting microtubules with forces of the same orientation. Our results thus elucidate the effective role of the membrane in determining the equilibrium arrangement of protrusions imposed by the cytoskeleton.

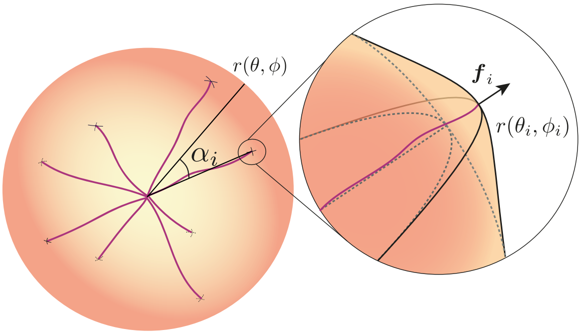

Figure 2.1: A schematic representation of a spherical membrane vesicle containing microtubules. **Inset**: local (small) deformation $R\,u(\theta_i,\phi_i)$, caused by a microtubule, which is exerting a force of magnitude f_i, along the radius of the vesicle.

2.2. MODEL

Our model is based on minimizing the energy of the system consisting of the membrane and force imposing microtubules, depicted in figure 2.1, to obtain the equilibrium shape. We use the Canham-Helfrich functional [18, 19]to characterize the energy, due to elastic deformations of a closed membrane vesicle.

$$E_{\text{el}} = \int \mathrm{d}S \left(2\kappa H^2 + \sigma\right) + \Delta P \int \mathrm{d}V, \tag{2.1}$$

with H, κ, σ and ΔP the mean curvature, bending modulus, surface tension and the pressure difference (between inside and outside of the vesicle), respectively. The first term in the elastic energy penalizes high curvature, while the second and third terms penalize the change in projected surface area and enclosed volume of the vesicle. In order to describe microtubules deforming the membrane, we impose deformations at n points $\mathbf{r}(\theta_i,\phi_i)$ as boundary conditions

$$E_{\text{PF}} = \int \mathrm{d}S\ (\mathbf{f}\cdot\boldsymbol{\Delta})R\,u(\theta,\phi), \tag{2.2}$$

where $\mathbf{f}\in\mathbb{R}^n$ is the vector of forces, exerted on the membrane by the microtubules. This vector contains the projections of the point forces along the radial direction of a spherical vesicle. We denote by $\boldsymbol{\Delta}$ the vector of Dirac-delta functions at the points (θ_i,ϕ_i),

$$\begin{aligned}\Delta_i &= \boldsymbol{\delta}(\hat{\mathbf{r}}(\theta_i,\phi_i)) \\ &= \frac{\sin(\theta)}{\sqrt{\det g}}\delta(\cos(\theta)-\cos(\theta_i))\delta(\phi-\phi_i) \\ &\overset{\text{def}}{=} \frac{\sin(\theta)}{\sqrt{\det g}}\bar{\Delta}_i.\end{aligned} \tag{2.3}$$

Here g is the metric tensor of the membrane, defined as $g_{\mu\nu} = \partial_\mu \boldsymbol{r} \cdot \partial_\nu \boldsymbol{r}$. The metric also appears in the surface area element $\mathrm{d}S = \sqrt{\det g}\mathrm{d}\theta\mathrm{d}\phi = \frac{\sqrt{\det g}}{\sin(\theta)}\mathrm{d}\Omega$.
Combining eqns. 2.1 and 2.2 gives us an expression for the total energy of a vesicle, as a function of the magnitudes and positions of n point forces. The road from the general energy functional to the macroscopic energy function can be broken down into seven steps, which are detailed in the appendix.
Most importantly, we can derive a linearized shape equation, by using the Monge gauge (eqn. S.5.13) and then minimizing the surface energy functional with respect to small deformations:

$$\nabla_s^4 u + (2-\bar{\sigma})\nabla_s^2 u - 2\bar{\sigma} u = \sum_{i=1}^{n} \bar{f}_i \bar{\Delta}_i = \bar{\mathbf{f}} \cdot \bar{\Delta}, \tag{2.4}$$

where $u(\theta,\phi)$ is the relative deviation of the vesicle from a perfect sphere of radius R at position (θ,ϕ) as defined in eqn. S.5.13. We also introduced the non-dimensionalized parameters $\bar{\sigma} = \frac{\sigma R^2}{\kappa}$ and $\bar{f}_i = \frac{f_i R}{\kappa}$. After finding the Green's function of the left hand side (eqn. S.5.25), its superposition gives the full solution of the inhomogeneous equation (eqn. S.5.29), i.e. the deformation field u at any point (θ,ϕ). After a couple more steps we obtain the final energy function

$$E_{\text{tot}} = -\kappa \sum_{j>i} \bar{f}_i \bar{f}_j \sum_{\ell=2}^{\ell_{\max}} c_\ell(\bar{\sigma}) P_\ell \left[\cos\left(\alpha_{ij}\right)\right], \tag{2.5}$$

where P_ℓ denotes ℓ-th Legendre polynomial and $\cos(\alpha_{ij})$ is the angle between the i-th and j-th microtubules at (θ_i,ϕ_i) and (θ_j,ϕ_j). The exact relation between these angles is given in eqn. S.5.28. Finally $c_\ell(\bar{\sigma})$ is an expansion coefficient determined by solving the shape equation. The upper cutoff $\ell_{\max}$ corresponds to a high frequency (low wave length) cutoff which is justified by the fact that the membrane is not actually continuous but consists of lipids. In all of the proceeding analysis we used $\ell_{\max} = 20$.

2.3. RESULTS AND DISCUSSION

2.3.1. ANALYTIC RESULTS FOR TWO MICROTUBULES

The system of two microtubules acting on a membrane can be treated fully analytically and can provide a valuable insight, for the $n > 2$ microtubule interactions. For simplicity we assume that both forces generated by the microtubules[1] have the same magnitude $\bar{f}$. Then eqn. 2.5 simplifies to

$$E_{\text{tot}} = -\kappa s \bar{f}^2 \sum_{\ell=2}^{\ell_{max}} c_\ell(\bar{\sigma}) P_\ell \left[\cos(\alpha)\right]. \tag{2.6}$$

Where $s = \text{sign}(\bar{f}_1 \bar{f}_2)$ and $\alpha \in [0,\pi]$ is the angle between the two forces. Since α is the only configuration variable, we can easily plot the total surface interaction energy of the system. We do this because we are only interested in the interaction energy between the point forces and not in the actual energy of the vesicle surface. The exclusion of the

[1] From now on we will refer to the forces generated by the microtubules simply as point forces.

“self-interaction energy", which is the diagonal part in eqn. S.5.31, does not mean that it is physically not measurable. It simply does not contribute to the rearrangement of point the forces. Note that the energy only depends on relative sign of the point forces s and not on each sign separately. Meaning that in our model it is the same when both point forces pull or push. This is a consequence of linearizion and would not hold true if the deformations $u(\theta,\phi)$ become comparable with the vesicle size R.

In the case of two point forces of the same sign (both pushing or pulling) we can see a

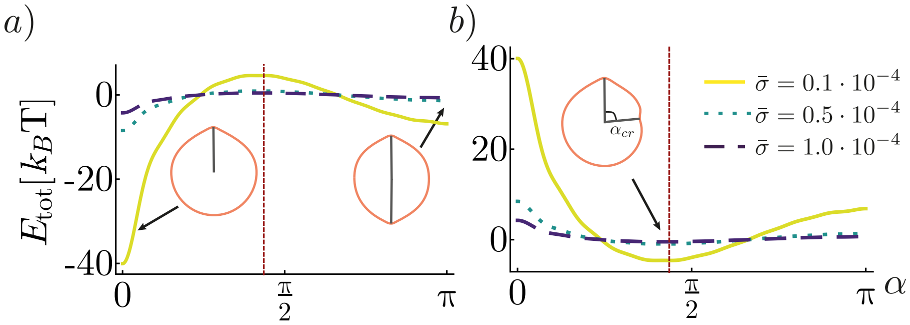

Figure 2.2: Total interaction energy for two point forces. The parameter s determines whether both forces are pushing (pulling) or one is pushing and the other pulling. **a)** $s = 1$. Both forces are pushing (pulling) and they experience strong short range attraction and weak long range repulsion. The critical angle determines weather the microtubules will coalesce or repel each other to the opposite poles. **b)** $s = -1$. In this case the forces have strong short range and weak long range repulsion, leading to a critical angle $\alpha_{\text{cr}} \approx 1.4$. In both subplots the vertical red (dashed) line denotes α_{cr}.

strong short range attraction together with a long range repulsion. This means that on a closed vesicle two point forces would either coalesce or repel each other until they reach the opposite poles of the vesicle, in both cases leading to emergent order (polar and nematic respectively). If the energy barrier at α_{cr} is too high ($E(\alpha_{\text{cr}}) - E(\pi) \gg k_B T$), the final state will depend on initial conditions. Forces that start close together will coalesce and forces that start further apart than α_{cr} will align at the opposite ends. In the case of one pulling and one pushing force we have only one ground state, in which the two point forces have a relative angle $\alpha_{\text{cr}} \approx 1.4$ rad.

Since the physical parameters of the system can be grouped into $\bar{f}$ and $\bar{\sigma}$, and κ is simply rescaling the energy, we can plot the full phase space of the system. Fig. 2.3 shows the magnitude of $\langle\alpha\rangle$, which is the expected equilibrium angle between two point forces in the system, for given pair of parameters $\bar{f}$ and $\bar{\sigma}$. As we can see, for the most of the phase space $\langle\alpha\rangle = 0$, meaning that the two point forces are coalesced in the equilibrium. However for small enough forces (or stiff enough membranes) $\langle\alpha\rangle \approx \frac{\pi}{2}$. In this disordered state the value of α is essentially random and the average is $\frac{\pi}{2}$ due to symmetry. Unfortunately the equilibrium average cannot predict the second (meta)stable state at $\alpha = \pi$, since the latter is not a global minimum. The most interesting feature of figure 2.3, however, is that the shape of the plot is effectively independent of $\bar{\sigma}$. This intriguing property

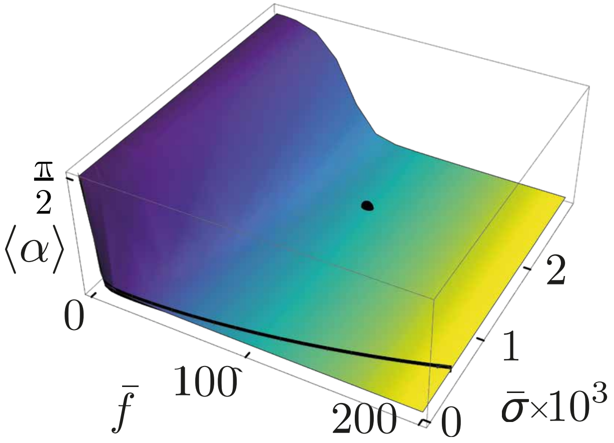

Figure 2.3: Expectation value of the equilibrium angle, between two point forces $\langle\alpha\rangle$, as a function of the magnitude of the force $\bar{f}$ and the membrane tension $\bar{\sigma}$. If the expected equilibrium angle between the point forces is 0 then they will coalesce. If $\langle\alpha\rangle \approx \frac{\pi}{2}$ then the system is in a disordered state where the orientation of two point forces is random and the $\frac{\pi}{2}$ state is "winning" in the average by having the largest degeneracy. Black line denotes the boundary of the tether pulling region, $\bar{f} = 2\pi\sqrt{2\bar{\sigma}}$. The black dot denotes the typical values used in our simulations, placing us firmly in the small deformation regime.

of the model is hiding in eqn. S.5.26. If we expand $c_\ell(\bar{\sigma})$ for large $\bar{\sigma}$ we get

$$c_\ell(\bar{\sigma}) = \frac{1}{4\pi\bar{\sigma}}\frac{2\ell+1}{\ell(\ell+1)-2} = \frac{\bar{c}_\ell}{4\pi\bar{\sigma}}. \tag{2.7}$$

This expansion is justified since the biologically relevant cases have $\bar{\sigma} \gtrsim 10^3$. In fact for very small $\bar{\sigma}$ we see the shape of the phase space plot depends on both $\bar{\sigma}$ and $\bar{f}$. For large values of $\bar{\sigma}$ eqn. 2.5 simplifies to

$$E_{\text{tot}} = \frac{\kappa\bar{f}^2}{4\pi\bar{\sigma}}\sum_{j>i} s_i s_j \gamma\left(\alpha_{ij}\right). \tag{2.8}$$

In 2.8 we again assumed for simplicity that all forces have the same magnitude, $s_i = \text{sign}(\bar{f}_i)$ and we defined the two point interaction propagator

$$\gamma(\alpha_{ij}) = -\sum_{\ell=2}^{\ell_{max}} \bar{c}_\ell P_\ell\left[\cos\left(\alpha_{ij}\right)\right], \tag{2.9}$$

which contains the interaction part of the energy and is separated from the physical parameters of the model. Hence the nature of these interactions can be studied, without the need to consider specific values of the material parameters.

As we stated in section 2.2, our analysis relies on the small deformation assumption. In order to remain in this regime we use forces throughout our analysis which are significantly below the force $\bar{f}_t = 2\pi\sqrt{2\bar{\sigma}}$ needed to stabilize a tether pulled from a vesicle, though the actual force needed to pull the tether is roughly 13% greater than this, for a point force as shown in [20]. It is also worth noting that in reality the forces acting on the membrane are not point like but rather distributed over a patch of finite size. Both experiments and theory show that in the case of such forces the barrier to pull a tether is much higher compared to $\bar{f}_t$ [21]. This means that the regime of small deformations is not only a theoretically convenient approximation but rather a biologically relevant and stable geometric phase of a deformed vesicle .We use $\bar{f}_t$ as our threshold for a “large" force (corresponding to a large deformation). In our analysis and simulations we use a micron sized membrane, i.e. $R \approx 10^3$ nm and material parameters and forces in typical range, $\kappa \approx 10\, k_{\mathrm{B}}T$, and $\sigma \approx 2\times 10^{-2}\, k_{\mathrm{B}}T/\mathrm{nm}^2$ [22, Table 2]. This leads to $\bar{\sigma} \approx 10^3$. For an estimate of the force magnitude, we use the force that a typical cytoskeletal filament can generate [23, Ch.3], which is around 5 pN and translates into $\bar{f} \approx 100$. This is far below the tether force of $\bar{f}_{\mathrm{t}}(\bar{\sigma} = 10^3) \approx 400$.

2.3.2. Interactions between multiple microtubules

Eqn. 2.8 for the total energy of a system with two microtubules suggests that the α_{ij} dependent part of the equation is a generic function that does not depend on system parameters. To calculate the total energy of n point forces, we simply add the two point interaction propagator for all interacting pairs, thus we expect the extrema of $\gamma(\alpha_{ij})$ to still be present in the total interaction term, which we name Γ

$$\Gamma(\{\alpha_{ij}\}) = \sum_{j>i} \gamma(\alpha_{ij}). \tag{2.10}$$

This means that the minimal energy states which are present in the n point force system are the same states as in the two point force system. Technically eqn. 2.10 does not prohibit the existence of additional energy minima. We therefore explore the state space of $n = 8-16$ microtubules using a Monte Carlo method. The results strongly suggest that there are no new stable configurations. The five configurations presented in fig.2.4 are the straightforward combinations of the three equilibrium configurations seen in fig.2.2.

To convince ourselves that the two point force analysis is also a good quantitative predictor of n point interactions we look at the dependence of the final configuration on the initial conditions. The prediction of eqn.2.8 is that each point force has a cone of attraction around it with an opening angle of α_{cr}. If two point forces (of the same kind, i.e, both pushing or both pulling on the membrane) fall into each other's cones of attraction they are likely to coalesce and if they fall outside of each other's cones they are more likely to end up at the opposite poles. To this end we ran several Monte Carlo simulations where we placed all particles inside a cone of opening angle α_{cone} at the beginning and then let the system converge against a final state. The difference in numbers of aligned pairs versus anti-aligned pairs can be estimated by a quantity proportional to the average relative angle $\langle\alpha_{ij}\rangle$. We hence define a polar order parameter

$$P = 1 - \frac{\langle\alpha_{ij}\rangle}{\pi/2} \tag{2.11}$$

which is one in the polar phase i.e. when $\langle\alpha_{ij}\rangle = 0$, and it is zero in the nematic and disordered phases when $\langle\alpha_{ij}\rangle = \frac{\pi}{2}$. Hence to truly distinguish between the polar and nematic ordered states we need a second parameter D which measures the level of noise in the system and acts as a “disorder" parameter. As such $(P = 1, D = 0)$ and $(P = 0, D = 0)$

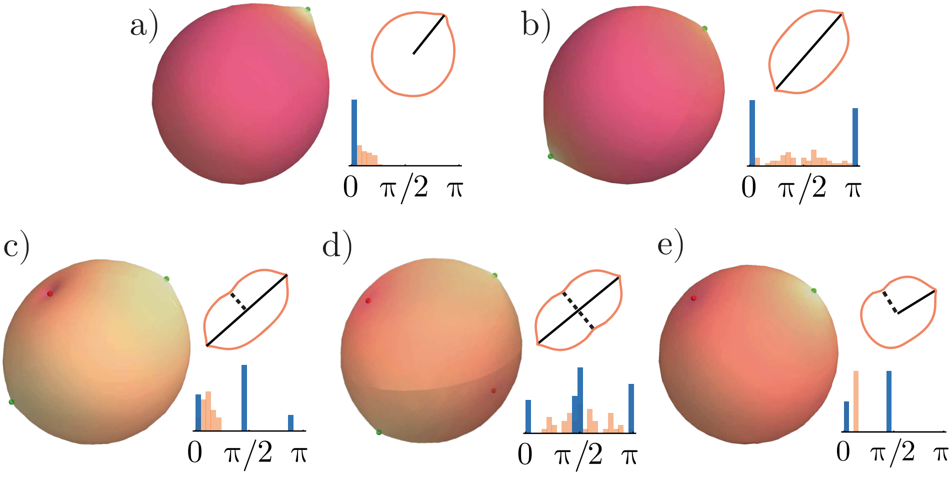

Figure 2.4: All five configurations of the final state of the Monte Carlo simulations of multiple point forces acting on the vesicle. Large 3D images are the actual plots of the final state of the simulation, accompanied by a sketch of the 2d cross section of the state (top) and a histogram (bottom) of the initial (light/orange bars) and final (dark/blue bars) distributions of the angles α_{ij} between the point forces. In all simulations we used the following parameter values: $\bar{f} = 100$, $\bar{\sigma} = 2000$, $R = 1\mu m$ and $\kappa = 10k_B T$. **a)** and **b)** 10 pushing microtubules. **c)** and **d)** 5 pushing and 5 pulling microtubules. **e)** 2 pushing and 1 pulling microtubules. **a)** Since all microtubules were close together initially they coalesced in one point. **b)** Here we have a broad initial distribution of microtubule positions consequently the final state has two points of bundled microtubules aligned at the opposite ends. **c)** Even if the initially all microtubules are close together, it is still likely that at least one of the flavors of microtubules (pushing or pulling) will create two poles of bundles. **d)** The most often observed configuration of the final state, where both flavors of microtubules are in nematic alignment. **e)** This state is very rare outside of 2 and 3 microtubule systems. Essentially for both flavors to be in polar alignment the initial conditions need to be perfectly tuned. Here we achieve this by having only one pulling force, hence depriving that flavor the possibility to make a nematic pattern and start with all microtubules close together such that the pushing ones are more likely to coalesce.

states correspond to polar and nematic orders respectively. The definition of D can be found in appendix S.6. In Fig. 2.5 we plot the dependence of the polar order parameter P on the opening angle of the initial state bounding cone and see that with growing cone angle α_{cone} we find a transition between polar and nematic end-states. This transition happens in the vicinity of α_{cr} as predicted from two point interaction analysis. Thus our point that $n > 2$ behavior does not qualitatively differ from two microtubule interactions is supported by this statistical analysis of the Monte Carlo simulations.

2.4. CONCLUSION

Together with actin and intermediate filaments, microtubules form an architecture that governs the shape of a cell, and therefore that of the plasma membrane surrounding it. The membrane, in turn, mediates the interaction between attached microtubules. Using analytical and numerical tools, we studied the effect of membrane mediated interactions on the rearrangement of microtubules. We found that force generating micro-

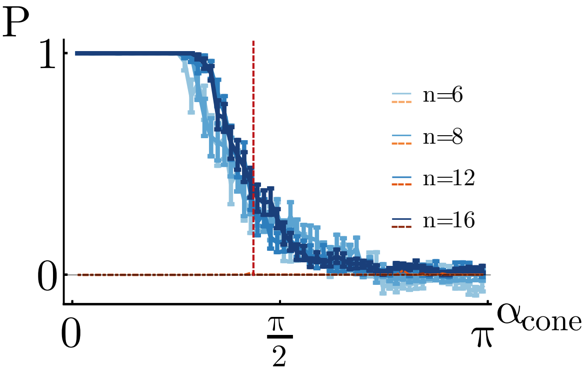

Figure 2.5: The dependence of the polar order P parameter on the opening angle of the cone of initial configurations α_{cone}, which represents the maximum mutual angle, any two point forces can have at the beginning of the simulation. Each point is an average over several runs (the number varies between 10 and 30 Monte Carlo runs depending on α_{cone}). The error bars display the sample standard error. Different shades of blue (solid with error bars) and orange (horizontal dot-dashed) lines depict the polar order and disorder parameters respectively, and the shade of the color indicates the amount of point forces pushing on the membrane. In all simulations we used the following parameter values: $\bar{f} = 100$, $\bar{\sigma} = 2000$, $R = 1\,\mu$m and $\kappa = 10\,k_{\mathrm{B}}T$. The vertical red (dashed) line marks the position of the critical angle between two interacting point forces $\alpha_{\mathrm{cr}} \approx 1.4$.

tubules, when colliding with a deformable obstacle like a fluid membrane, can coordinate their growing state through the shape of distorted membrane between them. Our results suggest that the elastic properties of cellular membranes facilitate the bundling of microtubules. In particular, we showed that two vesicle-encapsulated microtubules attract each other for small angular separations and repel for large angles. As we demonstrated for up to 16 microtubules numerically and motivated analytically for any number, the outcome of collective interactions between multiple filaments is microtubule coalescence, which may be harnessed for protrusion formation [14, 24]. Putting all the results together, our study suggests a possible mechanism underlying the preference of filaments for organizing in parallel configurations [25].

ACKNOWLEDGMENTS

AV was supported by the Netherlands Organisation for Scientific Research (NWO/ OCW), as part of the Frontiers of Nanoscience program and GD by the “BaSyC – Building a Synthetic Cell” Gravitation grant (024.003.019) of the Netherlands Ministry of Education, Culture and Science (OCW) and the Netherlands Organisation for Scientific Research (NWO).

S.5. DERIVATION OF MACROSCOPIC ENERGY

In this part of the appendix we want to step through the derivation of the macroscopic energy. I.e. we want to show the steps that bring us from eqn.'s 2.1 and 2.2 to the eqn. 2.5. The total energy which is simply the sum of eqns. 2.1 and 2.2 reads as

$$E_{\text{tot}} = E_{\text{el}} + E_{\text{PF}} = \int \mathrm{d}S \left(2\kappa H^2 + \sigma + (\mathbf{f}\cdot\boldsymbol{\Delta})\,|\boldsymbol{r}_{\text{ref}} - \boldsymbol{r}_{\text{def}}|\right) + \Delta P \int \mathrm{d}V. \tag{S.5.12}$$

In the deformation term we used a coordinate free way to describe deviations from the reference state $|\boldsymbol{r}_{\text{ref}} - \boldsymbol{r}_{\text{def}}|$, where $\boldsymbol{r}_{\text{ref}}$ and $\boldsymbol{r}_{\text{def}}$ describe the membrane surface in the reference and deformed states respectively. To proceed we need to have a way to talk about the geometric quantities like H dS and V, this means that we need to choose a coordinate system and parametrize our membrane surface in that system, i.e. define a specific functional form of $\boldsymbol{r}$.

S.5.1. STEP 1: MONGE GAUGE IN SPHERICAL COORDINATES

We want to make use of the small deformation regime later to expand the surface energy and derive an analytically solvable shape equation. The natural parametrization for systems slightly deviating from a reference state is the Monge gauge. We use the spherical coordinates version of it to describe the membrane

$$r(\theta,\phi) = R\left[1 + u(\theta,\phi)\right], \tag{S.5.13}$$

with R the radius of the undeformed (*reference*-)sphere and $u(\theta,\phi)$ the relative deformation field, with respect to that reference-sphere. It is worth noting that this description itself is very general and can describe almost all shapes (apart from the ones with overhangs[2]). The small deformation regime is an additional assumption of

$$u \ll 1. \tag{S.5.14}$$

For this parametrization we can see how the magnitude of the deformation becomes:

$$|\boldsymbol{r}_{\text{ref}} - \boldsymbol{r}_{\text{def}}| = Ru(\theta,\phi), \tag{S.5.15}$$

as used in the main text.

