Bachelor Thesis

Membrane mediated interactions and pattern formation of conical inclusions on a flat membrane

Roel Marijn Vos

Student number: 4391489
E-mail address: roelmarijn@gmail.com

Delft University of Technology
Department of Bionanoscience
Department of Applied Mathematics

Supervisor Applied Physics and Nanobiology:
Dr. Timon Idema (daily supervisor)

Supervisor Applied Mathematics:
Dr. Johan Dubbeldam

August 23, 2018

Preface

This document was written as part of my thesis project for the three bachelors of Nanobiology, Applied Physics and Applied Mathematics at the Delft University of Technology. In the past year that I have been working on this project, I have had a lot of fun and, more importantly, I have learned a lot. It was with youthful naivety that I started this project, and it is with somewhat more experienced naivety that I now finish it.

I started the project with the idea that I would spend most of my time on interactions on cylindrical and spherical membranes. But first, I would spend a few weeks getting to know the subject by studying interactions on the simpler flat membrane. As is often the case in research, things did not go as planned. The flat membranes turned out to be a lot more interesting than expected. By the time I finished simulating all systems that seemed interesting, I started to doubt the model itself. This led to a particularly exciting time, as my youthful naivety led me to believe that I would undermine several years of research by showing the limited reliability of this model.

But all is well that ends well. Instead of proving the unreliability of the model, I ended up producing substantial evidence that the model is, in fact, reliable. At this point, nearly a full year has gone by since I began and I never once looked at cylindrical or spherical membranes. But even so, I see my project as quite a success and also as an excellent example of the unpredictability of research.

During my project, I have been supported by many people. First, I would like to thank Timon Idema for giving me the opportunity to work in his team on this project and for being my daily supervisor. Thank you for always having the time to discuss problems I encountered. Second, I would like to thank Afshin Vahid Belarghou for giving me advice and helping out when needed. Third, I would like to than George Dadunashvili for his advice and his help with Wolfram Mathematica. Furthermore, I would like to thank Daniel McCusker, Steven Lankhorst and the other members of the group for their company and discussions. I enjoyed our weekly team meetings immensely. Finally, I would like to thank my family for their support, and in particular my mother for tirelessly producing advice and feedback on my writing. I especially want to thank my girlfriend Rosalie for her unconditional support and comfort through both the ups and downs of this project.

In this thesis, I report on all findings that I made during this project and I describe the methods used to find them. In addition to that, I also explain the choices I made during the project and why I made them. Furthermore, I did my utmost to properly describe all that I understand about the properties of the models and methods and why they behave the way they do. This last part is meant to help future generations working on membrane mediated interactions and is what I consider to be the most valuable parts of this thesis. Now all that remains is to wish you as much enjoyment reading this thesis as I had while writing it.

Abstract

In this thesis, we investigate interactions between conical inclusions in a lipid bilayer membrane and make predictions about the patterns they form. To find these patterns, we derive an expression for the energy of a membrane as a function of the inclusion locations and search numerically for the pattern that gives minimum energy.

The energy of a membrane with conical inclusions can be derived using the point particles model with corresponding formalism developed by Dommersnes and Fournier [1]. In this thesis, we apply this formalism to the finite size particles model described by Weikl et al. [2]. We compare the results of both models for a system of three inclusions, to validate the point particles model's ability to accurately predict equilibrium patterns for conical inclusions. For most non-conical inclusions, however, the point particles model proves inadequate, leaving only the computationally intensive finite size particles model to be used for more complex inclusions.

We develop a new numerical method for finding equilibrium patterns: the gradient descent method. This method is several hundred times faster than the standard Metropolis algorithm, and gives acceptable results. For large systems of inclusions, the method is very sensitive to local minima and has difficulties merging small groups. The addition of noise in the Brownian motion method proves to be unable to resolve the local minima sensitivity, but we speculate that small bursts of high noise or grouping stable inclusions structures and moving the groups as a whole may be more effective.

Using the point particles model, we found that four-inclusion square-shaped structures and six-inclusion butterfly-shaped structures are favored in all systems with more than six inclusions.