S.5.2. STEP 2: SECOND ORDER EXPANSION IN ENERGY

Assuming small deformations, we expand H^2 and $\sqrt{\det g}$ up to the second order, and $g_{\mu\nu} = \partial_\mu \boldsymbol{r} \cdot \partial_\nu \boldsymbol{r}$ and $H = \frac{1}{2} g^{\mu\nu} C_{\mu\nu}$, and

$$C_{\mu\nu} = \frac{\partial^2 \boldsymbol{r}}{\partial_\mu \partial_\nu} \cdot \frac{\partial_\mu \boldsymbol{r} \times \partial_\nu \boldsymbol{r}}{\left|\partial_\mu \boldsymbol{r} \times \partial_\nu \boldsymbol{r}\right|}, \tag{S.5.16}$$

is the curvature tensor, also known as the 2nd fundamental form, with $\mu,\nu \in \{\theta,\phi\}$. In order to reliably perform this expansions we used MATHEMATICA, which gave the following

[2] In 3D this would be a shape enclosing a volume which does not contain a point that can be connected to all points on the surface with a straight line that crosses this surface only at that point.

results[3]:

$$\begin{aligned}\frac{\sqrt{\det(g)}}{R^2} &= \frac{1}{2}\csc(\theta)u_\phi^2 + (u+1)^2\sin(\theta)\\ &+\frac{1}{4}\csc(\theta)u_\theta^2\left(2\sin^2(\theta)+((2-3u)u-1)u_\phi^2\right),\end{aligned}\tag{S.5.17}$$

$$\begin{aligned}(2HR)^2 &= 3u^2-2u+1\\ &+\frac{1}{4}\left(\cot(\theta)u_\theta+u_{\theta\theta}\right)\left(\cot(\theta)u_\theta+u_{\theta\theta}+12u-4\right)\\ &+\frac{1}{4}\csc^2(\theta)u_{\phi\phi}\left(2\left(\cot(\theta)u_\theta+u_{\theta\theta}+6u-2\right)+\csc^2(\theta)u_{\phi\phi}\right),\end{aligned}\tag{S.5.18}$$

where we abbreviated the derivatives as indices i.e. $f_\mu = \partial_\mu f$. The expansions were performed up to the second order in u and all of its derivatives. Upon defining non-dimensionalized gradient and Laplacian operators in spherical coordinates, also known as surface gradient and Laplacian,

$$\left|\nabla_s f\right|^2 = f_\theta^2 + \sin^{-2}(\theta) f_\phi^2 \tag{S.5.19}$$

$$\nabla_s^2 f = \cot(\theta) f_\theta + f_{\theta\theta} + \sin^{-2}(\theta) f_{\phi\phi}^2 \tag{S.5.20}$$

and reshuffling some terms we get

$$\begin{aligned}E_{tot} = \kappa\int \mathrm{d}\Omega \Bigg[& 2\left(1+\frac{1}{4}\left(\nabla_s^2 u\right)^2 - \nabla_s^2 u \boxed{+u\nabla_s^2 u + \frac{1}{2}|\nabla_s u|^2}\right)\\ & +\frac{\sigma R^2}{\kappa}\left((1+u)^2 \boxed{+\frac{1}{2}|\nabla_s u|^2}\right)\\ & -\frac{\Delta P R^3}{3}(1+u)^3 - \sum_{i=1}^{n}\frac{f_i R}{\kappa}\bar{\Delta}_i u\Bigg]\\ = \kappa\int \mathrm{d}\Omega \Bigg[& 2\left(1+\frac{1}{4}\left(\nabla_s^2 u\right)^2 - \nabla_s^2 u + \boxed{\frac{1}{2}u\nabla_s^2 u}\right)\\ & +\frac{\sigma R^2}{\kappa}\left((1+u)^2 - \boxed{\frac{1}{2}u\nabla_s^2 u}\right)\\ & -\frac{\Delta P R^3}{3}(1+u)^3 - \sum_{i=1}^{n}\frac{f_i R}{\kappa}\bar{\Delta}_i u\Bigg].\end{aligned}\tag{S.5.21}$$

In the last equality sign we used partial integration to eliminate the absolute value signs

$$\int \mathrm{d}\Omega\, |\nabla_s u|^2 = \cancelto{0}{\int \mathrm{d}\Omega \nabla_s\cdot(u\nabla_s u)} - \int \mathrm{d}\Omega\, u\nabla_s^2 u$$

[3] this equations are directly exported form MATHEMATICA and are hence not in the most readable form.

S.5.3. STEP 3: ENERGY VARIATION

In previous step we arrived at a simplified energy functional in terms of a deformation field $u(\theta,\phi)$. Next step is to find a surface configuration described by $u(\theta,\phi)$ that minimizes this energy. For this end we vary the energy with respect to this field (i.e. $u \to u+\delta u$), and further leverage the small deformation limit to replace ΔP with the Laplace pressure with the value for the sphere $\Delta P \approx 2\sigma/R$ we get a shape equation.

$$\nabla_s^4 u + (2-\bar{\sigma})\nabla_s^2 u - 2\bar{\sigma} u - 2\bar{\sigma} u^2 = \bar{\mathbf{f}}\cdot\bar{\Delta} \tag{S.5.22}$$

S.5.4. STEP 4: LINEARIZATION

Upon throwing away the nonlinear term in the shape equation S.5.22, we arrive at eqn. 2.4.

S.5.5. STEP 5: GREEN'S FUNCTION

To solve the eqn. 2.4 we first find its Green's function. This is a function $G(\theta-\theta_i,\phi-\phi_i)$ that solves the equation:

$$\nabla_s^4 G + (2-\bar{\sigma})\nabla_s^2 G - 2\bar{\sigma} G = \delta(\phi-\phi_i)\delta\left(\cos(\theta)-\cos(\theta_i)\right) \tag{S.5.23}$$

After expanding the delta functions in terms of spherical harmonics

$$\delta(\phi-\phi_i)\delta\left(\cos(\theta)-\cos(\theta_i)\right) = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} Y_l^m(\theta,\phi)Y_l^{m*}(\theta_,\phi_i)$$

where ‘ $*$ ‘ denotes complex conjugate, we get a solution

$$G(\theta-\theta_i,\phi-\phi_i) = \sum_{\ell=0}^{\infty}\sum_{m=-\ell}^{\ell} c_\ell(\bar{\sigma})Y_l^m(\theta,\phi)Y_l^{m*}(\theta_,\phi_i). \tag{S.5.24}$$

The first two modes in the expansion however need to be excluded. Since the zeroth mode ($\ell=0$) corresponds to the motion of the center of mass, which is irrelevant for our purposes and the first mode ($\ell=1$) corresponds to the conservation of the volume which is already accounted for in the energy functional with the ΔP term. Furthermore we use the addition theorem for spherical harmonics to simplify the second sum to ℓ-th degree Legendre polynomial, and finally arrive at

$$G(\theta-\theta_i,\phi-\phi_i) = \sum_{\ell=2}^{\infty} c_\ell(\bar{\sigma})P_\ell\left[\cos(\alpha_i)\right] \tag{S.5.25}$$

with the summand coefficients given by

$$c_\ell(\bar{\sigma}) = \frac{2\ell+1}{4\pi\left(\ell^2(\ell+1)^2-(2-\bar{\sigma})\ell(\ell+1)-2\bar{\sigma}\right)}, \tag{S.5.26}$$

and α_i defined as the angle between the i-th deforming point force and and any other point on the membrane given by θ and ϕ.

$$\cos(\alpha_i) = \sin(\theta)\sin(\theta_i)\cos(\phi-\phi_i)+\cos(\theta)\cos(\theta_i). \tag{S.5.27}$$

If this other point happens to be a position of j-th point force, then we get

$$\cos(\alpha_{ij}) = \sin(\theta_j)\sin(\theta_i)\cos(\phi_j-\phi_i)+\cos(\theta_j)\cos(\theta_i). \tag{S.5.28}$$

This makes $G(\theta-\theta_i,\phi-\phi_i)$ a propagator of deformation from (θ_i,ϕ_i) to (θ,ϕ).

S.5.6. STEP 6: GENERAL SOLUTIONS

The superposition of Greens' functions gives the deformation field u:

$$u(\theta,\phi)=\sum_{i=1}^{n} f_i G(\theta-\theta_i,\phi-\phi_i)=\bar{\mathbf{f}}\cdot\bar{\mathbf{G}}, \tag{S.5.29}$$

and hence

$$\mathbf{u}_0=\mathbf{M}\cdot\bar{\mathbf{f}} \tag{S.5.30}$$

Where $u_{0i}=u(\theta_i,\phi_i)$ is the vector of deformations at positions (θ_i,ϕ_i) and $M_{ij}=G(\theta_i-\theta_j,\phi_i-\phi_j)$ is the deformation interaction matrix. The diagonal elements M_{ii} are the self interaction terms and do not depend on the positions of the deformations.

Substituting u back in the energy functional gives us the macroscopic surface energy which only depends on the position and magnitude of deformations

$$E_{tot}=\frac{\kappa}{2}\mathbf{u}_0^T\cdot\mathbf{M}^{-1}\cdot\mathbf{u}_0+8\pi\left(1+\frac{\bar{\sigma}}{3}\right) \tag{S.5.31}$$

S.5.7. STEP 7: LEGENDRE TRANSFORMATION

We use the Legendre transformation to transform $\mathbf{u}_0$ into its conjugate variable $\bar{\mathbf{f}}$, then **omit all constant**[4] **terms** in the energy, use the symmetry of $\mathbf{M}$ and substitute the definition of the Green's function from the eqn.S.5.25

$$E_{tot}=-\kappa\sum_{j>i}\bar{f}_i\bar{f}_j\sum_{\ell=2}^{\ell_{max}}c_\ell(\bar{\sigma})P_\ell\left[\cos\left(\alpha_{ij}\right)\right]$$

Actually σ can be either understood as penalizing the projected area of the vesicle, i.e. the energy cost per surface area to flatten out the thermal fluctuations of the vesicle. Or as a chemical potential giving the energy cost of incorporating new lipids in the membrane. The actual stretching of the membrane, i.e increase of the surface area per lipid is energetically so costly that the membrane will rapture long before that mode of deformation is accessed [22, Table 2]. Point forces act along the radial direction of a spherical vesicle, i.e. $f_i=\mathbf{f}_i\cdot\hat{\mathbf{R}}_i$. Here $\mathbf{f}_i$ is the i-th point force and $\hat{\mathbf{R}}_i$ is the radial unit vector with spherical angles $\left(\theta_i,\phi_i\right)$.

S.6. DISORDER PARAMETER

To analyze the results of the Monte Carlo simulations we introduced the polar order parameter P in eqn. 2.11. Unfortunately $P=0$ is ambiguous since it could correspond to either nematic order or disorder. To distinguish between these states we defined a disorder parameter

$$D=\frac{1}{\pi/\sqrt{12}}\frac{\mathrm{STD}(\alpha_{ij}^{<})+\mathrm{STD}(\alpha_{ij}^{>})}{2}, \tag{S.6.32}$$

where the prefactor $\frac{1}{\pi/\sqrt{12}}$ is the standard deviation of an uniform distribution on the interval $[0,\pi]$ and serves as a normalization factor. STD stands for the standard deviation

[4] Constant with respect to the position of deformations $\{(\theta_i,\phi_i)\}$.

of a sample and $\alpha_{ij}^{<}$ and $\alpha_{ij}^{>}$ are the mutual angles in the final state that are smaller or greater than $\pi/2$. They can be formally defined as

$$\alpha_{ij}^{\lessgtr} = \{\alpha_{ij} \lessgtr \frac{\pi}{2} | \forall i, j\}. \tag{S.6.33}$$

This parameter measures how narrow the peaks of the final distribution of α_{ij} are around the states 0 and π. Broad peaks would mean microtubules that did not fully coalesce, i.e., a disordered final state.

Author Contributions

AV and TI formulated the initial idea for the project. AV derived the linearized shape equation 2.4, found the Green's function eq. (S.5.25) and analyzed the two point interaction shape presented in fig. 2.2. GD Performed the analysis of the two point interactions presented in fig. 2.3. All simulations and extension to many point force interactions were performed by GD. AV GD and TI wrote the introduction. GD and TI wrote the remainder of the text.

BIBLIOGRAPHY

[1] D. A. Fletcher and R. D. Mullins, “Cell mechanics and the cytoskeleton”, Nature **463**, 485 (2010).

[2] C. Revenu, R. Athman, S. Robine, and D. Louvard, “The co-workers of actin filaments: from cell structures to signals”, Nature reviews. Molecular cell biology **5**, 635 (2004).

[3] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell, *The actin cytoskeleton* (WH Freeman, 2000).

[4] A. J. Ridley, “Rho gtpases and actin dynamics in membrane protrusions and vesicle trafficking”, Trends in Cell Biology **16**, 522 (2006).

[5] C. Le Clainche and M.-F. Carlier, “Regulation of actin assembly associated with protrusion and adhesion in cell migration”, Physiological Reviews **88**, 489 (2008).

[6] D. K. Fygenson, J. F. Marko, and A. Libchaber, “Mechanics of microtubule-based membrane extension”, Physical Review Letters **79**, 4497 (1997).

[7] V. Emsellem, O. Cardoso, and P. Tabeling, “Vesicle deformation by microtubules: a phase diagram”, Physical Review E **58**, 4807 (1998).

[8] L. Mesarec, W. Góźdź, S. Kralj, M. Fošnarič, S. Penič, V. Kralj-Iglič, and A. Iglič, “On the role of external force of actin filaments in the formation of tubular protrusions of closed membrane shapes with anisotropic membrane components”, European Biophysics Journal **46**, 705 (2017).

[9] M. S. e Silva, J. Alvarado, J. Nguyen, N. Georgoulia, B. M. Mulder, and G. H. Koenderink, “Self-organized patterns of actin filaments in cell-sized confinement”, Soft Matter **7**, 10631 (2011).

[10] E. Atilgan, D. Wirtz, and S. X. Sun, “Mechanics and dynamics of actin-driven thin membrane protrusions”, Biophysical Journal **90**, 65 (2006).

[11] J. W. Kerssemakers, E. L. Munteanu, L. Laan, T. L. Noetzel, M. E. Janson, and M. Dogterom, “Assembly dynamics of microtubules at molecular resolution”, Nature **442**, 709 (2006).

[12] J. Howard and A. A. Hyman, “Dynamics and mechanics of the microtubule plus end”, Nature **422**, 753 (2003).

[13] M. Liu, V. C. Nadar, F. Kozielski, M. Kozlowska, W. Yu, and P. W. Baas, “Kinesin-12, a mitotic microtubule-associated motor protein, impacts axonal growth, navigation, and branching”, Journal of Neuroscience **30**, 14896 (2010).

[14] T. M. Svitkina, E. A. Bulanova, O. Y. Chaga, D. M. Vignjevic, S.-i. Kojima, J. M. Vasiliev, and G. G. Borisy, “Mechanism of filopodia initiation by reorganization of a dendritic network”, J Cell Biol **160**, 409 (2003).

[15] C. Conde and A. Cáceres, "Microtubule assembly, organization and dynamics in axons and dendrites", Nature reviews. Neuroscience **10**, 319 (2009).

[16] P. G. Dommersnes and J.-B. Fournier, "The many-body problem for anisotropic membrane inclusions and the self-assembly of "saddle" defects into an "egg carton"", Biophysical Journal **83**, 2898 (2002).

[17] A. Vahid and T. Idema, "Pointlike inclusion interactions in tubular membranes", Physical Review Letters **117**, 138102 (2016).

[18] P. B. Canham, "The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell", Journal of theoretical biology **26**, 61 (1970).

[19] W. Helfrich, "Elastic properties of lipid bilayers: theory and possible experiments", Zeitschrift für Naturforschung C **28**, 693 (1973).

[20] I. Derényi, F. Jülicher, and J. Prost, "Formation and interaction of membrane tubes", Physical review letters **88**, 238101 (2002).

[21] G. Koster, A. Cacciuto, I. Derényi, D. Frenkel, and M. Dogterom, "Force barriers for membrane tube formation", Physical review letters **94**, 068101 (2005).

[22] R. Phillips, "Membranes by the numbers", in *Physics of biological membranes,* edited by P. Bassereau and P. Sens (Springer International Publishing, Cham, 2018), pp. 73–105.

[23] R. Milo and R. Phillips, *Cell biology by the numbers* (Garland Science, 2015).

[24] J. Weichsel and P. L. Geissler, "The more the tubular: dynamic bundling of actin filaments for membrane tube formation", PLoS Computational Biology **12**, e1004982 (2016).

[25] A. P. Liu, D. L. Richmond, L. Maibaum, S. Pronk, P. L. Geissler, and D. A. Fletcher, "Membrane-induced bundling of actin filaments", Nature Physics **4**, 789 (2008).

3

ENTROPY OF MIXING GOVERNS THE STABILITY OF PROTEIN DECORATED MEMBRANE TUBES

Membrane tubes are a common sight in biological context, from organelle shapes like the tube network of smooth endoplasmic reticulum to the filopodia in moving cells, tubes play an important part of life on the nano- and micro-scales. Because of this importance the study of formation and stability of tubular structures has been an active research topic in biophysics for more than two decades. One important factor in stability of membrane tubes is the interaction with curvature inducing proteins, like the ones containing banana shaped BAR domains. These protein decorated membranes can sustain highly curved shapes like tubes because the proteins can scaffold the membrane and induce curvature. But the story is more complex than the proteins just scaffolding the tube. Just like the proteins influence the membrane shape, the membrane shape influences the local protein concentrations and thus a feedback loop is created that can lead to complex behaviour and pattern emergence. Here we start from general principles that have been well established and systematically derive conditions for stability of homogeneous protein distributions. Surprisingly we find indications for two distinct mechanisms of instability that can lead to demixing.

1. INTRODUCTION

Biological membranes are a quasi two-dimensional fluids that behave like an elastic solid when deformed out of plane. When coupled with objects like colloids or proteins that can locally induce curvature, the membranes act like a background potential and mediate effective interactions between these deforming objects. This effect arises from the membrane's tendency to reduce its overall curvature, is known as curvature mediated interaction and leads to a wide variety of emergent patterns in biology [1–5]. One very commonly occurring shape is that of tubes, seen in cellular organelles like the mitochondrion and the endoplasmic reticulum [6, Ch.12]. Consequently, the study of creation and stability of membrane tubes has been an active field of research for last two decades. In this chapter we look at a model system of membrane tubes pulled from vesicles (see fig. 1.1). One can add proteins to the solution surrounding membrane tubes (or encapsulate the proteins inside the vesicles the tubes are pulled from), which have strong binding affinity to the membrane and induce curvature (see fig. 1.2 b) and c)). A phenomenon that is observed in this model system is the phase separation of these curvature inducing proteins into high and low density phases on the tube which in turn leads to highly and weakly curved regions [7, 8]. It has long been established both theoretically and experimentally that such tubular shapes can be formed and maintained through a variety of mechanisms; like through application of pushing and pulling forces [9–11], enrichment of anisotropic lipids in the membrane [12, 13] and through action of rod shaped proteins with intrinsic curvature [14, 15]. We also know that the converse can be true and an initial shape deformation can lead to demixing of multi-component membranes [16–18] or spatial pattern formation of membrane associated proteins [7]. Generally the line between curvature sensing and curvature induction by membrane associated proteins is blurred, and depending on concentration or preexistence of deformations, proteins can either sense or induce curvature [19]. Starting in the early 80's with the work of Leibler [20] there have been many analytical models describing interactions between curvature inducing proteins and elastic membranes. These models are based on the idea that the proteins modify the material parameters of the membrane; changing the spontaneous curvature of the membrane by introducing additional asymmetry on one side (see fig. 1.2 b)) or influencing the bending rigidity of the membrane. The models for two-component membranes are mathematically similar, replacing the bare membrane and protein decorated membrane by lipid A and lipid B. Starting from a

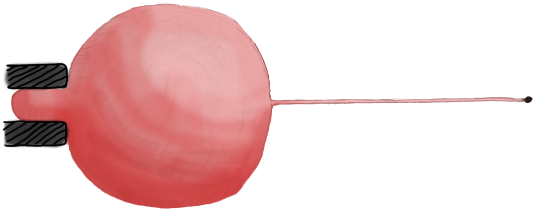

Figure 1.1: Sketch of the model system. When one pulls on a vesicle, the initial deformation looks like a dimple, however at a large enough strain a shape transition occurs and a membrane tube is formed.

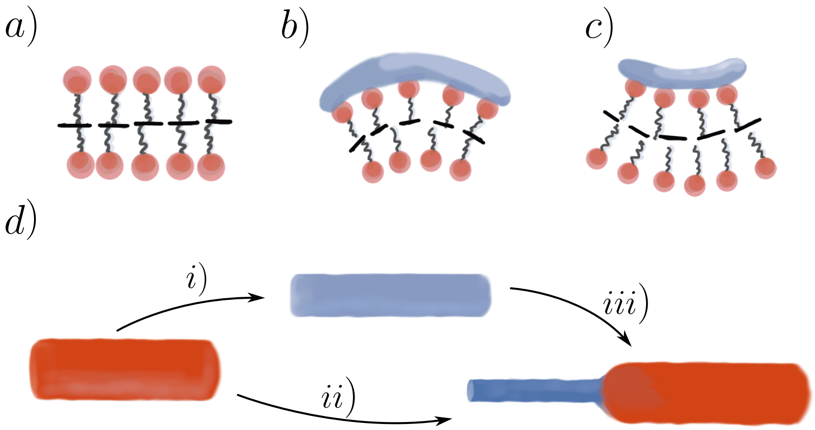

Figure 1.2: Sketch of the model system. Black dashed lines in drawings **a)**, **b)** and **c)** are the neutral surfaces of the membrane. **a)** Bare membrane without proteins with material parameters, bending rigidity (κ_b), spontaneous curvature (H_{0b}) and tension (σ_b). b) Membrane decorated with proteins that induce positive curvature, these proteins would prefer to attach on the outside of a tube. **c)** Membrane decorated with proteins that induce negative curvature, these proteins would prefer to attach on the inside of a tube. **b)** and **c)** have the same theoretical description, the presence of the proteins can modify the material parameters to new values, κ_p, H_{0p} and σ_p. **d)** Distinct macroscopic states of the system. (left) Bare membrane, (middle) homogeneously distributed proteins inducing same curvature everywhere (right) proteins are demixed into the high density and low density (or bare) phases and accordingly affect the tube curvature differently. When starting from a bare membrane, addition of proteins can lead to a transition into a state of homogeneously distributed proteins i) or a state of demixed proteins ii). In this work we start from the homogeneous state and analyze its stability by studying the demixing transition iii).

general model, we want to analyze which material parameters of the membrane need to be modified by proteins to reproduce experimentally observed demixing on membrane tubes.

2. MODEL

Our goal is to describe the separation of curvature inducing proteins on membrane tubes into high density, highly curved and low density, less curved regions, which corresponds to transition *ii)* in fig. 1.2 d). It is mathematically convenient to model a tube which is homogeneously covered by a low density of proteins (consequently only slightly reducing the overall tube radius) and then analyze the stability of this state, i.e. study the transition *iii)*. If the free energy density of the homogeneously covered tube is concave then the system will phase separate. A general free energy of membrane bound diffusing proteins can be written as a combination of the Canham-Helfrich free energy for fluid membranes [21, 22]

$$E_{\mathrm{CH}} = \int \mathrm{d}A \left[\frac{\kappa}{2} (H - H_0)^2 + \sigma \right] - fL, \tag{2.1}$$

and the Flory-Huggins free energy for polymer mixing [23, 24]

$$E_{\mathrm{FH}} = \frac{k_{\mathrm{B}}T}{a_p}\int \mathrm{d}A\big(c\log(c) + (1-c)\log(1-c) + \chi c(1-c)\big). \tag{2.2}$$

The constants κ and H_0 in eq. (2.1) are the membrane bending rigidity and spontaneous curvature, H is the local mean curvature of the membrane, σ is a Lagrange multiplier that constrains the area of the membrane, corresponding to the membrane tension and f is the force required to pull out a tube of length L from the vesicle. The eq. (2.2), describes the free energy of the protein density distribution, given by c, the relative area coverage fraction of the proteins, where $c = 0$ represents an empty region and $c = 1$ represents a region maximally packed with proteins. The free energy has an overall scaling factor $\frac{k_{\mathrm{B}}T}{a_p}$, where k_{B} is the Boltzmann constant T the absolute temperature and a_p the area occupied by a single protein. χ is the interaction parameter, encoding direct protein-protein interactions like electrostatic repulsion. We are only concerned with membrane mediated interactions here thus we will assume $\chi = 0$ for the remainder of this work. The integrals in both equations (2.1) and (2.2) are integrated over the full area of the membrane tube. We also restrict ourselves to systems where proteins can scaffold the membrane tube and thus reduce the pulling force necessary to sustain the tube to zero (i.e. $f = 0$) [7, 15]. To couple the membrane deformations and the protein concentration, we need to make the parameters in eq. (2.1) dependent on local protein concentration c. In the most general form this leads to the total system free energy

$$E = \int \mathrm{d}A\left[\frac{\kappa(c)}{2}\left(H - H_0(c)\right)^2 + \sigma(c) + \frac{k_{\mathrm{B}}T}{a_p}\big(c\log(c) + (1-c)\log(1-c)\big)\right]. \tag{2.3}$$

While each material parameter can have a complex dependence on the protein concentration c here we will consider a simple linear dependence

$$\kappa(c) = \kappa_b + c(\kappa_p - \kappa_b) \tag{2.4a}$$

$$H_0(c) = H_{0b} + c(H_{0p} - H_{0b}) \tag{2.4b}$$

$$\sigma(c) = \sigma_b + c(\sigma_p - \sigma_b). \tag{2.4c}$$

Since the the relative protein concentration can only take values between 0 and 1 ($c \in [0,1]$), the equations (2.4) interpolate between the material parameter values for the bare membrane (subscript b) and modified parameters for the protein decorated membrane (subscript p). The model system that we want to describe consists of symmetric membranes, thus we can eliminate the spontaneous curvature of the bare membrane i.e. $H_{0b} = 0$. Furthermore, we assume that the proteins do not modify the tension of the vesicle, i.e. $\sigma_p = \sigma_b$. This is based on the fact we consider proteins that do not interact directly, thus we do not expect them to build any superstructure that can withstand forces in the lateral direction like a membrane cytoskeleton would. This leaves us with five model parameters to consider: κ_b, κ_p, H_{0p}, σ_p and a_p.

To study the demixing of a homogeneous state we can look at the convexity of the mean field free energy density. This means that we can replace the spatial integral, the local curvature and the local protein concentration (that would depend on their local cylindrical coordinates θ and z) with their mean values

$$\begin{aligned}\int \mathrm{d}A &\rightarrow 2\pi RL\\ H(\theta,z) &\rightarrow \frac{1}{R}\\ c(\theta,z) &\rightarrow c\end{aligned}$$

This leads to the free energy

$$E = 2\pi RL\left[\frac{\kappa_b + c(\kappa_p-\kappa_b)}{2}\left(\frac{1}{R}-cH_{0p}\right)^2 + \sigma_b + \frac{k_\mathrm{B}T}{a_p}\left(c\log(c)+(1-c)\log(1-c)\right)\right],$$

which we nondimensionalize by scaling the energy by κ_b and expressing all length in terms of R_0, the radius of the undecorated tube, introducing

$$\bar{R} = \frac{R}{R_0} \tag{2.5a}$$

$$\bar{L} = \frac{L}{R_0}. \tag{2.5b}$$

$$\bar{E} = \frac{E}{\kappa_b} \tag{2.5c}$$

$$\bar{\kappa}_p = \frac{\kappa_p}{\kappa_b} \tag{2.5d}$$

$$\bar{H}_{0p} = H_{0p}R_0 \tag{2.5e}$$

$$\bar{\sigma}_b = \frac{\sigma_b R_0^2}{\kappa_b} = \frac{1}{2} \tag{2.5f}$$

$$\alpha = \frac{k_\mathrm{B}TR_0^2}{\kappa_b a_p} \tag{2.5g}$$

to write

$$\bar{E} = 2\pi\bar{R}\bar{L}\left[\frac{1+c(\bar{\kappa}_p-1)}{2}\left(\frac{1}{\bar{R}}-c\bar{H}_{0p}\right)^2 + \frac{1}{2} + \alpha\left(c\log(c)+(1-c)\log(1-c)\right)\right]. \tag{2.6}$$

To eliminate the rescaled tension in eq. (2.5f), we used the known relation between the radius bending rigidity and tension of a bare membrane tube [11], given by $R_0 = \sqrt{\kappa_b/(2\sigma_b)}$. Finally, we can also eliminate the rescaled radius, since we know that the membrane tubes will adapt the radius to minimize the energy. The condition

$$\partial_{\bar{R}}\bar{E}|_{\bar{R}=\bar{R}^*} = 0 \tag{2.7}$$

thus gives us the membrane radius $\bar{R}^*$ (see section S.5 for the full expression). Substituting the radius $\bar{R}^*$ back into eq. (2.6) and dividing by the non dimensional tube area $\bar{A} = 2\pi\bar{R}\bar{L}$ gives us an expression for free energy density

$$\bar{e}\,(c) = \frac{\bar{E}}{\bar{A}} \tag{2.8}$$

which only depends on three model parameters $\bar{\kappa}_p$, $\bar{H}_{0p}$ and α. The full expression can be seen in the supplement section S.6. An example of such energy for parameter values of $\bar{\kappa}_p = 1.5$, $\bar{H}_{0p} = 1.5$ and $\alpha = 1$ is given in fig. 2.3 a). We see that the free energy density has a concave region (second derivative is negative) around $c \approx 0.3$ indicated by red color. Negative second derivative indicates that the system is unstable for these parameter values and the proteins will demix into high and low density phases, which can be determined through the common tangent construction and for this example are $c_1 \approx 0.23769$ for the low density and $c_2 \approx 0.45157$ for the high density. A stability phase space for fixed $\bar{H}_{0p}$ and α can be seen in the carrot plot (fig. 2.3 b)), where the red color indicates the region of local instability, i.e. the boundary line between red and blue regions is the spinodal. The binodal line, which would indicate a globally unstable region is not shown in the plot. Since in our system the protein concentration on the tube can change freely, we are interested in the presence of instability for any concentration c. Thus, it is sufficient for us to find the region of local instability, which does not require the common tangent construction.

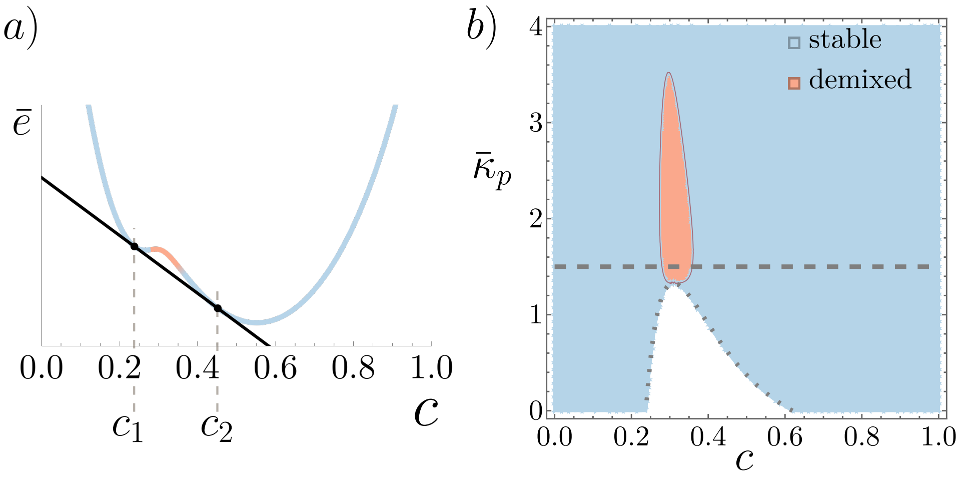

Figure 2.3: Spinodal demixing. **a)** Scaled free energy density per unit area $\bar{e}$ as a function of the protein concentration c (Solid blue line with a red region). The parameter values are chosen as $\bar{\kappa}_p = 1.5$, $\bar{H}_{0p} = 1.5$ and $\alpha = 1$. The concave region of the free energy density (second derivative is negative) around $c \approx 0.3$ is indicated by red color. The solid black line is the common tangent of the energy density at the low ($c_1 \approx 0.23769$) and high protein concentrations ($c_2 \approx 0.45157$). **b)** Phase diagram of the stable and unstable regions. The blue region has a positive second derivative of the free energy, i.e., a stable region. The red region has a negative second derivative of the free energy density, i.e. the homogeneous state is unstable and will decompose into high and low density phases trough spinodal decomposition. The uncolored area with the dotted boundary indicates the region where the stability analysis breaks down because the initial state is not well defined (see section S.5 for details). The X-axis: area coverage fraction of the proteins c, which varies from zero, no coverage, to one, full coverage. The Y- axis: bending rigidity of the protein covered tube $\bar{\kappa}_p$ measured in units of the bending rigidity of the bare membrane κ_b. The remaining parameters are fixed to $\bar{H}_{0p} = 1.5$ and $\alpha = 1$. Gray dashed line corresponds to the value of $\bar{\kappa}$ from part a).

3. Results and Discussion

Looking at the fig. 2.3 we can see that the system is unstable for quite a large range of κ_p, but this figure only shows us the stability of the system for fixed values of $\bar{H}_{0p}$ and α. In order to get a full picture we can distill the information provided by the entirety of the energy plot like in fig. 2.3 a) into one number. If the energy is concave for any range of c for a triplet of parameters $(\bar{\kappa}_p, \bar{H}_{0p}, \alpha)$ then we consider the system to be unstable for that triplet. Then sweeping over a wide range of combinations of triplets, we can build a view into the volume of the stability phase space, seen in figure fig. 3.4 a). Blue points correspond to a locally stable system and red points correspond to locally unstable ones. Strikingly we see that there are two distinct collections of red points, one large patch at large values of α and another smaller patch at value of α very close to zero. This means that there are two distinct instabilities through which the homogeneous concentration of the proteins demix. These unstable regions are separated by a large stable region. The parameter α that separates the two unstable regions represents the relative importance of entropic and mechanical terms in the energy (2.6). We can see from the definition (2.5g) that α is large if the initial tube radius squared is large compared to the protein area, or if the bending rigidity of the bare membrane is small compared to thermal fluctuations. This is to say that for small proteins on a large soft tube, the configuration matters more than for large proteins on small rigid tubes. What we can see in fig. 3.4 is that the configuration entropy has as strong impact on the stability of the system. The importance of mixing entropy for demixing is not very surprising on its own. In a theoretical study of tubulation induced by proteins which contain BAR domains (banana shaped curvature inducing proteins) it has already been noted by Góźdź et al. [18] that the mixing entropy can have a stabilizing effect on the homogeneity of protein distributions, whereas the absence of the entropic term in the energy leads to the formation of membrane tethers at lower overall concentration of BAR domain containing proteins. If the entropic term was considered in the energy then a much higher concentration of proteins was required to induce tubulation. What our work shows is

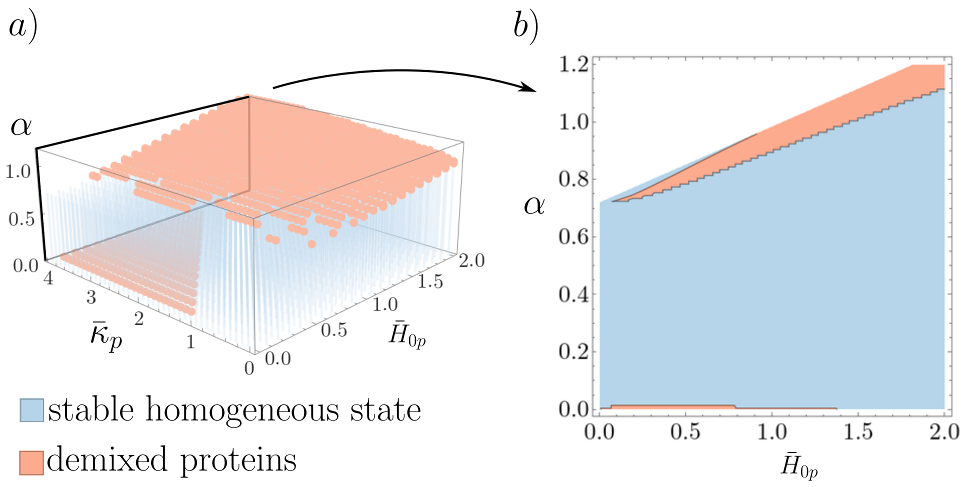

Figure 3.4: Stability of the homogeneous protein concentration on the tube. **a)** Full phase diagram of stability. For each triplet of parameters eiter a red or a blue point is plotted. The blue region is stable while the red region is unstable. Some combination of the parameters yielded complex values for the tube radius and accordingly the energy, those are outside the validity of our model and are not plotted (see section S.5 for more details). There are two distinct regions of instability. **b)** Single cross section of the full phase space. $\bar{H}_{0p}$-α plane for fixed $\bar{\kappa}_p = 4$. The ragged boundary edges in both plots are due to the limited resolution of the grid $(\bar{\kappa}_p, \bar{H}_{0p}, \alpha)$, and become smoother when the number of grid points is increased.

that these two types of behaviour are not merely two limits of the same phase transition but two different and disconnected branches, which implies that the phase transition mechanism is a different one. When we look at horizontal cross-sections from fig. 3.4 a) at high ($\alpha = 1$) and low ($\alpha = 0$) entropic contributions (see fig. 3.5 (left) and (right) respectively), we observe two different shapes of demixing regions indicating different responses to changes in parameters. At low entropy the protein induced bending rigidity $\bar{\kappa}_p$ and protein induced curvature $\bar{H}_{0p}$ have a purely antagonistic relationship. Induced spontaneous curvature only acts as a stabilizer, in contrast to bending rigidity which promotes mixing. This behaviour is reminiscent of formation of patches of liquid ordered (LO) and liquid disordered (LD) domains in multi component lipid membranes, where LO phase is characterized by a strong bending rigidity and the LD phase by weak bending rigidity[1] [25]. The relationship between $\bar{H}_{0p}$ and $\bar{\kappa}_p$ is more complicated in the parameter regime where the mixing entropy has a significant contribution. Now both induced curvature and bending rigidity act as stabilizers and the demixing transition only occurs for specific combinations of these parameters. Furthermore, the relationship is now inverse, i.e. the stronger the induced curvature, the less induced bending rigidity is required to stabilize the system.

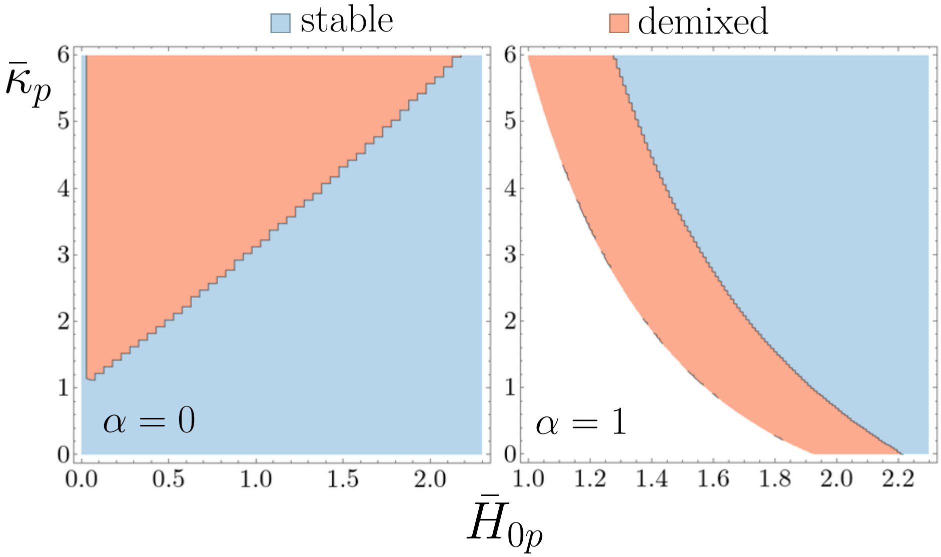

Figure 3.5: Phase diagram of stability. In the red region the homogeneous state demixes while it is stable in the blue region. (left) Horizontal cross section of the full phase space for $\alpha = 0$. (right) Horizontal cross section of the full phase space for $\alpha = 1$.

[1] Our model predicts a stable phase for $\hat{H}_{0p} = 0$ which may seem contradictory to the comparison with LO-LD domain formation since these domains form for symmetric lipids without any spontaneous curvature, however in that case line tension between the domains appears and acts as a stabilizing factor. Since we do not account for line tension our theory, a difference in the behaviour at $\hat{H}_{0p} = 0$ is to be expected.

4. CONCLUSION

In this chapter we started form a general energy functional of a system for curvature inducing proteins coupled to membrane deformations, systematically expanded all model parameters in powers of protein area coverage fraction c and then evaluated the stability of the system for different model parameters, by monitoring the convexity of the free energy density per unit area. Our results are consistent with previous work, that identifies the entropy of mixing as an important influencer of the stability behaviour, and further sheds light on this phenomenon. The systematic elimination of model parameters through model simplification and nondimensionalisation allowed us to visualize the full phase space of our system and realize that the high and low entropy cases are not merely two limits of the same behaviour but two different and disconnected regions of instability that likely come about through different mechanisms.

S.5. TUBE RADIUS: FULL EXPRESSION AND STABILITY ANALYSIS

The expression for the tube radius that minimizes free energy is given by

$$\bar{R}^* = \left(\frac{c(\bar{\kappa}_p - 1) + 1}{c^2 \bar{H}_{0p}{}^2 (c(\bar{\kappa}_p - 1) + 1) + 2\alpha \log(1-c) - 4\alpha c \tanh^{-1}(1-2c) + 1} \right)^{\frac{1}{2}}. \tag{S.5.9}$$

Since eq. (S.5.9) contains a square root it can become complex for some parameter combinations. These regions of complexity imply that our starting point of a homogeneously covered membrane tubes is not a minimal shape and thus our stability analysis is not valid in this parameter regime. The numerator of eq. (S.5.9) can never become negative since we only consider $\bar{\kappa} > 0$ and $0 < c < 1$. This means that the region boundary of model breakdown is given by the denominator of eq. (S.5.9). Plotting the parameters α, $\bar{H}_{0p}$ and $\bar{\kappa}_p$ against the protein density (see fig. S.5.6) reveals the regions of instability, from which we can read off to which parameter values we should restrict our analysis. We can plot the full parameter space (see fig. S.5.7) of the model breakdown if we assign each triplet a binary value (breakdown or not) if at any protein concentration for that triplet eq. (S.5.9) can become complex.

S.6. FULL EXPRESSION OF THE ENERGY

After plugging in the expression (S.5.9) into eq. (2.8) we get the full expression for the energy density

$$\begin{aligned} \bar{e}(c) = {} & \frac{1}{2} + \alpha \left[(1-c)\log(1-c) + c\log(c) \right] \\ & + \frac{1}{2}\left(\sqrt{c^2 \bar{H}_{0p}^2 (c(\bar{\kappa}_p - 1) + 1) + 2\alpha \log(1-c) - 4\alpha c \tanh^{-1}(1-2c) + 1} - c\bar{H}_{0p}\sqrt{c(\bar{\kappa}_p - 1) + 1} \right)^2. \end{aligned} \tag{S.6.10}$$

The minimization of the energy and all the numerical calculations for the phase space generation were done in a mathematicaMATHEMATICA [26] notebook which is available online [27].

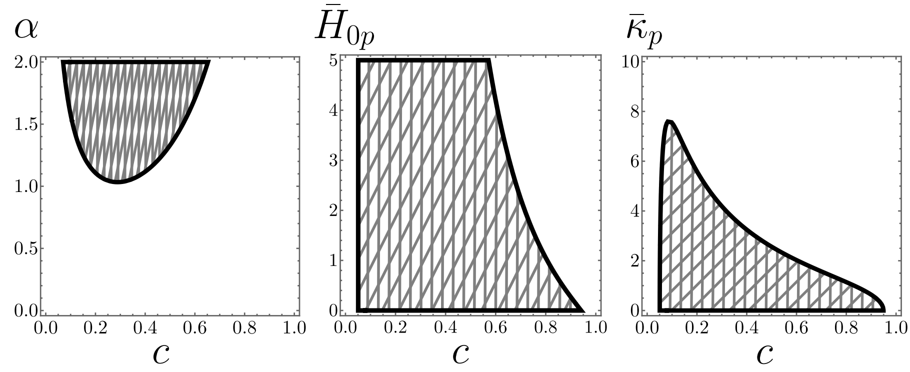

Figure S.5.6: Phase diagram of inaccessible regions. In hatched regions the combination of the parameters is such that the expression for the tube radius eq. (S.5.9) becomes complex. In these regions our approach breaks down. The X-axis indicates the area coverage fraction of the proteins c, which varies from zero, no coverage, to one, full coverage. The Y- axis indicates one of the model parameters, while the other two are fixed. (left) α is variable while $\bar{H}_{0p} = 1.5$ and $\bar{\kappa}_p = 2$. (middle) $\bar{H}_{0p}$ is variable while $\alpha = 2.5$ and $\bar{\kappa}_p = 2$. (right) $\bar{\kappa}_p$ is variable while $\alpha = 2.5$ and $\bar{H}_{0p} = 1.5$.

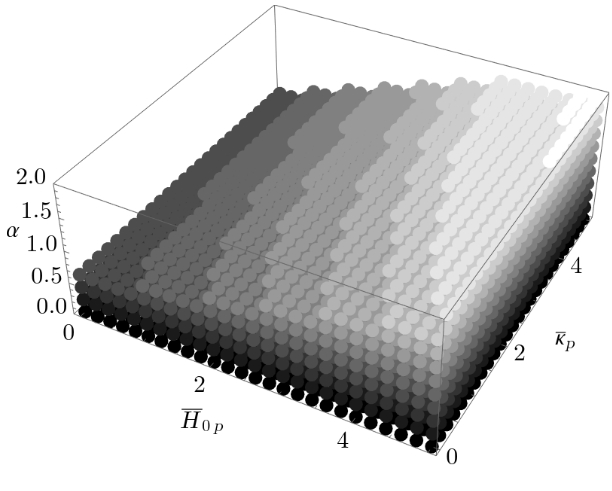

Figure S.5.7: Full phase diagram of inaccessible regions. Solid points indicate the combination of parameters for which the expression for the tube radius eq. (S.5.9) is not complex.

BIBLIOGRAPHY

[1] R. Shlomovitz and N. Gov, “Membrane-mediated interactions drive the condensation and coalescence of ftsz rings”, Physical biology **6**, 046017 (2009).

[2] C. Van Der Wel, A. Vahid, A. Šarić, T. Idema, D. Heinrich, and D. J. Kraft, “Lipid membrane-mediated attraction between curvature inducing objects”, Scientific reports **6** (2016).

[3] J. Agudo-Canalejo and R. Golestanian, “Pattern formation by curvature-inducing proteins on spherical membranes”, New Journal of Physics **19**, 125013 (2017).

[4] A. Vahid, A. Šarić, and T. Idema, “Curvature variation controls particle aggregation on fluid vesicles”, Soft Matter **13**, 4924 (2017).

[5] Z. Jarin, A. J. Pak, P. Bassereau, and G. A. Voth, “Lipid-composition-mediated forces can stabilize tubular assemblies of i-bar proteins”, Biophysical journal **120**, 46 (2021).

[6] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, “Molecular biology of the cell”, 5th ed.(Garland science, Taylor & Francis Group) (2008).

[7] C. Prévost, H. Zhao, J. Manzi, E. Lemichez, P. Lappalainen, A. Callan-Jones, and P. Bassereau, “IRSp53 senses negative membrane curvature and phase separates along membrane tubules”, Nature Communications **6**, 8529 (2015).

[8] M. Simunovic, E. Evergren, I. Golushko, C. Prévost, H.-F. Renard, L. Johannes, H. T. McMahon, V. Lorman, G. A. Voth, and P. Bassereau, “How curvature-generating proteins build scaffolds on membrane nanotubes”, Proceedings of the National Academy of Sciences **113**, 11226 (2016).

[9] I. Derényi, F. Jülicher, and J. Prost, “Formation and interaction of membrane tubes”, Physical review letters **88**, 238101 (2002).

[10] A. Roux, G. Cappello, J. Cartaud, J. Prost, B. Goud, and P. Bassereau, “A minimal system allowing tubulation with molecular motors pulling on giant liposomes”, Proceedings of the National Academy of Sciences **99**, 5394 (2002).

[11] G. Koster, A. Cacciuto, I. Derényi, D. Frenkel, and M. Dogterom, “Force barriers for membrane tube formation”, Physical review letters **94**, 068101 (2005).

[12] I. Tsafrir, Y. Caspi, M.-A. Guedeau-Boudeville, T. Arzi, and J. Stavans, “Budding and tubulation in highly oblate vesicles by anchored amphiphilic molecules”, Physical review letters **91**, 138102 (2003).

[13] V. Kralj-Iglič, H. Hägerstrand, P. Veranič, K. Jezernik, B. Babnik, D. R. Gauger, and A. Iglič, “Amphiphile-induced tubular budding of the bilayer membrane”, European Biophysics Journal **34**, 1066 (2005).

[14] J. Saarikangas, H. Zhao, A. Pykäläinen, P. Laurinmäki, P. K. Mattila, P. K. Kinnunen, S. J. Butcher, and P. Lappalainen, “Molecular mechanisms of membrane deformation by i-bar domain proteins”, Current biology **19**, 95 (2009).

[15] H. Noguchi, “Membrane tubule formation by banana-shaped proteins with or without transient network structure”, Scientific reports **6** (2016).

[16] A. Iglič, H. Hägerstrand, P. Veranič, A. Plemenitaš, and V. Kralj-Iglič, “Curvature-induced accumulation of anisotropic membrane components and raft formation in cylindrical membrane protrusions”, Journal of Theoretical Biology **240**, 368 (2006).

[17] M. Heinrich, A. Tian, C. Esposito, and T. Baumgart, “Dynamic sorting of lipids and proteins in membrane tubes with a moving phase boundary”, Proceedings of the National Academy of Sciences **107**, 7208 (2010).

[18] W. Góźdź, N. Bobrovska, and A. Ciach, “Separation of components in lipid membranes induced by shape transformation”, The Journal of Chemical Physics **137**, 015101 (2012).

[19] S. C. J. Helle, Q. Feng, M. J. Aebersold, L. Hirt, R. R. Grüter, A. Vahid, A. Sirianni, S. Mostowy, J. G. Snedeker, A. Šarić, et al., “Mechanical force induces mitochondrial fission”, Elife **6**, e30292 (2017).

[20] S. Leibler, “Curvature instability in membranes”, Journal de Physique **47**, 507 (1986).