Contents

1 Introduction

All living beings are made out of cells. Many parts of these cells contain the lipid bilayer sheets that we call membranes. In recent decades there has been an increasing interest in the properties of these membranes. As the understanding of the behavior of lipid bilayer membranes deepens, research is starting to shift towards the effect of inclusions that deform the membrane shape. One very important type of inclusion is the great variety of membrane proteins that are embedded in virtually all biological membranes. Membrane proteins are of vital importance to the cell because they mediate a multitude of essential tasks, like transport, signaling and shaping of the membrane itself. Many membrane proteins are mobile and can drift around in the fluid bilayer that is the membrane. Combined with the interactions that exist between these membrane proteins, spontaneous behavior is achieved that is often of considerable functional influence [3] as cited by [4].

The interaction between membrane proteins comes in two forms: direct interactions, that consist of van der Waals and electrostatic forces, and indirect interactions, which are mediated by the membrane. In this study we will focus on the indirect interactions. The indirect interactions are mediated by the membrane which seeks to minimize the deformations imposed by the proteins. This mechanism can result in intricate interactions involving large numbers of proteins, which can give rise to complex patterns that can be important to the functioning of a membrane protein. Examples include alamethycin channel activity which depends on clustering; clustering at the junctions between cells of gap-junction adhesion proteins; clustering of bacteriorhodopsin into trimes which in turn cluster to form a crystalline lattice; and lipid segregation that influences membrane shape [3] as cited by [4]. Because of their importance, a variety of models has been developed to predict the interactions and the pattern formation of these membrane proteins, like for example [1,2,4–6]. Direct interactions, which consist of electrostatic and van der Waals interactions between the proteins, are beyond the scope of this study.

One approach to model membrane inclusions is a purely numerical one that discretizes the membrane and uses a Metropolis algorithm to find the membrane shape and the inclusion configuration that has minimal deformations and is therefore the minimum energy state. These models can go to as much detail as atomic scale, leading to highly accurate simulations that are very computationally expensive [7–10]. As such, for most simulations, the membrane is discretized more coarsely to decrease computation time, like for example [6, 11, 12].

Alternatively, one could take a much more analytical approach by using a continuous model. In such a model, the membrane is described as a continuous surface whose shape is determined by minimizing the energy functional. The inclusions are then moved around in an attempt to find the configuration where the lowest membrane energy is achieved. In these models, the inclusions are usually modeled by imposing boundary conditions on the membrane shape [1, 2, 13]. An alternative approach is to apply a force field to the membrane that simulates the effect of an inclusion on the membrane shape [5, 14].

In this project we will be exploring two different continuous models to predict the patterns formed by the spontaneous aggregation of conical inclusions. The first model is based upon the formalism developed by Dommersnes and Fournier (2002) [1]. In this model, membrane inclusions are approximated as point particles which allows for relatively low computation times. The second model was described by Weikl et al. (1998) [2]. In this model, particles have finite size which sacrifices computation speed for better accuracy and realism.

Both models are based upon the energy functional developed by Helfrich [15], in which we will add a term for surface tension. Also, we will investigate the influence of a harmonic coupling between the membrane and a flat surface using the finite size particles model. Furthermore, we will develop a new numerical method, the gradient descent method, that speeds up simulations up several hundred times beyond the standard Metropolis algorithm simulations. The main research questions we will answer in this thesis are:
1) Is the point particles approximation proposed by Dommersnes and Fournier [1] accurate enough to predict equilibrium configurations with similar accuracy to the finite size particles model?
2) Which is the best numerical method to find the equilibrium configuration for inclusions in the point particles model?
3) What patterns will the inclusions form at equilibrium?

In chapter 2 we give detailed mathematical derivations of the membrane shape for both the point particles and the finite size particles model. In chapter 3 we discuss several numerical methods applied to the point particles model that were used to find the inclusion configuration for which the energy was minimized. In chapter 4 we first evaluate the quality of the point particles model by comparing its results for systems of 3 inclusions or less to those of the finite size particles model. We then continue chapter 4 by giving the results for larger systems obtained with the point particles model and we compare the numerical methods that were used to obtain them. Lastly, in chapter 5, we will discuss the results and gives some possible directions for future research.

2 Shape and bending energy for membranes with conical inclusions

In this chapter, we will first give a derivation of the energy functional of a membrane and a description of conical inclusions embedded in the membrane. After that, we will give a detailed derivation of the formalism used to describe the energy state of a membrane with conical inclusions for the point particles model as derived by Dommersnes and Fournier [1]. Finally we will extend this formalism to make it possible to apply it to the finite size particles model as described by Weikl et al. [2].