[21] P. B. Canham, “The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell”, Journal of theoretical biology **26**, 61 (1970).

[22] W. Helfrich, “Elastic properties of lipid bilayers: theory and possible experiments”, Zeitschrift für Naturforschung C **28**, 693 (1973).

[23] P. J. Flory, “Thermodynamics of high polymer solutions”, The Journal of chemical physics **10**, 51 (1942).

[24] M. L. Huggins, “Thermodynamic properties of solutions of long-chain compounds”, Annals of the New York Academy of Sciences **43**, 1 (1942).

[25] S. Semrau, T. Idema, T. Schmidt, and C. Storm, “Membrane-mediated interactions measured using membrane domains”, Biophysical journal **96**, 4906 (2009).

[26] Wolfram Research,Inc., *Mathematica, Version 12.0*, Champaign, IL, 2019.

[27] G. Dadunashvili and T. Idema, “Mathematica file for Entropy of mixing governs the stability of protein decorated membrane tubes”, version 1.0.0, `10.5281/zenodo.6477250` (2022).

4

FLIPPY: USER FRIENDLY AND OPEN SOURCE FRAMEWORK FOR LIPID MEMBRANE SIMULATIONS

Eukaryotic cells are both encapsulated and compartmentalized by lipid bilayer membranes. More than just acting as boundaries, the shapes of these membranes influence the function of the cell and its compartments. Physically, membranes are two dimensional fluids with complex elastic behavior, which makes it impossible, for all but a few simple cases, to predict membrane shapes analytically. Instead, the shape and behavior of biological membranes can be determined by simulations. The setup and use of such simulations however require a significant background in programming. Open-source and user-friendly packages for these simulations are currently lacking. Here, we present "flippy", an open source package for simulating lipid membrane shapes, their interaction with proteins or external particles, and the effect of external forces. Our goal is to provide a tool which is easy to use without sacrificing performance or versatility. "flippy" is an implementation of a dynamically triangulated membrane. We use a mathematically rigorous way of discretizing the surface, which is independent of the topology and thus can be used for flat, spherical or tubular membranes. This discretization also allows the inclusion of local spontaneous curvature. Finally, in flippy we can also include regions of purely elastic (non-fluid) membranes, and thus explore a rich variety of shapes as encountered in living systems.

This chapter is based on: G.Dadunashvili, T Idema, flippy: User friendly and open source framework for lipid membrane simulations (in preparation)

1. INTRODUCTION

Lipid bilayer membranes form the envelopes of all known cells, and of many organelles contained in cells. These biological membranes are highly flexible materials, capable of adopting many nontrivial shapes, corresponding to specific cell functions and in response to environmental circumstances. Therefore, we can infer which processes are going on inside a cell or organelle from the shapes of their membranes [1]. The inverse of this is also true in synthetic biology. In recent years attempts have been made to create aspects of membrane bounded, self replicating systems from bottom up [2–5]. Achieving symmetric division, that is stable over many generations, is a major challenge in bottom up assembly of living cells. A key part of the problem is to understand how membrane reshaping is driven by external mechanical and chemical cues. Unfortunately predicting shapes of membranes analytically is very difficult, and therefore limited to cases of membranes with few constraints and high symmetry. Even numeric solutions to analytic equations are usually only possible in highly symmetric cases. In order to predict membrane shapes for generic problems we need to use simulations. If we are interested in large scale membrane reshaping then full atomistic simulations are out of question due to computational constraints. Luckily there are several types of simulations to choose from, which describe the membrane on a coarse grained level, like self assembled membranes [6], phase fields based methods [7] and dynamically triangulated membrane Monte Carlo (DTMMC) simulations [8]. The latter method is very appealing from a theoretical perspective since it is based on minimizing membrane surface energy, which makes the interpretation of results easy and leaves an opportunity to connect the findings to an analytical model [9]. Thus, it is not surprising that the method of dynamically triangulated membrane simulations has found broad adoption in the field and has been used to model diverse set of experimental systems from membranes responding to osmotic conditions [9] and shear flow [10], to interactions of membranes with colloidal particles [11–15] and with proteins [16–18].

The apparent popularity of DTMMC simulations coupled with the rareness of publication of simulation code is the primary reason that compelled us to write the flippy software package. The biggest strength of an open source project is that it can grow and improve through the input of the community. While DTMMC simulations are popular, they are hard to write, and even harder to optimize. Even with only basic functionality, a DTMMC code quickly becomes large and hard to maintain. Therefore, to make further progress in development of DTMMC simulations, we need an open source library with a vibrant community and developer base around it. flippy is designed with this objective in mind.

2. DESIGN AND IMPLEMENTATION

2.1. DESIGN GOALS

In our opinion, an ideal simulation framework for membranes would not involve programming at all on the side of the end user. Simulating a membrane under a certain physical constraint would be akin to conducting the same experiment. This approach would drastically reduce the barrier to simulations. A fully interactive framework would only require the understanding of the experimental setup, and enable experimental researchers to directly compare their results to simulations. With this end-goal in mind, we designed a relatively small software library with a manageable amount of content. This starting point allows us to grow the library slowly, based on user feedback. Even though using a c++ library requires much more knowledge than just using interactive software, we keep user-friendliness and a high level of abstraction as our primary goals. We want our package to help biophysicists to get to implementing the specifics of their system, without needing to reinvent the wheel and code the whole dynamically triangulated membrane from scratch.

2.2. Mathematical basis of the implementation

Since we aspire to a user-friendly framework, flippy needs to implement utilities that will be used in almost every DTMMC simulation. This means that the triangulation provided by flippy needs to keep track of several key geometric quantities, like the local curvature vector, local area and local unit bending energy of each node. We also keep track of the global counterparts of these quantities, i.e. the total area and total unit bending energy of the triangulated shape. Since total curvature vector does not have a widely used physical interpretation, flippy does not keep track of that quantity. In addition, flippy also needs to keep track of the global volume of a triangulation , if the volume is mathematically defined for that shape. For the definitions of these quantities, we follow Guegen et al.[9]. Each node i has a local neighbourhood of nodes (its next neighbours). These nodes are stored in a counterclockwise manner (see fig. 2.1A), such that a cross product of edge vectors $\vec{\ell}_{ij}$ and $\vec{\ell}_{ij+1}$ that point from node i to nodes j and $j+1$ respectively, always points to the outside of a triangulation. The definition of an edge vector $\vec{\ell}_{ij}$ is given by

$$\vec{\ell}_{ij} = \vec{x}_j - \vec{x}_i, \tag{2.1}$$

where $\vec{x}_i$ is the lab frame coordinate of the i-th node. The counterclockwise orientation of next neighbours guarantees that all face normal vectors

$$\vec{n}_{i,j,j+1} = \vec{\ell}_{ij} \times \vec{\ell}_{ij+1} \tag{2.2}$$

can be relied to point to the outside of the shape. We want to assign an area A_i to each node, which does not overlap with the areas of the neighbouring nodes (highlighted red in fig. 2.1B). This means that we need to construct a Voronoi cell around each node. Following the construction in [19], the area of a Voronoi cell of node i inside a triangle $i, j, j+1$ is given by

$$A_{ij} = \frac{1}{8}\left(\cot\left(\alpha_{ij}^{j+1}\right) \|\vec{\ell}_{ij}\|^2 + \cot\left(\alpha_{ij+1}^{j}\right) \|\vec{\ell}_{ij+1}\|^2\right), \tag{2.3}$$

if the triangle $i, j, j+1$ is not obtuse (in the case of the obtuse triangle the Voronoi area extends outside that triangle and thus overlaps with areas of other nodes). The angles α_{ij}^{j+1} and α_{ij+1}^{j}, are the angles of the corners at the neighbours $j+1$ and j opposite to the edges i, j and $i, j+1$ respectively (see fig. 2.1B). Accounting for the possibility of obtuse angles, we get a conditional expression for the area A'_{ij}, belonging to the node i inside the triangle $i, j, j+1$ as

$$A'_{ij} = \begin{cases} A_{ij}, & \text{if triangle } i, j, j+1 \text{ is not obtuse} \\ \frac{1}{2} A_{i,j,j+1}, & \text{if triangle } i, j, j+1 \text{ is obtuse at node } i \\ \frac{1}{4} A_{i,j,j+1}, & \text{if triangle } i, j, j+1 \text{ is obtuse at node } j \text{ or } j+1 \end{cases} \tag{2.4}$$

where $A_{i,j,j+1}$ is the area of the triangle $i, j, j+1$ (see fig. 2.1C). Summing the node areas inside each triangle gives us the total node area

$$A_i = \sum_j A'_{ij}. \tag{2.5}$$

The curvature vector of the node i is then defined as

$$\vec{K}_i = \frac{1}{2A_i} \sum_j \left(\cot \alpha_{ij}^{j+1} + \cot \alpha_{ij}^{j-1} \right) \vec{\ell}_{ij}. \tag{2.6}$$

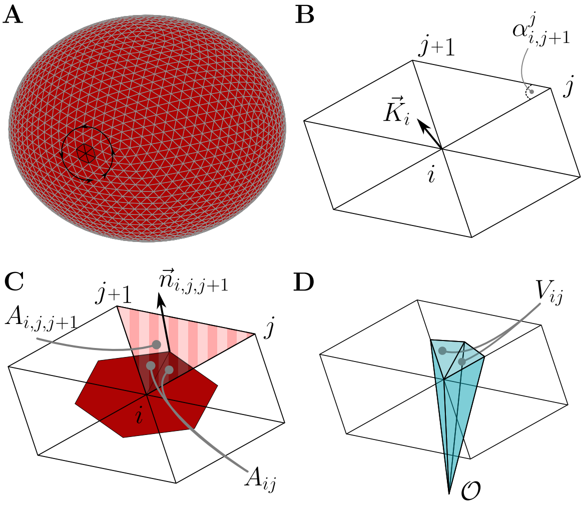

Figure 2.1: **Visualization of the triangulation.** **A**: Triangulated sphere with $N_{\text{nodes}} = 2252$. Black edges highlight the local neighbourhood of a node. Circular arrows show the counterclockwise orientation of the nodes. This choice guarantees that all normal vectors point to the outside of the sphere. **B**: An arbitrary node i, with its curvature vector $\vec{K}_i$ and a highlighted angle $\alpha^{j}_{i,j+1}$ at neighbour j opposite to the edge $i, j+1$. Superscript j denotes the neighbouring node to which the angle belongs and subscript $i, j+1$ denotes the edge opposite of the angle. **C**: Node i with its associated Voronoi area A_i highlighted in red. The node has an associated area inside each triangle it is part of. We also highlight the triangle $i, j, j+1$ (light red with stripes) with the face normal $\vec{n}_{i,j,j+1}$ and the area $A_{i,j,j+1}$. The part of this triangle that is associated to node i is highlighted in dark red and has the area A_{ij}. The convention here is to use the central node and the right most node in the subscript. Since the nodes are ordered counterclockwise this convention is unambiguous. **D**: Volume associated to node i is made up of tetrahedrons, that have as their base the triangles that make up the Voronoi cell of the node. The head of the tetrahedron points to some lab frame origin $\mathcal{O}$. V_{ij} is the part of the volume associated to node i that has its base in the triangle $i, j, j+1$.

This vector can be used to calculate the mean curvature at the node

$$|H_i| = \frac{1}{2}\|\vec{K}_i\|. \tag{2.7}$$

For closed shapes, flippy can also calculate their volume. To do so, we connect every triangle that makes up a node's Voronoi cell to the origin of the lab frame to create tetrahedra with signed volumes (see fig. 2.1D)

$$V_{ij} = A_{ij}\vec{x}_i \cdot \frac{\vec{n}_{ij,j+1}}{\|\vec{n}_{ij,j+1}\|}. \tag{2.8}$$

The local sum over the volumes of these tetrahedra gives us the node associated volume

$$V_i = \vec{x}_i \cdot \sum_j A_{ij}\frac{\vec{n}_{ij,j+1}}{\|\vec{n}_{ij,j+1}\|}, \tag{2.9}$$

which could be positive or negative depending on the position of the lab frame. Thus the local node volume does not necessarily have a physical interpretation, but the sum over all node volumes gives the correct total volume enclosed by the triangulation.

2.3. LANGUAGE CHOICE

flippy is implemented in c++, more specifically in the most modern standard of it (as of writing c++ 20). Since we aspire to develop a user-friendly package with low barrier to entry, one might expect flippy to be implemented in a scripting language that has a wide adoption. While it is true that a scripting language like Python would make the installation and adoption of the package much easier, the sacrifice in speed would make the code far less useful. Thus having an implementation in a compiled language is necessary to create fast simulations that enable users to quickly try out different parameters and get quick feedback. This is also more compatible with our end goal of creating a real time interactive simulation framework, since then the language implementation will be completely hidden from the end user, but the improvement in speed will be crucial.

DEPENDENCIES

The fact that c++ does not have a centralized package manager usually makes it hard to obtain or use external libraries. In our experience this is in fact the largest inconvenience, related to using the c++ language. In order to minimize this friction as much as possible, we opted to implement a header only library and eliminate almost all external dependencies. Our code only relies on an external JSON parser to easily save simulation data, but this parser is itself licenced under the same open source licence as flippy which enabled us to bundle it with flippy [20]. This makes using our software package as easy as it gets for c++ libraries.

2.4. TESTING AND BUG REPORTING

Every large code base is prone to hidden bugs and unexpected behaviours in new use cases. In order to minimize errors we implemented an extensive unit testing framework, and we are happy to report that our code base has over 95% coverage. This means that almost every function that is implemented in our base is covered by at least one test case, and we intend our libray of unit tests to continuously grow. We are aware that unit tests cannot guarantee that the code is bug free, and we intend to use the bug reporting facilities of the github repository to enable our users to report bugs and help improve the package.

3. RESULTS

In this section we want to demonstrate flippy's ability to abstract away the implementation details of a dynamic triangulation and Monte Carlo updating scheme. To this end, we want to go through the process of simulating a simple experimental system of a deflated giant unilamellar vesicle (GUV), and use flippy to predict the equilibrium shape of the vesicle. The full code of this simulation can be obtained on github, for more details please see the *Availability and future directions* section. In the following text we will only present the key elements of the code.

The system of a deflated GUV can be modeled by the following surface energy

$$E_{\text{surf}} = \frac{\kappa}{2}\int \mathrm{d}A(2H)^2 + K_A\frac{(A-A_t)^2}{A_t} + K_V\frac{(V-V_t)^2}{V_t}, \tag{3.10}$$

Where κ is the bending rigidity and H is the local mean curvature of the membrane. The integral $\int \mathrm{d}A$ goes over the whole area of the vesicle. This part of the energy describes the tendency of the biological membranes to minimize their local square mean curvature [21, 22]. The Lagrange multipliers K_A and K_V fix the area A and volume V to their target values $A_t = A_0 = 4\pi R_0^2$ and $V_t = 0.6V_0 = 0.6\frac{4\pi}{3}R_0^3$, where R_0 is the radius of the initial (pre deflation) spherical GUV. Since the deflation of the vesicle does not change its area, we keep it fixed to the initial value. However, the target value of the volume is fixed to 60% of the initial volume, to account for deflation. We picked 60% of the initial volume because, for this value we expect the equilibrium configuration to be a biconcave shape, providing an easy visual way to judge the success of the simulation.

In order to model this system with flippy we need to define an energy function. Since this energy will be used by the `MonteCarloUpdater`, its signature needs to follow a specific convention

```
double surface_energy(fp::Node<double, int> const&,
                      fp::Triangulation<double, int> const& ,
                      EnergyParameters const& )
```

where the first argument needs to be a flippy `Node` type, representing the node which is being updated, the second argument needs to be a `Triangulation` type representing the triangulation that is being updated, and the third argument can be any type, and is intended to be a user defined data struct containing all the parameters of the energy function. The actual function body is then a straight forward implementation of eq. (3.10)

```
double surface_energy([[maybe_unused]]fp::Node<double, int> const& node,
                      fp::Triangulation<double, int> const& trg,
                      EnergyParameters const& prms){
```

```
    double V = trg.global_geometry().volume;
    double A = trg.global_geometry().area;
    double dV = V-prms.V_t;
    double dA = A-prms.A_t;
    double energy = prms.kappa*trg.global_geometry().unit_bending_energy +
                    prms.K_V*dV*dV/prms.V_t + prms.K_A*dA*dA/prms.A_t;
    return energy;
}
```

here the first variable in the function signature is designated `[[maybe_unused]]`, since in this particular implementation of the energy we are not interested in local properties of any given node and thus do not use this variable.

The second step in the implementation of the model is to declare a triangulation:

```
fp::Triangulation<double, int> tr(n_triang, R_0, r_Verlet);
```

where the template parameters `double` and `int` specify which internal representation of floating point and integer numbers the `Triangulation` class is supposed to use. The first argument of the instantiation `n_triang` specifies the level of triangulation, which sets the fineness of the mesh. The second argument `R_0`, sets the initial radius of the triangulated sphere and the last argument `r_Verlet`, relates to the implementation of membrane self intersection avoidance. flippy implements a Verlet list to check spatial closeness of the nodes efficiently [23].

The third step is to declare a Monte Carlo updater, that will use the energy function to update the triangulation according to a Metropolis algorithm [24]:

```
fp::MonteCarloUpdater<double, int, EnergyParameters,
                      std::mt19937, fp::SPHERICAL_TRIANGULATION>
    mc_updater(tr, prms, surface_energy, rng, l_min, l_max);
```

The signature of this class instantiation is quite large since the updater needs to have knowledge of the energy function, all necessary update parameters and the triangulation. The first two template parameters specify the internal representation of numbers just like in the case of the `Triangulation` class, it is important that these parameters are the same in both case. `EnergyParameters` specifies the user defined struct type name, that contains the parameters used inside the energy function. `std::mt19937` specifies the type of the random number generator that we will be providing to the updater, for generating random numbers for the Metropolis algorithm. The last parameter specifies the type of the triangulation (currently spherical and planar triangulations are possible). The instance of the updater itself has six arguments, the first four provide the updater with references to the already declared instances of triangulation class, energy parameters struct, energy function and random number generator. The last two arguments specify minimum and maximum allowed distances between the nodes of the triangulation. All the complexity of creating and maintaining a dynamic triangulation is hidden in the above three conceptual steps; define the energy, initiate a triangulation, and initiate an updater that will use the energy to update the triangulation. All that is left to do is to create an update loop that specifies in what order and how often we want to update the triangulation. A simple example of this would be the following:

```
for(int mc_step=0; mc_step<max_mc_steps; ++mc_step){
    for (int node_id: shuffled_ids) {
        displ = {displ_distr(rng), displ_distr(rng), displ_distr(rng)};
        mc_updater.move_MC_updater(guv[node_id], displ);
    }
    std::shuffle(shuffled_ids.begin(), shuffled_ids.end(), rng);
    for (int node_id: shuffled_ids) {
        mc_updater.flip_MC_updater(guv[node_id]);
    }
}
```

where in every update step, we loop over each node and use the methods that are part of the MonteCarloUpdater class to move nodes and flip bonds. Between the move and flip passes over the nodes we shuffle the shuffled_ids vector, which was defined before the loop and contains the ids of the nodes.

The MonteCarloUpdater class is provided by flippy because, a Metropolis updating scheme is a popular one. However, the Triangulation class itself is completely agnostic towards the updating scheme that is used on it. The user is free to implement another updating scheme if the Metropolis algorithm is not suitable to their problem, and still be able to use the triangulation provided by flippy. This also enables us to easily extend flippy with new updaters.

Finally, we want to address the question of data serialization. To make the saving of the state of the simulation easy, flippy's Triangulation class has a method that saves the representation of the data as a json object, which is a text-based human-readable data format [25]. flippy comes bundled with an open source json parser [20]. A single statement is sufficient to create json data of the current state of the triangulation

```
fp::Json data = tr.make_egg_data();
```

and a utility function in flippy allows to save this data to a text file as follows:

```
fp::json_dump("test_run_final", data);
```

The make_egg_data methods naming refers to the fact that this json data contains the necessary information to reinitialize the triangulation (like an egg contains all the nutrients for the chicken that will hatch from it), thus allowing the user to continue simulation from a save-file. If we use this simple code [26] (which is comfortably below 100 lines, including all imports, variable definitions and comments) we will obtain (in few minutes) the expected biconcave shape (see fig. 3.2 B and C). This example shows clearly that flippy is capable of simulating a simple physical system in few lines of code, where all unnecessary complexity is abstracted away in the library and all the complexity that is still left in the user written code, contains necessary information about specific characteristics of the simulated system in question. And importantly this abstraction and simplicity does not come at the cost of unreasonable runtime of the simulation.

4. Availability And Futuredirections

The source code of flippy is currently available on github [27]. The code used for the simulation in the *Results* section is also part of flippy's github repository, and the most up-to-date version of it can be found in the demo/biconcave_shapes_MC folder of the repository [27]. The version of the code that was most up to date at the time of writing this paper, and was used to generate the code snippets in the *Results* section as well as

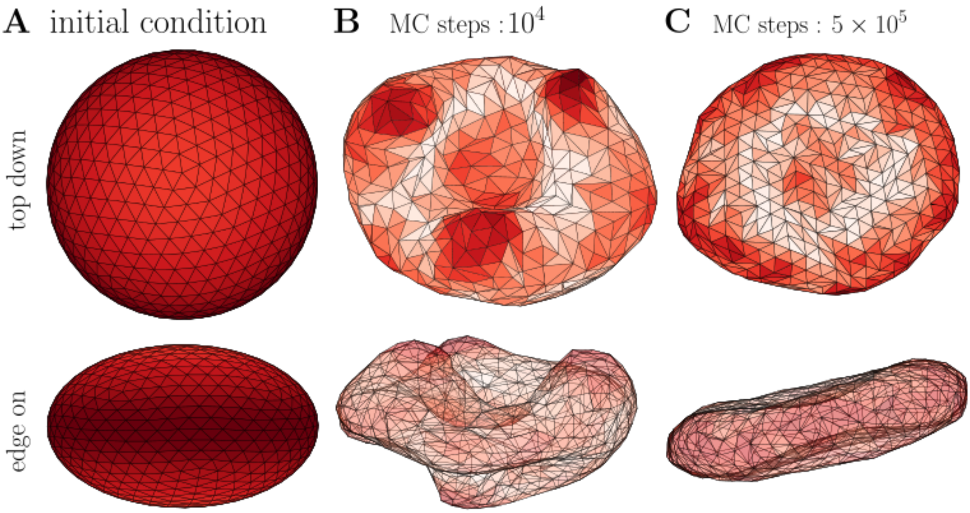

Figure 3.2: **Result of the Monte carlo Simulation** The triangulation consisted of 642 nodes. **A**: Initial condition. A slightly oblate spheroid. **B**: Simulation that ran for 10^4 Monte Carlo steps per node (took 10 seconds). **C**: Simulation that ran for 10^5 Monte Carlo steps per node (took approximately 7 minutes). C is not a longer run of B but a new simulation that ran longer than B. All simulations were done on Intel i7-8650U 1.9GHz processor.

the data for fig. 3.2, can be found in reference [26] We use this code for several projects internally and have developed a robust workflow for introducing updates. New features are first used and tested by us before incorporating it into flippy, which makes easier to detect and eliminate problems that slip by our unit testing framework. We intend to maintain and develop the code for years to come, while feedback and contributions from users are highly encouraged.

BIBLIOGRAPHY

[1] F. Frey and T. Idema, "More than just a barrier: using physical models to couple membrane shape to cell function", Soft Matter **17**, 3533 (2021).

[2] E. Godino, J. N. López, D. Foschepoth, C. Cleij, A. Doerr, C. F. Castellà, and C. Danelon, "De novo synthesized min proteins drive oscillatory liposome deformation and regulate FtsA-FtsZ cytoskeletal patterns", Nature Communications **10**, 4969 (2019).

[3] T. Litschel, B. Ramm, R. Maas, M. Heymann, and P. Schwille, "Beating vesicles: encapsulated protein oscillations cause dynamic membrane deformations", Angewandte Chemie International Edition **57**, 16286 (2018).

[4] J. Steinkühler, R. L. Knorr, Z. Zhao, T. Bhatia, S. M. Bartelt, S. Wegner, R. Dimova, and R. Lipowsky, "Controlled division of cell-sized vesicles by low densities of membrane-bound proteins", Nature communications **11** (2020).

[5] E. Rideau, R. Dimova, P. Schwille, F. R. Wurm, and K. Landfester, "Liposomes and polymersomes: a comparative review towards cell mimicking", Chemical society reviews **47**, 8572 (2018).

[6] H. Yuan, C. Huang, J. Li, G. Lykotrafitis, and S. Zhang, "One-particle-thick, solvent-free, coarse-grained model for biological and biomimetic fluid membranes", Physical review E **82**, 011905 (2010).

[7] T. Biben, K. Kassner, and C. Misbah, "Phase-field approach to three-dimensional vesicle dynamics", Physical Review E **72**, 041921 (2005).

[8] G. Gompper and D. M. Kroll, "Network models of fluid, hexatic and polymerized membranes", Journal of Physics: Condensed Matter **9**, 8795 (1997).

[9] G. Gueguen, N. Destainville, and M. Manghi, "Fluctuation tension and shape transition of vesicles: renormalisation calculations and monte carlo simulations", Soft Matter **13**, 6100 (2017).