2.1 Bending energy of a membrane with conical inclusions

2.1.1 The energy functional of a membrane

The goal of this project is to predict the patterns that are formed by inclusions on a membrane. To do so, we need a way to predict the equilibrium shape of the the membrane as well at the equilibrium configuration of the inclusions embedded in it. The equilibrium is defined as the state with the lowest energy. Therefore, in order to predict the equilibrium, we need to be able to calculate the energy of the system for any membrane shape. In this case, the energy consists of the bending energy of the membrane. Helfrich developed an energy functional that describes this bending energy as a function of the principal curvatures of the membrane [15]. This energy functional is given by:

$$E = \int \left(\frac{\kappa}{2}(2H)^2 + \sigma \right) dS \tag{2.1}$$

where κ (Nm) is the bending modulus, H (1/m) is the mean curvature, σ (N/m) is the surface tension and dS (m^2) is an infinitely small area element. We assume that the membrane is infinitely large and flat in isolation, and that deviations from its equilibrium state can be described as a function $z = u(x, y)$ (m). For $|\nabla u| << 1$, H can be approximated by (for an example of the derivation, see [16]):

$$H \approx \frac{1}{2}(u_{xx} + u_{yy}) = \frac{1}{2}(\nabla^2 u) \tag{2.2}$$

Furthermore, expressing dS in terms of dx and dy results in:

$$dS = \sqrt{1 + u_x^2 + u_y^2}dxdy = \sqrt{1 + |\nabla u|^2}dxdy \overset{|\nabla u|<<1}{\approx} \left(1 + \frac{1}{2}|\nabla u|^2\right) dxdy \tag{2.3}$$

Substituting equations (2.2) and (2.3) into equation (2.1) then yields:

$$E = \frac{1}{2} \int \int \left(\kappa(\nabla^2 u)^2 + \sigma|\nabla u|^2 \right) dxdy \tag{2.4}$$

2.1.2 Conical inclusions on a membrane

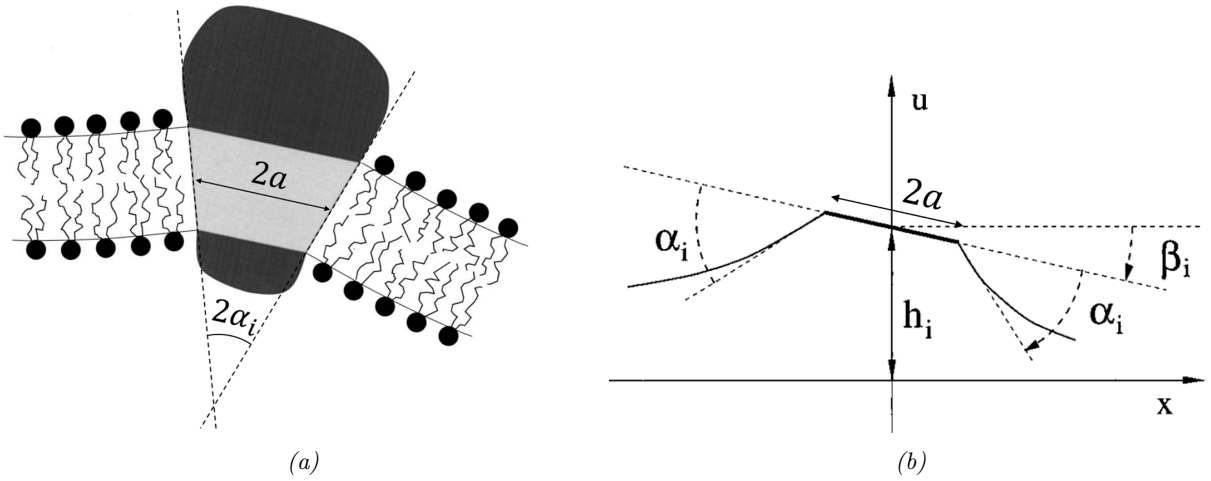
A membrane in isolation is boring and flat. Therefore, we will add inclusions to the membrane to spice things up a bit. We will be using the relatively simple conical inclusion as depicted in figure 1a. $2\alpha_i$ (-) is the angle at the point of the cone, and the radius of the inclusion a (m) is defined as the radius in the middle of the lipid bilayer membrane. The membrane sticks to the sides of the inclusion, forcing its orientation in the direction perpendicular to the tilted surface of the inclusion.

We will be using the idealized rotationally symmetric description of this inclusion as depicted in figure 1b. The thickness of the membrane is neglected, approximating it as an infinitely thin sheet. The membrane is forced to be perpendicular to the surface of the inclusion, and therefore it makes a contact angle α_i with the inclusion on all sides. The conical inclusion is assumed to be a rigid disc tilted at an angle β_i (-) with respect to the x-axis, and an angle γ_i (-) with respect to the y-axis. The center of the inclusion is at a distance h_i (m) to the $u = 0$ plane.

This inclusion has two possible orientations: point down like in figure 1, which will be called the negative orientation, and point up, which will be called the positive orientation. By definition, a negatively oriented inclusion has angle $\alpha_i < 0$ and a positively oriented inclusion has angle $\alpha_i > 0$. In the models that

we will describe in the next two sections, the main aim is to model the conical inclusions we described here. However, with some adjustments, these models can also be applied to some differently shaped inclusions.

We will be using the expression for the bending energy of a membrane given by equation (2.4). Without inclusions, a membrane will be flat at $u = 0$. In the next two sections, we will model the effect of inclusions by imposing constraints to the membrane shape u. The challenge is to derive an expression for u that minimizes the energy functional while obeying these boundary conditions. We will first present this derivation for the point particles model in section 2.2. After that we will expand the model to the more accurate finite size particles model in section 2.3.



(a) *(b)*

Figure 1: A conical inclusion with point angle $2\alpha_i$ embedded in a membrane. The inclusion is pointed downwards, which will be called the negative orientation. (a) The radius a (m) of the inclusion is defined as the radius in the middle of the lipid bilayer. Modified after [2] (b) The cone inclusion is approximated as a rotationally symmetric rigid disc and the membrane thickness is neglected. The inclusion is tilted at an angle β_i to the the x-axis and γ_i to the y-axis (perpendicular to the paper). It makes a contact angle α_i with the membrane and h_i (m) is the distance between the center of the inclusion and the $u = 0$ plane. Modified after [2].

2.2 The point particles model

In this section we will apply the formalism developed by Dommersnes and Fournier [1]. In this formalism, the inclusions are approximated as point particles ($a = 0$) which will result in an analytical expression for the minimum energy shape of the membrane given any set of inclusion locations.

2.2.1 Minimizing the energy

One may postulate that for long range interactions, conical inclusions can be approximated as point particles. That is to say, $a \approx 0$. If we do this, the curvature imposed by the inclusion needs to be described by a point constraint.

From the left to the right of an inclusion, the inclination of the membrane shifts a total of $2\alpha_i$ over a distance $2a\cos(\beta_i) \approx 2a$ for small β_i. Thus, the second derivative u_{xx} of u should integrate to $2\alpha_i$ over this interval. We can approximate this by imposing the boundary condition $u_{xx} = \alpha_i/a$ at the center of the inclusion. A similar argument can be made for all other directions, so we get $u_{yy} = \alpha_i/a$ and $u_{xy} = 0$.

If there are N inclusions on the membrane at positions $\mathbf{r}_i$ (m), the imposed curvature on the membrane can be described as the $3N$ vector:

$$\mathbf{U} = \begin{pmatrix} u_{xx}(\mathbf{r}_1) \\ u_{xy}(\mathbf{r}_1) \\ u_{yy}(\mathbf{r}_1) \\ \vdots \end{pmatrix} \tag{2.5}$$

where the three components should be repeated for $\mathbf{r}_2, \mathbf{r}_3, ..., \mathbf{r}_N$. The constraints are therefore:

$$U_\alpha = K_\alpha \tag{2.6}$$

where K_α (1/m) are the prescribed curvature components, or in the case of conical inclusions:

$$\mathbf{K} = \begin{pmatrix} \alpha_1/a \\ 0 \\ \alpha_1/a \\ \vdots \end{pmatrix} \tag{2.7}$$