[10] H. Noguchi and G. Gompper, "Fluid vesicles with viscous membranes in shear flow", Physical review letters **93**, 258102 (2004).

[11] A. Šarić and A. Cacciuto, "Mechanism of membrane tube formation induced by adhesive nanocomponents", Physical review letters **109**, 188101 (2012).

[12] A. Šarić and A. Cacciuto, "Fluid membranes can drive linear aggregation of adsorbed spherical nanoparticles", Physical Review Letters **108**, 118101 (2012).

[13] A. H. Bahrami, R. Lipowsky, and T. R. Weikl, "Tubulation and aggregation of spherical nanoparticles adsorbed on vesicles", Physical review letters **109**, 188102 (2012).

[14] C. Van Der Wel, A. Vahid, A. Šarić, T. Idema, D. Heinrich, and D. J. Kraft, "Lipid membrane-mediated attraction between curvature inducing objects", Scientific reports **6** (2016).

[15] A. Vahid, A. Šarić, and T. Idema, "Curvature variation controls particle aggregation on fluid vesicles", Soft Matter **13**, 4924 (2017).

[16] S. C. J. Helle, Q. Feng, M. J. Aebersold, L. Hirt, R. R. Grüter, A. Vahid, A. Sirianni, S. Mostowy, J. G. Snedeker, A. Šarić, et al., "Mechanical force induces mitochondrial fission", Elife **6**, e30292 (2017).

[17] G. Kumar, N. Ramakrishnan, and A. Sain, "Tubulation pattern of membrane vesicles coated with biofilaments", Physical Review E **99**, 022414 (2019).

[18] M. Fošnarič, S. Penič, A. Iglič, V. Kralj-Iglič, M. Drab, and N. S. Gov, "Theoretical study of vesicle shapes driven by coupling curved proteins and active cytoskeletal forces", Soft Matter **15**, 5319 (2019).

[19] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr, "Discrete differential-geometry operators for triangulated 2-manifolds", in *Visualization and mathematics iii* (Springer, 2003), pp. 35–57.

[20] N. Lohmann, *JSON for Modern C++*, version 3.10.5, Jan. 2022.

[21] P. B. Canham, "The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell", Journal of theoretical biology **26**, 61 (1970).

[22] W. Helfrich, "Elastic properties of lipid bilayers: theory and possible experiments", Zeitschrift für Naturforschung C **28**, 693 (1973).

[23] L. Verlet, "Computer" experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules", Physical review **159**, 98 (1967).

[24] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines", The journal of chemical physics **21**, 1087 (1953).

[25] ECMA, *Standard ecma-404 the json data interchange syntax*, 2017.

[26] G. Dadunashvili, *Demo code: generating biconcave shapes using flippy*, This URL links to the version of the code that was used in this paper.

[27] G. Dadunashvili, *Flippy-software-package/flippy: v1.0.0*, version v1.0.0, Mar. 2022.

5

EMERGENT ORDER IN A CONTINUOUS STATE ADAPTIVE NETWORK MODEL OF LIVING SYSTEMS

Order can spontaneously emerge from seemingly noisy interactions between biological agents, like a flock of birds changing their direction of flight in unison, without a leader or an external cue. We are interested in the generic conditions that lead to such emergent phenomena. To find these conditions, we use the framework of complex networks to characterize the state of agents and their mutual influence. We formulate a continuous state adaptive network model, from which we obtain the phase boundaries between swarming and disordered phases and characterize the order of the phase transition.

This chapter is based on: C.T. van de Kamp, G. Dadunashvili, J.L.A. Dubbeldam, T Idema, *Emergent order in a continuous state adaptive network model of living systems.*

1. INTRODUCTION

Swarming is a collective phenomenon which is encountered in many biological systems, such as schools of fish, swarms of locusts and flocks of birds. Some swarming phenomena in biological systems can be guided by external cues, or a leading individual, but a large variety of biological systems display spontaneously emerging, self-organized group behavior [1]. Even unicellular organisms, such as the bacterium *E. coli* and the amoeba *D. discoideum*, have been observed to display such collective behavior [2].

Models of swarm formation can be broadly put into two categories. The first group we call mechanistic models. Their microscopic interactions are solely based on first principles, and they do not encode the tendency to order explicitly. This works well if all microscopic forces that determine the dynamics are known. Such models have been successfully used to describe the dynamics of self propelled particles of different shapes that exhibit exclusion interactions [3–5]. The second group, that we call heuristic models, are well exemplified by opinion formation in social groups. We know that people can convince others to join their cause. However, we do not yet understand the details of how one individual can cause another to form an opinion, thus we have a need for a heuristic rule that postulates how two individuals can form a consensus. Such heuristic treatment is in general needed when it is clear that there is a microscopic tendency to align, but we do not know the details of interactions; in context of opinion dynamics, such a model was first put forward by Vicsek et al. [6].

Giving up the connection to the mechanistic first principles opens a door to an interesting opportunity. One can recognize that the heuristic of *be more like your neighbor* is now the central aspect of the evolution of the system, instead of spatial dynamics of the particles. Thus we are shifting our attention from spatial dynamics to the topology of interactions, a problem for which the language of complex networks is uniquely well suited. In this framework the system is represented by a dynamic network. The agents, corresponding to the nodes of the network, can occupy different states, representing the direction of movement, and the changing topology of the network encodes the temporal dynamics of the interactions. An example of such a system can be seen in fig. 1.1. One such dynamic network model, for a system with a discrete state space, has been proposed by Chen, Huepe and Gross [7]. This model predicts a first or second order phase transition from a disordered to an ordered state, depending on the number of possible internal states. The first order phase transition has previously been observed in agent based models that follow the spatial dynamics of individuals, and has been confirmed both in experiments with active matter and observations of living systems [8–10]. The character of the transition can also depend on finite system size effects [11, 12] and on subtle changes in the way the noise is incorporated [13]. Earlier network models [7, 14, 15] have focused on a discrete state space. All the aforementioned biological systems, however, have a continuous state space. Our intuition from equilibrium statistical mechanics tells us that the nature of the state space can matter a lot [16–18]. It has been demonstrated that in nonequilibrium systems, details affect the type of phase transition. Therefore, in this paper we investigate the swarming transition in a continuous state space.

In this letter we present the numerical solutions of our model accompanied with analytic expressions for the phase space boundary, that we found through stability analysis

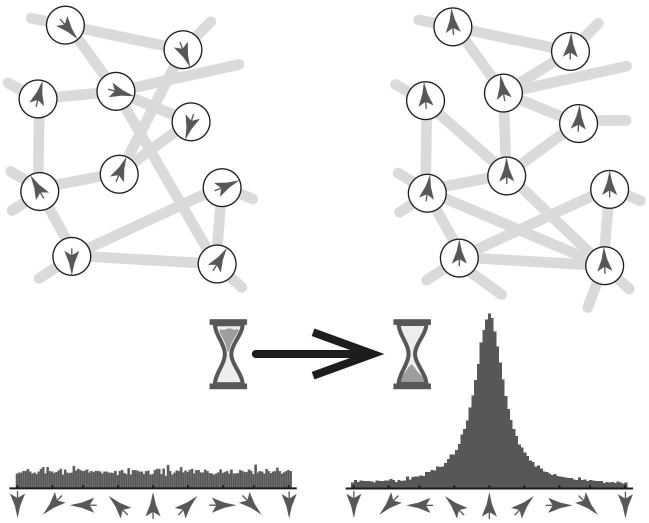

Figure 1.1: An illustration of a swarming phase transition in an adaptive network. **Top:** Each node represents an agent that can point in any direction on a plane. These agents are connected to each other through links, which represent mutual awareness. **Top left:** The agents do not have a common heading direction and the system is disordered. **Top right:** Through interactions the agents can form a coherently moving swarm and choose an overall heading direction. **Bottom:** How often a certain state occurs in the network tells us if the network is ordered or disordered. **Bottom left:** different states occur with roughly the same frequency, whereas on the **Bottom right:** one direction is the most common.

of the disordered phase. Our results clearly show that continuous state adaptive network models are able to describe spontaneous emergence of order in active systems. We find that the balance of time scales in the model determines the type of phase transition. Our model has two main time scales, the time scale of information propagation through the network, and the time scale of network reshaping. When the latter is significantly faster than the former, we get a mean field like situation, where the network is updated so fast that every node has contact with a random selection of nodes, replacing the link structure by an average connectivity number. The phase transition in this case becomes second order, while in the case that the time scales are of comparable magnitude, we find a first order transition. A meta phase transition between the first and second order regimes, similar to this one has been reported before in the context of active adaptive systems [12].

2. MODEL

We model a system of self-propelled particles with a constant speed and changing direction in two dimensions. Each particle corresponds to a node in a network, with an internal state that represents the direction of movement. We represent this internal state by an angle $\theta \in (-\pi, \pi]$. Nodes may be connected by links, indicating mutual awareness. We will refer to individuals connected by a link as neighbors.

In our system not only the states of individual agents can evolve dynamically, but

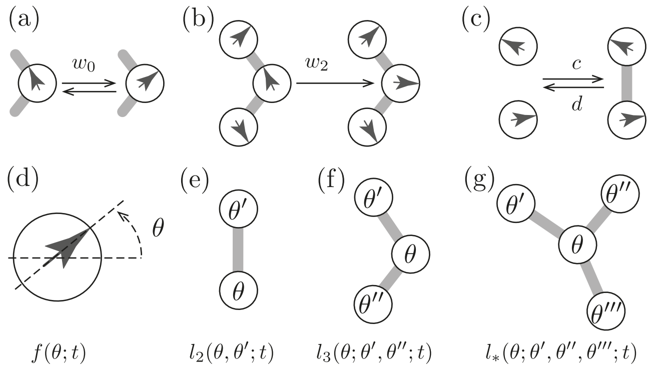

Figure 2.2: **(a-c)** Illustration of the dynamic rules of the model. The internal state of each node (circle) is represented by the direction of the arrow. These dynamics take place irrespective of any additional links that may be present, but are not drawn. **(d-g)** Visualization of interaction subgraphs. Every subgraph depicts a node in state θ that is interacting with its neighbors. $f(\theta)$, $l_2(\theta,\theta';t)$, $l_3(\theta;\theta',\theta'';t)$ and $l_*(\theta;\theta',\theta''\theta''';t)$ are the relative densities of the respective subgraphs in the network at time t.

also the relationship of the agents with their surroundings. Analogously to [7], we distinguish four types of dynamics, given by four rules:
Rule 1 Individuals spontaneously change their heading direction to another uniformly chosen direction with rate w_0.
Rule 2 Individuals adopt to the average direction of two neighbors with rate w_2.
Rule 3 Arbitrarily chosen not-neighboring individuals become neighbors with a coupling rate c.
Rule 4 Arbitrarily chosen neighbors loose mutual awareness with a decoupling rate d.

An illustration of the model is given in fig. 2.2. The first two rules are comparable to the rules in the Vicsek model [6]; rule 1 is similar to the noise that is added to each particle's updated heading direction, whereas rule 2 corresponds to the tendency of individuals to align with their neighbors within a certain radius. The radius is modeled in the network with the links, since we do not keep track of the physical position of individuals in space. The interactions described by rule 3 and rule 4 are needed since non-neighboring individuals, moving in different directions, might become aware of each other and conversely individuals which are neighboring, but head in different directions, may loose mutual awareness; therefore, their link must be created or removed respectively. Since we want to look at the behavior of groups comprised of agents of the same species, we assume that the rates are global. We define the state distribution function $f(\theta;t)$ which represents the fraction of the agents in the state $\theta \in (-\pi,\pi]$ at time t. Thus the integral of $f(\theta,t)$ over the whole domain is normalized at any time. The cartoon in fig. 1.1 is an illustration of such a state density that starts out as a random, noisy distribution over the states and evolves into a state corresponding to a collective motion in one predominant direction. Moreover, we define the link distribution function $l_2(\theta,\theta';t)$ as the density of neighboring individuals, one of which is heading in direction θ while the other one has direction θ', at time t. Note that in contrast to $f(\theta;t)$, the link density $l_2(\theta,\theta';t)$ is not normalized since the overall number of links can grow or shrink, making the normalization of l_2 time dependent. We also define higher order interaction terms like the path-like three node and star-like four node subgraph densities $l_3(\theta;\theta',\theta'';t)$ and $l_*(\theta;\theta',\theta'',\theta''';t)$ analogously. A visual representation of these subgraphs can be seen in fig. 2.2. We note that the link density functions obey certain symmetries. In particular, we have $l_2(\theta,\theta';t) = l_2(\theta',\theta;t)$ and for higher order terms all non central nodes can be exchanged in any order.

One can derive the master equations for the continuous state model using the four rules, which govern the dynamics of the state and link density distributions. These equations can be interpreted as the extension of the master equations in [7], into the continuous state set.

$$\partial_t f(\theta;t) = w_0\left(\frac{1}{2\pi} - f(\theta;t)\right) + w_2 F^{\text{int}}[l_3;\theta;t] \tag{2.1a}$$

$$\begin{aligned}\partial_t l_2(\theta,\theta';t) = {} & w_0 L^{\text{noise}}\left[l_2;\theta,\theta';t\right] + w_2 L^{\text{int}}\left[l_3,l_*;\theta,\theta';t\right] \\ & + c\, f(\theta;t)\, f(\theta';t) - d\, l_2(\theta,\theta';t).\end{aligned} \tag{2.1b}$$

The first term of eq. (2.1a) scales with w_0 and models the spontaneous direction changes of nodes. F^{int} in the second term is a functional that describes three-body interactions that cause a change of the state of an agent. The eq. (2.1b) contains four contributions. L^{noise} accounts for changes in link density, due to the random changes of states of already linked agents. L^{int} represents the changes of the link density due to the changes of states of already linked agents, but in this case the changes were induced by interactions between neighbors. The last two terms represent the change in link density due to link creation or deletion and scale with the rates c and d respectively. Both L^{noise} and L^{int} integrate to zero over the whole domain (see S.5) i.e. $\int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta'\, L = 0$, thus we can show that the overall link density, that we define as

$$k(t) = \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta'\; l_2(\theta,\theta';t), \tag{2.2}$$

always evolves to a steady state. Integrating eq. (2.1b) over θ and θ' yields a simple differential equation $\partial_t k(t) = c - dk(t)$ which has the solution

$$k(t) = \left(k(0) - \frac{c}{d}\right)\exp(-dt) + \frac{c}{d}. \tag{2.3}$$

Thus we find for the steady state link density $k_s = c/d$. During our analysis we found that the system was not qualitatively affected by changes in the value of k_s (see S.5), so from now on, unless explicitly stated otherwise, we will assume that $k_s = 1$, i.e. $c = d$. We will refer to both rates as the rate of link dynamics. The parameter space can be reduced further by rescaling time with the rate w_2. Going forward we will set $w_2 = 1$, which implies that time is measured in units of $1/w_2$ and all other rates are measured in multiples of w_2.

In order to solve eq. (2.1) we need to relate higher order subgraph density functions

l_3 and l_* to the link density function l_2

$$l_3(\theta;\theta',\theta'';t) = \frac{l_2(\theta,\theta';t)\ l_2(\theta,\theta'';t)}{f(\theta;t)}, \tag{2.4}$$

$$l_*(\theta;\theta',\theta'',\theta''';t) = \frac{l_2(\theta,\theta';t)\ l_2(\theta,\theta'';t)\ l_2(\theta,\theta''';t)}{f(\theta;t)^2}. \tag{2.5}$$

We call the resulting model the moment closure approximation (MCA) model. The MCA model can be solved directly, and we can obtain a phase space, with ordered and disordered phases. In the case of a high rate of link dynamics, i.e. $d \to \infty$, the time evolution of the link density in eq. (2.1b) leads to the steady state of

$$l_2(\theta,\theta';t) = \frac{c}{d} f(\theta;t) f(\theta';t). \tag{2.6}$$

Equation (2.6) shows that for large sampling rates, the probability of two states being linked is proportional to the relative abundances of those states. In this case eq. (2.1a) becomes independent of the link density and reduces to a single differential equation for $f(\theta;t)$. We recognize this as the mean field approximation of the full model. We call the steady state solution of the mean field model $f_s(\theta)$, which is disordered if all states occur with same frequency i.e. $f_s(\theta) = \frac{1}{2\pi}$. We call the solution ordered if one state is more abundant than all others, see fig. 1.1. To properly quantify this notion of order we introduce an order parameter a_1, which is the amplitude of the first symmetric Fourier mode in the series expansion of $f_s(\theta)$, given by

$$f_s(\theta) = \frac{1}{2\pi}\left[1 + 2\sum_{n\geq 1} a_n \cos(n(\theta - \theta_s))\right]. \tag{2.7}$$

In eq. (2.7) the a_n are the mode amplitudes and θ_s is the direction preferred in the steady state. Since $\cos(\theta - \theta_s)$ is a function with a single peak around θ_s, the contribution of the first mode to the series expansion of $f_s(\theta)$ gives us a quantitative understanding of how ordered the system is. The atypical normalization of the Fourier series is chosen to have $0 \leq a_1 \leq 1$. Thus $a_1 = 0$ is associated with the disordered and $a_1 = 1$ with the ordered state.

3. RESULTS

We first solved the model in the mean field approximation, since in this case we only have an equation for $f(\theta;t)$ to solve, with w_0 as its only control parameter. Using the Fourier series expansion in eq. (2.7), we can reduce the differential equation for the mean field system to a set of algebraic equations for the Fourier modes in steady state

$$\partial_t a_n = \left(\frac{\sin(\frac{n\pi}{2})}{n\pi} - \frac{1}{4} - w_0\right) a_n + \sum_{p,q>0}^{\infty} \frac{\gamma_{npq}}{(2\pi)^2} a_p a_q, \tag{3.8}$$

where $\gamma_{npq} =$ $\int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\phi \cos\left[p\left(\theta + \frac{\phi}{2}\right)\right] \cos\left[q\left(\theta - \frac{\phi}{2}\right)\right] \cos(n\theta)$. From eq. (3.8) we can extract the exact value of the critical noise

$$w_0^{\mathrm{cr}} = \frac{1}{\pi} - \frac{1}{4}, \tag{3.9}$$

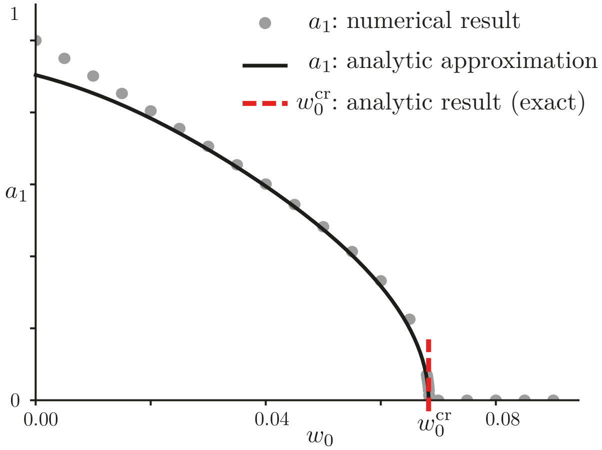

Figure 2.3: Bifurcation diagram of the system in the mean field approximation. The order parameter a_1 is plotted versus control parameter w_0. The dots represent numerical solutions, whereas the curve represents the best analytic approximation, after closing the system of Fourier coefficients at the fourth order.

and the scaling behavior of the order parameter close to the critical point $a_1 \propto (w_0 - w_0^{\rm cr})^{1/2}$. Our analytic approximation of a_1 is obtained by closing eq. (3.8). This procedure provides a generic n-th order polynomial for a_1 if closed at that order. Thus the best possible analytic solution is a root of a fourth order polynomial, which is plotted in fig. 2.3 as a solid black line. This solution gets progressively worse away from the critical point, but close to it our numerical and analytic results are in great agreement. Furthermore, we see that this phase transition is unambiguously second order and has a critical exponent of 1/2, which was obtained analytically and is exact (see S.5).

For the MCA model we have an additional control parameter d, which sets the time scale of the link dynamics. Therefore, we get a critical line

$$d_{\rm cr} = -\frac{9}{4} - w_0 + \frac{\pi - 2}{2\pi(w_0 - w_0^{\rm cr})} \tag{3.10}$$

which can be seen in fig. 3.4, as the red dash-dotted curve. The vertical red dashed line is drawn at the value $w_0^{\rm cr}$, this value is reached when d goes to infinity. This critical line is obtained by linear stability analysis of the disordered state and is exact, just like the mean field model. The disordered state becomes unstable left of the critical line. Our numerical solutions were obtained by solving the system from an ordered initial condition. Thus the numerical solutions approach the phase transition from the ordered side. The existence of numerical results with nonzero order parameter, on the right of the line indicates hysteresis. Unfortunately, the methods that allowed us to obtain analytic values of the order parameter near criticality in the mean field model do not work here. However the numerical results in fig. 3.4, clearly show a discontinuous change in the value of the order parameter at the critical line. Together with the bistable region, this is a clear sign of a first order phase transition. To exclude artifacts related to slow convergence to steady state, we repeated the runs for 3×10^5 and 9×10^5 time steps measured in units of $1/w_2$. The results remained unchanged (see S.5).

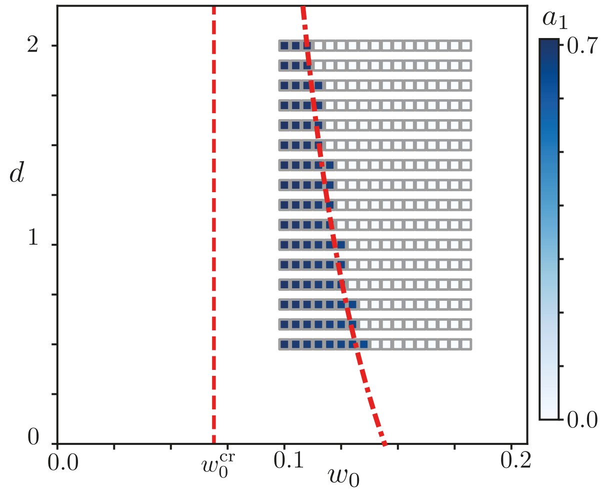

Figure 3.4: Phase diagram for the MCA system in which both parameters w_0 and d are varied. Squares represent numerical solutions. The curve is analytically determined and converges to w_0^{cr} for $d \to \infty$. Left from this curve the disordered state is linearly unstable which leads to $a_1 > 0$, such that the final system state is ordered. Right from this curve the disordered state is linearly stable, but the system can remain ordered (blue squares) if the initial condition of the system was ordered.

4. Discussion

The continuous state adaptive network model can describe the emergence of swarming. The key parameter in this transition is the relative strength of the noise compared to the agent interactions. The critical noise value, at which the swarming transition occurs, can be changed by tuning the timescale of the network dynamics. However, if this timescale, given by $1/d$, becomes much smaller than the timescale of the node dynamics, set by $1/w_2$, two interesting things happen. First, the region of stability shrinks and the swarming phase transition changes from a first, to a second order transition. The shrinking of the stable region, for fast link dynamics, is surprising because, the limit of this regime is the mean field approximation. Using the intuition from equilibrium statistical mechanics, one would expect the critical noise to be higher in the case of the mean field model, since we expect mean field models to overestimate the tendency to order. However, we get the opposite result. The critical noise is maximal for $d = 0$; most likely due to the decoupling of the link dynamics from the state dynamics. This decoupling can slow down the propagation of orientational information through the system. The second striking effect seen in our system is the meta phase transition. While changing the rate of the link dynamics, the swarming transition becomes second order. Further investigation is necessary, but it seems that this effect can be explained by the observation that the type of noise affects the type of phase transition. According to Pimentel et al. [19] intrinsic noise in agent states leads to a second order transition, while extrinsic noise, caused by imperfect sampling of neighbors, leads to a first order phase transition. This fits well with our findings, since the mean field system only has intrinsic noise caused by spontaneous direction change, while the MCA system introduces an additional noise source, through changing topology. We do not know if this meta transition happens at a finite value of d or strictly at $d \to \infty$, but regardless of the point where the meta transition happens, it constitutes a significant change in the behavior of the system. Biological systems that are described by this model, would therefore not always need nucleation points, such as small co-moving groups, for global swarms to emerge. If such systems could facilitate a quickly changing neighborhood topology, they would be able to smoothly transition into a swarm state.

The new model that we have put forward in this letter demonstrates that the timescale separation between the topology- and state-dynamics of the network has a strong impact on the nature of the swarming transition.

ACKNOWLEDGMENTS

We thank Felix Frey for helpful discussions. G.D. was supported by the "BaSyC – Building a Synthetic Cell" Gravitation grant (024.003.019) of the Netherlands Ministry of Education, Culture and Science (OCW) and the Netherlands Organisation for Scientific Research (NWO).

C.K. and G. D. contributed equally to this work.