To minimize the energy in equation (2.4) while obeying these boundary conditions, we introduce $3N$ Lagrange Multipliers Λ_α:

$$\begin{aligned} E^* &= \frac{1}{2}\int\int \left(\kappa(\nabla^2 u)^2 + \sigma|\nabla u|^2\right) dxdy - \Lambda_\alpha U_\alpha \\ &= \frac{1}{2}\int\int \left(\kappa(\nabla^2 u)^2 + \sigma|\nabla u|^2 - \Lambda_\alpha u D_\alpha\right) dxdy \end{aligned} \tag{2.8}$$

where summation over $\alpha = 1, 2, ..., 3N$ is implied. D_α are the elements of the vector $\mathbf{D}$ which is defined as:

$$\mathbf{D} = \begin{pmatrix} \delta_{xx}(\mathbf{r} - \mathbf{r}_1) \\ \delta_{xy}(\mathbf{r} - \mathbf{r}_1) \\ \delta_{yy}(\mathbf{r} - \mathbf{r}_1) \\ \vdots \end{pmatrix} \tag{2.9}$$

A small change in the membrane shape δu will lead to a small change δE^* in equation (2.8):

$$\begin{aligned} E^* + \delta E^* &= \frac{1}{2}\int\int \left(\kappa(\nabla^2(u+\delta u))^2 + \sigma|\nabla(u+\delta u)|^2 - \Lambda_\alpha(u+\delta u)D_\alpha\right) dxdy \\ &= E^* + \frac{1}{2}\int\int \left(2\kappa\nabla^2 u\nabla^2\delta u + 2\sigma\nabla u\cdot\nabla\delta u - \Lambda_\alpha\delta u D_\alpha\right) dxdy \end{aligned} \tag{2.10}$$

Canceling out the E^* on both sides and using integration by parts twice on the first term and once on the second term yields:

$$\begin{aligned} \delta E^* &= \int\int \left(\kappa\nabla^4 u\delta u - \sigma\nabla^2 u\delta u - \frac{1}{2}\Lambda_\alpha\delta u D_\alpha\right) dxdy \\ &= \int\int \left(\kappa\nabla^4 u - \sigma\nabla^2 u - \frac{1}{2}\Lambda_\alpha D_\alpha\right)\delta u dxdy \end{aligned} \tag{2.11}$$

In the minimum energy state, δE^* has to be zero for all changes δu. Therefore, the integrand has to be zero everywhere. This yields:

$$\kappa\nabla^4 u - \sigma\nabla^2 u = \frac{1}{2}\Lambda_\alpha D_\alpha \tag{2.12}$$

Dividing by κ, substituting $\lambda = \sqrt{\frac{\sigma}{\kappa}}$ (1/m) and absorbing the $\frac{1}{2\kappa}$ term into the Lagrange multipliers Λ_α results in:

$$(\nabla^4 - \lambda^2\nabla^2)u = \Lambda_\alpha D_\alpha \tag{2.13}$$

2.2.2 Green's function

To solve the differential equation (2.13) we follow an approach similar to the one used by Evans et al. [5]. We first derive the related Green's function:

$$(\nabla^4 - \lambda^2\nabla^2)G(\mathbf{r}) = \delta(\mathbf{r}) \tag{2.14}$$

Taking the Fourier transform of both sides yields:

$$(q^4 + \lambda^2 q^2)F\{G\} = 1 \tag{2.15}$$

and therefore

$$G(\mathbf{r}) = \frac{1}{(2\pi)^2}\int \frac{e^{i\mathbf{q}\cdot\mathbf{r}}}{q^2(q^2+\lambda^2)}d^2q \tag{2.16}$$

Using partial fraction decomposition we obtain:

$$\begin{aligned} G(\mathbf{r}) &= \frac{1}{(2\pi)^2}\int \frac{1}{\lambda^2}\left(-\frac{1}{(q^2+\lambda^2)} + \frac{1}{q^2}\right)e^{i\mathbf{q}\cdot\mathbf{r}}d^2q \\ &= \frac{1}{\lambda^2}\left(\frac{1}{(2\pi)^2}\int \frac{e^{i\mathbf{q}\cdot\mathbf{r}}}{(-q^2-\lambda^2)}d^2q - \frac{1}{(2\pi)^2}\int \frac{e^{i\mathbf{q}\cdot\mathbf{r}}}{-q^2}d^2q\right) \end{aligned} \tag{2.17}$$

Note that although the second integral diverges near $q = 0$, the related differential equation still has a solution. The divergence is caused by the Fourier transform of the Laplacian in equation (2.15), which only holds for $q \neq 0$. We can rewrite these integrals as the solutions to the related differential equations. We then get:

$$G(\mathbf{r}) = \frac{1}{\lambda^2}(G_1 - G_2) \tag{2.18}$$

where G_1 and G_2 are the solutions to the differential equations:

$$\begin{aligned} (\nabla^2 - \lambda^2)G_1(\mathbf{r}) &= \delta(\mathbf{r}) \\ \nabla^2 G_2(\mathbf{r}) &= \delta(\mathbf{r}) \end{aligned} \tag{2.19}$$