S.5. SUPPLEMENTAL MATERIAL

Here we want to present the governing equations for the time evolution of the state density $f(\theta;t)$ and the link density $l_2(\theta,\theta';t)$, i.e. eq. (1). These equations describe the dynamics according to the four rules, as given in the main text. Each rule has an associated type of dynamics and thus an associated term in the equations of both densities, which we indicate with a corresponding superscript. E.g. $\partial_t f^{(1)}(\theta;t)$ accounts for the effects of rule 1 on the time evolution of $f(\theta;t)$. Then the general form of our equations for state and link densities reads as follows:

$$\partial_t f(\theta;t) = \sum_{i=1}^{4} \partial_t f^{(i)}(\theta;t) \tag{S.5.11a}$$

$$\partial_t l_2(\theta,\theta';t) = \sum_{i=1}^{4} \partial_t l_2^{(i)}(\theta,\theta';t) \tag{S.5.11b}$$

DYNAMICS ASSOCIATED WITH RULE 1

The effect of the dynamics of type 1 on the state density function $f(\theta;t)$ is captured by the following equation

$$\partial_t f^{(1)}(\theta;t) = w_0 \int_{-\pi}^{\pi} \mathrm{d}\theta' \frac{1}{2\pi} \left[f(\theta';t) - f(\theta;t) \right] \tag{S.5.12}$$

where $1/(2\pi)$ is the probability measure of a state change, which is a uniform distribution on the state space, since rule 1 prescribes random state changes that do not depend on the initial and final state. Using the fact that $f(\theta;t)$ is normalized we get

$$\partial_t f^{(1)}(\theta;t) = w_0 \left(\frac{1}{2\pi} - f(\theta;t) \right). \tag{S.5.13}$$

The change in link density due to interactions of type 1 is described by

$$\partial_t l_2^{(1)}(\theta,\theta';t) = w_0 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \frac{1}{2\pi}\left[l_2(\theta,\theta'';t) + l_2(\theta',\theta'';t) - 2l_2(\theta,\theta';t)\right], \tag{S.5.14}$$

where the first two terms account for the creation of a $\theta - \theta'$ pair, through the change of a single state. The last term accounts for the destruction of already existing $\theta - \theta'$ links, through the change of one of the states, which can happen two different ways, and hence the factor of two. Performing the integral on the last term gives

$$\partial_t l_2^{(1)}(\theta,\theta';t) = \frac{w_0}{2\pi}\int_{-\pi}^{\pi} \mathrm{d}\theta'' \left[l_2(\theta,\theta'';t) + l_2(\theta',\theta'';t)\right] - 2w_0 l_2(\theta,\theta';t). \tag{S.5.15}$$

DYNAMICS ASSOCIATED WITH RULE 2

The next step will be to obtain similar equations for the second type of dynamics. For the state density $f(\theta;t)$, these interactions can be described as follows,

$$\partial_t f^{(2)}(\theta;t) = w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_0^{\frac{\pi}{2}} \mathrm{d}\xi \left[l_3(\theta';\theta-\xi,\theta+\xi;t) - l_3(\theta;\theta'-\xi,\theta'+\xi;t)\right]. \tag{S.5.16}$$

First term represents nodes in an arbitrary state θ', in between two nodes that average to the state θ. This way an extra θ node is created due to three body interactions at rate w_2. Second term corresponds to removal of θ nodes due to these interactions. We need to be careful that we take a proper circle average of the directions. This can be implemented in the equations by integrating over ξ from 0 to $\pi/2$. The next differential equation captures the change in the link density function $\partial_t l_2(\theta,\theta';t)$ due to three body interactions

$$\begin{aligned}
\partial_t l_2^{(2)}(\theta,\theta';t) = \\
& w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_0^{\pi/2} \mathrm{d}\xi \; l_*(\theta'';\theta,\theta'+\xi,\theta'-\xi;t) && \text{(S.5.17a)}\\
& + w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_0^{\pi/2} \mathrm{d}\xi \; l_*(\theta'';\theta',\theta+\xi,\theta-\xi;t) && \text{(S.5.17b)}\\
& - w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_0^{\pi/2} \mathrm{d}\xi \; l_*(\theta;\theta',\theta''+\xi,\theta''-\xi;t) && \text{(S.5.17c)}\\
& - w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_0^{\pi/2} \mathrm{d}\xi \; l_*(\theta';\theta,\theta''+\xi,\theta''-\xi;t) && \text{(S.5.17d)}\\
& + w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \; l_3(\theta'';\theta,-\theta+2\theta';t) && \text{(S.5.17e)}\\
& + w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \; l_3(\theta'';\theta',-\theta'+2\theta;t) && \text{(S.5.17f)}\\
& - w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \; l_3(\theta;\theta',\theta'';t) && \text{(S.5.17g)}\\
& - w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \; l_3(\theta';\theta,\theta'';t). && \text{(S.5.17h)}
\end{aligned}$$

The eq. (S.5.17a) and eq. (S.5.17b) account for four-point subgraphs where one of the nodes is already pointing in the right direction θ or θ' and the center one can average with the remaining two neighbors to change into θ' or θ respectively and thus create a θ – θ' link. The terms of eq. (S.5.17c) and eq. (S.5.17d), describe situations where a $\theta - \theta'$ link already exists in a star like subgraph and gets destroyed. Again, we integrate over all ξ for which our averaging operation is defined properly. Three-point subgraphs with the potential to form an θ–θ' link are taken into account in eq. (S.5.17e) and eq. (S.5.17f). In these cases θ'' changes to θ or θ', while the one of the neighbor nodes is already in θ' or θ configuration respectively. Finally, eq. (S.5.17g) and eq. (S.5.17h) account the loss term on the three-point subgraphs level.

DYNAMICS ASSOCIATED WITH RULE 3 AND RULE 4

Since the dynamics described in rule 3 and rule 4 only affect links,

$$\partial_t f^{(3)}(\theta;t) = \partial_t f^{(4)}(\theta;t) = 0. \tag{S.5.18}$$

Link creation dynamics, between two nodes of type θ and θ' is given by the density of already existing nodes and the link creation rate c

$$\partial_t l_2^{(3)}(\theta,\theta';t) = c f(\theta;t) f(\theta';t). \tag{S.5.19}$$

Link deletion dynamics is given by the number of already existing links and the deletion rate d

$$\partial_t l_2^{(4)}(\theta,\theta';t) = -d l_2(\theta,\theta';t). \tag{S.5.20}$$

FULL EQUATIONS

Putting everything together in eq. (S.5.11) we arrive at

$$\partial_t f(\theta;t) = w_0\left(\frac{1}{2\pi} - f(\theta;t)\right) + w_2\int_{-\pi}^{\pi} \mathrm{d}\theta' \int_0^{\frac{\pi}{2}} \mathrm{d}\xi \left[l_3(\theta';\theta-\xi,\theta+\xi;t) - l_3(\theta;\theta'-\xi,\theta'+\xi;t)\right], \tag{S.5.21a}$$

for the state density distribution and we can identify the definition of the functional $F^{\text{int}}[l_3;\theta;t]$ from the second term as

$$F^{\text{int}}[l_3;\theta;t] = \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_0^{\frac{\pi}{2}} \mathrm{d}\xi \left[l_3(\theta';\theta-\xi,\theta+\xi;t) - l_3(\theta;\theta'-\xi,\theta'+\xi;t)\right]. \tag{S.5.22}$$

For the link density distribution we get

$$
\begin{align}
\partial_t l_2^{(2)}(\theta,\theta';t) = & \\
& \frac{w_0}{2\pi}\int_{-\pi}^{\pi} \mathrm{d}\theta'' \big[l_2(\theta,\theta'';t) + l_2(\theta',\theta'';t)\big] - 2w_0 l_2(\theta,\theta';t) \tag{S.5.23a}\\
& + w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \Big\{ l_3(\theta'';\theta,-\theta+2\theta';t) + l_3(\theta'';\theta',-\theta'+2\theta;t) - l_3(\theta;\theta',\theta'';t) \\
& \qquad - l_3(\theta';\theta,\theta'';t) \\
& \qquad + \int_0^{\pi/2} \mathrm{d}\xi \big[l_*(\theta'';\theta,\theta'+\xi,\theta'-\xi;t) + l_*(\theta'';\theta',\theta+\xi,\theta-\xi;t) \\
& \qquad\qquad - l_*(\theta;\theta',\theta''+\xi,\theta''-\xi;t) - l_*(\theta';\theta,\theta''+\xi,\theta''-\xi;t)\big]\Big\} \tag{S.5.23b}\\
& + c f(\theta;t) f(\theta';t) - d l_2(\theta,\theta';t). \tag{S.5.23c}
\end{align}
$$

The functional $w_0 L^{\text{noise}}$ is given by eq. (S.5.23a) and $w_2 L^{\text{int}}$ is given by eq. (S.5.23b). As mentioned in the main text L^{noise} and L^{int} integrate to zero. We demonstrate this later in section S.5, but it should already be conceptually clear, since these two functionals account for conservative dynamics that come from the state change. No links are created or destroyed during this type of changes, and thus we do not expect this to cause any global changes in the overall link density.

Properties of L^{noise} and L^{int}

We first show that L^{noise} integrates to zero. This is done by straight forward integration,

$$
\begin{aligned}
\int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' L^{\text{noise}}[l_2;\theta,\theta';t] = & \\
\frac{1}{2\pi}\int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta'' \big[l_2(\theta,\theta'';t) + l_2(\theta',\theta'';t)\big] - 2\int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' l_2(\theta,\theta';t) & \\
= k(t) + k(t) - 2k(t) = 0, &
\end{aligned}
$$

where we used the definition of $k(t)$, as provided in eq. (2) of the main text. The proof that L^{int} integrates to zero is also a straight forward integration. Writing everything out

$$\int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' L^{\text{int}}[l_3, l_*; \theta, \theta'; t] =$$

$$\int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_{0}^{\pi/2} \mathrm{d}\xi\, l_*(\theta''; \theta, \theta' + \xi, \theta' - \xi; t) \tag{S.5.24a}$$

$$+ \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_{0}^{\pi/2} \mathrm{d}\xi\, l_*(\theta''; \theta', \theta + \xi, \theta - \xi; t) \tag{S.5.24b}$$

$$- \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_{0}^{\pi/2} \mathrm{d}\xi\, l_*(\theta; \theta', \theta'' + \xi, \theta'' - \xi; t) \tag{S.5.24c}$$

$$- \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_{0}^{\pi/2} \mathrm{d}\xi\, l_*(\theta'; \theta, \theta'' + \xi, \theta'' - \xi; t) \tag{S.5.24d}$$

$$+ \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta''\, l_3(\theta''; \theta, -\theta + 2\theta'; t) \tag{S.5.24e}$$

$$+ \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta''\, l_3(\theta''; \theta', -\theta' + 2\theta; t) \tag{S.5.24f}$$

$$- \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta''\, l_3(\theta; \theta', \theta''; t) \tag{S.5.24g}$$

$$- \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta''\, l_3(\theta'; \theta, \theta''; t), \tag{S.5.24h}$$

one can see that one gets pairs of equations eq. (S.5.24a), eq. (S.5.24c); eq. (S.5.24b), eq. (S.5.24d); eq. (S.5.24e), eq. (S.5.24g) and eq. (S.5.24f), eq. (S.5.24h) that only differ by relabeling of their integration variables but have opposite signs, so everything adds up to zero.

S.5.1. MOMENT CLOSURE APPROXIMATION

In this section we derive eq. (3) and eq. (4) from the main text. Moment closure effectively means that the probability of finding higher order structures decomposes into the product of the probabilities of finding lower order structures. For triplets of $\theta' - \theta - \theta''$ where θ is the middle node, like depicted in figure (2f) of the main text, this means

$$p(\theta; \theta', \theta''; t) = p(\theta', \theta'' | \theta; t)\, p(\theta; t) \tag{S.5.25a}$$

$$= p(\theta' | \theta'', \theta; t)\, p(\theta'' | \theta; t)\, p(\theta; t) \tag{S.5.25b}$$

$$= p(\theta' | \theta; t)\, p(\theta'' | \theta; t)\, p(\theta; t), \tag{S.5.25c}$$

where $p(\theta; \theta', \theta''; t)$ is the full probability of finding the triplet $\theta' - \theta - \theta''$, at time t. Note that the semicolon after the first argument does not denote a conditional probability, but it simply separates the central node from the peripheral nodes. $p(\theta', \theta'' | \theta)$ is the conditional probability that the peripheral nodes have the angles θ' and θ'', given that the central node has the angle θ. $p(\theta;t)$ is the prior probability of finding a node, which has the angle θ, i.e. $p(\theta;t) = f(\theta;t)$. Going from eq. (S.5.25b) to eq. (S.5.25c) we made the closure assumption $p(\theta'|\theta'',\theta;t) = p(\theta'|\theta;t)$, saying that the probability of finding one peripheral node does not depend on the state of the other peripheral node. Now we simply relate the probability of finding a state, to the density of states

$$\frac{l_3(\theta;\theta',\theta'';t)}{\|l_3\|} = p(\theta;\theta',\theta'';t) \tag{S.5.26}$$

$$\frac{l_2(\theta;\theta';t)}{k} = p(\theta;\theta';t) = p(\theta'|\theta;t)p(\theta;t), \tag{S.5.27}$$

$$\tag{S.5.28}$$

where k is the norm of the link density as defined in eq. (2) of the main text and $||l_3||$ is the norm of the triplet density, defined as

$$\|l_3\| = \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta''\, l_3(\theta;\theta',\theta'';t). \tag{S.5.29}$$

Putting everything together, we get

$$l_3(\theta;\theta',\theta'';t) = \frac{\|l_3\|}{k^2} \frac{l_2(\theta,\theta';t)\,l_2(\theta,\theta'';t)}{f(\theta;t)}. \tag{S.5.30}$$

Repeating the same steps for the star like, four point subgraphs we get,

$$\begin{aligned}
\frac{l_*(\theta;\theta',\theta'',\theta''';t)}{\|l_*\|} &= p(\theta;\theta',\theta'',\theta''';t) \\
&= p(\theta',\theta'',\theta'''|\theta;t)p(\theta;t) \\
&= p(\theta'|\theta;t)p(\theta''|\theta;t)p(\theta'''|\theta;t)p(\theta;t) \\
&= \frac{p(\theta',\theta;t)}{p(\theta;t)} \frac{p(\theta'',\theta;t)}{p(\theta;t)} \frac{p(\theta''',\theta;t)}{p(\theta;t)} p(\theta;t) \\
&= \frac{l_2(\theta,\theta';t)\,l_2(\theta,\theta'';t)\,l_2(\theta,\theta''';t)}{f^2(\theta;t)k^3},
\end{aligned} \tag{S.5.31}$$

where $\|l_*\|$ is the norm of the density of star like subgraphs and is defined as

$$\|l_*\| = \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_{-\pi}^{\pi} \mathrm{d}\theta'' \int_{-\pi}^{\pi} \mathrm{d}\theta'''\; l_*(\theta;\theta',\theta'',\theta''';t). \tag{S.5.32}$$

Rearranging terms in S.5.31 gets us to,

$$l_*(\theta;\theta',\theta'',\theta''';t) = \frac{\|l_*\|}{k^3} \frac{l_2(\theta,\theta';t)\,l_2(\theta,\theta'';t)\,l_2(\theta,\theta''';t)}{f^2(\theta;t)}. \tag{S.5.33}$$

The only differences between the here derived eq. (S.5.30) and eq. (S.5.33) and the closure relations given in eq. (4) and eq. (5) of the main text, are the prefactors of $\|l_3\|/k^2$ and $\|l_*\|/k^3$. These prefactors represent the topology of the underlying graphs, on which our dynamics take place. k is the link density, i.e. the number of links in the network relative to the number of nodes, and since each link is shared by two nodes, we can estimate the average node degree $n = 2k$. $\|l_3\|$ is the number of triplets a node is the center of and $\|l_*\|$ is the number of quadruplets a node is the center of. In a fully connected graph with the node degree n each node will be the center of $\binom{n}{2}$ triplets and $\binom{n}{3}$ quadruplets. This means that the upper bounds on our prefactors are,

$$\frac{\|l_3\|}{k^2} \leq \frac{n(n-1)/2}{(n/2)^2} = 2+\mathcal{O}(n^{-1}) \tag{S.5.34}$$

$$\frac{\|l_*\|}{k^3} \leq \frac{n(n-1)(n-2)/6}{(n/2)^3} = \frac{4}{3}+\mathcal{O}(n^{-1}). \tag{S.5.35}$$

The actual values will be dependent on the exact realization of the random network topology, and one needs to make an assumption at this point. For our model we set both of these ratios to one,

$$\frac{\|l_3\|}{k^2} = 1 \tag{S.5.36}$$

$$\frac{\|l_*\|}{k^3} = 1. \tag{S.5.37}$$

Since these values are below the fully connected network bounds, we interpret them as equivalent to choosing a sparse network topology.

Using the moment closure relations as given in eq. (4) and eq. (5) of the main text, we get a closed system of differential equations for the state and link densities,

$$
\begin{aligned}
\partial_t f(\theta;t) =& w_0 \left(\frac{1}{2\pi} - f(\theta;t) \right) \\
&+ w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta' \int_0^{\frac{\pi}{2}} \mathrm{d}\xi \left[\frac{l_2(\theta',\theta-\xi;t)\, l_2(\theta',\theta+\xi;t)}{f(\theta';t)} - \frac{l_2(\theta,\theta'-\xi;t)\, l_2(\theta,\theta'+\xi;t)}{f(\theta;t)} \right]
\end{aligned}
\tag{S.5.38a}
$$

$$
\begin{aligned}
\partial_t l_2^{(2)}(\theta,\theta';t) =& \\
& \frac{w_0}{2\pi} \int_{-\pi}^{\pi} \mathrm{d}\theta'' \left[l_2(\theta,\theta'';t) + l_2(\theta',\theta'';t) \right] - 2 w_0 l_2(\theta,\theta';t) \\
& + w_2 \int_{-\pi}^{\pi} \mathrm{d}\theta'' \Bigg\{ \frac{l_2(\theta'',\theta;t)\, l_2(\theta'',-\theta+2\theta';t)}{f(\theta'';t)} + \frac{l_2(\theta'',\theta';t)\, l_2(\theta'',-\theta'+2\theta;t)}{f(\theta'';t)} \\
& \qquad - \frac{l_2(\theta,\theta';t)\, l_2(\theta,\theta'';t)}{f(\theta;t)} - \frac{l_2(\theta',\theta;t)\, l_2(\theta',\theta'';t)}{f(\theta';t)} \\
& \qquad + \int_0^{\pi/2} \mathrm{d}\xi \Big[\frac{l_2(\theta'',\theta;t)\, l_2(\theta'',\theta'+\xi;t)\, l_2(\theta'',\theta'-\xi;t)}{f^2(\theta'';t)} \\
& \qquad\qquad + \frac{l_2(\theta'',\theta';t)\, l_2(\theta'',\theta+\xi;t)\, l_2(\theta'',\theta-\xi;t)}{f^2(\theta'';t)} \\
& \qquad\qquad - \frac{l_2(\theta,\theta';t)\, l_2(\theta,\theta''+\xi;t)\, l_2(\theta,\theta''-\xi;t)}{f^2(\theta;t)} \\
& \qquad\qquad - \frac{l_2(\theta',\theta;t)\, l_2(\theta',\theta''+\xi;t)\, l_2(\theta',\theta''-\xi;t)}{f^2(\theta';t)} \Big] \Bigg\} \\
& + c f(\theta;t) f(\theta';t) - d\, l_2(\theta,\theta';t).
\end{aligned}
\tag{S.5.38b}
$$

We used eq. (S.5.38) to obtain the numeric results for the MCA model, since it has good numerical stability. But it is possible to simplify the equations further by exploiting the symmetry of l_2 in its arguments and factor many terms. Which we will make use of in section S.5.3 and section S.5.4 to find the phase transition analytically.

S.5.2. MEAN FIELD APPROXIMATION

Since the closed system from eq. (S.5.38) is still difficult to analyze, both numerically and analytically, one can simplify the model further and close it on the level of the state density $f(\theta;t)$. This is a good approximation for diverging rates of link dynamics i.e. $d \to \infty$ and $c \to \infty$, while $c/d = k_s = \text{const}$. Then the last two terms in eq. (S.5.38b) dominate the dynamics of l_2 and lead to the steady state,

$$
l_2(\theta,\theta';t) = \frac{c}{d} f(\theta;t) f(\theta';t) = k_s f(\theta;t) f(\theta';t), \tag{S.5.39}
$$

where l_2 is directly determined by the state density f, which can still change through it's own dynamics. Substituting eq. (S.5.39) into eq. (S.5.38a) we get,

$$\begin{aligned}\partial_t f(\theta;t) &= w_0\left(\frac{1}{2\pi}-f(\theta;t)\right)\\ &+ w_2k_s^2\left\{\int_0^{\frac{\pi}{2}}\mathrm{d}\xi\, f(\theta-\xi;t)f(\theta+\xi;t)-f(\theta;t)\int_{-\pi}^{\pi}\mathrm{d}\theta'\int_0^{\frac{\pi}{2}}\mathrm{d}\xi\, f(\theta'-\xi;t)f(\theta'+\xi;t)\right\}.\end{aligned} \tag{S.5.40}$$

We used eq. (S.5.40) for numerical solutions, because it is numerically stable. But if we recognize that

$$\int_{-\pi}^{\pi}\mathrm{d}\theta'\int_0^{\frac{\pi}{2}}\mathrm{d}\xi\, f(\theta'-\xi;t)f(\theta'+\xi;t)=\frac{1}{4}, \tag{S.5.41}$$

then, the mean field equation simplifies to

$$\partial_t f(\theta;t)=\frac{w_0}{2\pi}-\left(w_0+\frac{w_2k_s^2}{4}\right)f(\theta;t)+\frac{w_2k_s^2}{4}\int_{-\pi}^{\pi}\mathrm{d}\xi\, f\left(\theta-\frac{\xi}{2};t\right)f\left(\theta+\frac{\xi}{2};t\right), \tag{S.5.42}$$

Where we used eq. (S.5.63) to change the integration boundary of the ξ integral and eq. (S.5.41) is a special case of eq. (S.5.65).

S.5.3. SOLUTION OF THE MEAN FIELD MODEL

CRITICAL NOISE

To obtain the value of the critical noise, w_0^{cr} we perform the linear stability analysis of eq. (S.5.42), by expanding the state density around the uniform state, in Fourier modes. In our simulations we had observed that the steady state distribution is always symmetric around the mean, thus we used the expansion in symmetric modes, as shown in eq. (7) of the main text. Here we also set $\theta_s=0$, without loss of generality, and start from

$$f(\theta;t)=\frac{1}{2\pi}\left[1+2\sum_{n=1}^{\infty}a_n\cos(n\theta)\right]. \tag{S.5.43}$$

Plugging eq. (S.5.43) in eq. (S.5.42) we get

$$\begin{aligned}2\sum_{n=1}^{\infty}\partial_t a_n\cos(n\theta) &= -\frac{w_2k_s^2}{4}-\left(w_0+\frac{w_2k_s^2}{4}\right)2\sum_{n=1}^{\infty}a_n\cos(n\theta)\\ &+\frac{w_2k_s^2}{4}\frac{1}{2\pi}\int_{-\pi}^{\pi}\mathrm{d}\xi\left[1+2\sum_{n=1}^{\infty}a_n\cos\left(n\left(\theta+\frac{\xi}{2}\right)\right)\right]\left[1+2\sum_{m=1}^{\infty}a_m\cos\left(m\left(\theta-\frac{\xi}{2}\right)\right)\right]\\ &=-\left(w_0+\frac{w_2k_s^2}{4}\right)2\sum_{n=1}^{\infty}a_n\cos(n\theta)+\frac{w_2k_s^2}{4}\frac{1}{2\pi}2\sum_{n=1}^{\infty}a_n\,2\int_{-\pi}^{\pi}\mathrm{d}\xi\cos\left(n\left(\theta+\frac{\xi}{2}\right)\right)\\ &+\frac{w_2k_s^2}{4}\frac{4}{2\pi}\sum_{m,n=1}^{\infty}a_na_m\int_{-\pi}^{\pi}\mathrm{d}\xi\cos\left(n\left(\theta+\frac{\xi}{2}\right)\right)\cos\left(m\left(\theta-\frac{\xi}{2}\right)\right),\end{aligned}$$

canceling the factor of two on both sides and using,

$$\int_{-\pi}^{\pi} \mathrm{d}\xi \cos\left(n\left(\theta+\frac{\xi}{2}\right)\right)=\frac{4\sin\left(\frac{n\pi}{2}\right)}{n}\cos(n\theta),$$

we get,

$$\sum_{n=1}^{\infty}\partial_t a_n\cos(n\theta)=-\left(w_0+\frac{w_2k_s^2}{4}\right)\sum_{n=1}^{\infty}a_n\cos(n\theta)+\frac{w_2k_s^2}{4}\sum_{n=1}^{\infty}a_n\frac{4\sin\left(\frac{n\pi}{2}\right)}{n\pi}\cos(n\theta)$$
$$+\frac{w_2k_s^2}{4\pi}\sum_{m,n=1}^{\infty}a_na_m\int_{-\pi}^{\pi}\mathrm{d}\xi\cos\left(n\left(\theta+\frac{\xi}{2}\right)\right)\cos\left(m\left(\theta-\frac{\xi}{2}\right)\right).$$

Finally to get rid of the sums we multiply both sides with $\cos(p\theta)$ and integrate over θ. This allows us to use

$$\int_{-\pi}^{\pi}\mathrm{d}\theta\cos(n\theta)\cos(p\theta)=\pi\delta_{np}, \tag{S.5.44}$$

to get the nonlinear amplitude equation

$$\partial_t a_n=\left(w_2k_s^2\left(\frac{\sin\left(\frac{n\pi}{2}\right)}{n\pi}-\frac{1}{4}\right)-w_0\right)a_n+\frac{w_2k_s^2}{(2\pi)^2}\sum_{p,q=1}^{\infty}a_pa_q\gamma_{npq}, \tag{S.5.45}$$

where we relabeled the indices p, n, m to n, p, q, and introduced

$$\gamma_{npq}=\int_{-\pi}^{\pi}\mathrm{d}\theta\int_{-\pi}^{\pi}\mathrm{d}\xi\cos\left(p\left(\theta+\frac{\xi}{2}\right)\right)\cos\left(q\left(\theta-\frac{\xi}{2}\right)\right).$$

To determine the linear stability of the disordered phase we only need to look at the linear part of eq. (S.5.45),

$$\partial_t a_n=(w_0^{\mathrm{cr}}(n)-w_0)a_n. \tag{S.5.46}$$

With $w_0^{\mathrm{cr}}(n)=w_2k_s^2\left(\frac{\sin\left(\frac{n\pi}{2}\right)}{n\pi}-\frac{1}{4}\right)$. The solution of eq. (S.5.46) is

$$a_n(t)=a_n(0)\exp\left[(w_0^{\mathrm{cr}}(n)-w_0)t\right]. \tag{S.5.47}$$

If $(w_0^{\mathrm{cr}}(n)-w_0)>0$ for any integer $n\geq 1$ than the amplitude of that mode will start growing and will destabilize the disordered state. For this to happen $w_0^{\mathrm{cr}}(n)$ has to be positive in the first place, which is only the case for $n=1$. Then the condition for the critical noise becomes

$$w_2k_s^2\left(\frac{1}{\pi}-\frac{1}{4}\right)-w_0>0, \tag{S.5.48}$$

and we get the general result for the critical noise

$$w_0^{\mathrm{cr}}=w_2k_s^2\left(\frac{1}{\pi}-\frac{1}{4}\right). \tag{S.5.49}$$

Approximate solution of the nonlinear equation

We start by evaluating integral for γ_{npq}, which leads to

$$\gamma_{npq} = \frac{2\pi}{p^2 - q^2}\left(\zeta_{npq} - \zeta_{nqp}\right), \tag{S.5.50}$$

where

$$\zeta_{npq} = \cos\left(\frac{p\pi}{2}\right)\sin\left(\frac{q\pi}{2}\right)\Big[(p-q)\left(\delta_{p,q-n}+\delta_{p,q+n}\right) - (p+q)\left(\delta_{p,-q+n}+\delta_{p,-q-n}\right)\Big] \tag{S.5.51a}$$

$$= \frac{1}{2}\left[\sin\left((p+q)\frac{\pi}{2}\right) - \sin\left((p-q)\frac{\pi}{2}\right)\right]\Big[(p-q)\left(\delta_{p,q-n}+\delta_{p,q+n}\right) - (p+q)\delta_{p,-q+n}\Big]. \tag{S.5.51b}$$

In eq. (S.5.51b) we dropped $\delta_{p,-q-n}$ since this would lead to a_{-p-n} which are zero because all indices must be positive. Plugging the definitions of γ_{npq} and ζ_{npq} in the nonlinear part of eq. (S.5.45) we get

$$\frac{1}{(2\pi)^2}\sum_{p,q} a_p a_q \gamma_{npq} = \frac{1}{(2\pi)^2}\sum_{p,q} a_p a_q \frac{2\pi}{p^2-q^2}\left(\zeta_{npq} - \zeta_{nqp}\right). \tag{S.5.52a}$$

$$= \frac{1}{2\pi}\sum_{p,q=1}^{\infty} a_p a_q \frac{2\zeta_{npq}}{p^2-q^2} \tag{S.5.52b}$$

$$= \frac{1}{2\pi}\sum_{p,q=1}^{\infty} a_p a_q \left[\sin\left((p+q)\frac{\pi}{2}\right) - \sin\left((p-q)\frac{\pi}{2}\right)\right]\left[\frac{\delta_{p,q-n}+\delta_{p,q+n}}{p+q} - \frac{\delta_{p,-q+n}}{p-q}\right] \tag{S.5.52c}$$

$$= \frac{\sin\left(\frac{n\pi}{2}\right)}{2\pi}\sum_{q=1}^{\infty} a_q \left(\frac{(-1+(-1)^q)a_{n-q}}{n-2q} + \frac{\left(1-(-1)^q\right)a_{q-n}}{2q-n} + \frac{(-1+(-1)^q)a_{q+n}}{n+2q}\right) \tag{S.5.52d}$$

$$= \frac{\sin\left(\frac{n\pi}{2}\right)}{2\pi}\left(\sum_{q=1}^{n-1}\frac{(-1+(-1)^q)a_{n-q}a_q}{n-2q} + \sum_{q=n+1}^{\infty}\frac{\left(1-(-1)^q\right)a_{q-n}a_q}{2q-n} + \sum_{q=1}^{\infty}\frac{(-1+(-1)^q)a_{q+n}a_q}{n+2q}\right) \tag{S.5.52e}$$

$$= \frac{\sin\left(\frac{n\pi}{2}\right)}{2\pi}\left(\sum_{q=1}^{n-1}\frac{(-1+(-1)^q)a_{n-q}a_q}{n-2q} + (1-(-1)^n)\sum_{q=1}^{\infty}\frac{(-1)^q a_{n+q}a_q}{n+2q}\right). \tag{S.5.52f}$$

We relabeled p and q in the second term of eq. (S.5.52a). In eq. (S.5.52e) we again make use of the property $a_n = 0\ \forall n \leq 0$ to constrain the boundaries of the first two sums. After shifting the index in the second sum $p \to p+n$ we arrive at the final form eq. (S.5.52f). Substituting eq. (S.5.52f) back into the amplitude equation, we get

$$\begin{aligned} &\left(w_2 k_s^2\left(\frac{\sin\left(\frac{n\pi}{2}\right)}{n\pi} - \frac{1}{4}\right) - w_0\right)a_n + \\ &w_2 k_s^2\frac{\sin\left(\frac{n\pi}{2}\right)}{2\pi}\left(\sum_{q=1}^{n-1}\frac{(-1+(-1)^q)a_{n-q}a_q}{n-2q} + (1-(-1)^n)\sum_{q=1}^{\infty}\frac{(-1)^q a_{n+q}a_q}{n+2q}\right) = 0, \end{aligned} \tag{S.5.53}$$

for steady state. Since $\sin\left(\frac{n\pi}{2}\right) = 0$ for even n, the nonlinear term is zero for even n unless $n = 2q$. This leads to

$$a_{2q} = \frac{a_q^2}{4\frac{w_0}{w_2 k_s^2} + 1}. \tag{S.5.54}$$

Truncating eq. (S.5.53) up to the second mode, we get a quadratic equation for the order parameter, which has the solution

$$a_1 \approx \frac{1}{4k_s}\left(\frac{3(1+4w_0)}{\pi w_2}(w_0^{\mathrm{cr}} - w_0)\right)^{\frac{1}{2}}. \tag{S.5.55}$$

Truncating eq. (S.5.53) up to 4th order leads to the best possible analytical approximation, that can be obtained through this method. The resulting equation for a_1 was obtained using the symbolic language MATHEMATICA [20] and is too large to reasonably be displayed in this supplement. The plot of the 4th order approximation of a_1 is displayed as a solid black line in figure 3 of the main text. Truncating at 5th order already leads to a generic quintic polynomial for a_1, which cannot be solved analytically [21].

S.5.4. PHASE TRANSITION IN THE MCA MODEL

Since our numerical data shows that the phase transition in the MCA model is first order, we expect to have two phase boundaries. One can be obtained from the stability analysis of the ordered phase and the other from the stability analysis of the disordered phase. Using the methods described in section S.5.3, we could only perform the stability analysis of the disordered phase. Analyzing the stability of the ordered phase would involve expanding the state density $f(\theta;t)$ around $\delta(\theta)$, which is was not feasible.

STABILITY ANALYSIS OF THE DISORDERED PHASE IN MCA MODEL

Again we assume the symmetry of $f(\theta;t)$, furthermore we assume that, at least to the linear order in Fourier modes, we can expand $l_2(\theta,\theta';t)$ separately in its arguments, i.e.

$$l_2(\theta,\theta';t) = \frac{k_s}{4\pi^2}\left(1 + 2\sum_{n=1}^{\infty} b_n\cos(n\theta) + 2\sum_{n=1}^{\infty} b_n\cos(n\theta')\right) + \mathcal{O}(b_n b_m) \tag{S.5.56}$$

$$= \frac{k_s}{4\pi^2}\left(1 + 2\sum_{n=1}^{\infty} b_n\left[\cos(n\theta) + \cos(n\theta')\right]\right), \tag{S.5.57}$$

where we also dropped the anti-symmetric part of the Fourier expansion. We are only interested in stability of the state density distribution, which we assume to be symmetric, and since to the linear order the time evolution of symmetric and anti-symmetric modes decouples, we make the analysis easier, by only considering the symmetric modes of the link density distribution. Note that to the linear order, there is no additional information stored in the second variable of l_2, this means that we can integrate it out to simplify eq. (S.5.38), which gives us

$$\partial_t f(\theta;t) = w_0\left(\frac{1}{2\pi} - f(\theta;t)\right) + \frac{k_s^2}{4}\left[\int_{-\pi}^{\pi} \mathrm{d}\theta' \frac{I_{\frac{1}{2},l}(\theta';\theta;t)}{f(\theta';t)} - \frac{l^2(\theta;t)}{f(\theta;t)}\right] \tag{S.5.58a}$$

$$\begin{aligned}\partial_t l(\theta;t) = {}& w_0\left(\frac{1}{2\pi} - l(\theta;t)\right) + d\big(f(\theta;t) - l(\theta;t)\big) + k_s\left[\int_{-\pi}^{\pi} \mathrm{d}\theta' \frac{I_{1,l}(\theta';\theta;t)}{f(\theta';t)} - \frac{l^2(\theta;t)}{f(\theta;t)}\right] \\ & + \frac{k_s^2}{4}\left[\int_{-\pi}^{\pi} \mathrm{d}\theta'\, l(\theta') \frac{I_{1,l}(\theta';\theta;t)}{f^2(\theta';t)} - \frac{l^3(\theta;t)}{f^2(\theta;t)}\right],\end{aligned} \tag{S.5.58b}$$

where, we already set $w_2 = 1$ which rescales time and

$$l(\theta;t) = \int_{-\pi}^{\pi} \mathrm{d}\theta'\; l_2(\theta,\theta';t) = \frac{1}{2\pi}\left(1 + 2\sum_{n=1}^{\infty} b_n \cos(n\theta)\right) + \mathcal{O}(b_n b_m), \tag{S.5.59a}$$

$$I_{\frac{1}{2},l}(\theta;\theta';t) = \int_{-\pi}^{\pi} \mathrm{d}\xi\; l_2(\theta,\theta' + \frac{\xi}{2};t)\; l_2(\theta,\theta' - \frac{\xi}{2};t), \tag{S.5.59b}$$

$$I_{1,l}(\theta;\theta';t) = \int_{-\pi}^{\pi} \mathrm{d}\xi\; l_2(\theta,\theta' + \xi;t)\; l_2(\theta,\theta' - \xi;t). \tag{S.5.59c}$$

We plugged the Fourier expansions of eq. (S.5.43) and eq. (S.5.59a) in eq. (S.5.58), and kept only linear terms in Fourier modes. Since eq. (S.5.58) consists of much longer equations then eq. (S.5.42), we automated this process with MATHEMATICA [20] and got,

$$\partial_t \begin{pmatrix} a_n \\ b_n \end{pmatrix} = \boldsymbol{M} \cdot \begin{pmatrix} a_n \\ b_n \end{pmatrix}, \tag{S.5.60}$$

where

$$\boldsymbol{M} = \begin{pmatrix} \frac{c^2 - 4d^2 \mathrm{w0}}{4\pi d^2} & -\frac{c^2\left(\pi q - 2\sin\left(\frac{\pi q}{2}\right)\right)}{2\pi^2 d^2 q} \\ \frac{c^2 + 2cd + 2d^3}{2\pi d^2} & \frac{4c^2 \sin\left(\frac{\pi q}{2}\right) - \pi q\left(3c^2 + 8cd + 4d^2(d + \mathrm{w0})\right)}{4\pi^2 d^2 q}. \end{pmatrix} \tag{S.5.61}$$

In this case the eigenvalues of $\boldsymbol{M}$ indicate the stability of the system. Only the largest eigenvalue can become positive, which gives us a condition

$$w_0^{\mathrm{cr}}(c,d) = \frac{-\pi\left(c^2 + 4cd + 2d^3\right) + 2c^2 + 2\sqrt{c^4 + 2\pi c^2 d^3 + \pi^2 d\left(2c^3 + 4c^2 d + 4cd^3 + d^5\right)}}{4\pi d^2}, \tag{S.5.62}$$

for critical noise. Below w_0^{cr} the disordered phase is linearly unstable. We plot the critical surface $w_0^{\mathrm{cr}}(c,d)$ in fig. S.5.5. We see that the critical surface trivially depends on c and d. For $c >> d$ the values of $w_0^{\mathrm{cr}}(c,d)$ become very large, which is expected since $c >> d$ is equivalent to a very dense network. In order to stay in the sparse network limit and simplify the expressions, we set $c = d$, which is the convention that we used in the main text. Making this substitution in eq. (S.5.62) and solving for d leads us to eq. (10) from the main text.

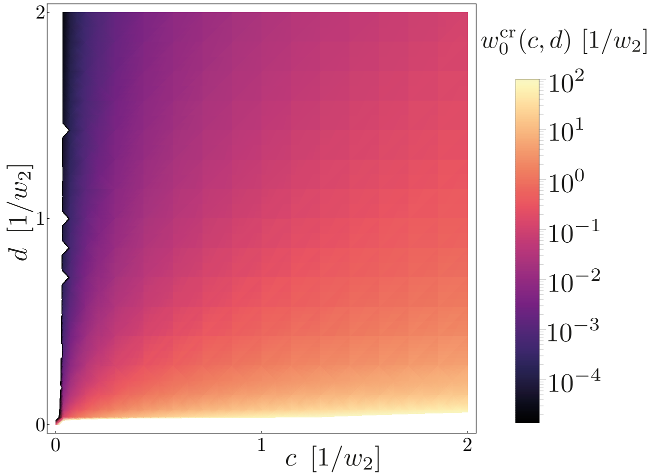

Figure S.5.5: Critical surface $w_0^{\mathrm{cr}}(c,d)$. For values below $w_0^{\mathrm{cr}}(c,d)$, disordered phase is linearly unstable.

S.5.5. Integral transformations

Here we show two general results that we used to arrive at simplified forms of differential equations. First we used the sign reversal symmetry of the ξ integral

$$\int_0^{\frac{\pi}{2}} \mathrm{d}\xi\, g(\theta-\xi)h(\theta+\xi) = \frac{1}{2}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \mathrm{d}\xi\, g(\theta-\xi)h(\theta+\xi) = \frac{1}{4}\int_{-\pi}^{\pi} \mathrm{d}\xi\, g\left(\theta-\frac{\xi}{2}\right)h\left(\theta+\frac{\xi}{2}\right), \tag{S.5.63}$$

and after defining

$$I(\theta) = \int_{-\pi}^{\pi} \mathrm{d}\xi\, g\left(\theta-\frac{\xi}{2}\right)h\left(\theta+\frac{\xi}{2}\right), \tag{S.5.64}$$

we show a general relation,

$$\int_{-\pi}^{\pi} \mathrm{d}\theta\, I(\theta) = \int_{-\pi}^{\pi} \mathrm{d}\theta \int_{-\pi}^{\pi} \mathrm{d}\xi\, g\left(\theta-\frac{\xi}{2}\right)h\left(\theta+\frac{\xi}{2}\right) \tag{S.5.65a}$$

$$= \int_{-\pi}^{\pi} \mathrm{d}\xi \int_{-\pi-\frac{\xi}{2}}^{\pi-\frac{\xi}{2}} \mathrm{d}\theta\, g(\theta)h(\theta+\xi) \tag{S.5.65b}$$

$$= \int_{-\pi}^{\pi} \mathrm{d}\xi \int_{-\pi}^{\pi} \mathrm{d}\theta\, g(\theta)h(\theta+\xi) \tag{S.5.65c}$$

$$= \int_{-\pi}^{\pi} \mathrm{d}\theta\, g(\theta) \int_{-\pi+\theta}^{\pi+\theta} \mathrm{d}\xi\, h(\xi) \tag{S.5.65d}$$

$$= \int_{-\pi}^{\pi} \mathrm{d}\theta\, g(\theta) \int_{-\pi}^{\pi} \mathrm{d}\xi\, h(\xi) \tag{S.5.65e}$$

$$= GH \tag{S.5.65f}$$

which holds for any functions $g(\theta)$ and $h(\theta)$ which are periodic on $(-\pi,\pi]$. We used the property of periodicity in eq. (S.5.65c) and eq. (S.5.65e) to discard the shift of the integral boundaries, since the integral is over the whole domain of a periodic function. And G and H denote the integrals over the full period of the functions g and h respectively.

Author Contributions

JD and TI formulated the initial idea for the project. CK formulated the equations of motion 2.1 for systems with continuous state based on [7], and performed preliminary numerical analysis. GD formalized the moment closure and mean field approximations to 2.1, performed all numerical analysis presented in this work, and derived all analytic results for the stability analysis. All authors contributed to planning, which kind of numerical analysis to perform, and the refinement of the research question once the preliminary results were present. GD drafted the text for all sections of this work. All authors contributed to the final version of the text.

BIBLIOGRAPHY

[1] T. Vicsek and A. Zafeiris, "Collective motion", Physics Reports **517**, 71 (2012).

[2] R. Bouffanais, *Design and Control of Swarm Dynamics*, Understanding Complex Systems (Springer Singapore, 2016).

[3] R. van Drongelen, A. Pal, C. P. Goodrich, and T. Idema, "Collective dynamics of soft active particles", Physical Review E **91**, 032706 (2015).

[4] D. R. McCusker, R. van Drongelen, and T. Idema, "Active particle dynamics beyond the jamming density", EPL (Europhysics Letters) **125**, 36001 (2019).

[5] D. v. H. t. Echten, G. Nordemann, M. Wehrens, S. Tans, and T. Idema, "Defect dynamics in growing bacterial colonies", arXiv:2003.10509 [cond-mat, physics:physics, q-bio] (2020).

[6] T. Vicsek, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, "Novel Type of Phase Transition in a System of Self-Driven Particles", Physical Review Letters **75**, 1226 (1995).

[7] L. Chen, C. Huepe, and T. Gross, "Adaptive network models of collective decision making in swarming systems", en, Physical Review E **94**, 022415 (2016).

[8] J. Buhl, D. J. T. Sumpter, I. D. Couzin, J. J. Hale, E. Despland, E. R. Miller, and S. J. Simpson, "From Disorder to Order in Marching Locusts", Science **312**, 1402 (2006).

[9] N. C. Makris, P. Ratilal, S. Jagannathan, Z. Gong, M. Andrews, I. Bertsatos, O. R. Godø, R. W. Nero, and J. M. Jech, "Critical Population Density Triggers Rapid Formation of Vast Oceanic Fish Shoals", Science **323**, 1734 (2009).

[10] L. Huber, R. Suzuki, T. Krüger, E. Frey, and A. R. Bausch, "Emergence of coexisting ordered states in active matter systems", Science **361**, 255 (2018).

[11] G. Grégoire and H. Chaté, "Onset of Collective and Cohesive Motion", Physical Review Letters **92**, 025702 (2004).

[12] M. Nagy, I. Daruka, and T. Vicsek, "New aspects of the continuous phase transition in the scalar noise model (SNM) of collective motion", Physica A: Statistical Mechanics and its Applications **373**, 445 (2007).

[13] M. Aldana, V. Dossetti, C. Huepe, V. M. Kenkre, and H. Larralde, "Phase Transitions in Systems of Self-Propelled Agents and Related Network Models", Physical Review Letters **98**, 095702 (2007).

[14] V. Sood and S. Redner, "Voter Model on Heterogeneous Graphs", Physical Review Letters **94**, 178701 (2005).

[15] C. Huepe, G. Zschaler, A.-L. Do, and T. Gross, "Adaptive-network models of swarm dynamics", New Journal of Physics **13**, 073022 (2011).

[16] L. Onsager, “Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition”, Physical Review **65**, 117 (1944).

[17] N. D. Mermin and H. Wagner, “Absence of Ferromagnetism or Antiferromagnetism in One- or Two-Dimensional Isotropic Heisenberg Models”, Physical Review Letters **17**, 1133 (1966).

[18] F. Merkl and H. Wagner, “Recurrent random walks and the absence of continuous symmetry breaking on graphs”, Journal of Statistical Physics **75**, 153 (1994).

[19] J. A. Pimentel, M. Aldana, C. Huepe, and H. Larralde, “Intrinsic and extrinsic noise effects on phase transitions of network models with applications to swarming systems”, Physical Review E **77**, 061138 (2008).

[20] Wolfram Research,Inc., *Mathematica, Version 12.0*, Champaign, IL, 2019.

[21] M. I. Rosen, “Niels hendrik abel and equations of the fifth degree”, The American Mathematical Monthly **102**, 495 (1995).

6

Preliminary results and Outlook

My PhD was part of the *BaSyC* consortium, which has the goal to **B**uild **a Sy**nthetic **C**ell. My project was part of the cell division work package and as I understood it at the time, my goal was to better understand cell division. As part of the first generation of PhDs hired by *BaSyC* it was clear to me that I was unlikely to see the completion of this immense task during my four years in Delft, and thus I would work on the fundamental parts of the problem . The question that I needed to answer early on was how I to interpret the general goal of "better understanding cell division". The answer to these questions was heavily informed by my training as a theoretical physicist and the existing expertise in the Idema group. Very early on I decided to contribute to the understanding of the physical mechanisms underlying cell division, by studying membrane mediated interactions, which can lead to self-assembly of protein structures on the membrane and its subsequent reshaping. In introductory **chapter 1**, I first elaborate on the biological context relevant to the membrane physics, then introduce the theory behind membrane mediated interactions and present several numerical methods that can be used to model membranes and their interactions with membrane deforming objects. The early projects that I was involved in were all based on analytical approaches which have the great advantage of delivering easily interpretable results. These projects are presented in chapters two and three. In **chapter 2** I studied the interaction of point like force generating objects on the membrane. These point like objects are supposed to model the action of filamentous proteins like the actin network or the microtubules that can push into the membrane and generate force through polymerization. We found that the local force generation by the filaments induces membrane deformations that create effective attractive potentials between the filaments and leads to their bundling. In **chapter 3** we looked at proteins that generate local deformations instead of forces and analyzed the role of feedback between the mixing entropy of such proteins and the change in curvature and structural properties of membranes. We showed that, depending on the characteristics of the proteins, the system can undergo two different types of phase separation, which may have different implications on the shape instability of the membrane.

While working with these analytic approaches their shortcomings became clear. The Canham-Helfrich energy (see eq. (1.2) in the introduction) gives rise a shape equations that is highly nonlinear and contains fourth order derivatives of the shape field. This hinders us to make any progress analytically, without strong simplifying assumptions. This limited the analysis in the second chapter to the small deformation regime, and we needed to employ the mean field approximation in the third chapter, which means that while we know that the demixing is happening, we can not be sure what the resulting patterns look like. To overcome the limitations of analytic approaches that necessitate strong approximations, it was clear that simulation methods would be necessary. Dynamically triangulated membrane Monte Carlo (DTMMC) simulations were my the method of choice. DTMMCs allow us to describe large scale membrane deformations and easily simulate interactions of the membrane with deforming objects while not sacrificing the easy interpretability, since they are based on minimizing the discretized version of the Canham-Helfrich energy, it is usually easy to interpret the observed effects. The popularity of this method speaks on its own. The only problem with simulations is a high entry barrier, necessitated by upfront setup of the simulation. In **chapter 4** I present flippy, an open source `c++` library that I built and published to make DTMMC simulations very easy to set up, since it abstracts away several hundreds of lines of technical code into few lines of high level definitions. flippy implements the dynamic triangulation and its update rules, such that the end user does not need to think about those tedious details and can focus on higher level issues that directly relate to the system properties that they are investigating. During the supervision of bachelor students working on this topic, I observed that flippy can indeed shorten a period of several weeks of simulation setup to few hours. In section 1 of this chapter I present the preliminary results that I obtained using flippy.

1. Preliminary results on vesicle division by external forces

1.1. Introduction

Division of membrane compartments is an important step for building a synthetic cell that can grow and reproduce. In 2018 Deshpande et al. [1] published experiments on mechanical division of liposomes through external forces. In these experiments a large membrane compartment, a liposome, is first deflated through osmotic pressure to give it excess area (fig. 1.1 T.1), then lead through a microfluidics setup and collided with a wedge (fig. 1.1 T.2), which splits the liposome in two daughter liposomes (fig. 1.1 T.3), in successful cases. This system presents an interesting opportunity to study membrane deformations in isolation, without having to account for other actors like proteins. Our central interest was the modeling of the deformations induced by the wedge. It is most convenient to predict the shape right before splitting (see fig. 1.1 (P.2)). This allows us to sidestep the problem with unknown gaussian curvature modulus and enables us to use the triangulated representation of the membrane that cannot easily accommodate topological changes, like splitting. We reproduced the four experimentally observed cases: (1) symmetric splitting, in which case the daughter vesicles have similar sizes (experimentally a 3% deviation in the diameter of the two daughter liposomes was reported); (2) asymmetric division, in which case only one daughter vesicle survives the division

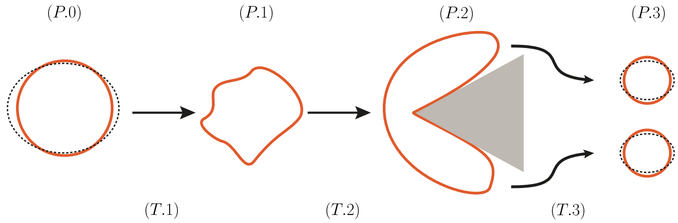

Figure 1.1: A cartoon depiction of different stages of the liposome during the splitting experiment. **P.0:** assembly phase where almost spherical liposomes are produced. **P.1:** the cells are deflated through osmotic pressure to a fraction v of their original volume to make volume and surface conserving splitting possible. **P.2:** the cell collides with a wedge and undergoes bending and stretching. **P.3:** if the split was successful two new cells are created which have half of the surface area of the original cell. z-axis is coming out of the plane and $z = 0$ is chosen s.t. the experimental setup is sliced through the middle. x-axis is chosen such that the splitter is flip symmetric with respect to it.

and the other bursts; (3) bursting, in which case the mother liposome bursts upon encountering the edge and (4) snaking, where the liposome does not divide and simply flows through on one of the sides of the wedge.

1.2. METHODS

To simulate the experimental system we implemented a Monte Carlo simulation in c++. The vesicle was implemented as a dynamic triangulation using flippy[1] [2], while the wedge was implemented as a rigid body, that had volume exclusion interactions with the beads of the membrane triangulation[2]. The membrane triangulation was updated according to the Canham-Helfrich bending energy [4, 5]

$$E_{\mathrm{CH}} = \int \left[\frac{\kappa}{2}\mathrm{d}A(2H)^2 + \bar{\kappa}K_{\mathrm{G}}\right], \tag{1.1}$$

where κ and $\bar{\kappa}$ are the bending rigidity and the saddle-splay moduls of the membrane and H and K_{G} are membrane's local mean curvature and the Gaussian curvature respectively. Since we are only modeling the deformation before splitting as seen in fig. 1.1 (P.2), we can disregard the Gaussian part of the bending energy, which only changes value during topological transitions. Bending energy can be discretized in a variety of ways on a triangular grid. We use the builtin bending energy from flippy (presented in detail in section 2.2), which follows the definition from Gueguen et al. [6]. In addition to the bending energy we also enforce the conservation of area A and volume V, with harmonic potentials

$$E_A = K_A\frac{(A-A_0)^2}{A_0} \tag{1.2}$$

$$E_V = K_V\frac{(V-V_t)^2}{V_t}. \tag{1.3}$$

The constants K_A and K_V penalize area and volume extension. Area is fixed to its initial value $A_0 = 4\pi R_0^2$ which is the area of a sphere with radius R_0. Volume is fixed to the target value $V_t = vV_0$, which is a fraction of the spherical volume $V_0 = \frac{4\pi}{3}R_0^3$. The parameter v giving that fixes the ratio of the target volume and spherical volume, is the most important parameter in the simulation. It is equivalent to the reduced volume that is usually defined as

$$v = 6\sqrt{\pi}\frac{V}{A^{\frac{3}{2}}}, \tag{1.4}$$

which gives the ration between the actual volume (V) of the liposome and the volume of a sphere with the same surface area (A). The variation in the reduced volume v is necessary to reproduce the different experimental outcomes. This is in qualitative agreement with the experiments, the reduced volume was correlated with successful division. The interactions between the wedge and the membrane triangulation beads[3] is mediated by

[1] We implemented the flippy software package ourself and it is described in detail in chapter 4.

[2] Code and data simulation data presented in this chapter can be seen online [3].

[3] The representation of the membrane as a triangulation of connected beads with a finite radius is discussed in chapter 4

a soft repulsion potential. For this interaction we used the repulsive part of the Lennard-Jones potential

$$E_{\text{int}} = \begin{cases} K_{\text{int}} \left(\frac{r_b}{d}\right)^{12} & \text{if } d < r_b \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

where K_{int} is the cost of full overlap per bead, d is the distance between the center of the bead and the closest wedge surface, r_b is the radius of the bead, which also happens to be the touching distance of the beads and the wedge surface. This potential is only turned on if the triangulation beads actually overlap with the wedge i.e. if $d < r_b$. The total system energy then reads

$$E_{\text{tot}} = E_{\text{CH}} + E_A + E_V + E_{\text{int}}. \tag{1.6}$$

This energy is used to update the membrane triangulation. In each update step we either move a bead or flip a bond between pairs of beads (see fig. 1.8 from section 1.4.2 for an illustration). The resulting change in the surface changes the total energy. We calculate the total energy before and after an update as $E_{\text{tot}}^{\text{old}}$ and $E_{\text{tot}}^{\text{new}}$ and respectively. Then the acceptance probability of the membrane update p_m is given by the expression

$$p_m = \max(\exp(\Delta E), 1), \tag{1.7}$$

with

$$\Delta E = E_{\text{tot}}^{\text{old}} - E_{\text{tot}}^{\text{new}}. \tag{1.8}$$

Finally we have the work W performed by the flow of the fluid which pushes the liposome into the wedge. We want to avoid the complications of modeling an actual fluid flow around a vesicle, instead we push the wedge into the liposome and add the work to the total energy. Then the acceptance probability of a wedge update p_w is given by the expression

$$p_w = \max(\exp(\Delta E + W), 1). \tag{1.9}$$

The values of the constants used in the simulation are provided in section S.1.4, together with some explanation of how the values were chosen.

1.3. RESULTS AND DISCUSSION

Monte Carlo simulations provide us with liposome shapes deformed by the wedge. We managed to reproduce all four cases of experimentally observed behaviour. In fig. 1.2 a) and b) we see the vesicle right before division. In both cases the wedge presses the membrane walls together, and we identify this as successful splitting. We observe that the lobes are not perfectly symmetric in either case and the asymmetry is more pronounced in fig. 1.2 b), which leads us to believe that the asymmetric division is largely controlled by the random configuration in which the vesicle encounters the splitter. The configuration in fig. 1.2 c) with reduced volume $v = 0.75$ is interpreted as the bursting event. Since after the monte carlo simulation equilibrates we see that the vesicle width is not sufficiently narrow at the tip of the wedge. Finally we show a snaking configuration in fig. 1.2 d). In this case the reduced volume was too close to that of a sphere ($v = 0.85$) and the vesicle was not significantly deformed when encountering the edge but was instead deflected to one of the sides.

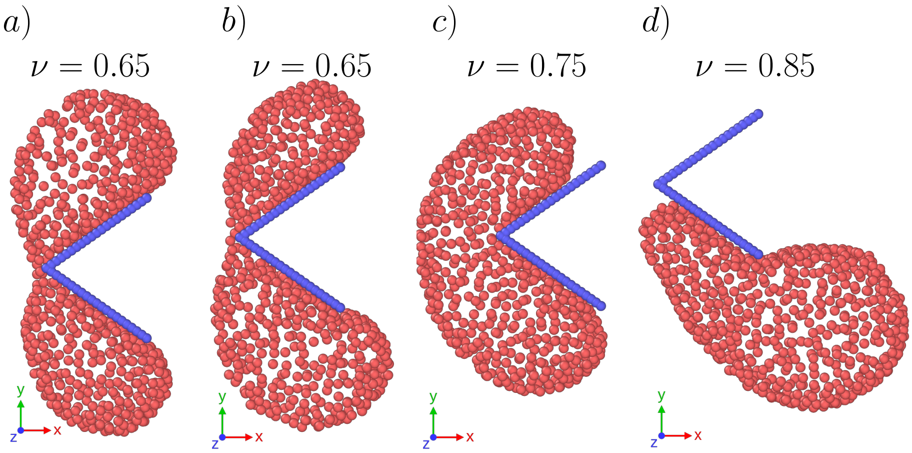

Figure 1.2: Four outcomes of the experiment reimplemented in the simulation. a) & b) Two runs of a simulation with same initial conditions and reduced volume ($\nu = 0.65$). We see that the lobes of the dividing liposome are more symmetric in a) than in b). c) & d) Bursting and snaking, same initial conditions as before, with different reduced volume parameters of $\nu = 0.75$ and $\nu = 0.85$ respectively.

The advantage of simulating a system is that we can get access to more information. In this case the shape of the liposome along the z-axes is the new information that was not accessible in the experiments, that only provided a top-down view. In fig. 1.3 we can see four progressive snapshots of the simulation corresponding to fig. 1.2 a). From this viewpoint we see that the wedge does not cut the liposome across its full height, but a neck starts to form which gradually narrows to a point where it can easily be pinched. This information together with the reported 10% leakage from the experiment strongly indicates that the division happens through splitting of liposomes that creates small pores on daughter vesicles, that are resealed again. According to the theory on the dynamics of transient pores by Brochard-Wyart et al. [7], the maximum radius ($r_{\max}$) of a pore that can automatically reseal itself, is given by a simple relation

$$r_{\max} = \frac{\gamma}{\sigma}, \tag{1.10}$$

where γ is the edge tension of a pore, which was estimated to be around 27pN for DOPC [8], the same type of lipid a used in the splitting experiments. σ denotes the membrane tension which is very hard to measure or estimate and can vary anywhere between 10^{-3} pN/nm and 10 pN/nm [9]. The value of σ will depend on the amount of stretching that the liposome experiences, and we believe that it will be the deciding factor for successful splitting. Further simulations are necessary to extract an estimate of the tension in the vesicle during division and correlate the resulting $r_{\max}$ values with successes and failures in vesicle division. However, here we can provide another result that supports the idea that in-plane tension plays the key role in the division process. We conducted the simulation from fig. 1.3 again, under the same conditions with single change. We turned off the bond flipping between the beads of the triangulation, meaning that the triangulated

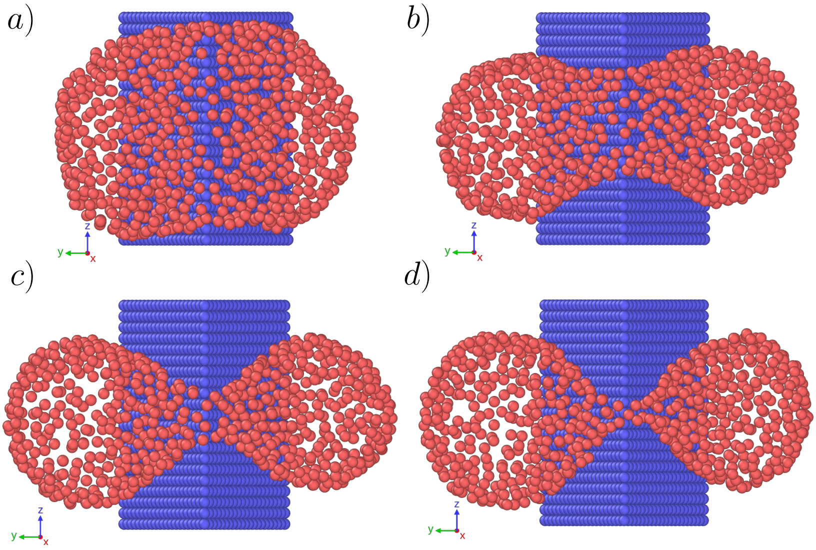

Figure 1.3: View of the collision revealing the shape of the liposome along the height of the wedge (z-axis). a) Initial deformation during early collision (frame 69). b) Neck starts to form (frame 90). c) Neck becomes very pronounced (frame 115) d) Final frame of the simulation with touching neck beads.

surface that describes the membrane shape was no longer fluid. This simple change completely destroyed the neck formation process, as seen in fig. 1.4. The height of the liposome in the middle remains the same between an early snapshot (fig. 1.4 a)) and the final snapshot (fig. 1.4 b)), a neck never forms and membrane walls never come very close.

In conclusion we can say that our preliminary results point strongly in favour of splitting and resealing of the liposome with an intermediate state of daughter liposomes with leaky pores, as the division mechanism. We believe that this process is strongly influenced by in plane tension and furthermore we see that the inhibition of fluidity of the membrane completely destroys the neck formation process that precedes the division of the vesicles. Suggesting that the optimal conditions for liposome division through external forces are provided by floppy liposomes composed of lipids with high fluidity.

S.1.4. SUPPLEMENT: SIMULATION PARAMETERS

DATA AVAILABILITY

The simulation code and data of the results presented in this chapter is available for download online [3]. The code is contained in the `liposome_splitting` sub-folder and the data is contained in the `data` sub-folder. The simulation parameters are provided to the binary via a configuration file `config.json`. The data folders for different simulations contain three files: `data_reload_0.json` which contains different types of data like the total bending energy, total area, total volume etc. This data was not used in this chapter but would be necessary for more quantitative analysis. The file `data.xyz`

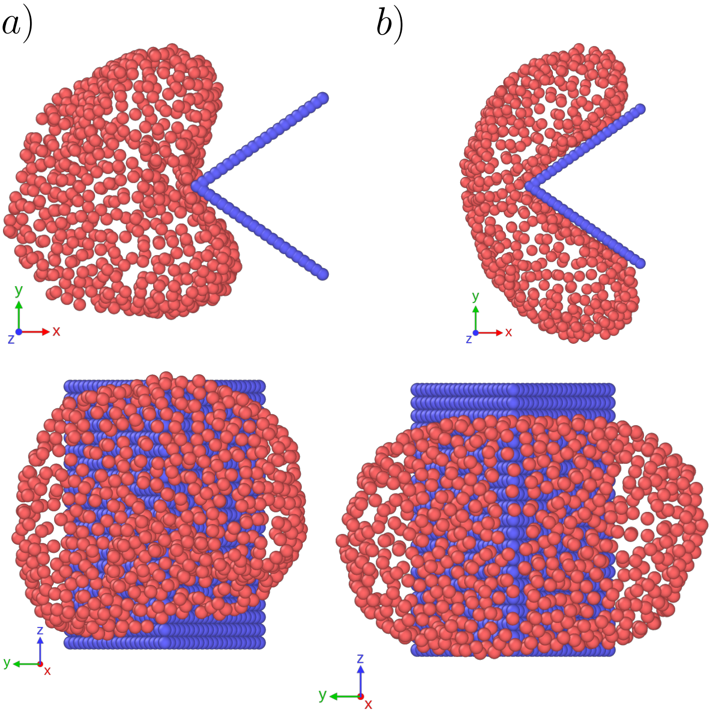

Figure 1.4: Triangulation of tethered beads that can not exchange their neighbours This simulation ran for the same amount of Monte Carlo time staps as all the previous ones. If we run the simulation twice as long (not shown here) then random fluctuations push more triangulation beads on one side of the splitter and the whole liposome eventually snakes around it. Top panels show the top-down view and the bottom panels show the back-to-front view of a colliding liposome. a) Initial deformation during early collision. b) final configuration at the and of the simulation. The bottom panels look similar at the beginning and at the end of the simulation, since no neck formation takes place.

contains coordinates of membrane beads and beads representing the wedge[4] for saved time frames; the_egg_last.json which contains the full triangulation details and the position of the wedge from which the simulation state can be reconstructed, thus enabling to continue the simulation from the last frame. Both data_reload_0.json and the_egg_last.json contain a sub-field called metadata which contains a local copy of the config.json file which seeded the parameter values of each simulation. Even the seed for the random number generator is saved, meaning that these simulations are fully reproducible.

In table 6.1 we show the most important set of parameters used in the simulations and give our motivations for choosing their specific values.

[4] Note that the internal representation of the wedge inside the simulation is not that of a collection of beads, but that of an prism made of planes. It is just much simpler to visualize the wedge as a collection of beads.

2. OUTLOOK

Looking forward, we should decide how theoretical physicists can contribute to the endeavour of artificial cell division in *BaSyC*. If we take the completion of the initial goals (i.e. actually building a synthetic cell) seriously, then I believe that there is very little room for purely theoretical projects in the cell division work package of *BaSyc*. Building something is a very applied task from the point of view of a theoretical physicist and to contribute to the process of building, the transition out of the fundamental research phase is crucial. Which entails heavy utilization of computational methods to model specific experimentally inspired scenarios of vesicle division. We developed flippy with exactly this application in mind. Lowering the barrier to model a diverse set of experimental conditions and a short feedback cycle between experiment and theory will be essential to construct a dividing membrane compartment.

parameter name in simulation	symbol in latex	value [units]	note
bead_diameter	$d_b = 2r_b$	1.0 [a.u]	Canham-Helfrich theory is scale invariant so there is no natural length scale in the simulations. Beads as the smallest objects are used as the length unit.
R_guv	R_0	10 [d_b]	This value was chosen such that the triangles were not too large.
N/A	$k_\mathrm{B}T$	1	This value is not explicitly set in the parameters but is implicit in the code. Sets the energy scale of all other parameters.
bending_rigidity	N/A	10 [$k_\mathrm{B}T$]	Typical literature value (at least order of magnitude) for lipid membrane bending rigidities [10].
K_A	N/A	1000 [$k_\mathrm{B}T/d_b^2$]	Large enogh to fix the target value well. Discovered through trial and error.
K_V	N/A	600 [$k_\mathrm{B}T/d_b^3$]	
K_int	K_{int}	50 [$k_\mathrm{B}T$]	We chose a large value to be sure that the exclusion interactions are close to hard body exclusion.
area_target	A_0	$4\pi R_0^2$	Area is fixed to that of the initial sphere
volume_target	N/A	varying between $0.65V_0$ and $0.85V_0$	Target volume is the control parameter. We change it between the regimes where there is plenty of excess area for two daughter spheres to where there is not enough.
work	N/A	10000 [$k_\mathrm{B}T$]	Large enough for updates to happen
max_to_average _bond_length _fraction	N/A	2.5	If the edges of the triangles in the membrane triangulation are allowed to stretch too much the membrane can start self intersecting and non physical configurations will start minimizing the energy. This sets the ration between the maximum bond length and the initial average bond length. This value was chosen after some trial, it works well.
N_bond_flips_per _mem_update	N/A	varying, 0 or 1	Number of bond flip updates per move update. Setting it to 0 turns of the bond flip update.
full_opening_angle	N/A	70°	Angle of the wedge tip, value is the same as in the experiment.

Table 6.1: Most important parameters used in the simulations.

Bibliography

[1] S. Deshpande, W. K. Spoelstra, M. Van Doorn, J. Kerssemakers, and C. Dekker, "Mechanical division of cell-sized liposomes", ACS nano **12**, 2560 (2018).

[2] G. Dadunashvili, *Flippy-software-package/flippy: v1.0.0*, version v1.0.0, Mar. 2022.

[3] G. Dadunashvili and T. Idema, *Code and Data Related to Liposome Splitting Simulations*, version 1, Apr. 2022.

[4] P. B. Canham, "The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell", Journal of theoretical biology **26**, 61 (1970).

[5] W. Helfrich, "Elastic properties of lipid bilayers: theory and possible experiments", Zeitschrift für Naturforschung C **28**, 693 (1973).

[6] G. Gueguen, N. Destainville, and M. Manghi, "Fluctuation tension and shape transition of vesicles: renormalisation calculations and monte carlo simulations", Soft Matter **13**, 6100 (2017).

[7] F. Brochard-Wyart, P. G. de Gennes, and O. Sandre, "Transient pores in stretched vesicles: role of leak-out", Physica A: Statistical Mechanics and its Applications **278**, 32 (2000).

[8] T. Portet and R. Dimova, "A New Method for Measuring Edge Tensions and Stability of Lipid Bilayers: Effect of Membrane Composition", en, Biophysical Journal **99**, 3264 (2010).

[9] W. Rawicz, K. C. Olbrich, T. McIntosh, D. Needham, and E. Evans, "Effect of chain length and unsaturation on elasticity of lipid bilayers", Biophysical Journal **79**, 328 (2000).

[10] R. Phillips, "Membranes by the numbers", in *Physics of biological membranes*, edited by P. Bassereau and P. Sens (Springer International Publishing, Cham, 2018), pp. 73–105.

ACKNOWLEDGEMENTS

First and foremost I want to thank my parents. Your support, encouragement and advice was integral to me to get this far.

დედა და მამა, ძალიან დიდი მადლობა თქვენი თანადგომისათვის. ჩემი სადოქტოროს დროს, როცა ყველაზე მეტად მიჭირდა ხოლმე, სახლში ჩამოსვლა და თქვენთან ერთად ერთი ორი კვირის გატარება ყოველთვის ძალიან მეხმარებოდა. განსაკუთრებული მადლობა ჩემი სადოქტოროს ბოლო თვეებში გამოჩენილი დახმარებისათვის, უთქვენოდ მე ამ თეზისის წერას ვერ დავამთავრებდი.

ასევე დიდი მადლობა მინდა გადავუხადო დარიკო ბებოს. აქ მთელი ამაგის ჩამოთვლას ვერ მოვასწრებ ასე რომ ორიათას ოცი წლის ზამთარში რომ ბორშის რეცეფტი ჩამაწერინე იმაზე გადაგიხდი მადლობას ბებო! ბევრი ცივი საღამო გავითბე მაგ რეცეპთით გაკეთებული ბორშით.

Next I want to thank my supervisor **Timon**. Working with you Timon was everything I hoped it would be. When I was finishing the first draft, of my first paper, you went on a one-weak vacation, and I promised you to have a finished draft for your return. I then had a new idea and spent the weak on working it out instead. I remember being nervous when I came to your office to tell you that I did some new research instead of writing up the already finished stuff. I was concerned that you would be annoyed by the delays I was causing. You just smiled and told me doing research was what I was there for. That conversation set the tone for the entirety of my PhD. You always let me explore my projects on my own terms and allowed me to spend time on my growth as a physicist, even if it came at your expense. You always put my development first and tolerated many of my tamper-tantrums while calmly guiding me through my PhD. Thanks for the past four years.

Thanks to the rest of the Idema group. **Felix** I always enjoyed our many discussions on topics ranging from the future of theoretical physics to the lack of proper food in TU Delft cafeterias. **Rachel** you were always ready to point out when you liked a figure on my slides. I always appreciated that because I put a lot of effort in them and no one else ever noticed. I also want to thank the students from the group, that I had the pleasure to work with. Especially **Joep**, **Storm**, **Dion** and **Emma**, I feel like I learned way more from you guys than you learned from me.

Finally, I want to thank my friends. They all make life much more livable! **Guga**, we have known each other from the third grade and knowing you is probably the reason why I became a physicist, which is crazy to think about. In the tenth grade I wanted to put all my extracurricular time in a historical essay writing competition. I remember telling you about this, and I remember you bluntly telling me that, that sounded very boring and that I should participate in this fun physics tournament with you, that you had just heard about. Somehow that “argument” convinced me, and now I am getting my doctorate in physics! During that physics tournament I also met **Mari** who is one of my paranymphs. Mari I am happy that you can be part of my defense, even though I never really explained to you what a paranymph does. Visiting you in Cologne on the way back from Germany, was an important tradition of my PhD, I always have a lot of fun when I hang out with you and I value your advice as a more senior scientist. Ofcourse special thanks go to my friends that I made during the PhD, **Maria**, **Christine** and **Diederik**. Meeting you guys at our graduate schools yearly retreats, was the best thing that the graduate school ever did for me, and it almost makes up for the other things it put me through. Joking aside having met you guys is without any competition, by far the best part of my PhD and I value your friendships a lot. And most of my memories from last four years has you guys in it. **Maria** you and I always managed to go through the same PhD problems at the same time, so it was always very cathartic to commiserate with you and I can't wait for you to start your PostDoc. **CD** our daily zoom calls really kept me sane during the early corona lockdown, and our many dinners and BBQ's were some of the most carefree times I had in Delft. I am very glad that in the crapshoot that is the academic job-market we somehow ended up decently close together geographically and I will be seeing you guys semi-regularly in the foreseeable future.

CURRICULUM VITÆ

George DADUNASHVILI

21-04-1992 Born in Tbilisi, Georgia.

EDUCATION

1998–2010 School

2010–2013 Bachelor of Science in Physics
Ludwig-Maximilians-universität München

Thesis: Effect of Grain Growth on Dust Content of Photoevaporative Flows in Protostellar Discs

Supervisor: Prof. dr. B. Ercolano

2013–2017 Master of Science in Physics
Ludwig-Maximilians-universität München

Thesis: Analysis of the genetic code:
Characterisation of robustness properties in the framework of graph theory

Supervisor: Prof. dr. E. Frey

2022 Ph.D. in Theoretical Physics
Delft University of Technology

Thesis: Physical principles of membrane reshaping
A theoretical study of self organization on membrane surfaces through membrane-mediated interactions

Promotor: Dr. T. Idema

List of Publications

Preprints

2. C van de Kamp, **G Dadunashvili**, J Dubbeldam, T Idema, *Emergent order in a continuous state adaptive network model of living systems*, preprint arXiv:2103.04312.

1. A Vahid, **G Dadunashvili**, T Idema, *Membrane-mediated interactions induce spontaneous filament bundling*, preprint bioRxiv.

Software

2. **G Dadunashvili**, T Idema, *flippy: a c++20 package for dynamically triangulated membrane simulations.* , available on github.

1. **G Dadunashvili**, T Idema, *diffconvolver: a very very very simple finite difference solver package for python*, available on github.

I have discovered such wonderful things that I was amazed... out of nothing I have created a strange new universe.

János Bolyai writing to his father Farkas