These are known to have solutions:

$$\begin{aligned} G_1(\mathbf{r}) &= -\frac{1}{2\pi}K_0(\lambda r) \\ G_2(\mathbf{r}) = \frac{1}{2\pi}\log(Br) &= \frac{1}{2\pi}\log(\lambda r) \end{aligned} \tag{2.20}$$

where $K_0(\lambda r)$ is the zeroth order modified Bessel function of the first kind. The other linearly independent solution of G_1, the modified Bessel function of the second kind $I_0(\lambda r)$, has been left out as it diverges as $r \to \infty$. Also, we have chosen $B = \lambda$ for convenience. Thus we get:

$$G(\mathbf{r}) = -\frac{1}{2\pi\lambda^2}\left(K_0(\lambda r) + \log(\lambda r)\right) \tag{2.21}$$

2.2.3 Membrane shape and energy

Using the properties of the Green's function, equation (2.13) yields:

$$u(\mathbf{r}) = \Lambda_\alpha G_\alpha(\mathbf{r}) \tag{2.22}$$

where

$$\mathbf{G}(\mathbf{r}) = \begin{pmatrix} G_{xx}(\mathbf{r}-\mathbf{r}_1) \\ G_{xy}(\mathbf{r}-\mathbf{r}_1) \\ G_{yy}(\mathbf{r}-\mathbf{r}_1) \\ \vdots \end{pmatrix} \tag{2.23}$$

To find the Lagrange multipliers, we apply the constraints imposed by the inclusions. To that end we define the vector of operators $\mathbf{L}$ as:

$$\mathbf{L} = \begin{pmatrix} \frac{\partial^2}{\partial x^2}\big|_{\mathbf{r}_1} \\ \frac{\partial^2}{\partial x \partial y}\big|_{\mathbf{r}_1} \\ \frac{\partial^2}{\partial y^2}\big|_{\mathbf{r}_1} \\ \vdots \end{pmatrix} \tag{2.24}$$

The constraints can then be written as:

$$L_\beta u = L_\beta(\Lambda_\alpha G_\alpha) = (L_\beta G_\alpha)\Lambda_\alpha = M_{\beta\alpha}\Lambda_\alpha = K_\beta \tag{2.25}$$

where the $3N \times 3N$ matrix $\mathbf{M}$ is defined by:

$$M_{\beta\alpha} = L_\beta G_\alpha \tag{2.26}$$

From equation (2.25), we can then determine the Lagrange multipliers:

$$\Lambda_\alpha = M^{-1}_{\beta\alpha} K_\beta \tag{2.27}$$

The membrane shape is then given by:

$$u(\mathbf{r}) = M^{-1}_{\beta\alpha} K_\beta G_\alpha(\mathbf{r}) \tag{2.28}$$

Given the positions of the inclusions, $\mathbf{M}$ and $\mathbf{G}(\mathbf{r})$ can be calculated. Then, using prescribed curvatures $\mathbf{K}$, the membrane shape can be obtained. Next, we want to calculate the bending energy E. Integrating equation (2.4) by parts yields:

$$E = \frac{1}{2}\int\int \left(\kappa\nabla^4 u - \sigma\nabla^2 u\right) u dx dy \tag{2.29}$$

Substituting equation (2.13) into this expression, then integrating by parts results in:

$$E = \frac{\kappa}{2}\Lambda_\alpha \int\int D_\alpha u dx dy = \frac{\kappa}{2}\Lambda_\alpha K_\alpha = \frac{\kappa}{2} M^{-1}_{\beta\alpha} K_\beta K_\alpha \tag{2.30}$$

2.2.4 The interaction matrix and the wave-vector cutoff

From the definition in equation (2.26) we derive:

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1N} \\ m_{21} & m_{22} & & m_{2N} \\ \vdots & & \ddots & \vdots \\ m_{N1} & \dots & \dots & m_{NN} \end{pmatrix} \tag{2.31}$$

where

$$m_{ij} = \begin{pmatrix} G_{xxxx}(\mathbf{r}_i-\mathbf{r}_j) & G_{xxxy}(\mathbf{r}_i-\mathbf{r}_j) & G_{xxyy}(\mathbf{r}_i-\mathbf{r}_j) \\ G_{xxxy}(\mathbf{r}_i-\mathbf{r}_j) & G_{xxyy}(\mathbf{r}_i-\mathbf{r}_j) & G_{xyyy}(\mathbf{r}_i-\mathbf{r}_j) \\ G_{xxyy}(\mathbf{r}_i-\mathbf{r}_j) & G_{xyyy}(\mathbf{r}_i-\mathbf{r}_j) & G_{yyyy}(\mathbf{r}_i-\mathbf{r}_j) \end{pmatrix} \tag{2.32}$$

m_{ij} diverges when $i = j$. To prevent this we apply a high wave-vector cutoff Λ to these parts of the matrix. Taking the fourth derivative of equation (2.16), we get:

$$G_{xxxx}(\mathbf{r}) = \int_0^{2\pi} \int_0^{\infty} \frac{1}{(2\pi)^2} \frac{q_x^4 e^{i\mathbf{q}\cdot\mathbf{r}}}{q(q^2+\lambda^2)} dq d\theta \tag{2.33}$$

Introducing the high wave-vector cutoff Λ and plugging in $\mathbf{r} = 0$ yields:

$$\begin{aligned} G_{xxxx}(0) &= \int_0^{2\pi} \int_0^{\Lambda} \frac{1}{(2\pi)^2} \frac{q_x^4}{q(q^2+\lambda^2)} dq d\theta \\ &= \int_0^{2\pi} \int_0^{\Lambda} \frac{1}{(2\pi)^2} \frac{q^4(\cos(\theta))^4}{q(q^2+\lambda^2)} dq d\theta \\ &= \frac{1}{(2\pi)^2} \int_0^{2\pi} (\cos(\theta))^4 d\theta \int_0^{\Lambda} \frac{q^3}{(q^2+\lambda^2)} dq \\ &= \frac{3}{16\pi} \int_0^{\Lambda} \frac{q^3}{(q^2+\lambda^2)} dq \\ &= \frac{3}{32\pi} \left(\Lambda^2 - \lambda^2 \log\left(1 + \frac{\Lambda^2}{\lambda^2}\right)\right) \end{aligned} \tag{2.34}$$

This works similarly for the other derivatives. We then find:

$$m_{ii} = \frac{1}{32\pi} \left(\Lambda^2 - \lambda^2 \log\left(1 + \frac{\Lambda^2}{\lambda^2}\right)\right) \begin{pmatrix} 3 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 3 \end{pmatrix} \tag{2.35}$$

2.2.5 Higher order symmetrical inclusions (monkey saddles)

To increase the variety of shapes an inclusion can impose, one might change the constraints in equation (2.6) to ones that impose higher order derivatives. For third order constraints (monkey saddles) the vector U describing the constraints (equation (2.5)) then changes to:

$$\mathbf{U} = \begin{pmatrix} u_{xxx}(\mathbf{r}_1) \\ u_{xxy}(\mathbf{r}_1) \\ u_{xyy}(\mathbf{r}_1) \\ u_{yyy}(\mathbf{r}_1) \\ \vdots \end{pmatrix} \tag{2.36}$$

where, once again,

$$U_\alpha = K_\alpha \tag{2.37}$$

The differential equation from equation (2.13) does not change, and therefore the Green's function in equation (2.21) also remains the same. The $3N \times 3N$ matrix from equation (2.31) changes to the $4N \times 4N$ matrix:

$$\mathbf{M} = \begin{pmatrix} m_{11} & m_{12} & \dots & m_{1N} \\ m_{21} & m_{22} & & m_{2N} \\ \vdots & & \ddots & \vdots \\ m_{N1} & \dots & \dots & m_{NN} \end{pmatrix} \tag{2.38}$$

where

$$m_{ij} = \begin{pmatrix} G_{xxxxxx}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxxxxy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxxxyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxxyyy}(\mathbf{r}_i - \mathbf{r}_j) \\ G_{xxxxxy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxxxyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxxyyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxyyyy}(\mathbf{r}_i - \mathbf{r}_j) \\ G_{xxxxyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxxyyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxyyyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xyyyyy}(\mathbf{r}_i - \mathbf{r}_j) \\ G_{xxxyyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xxyyyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{xyyyyy}(\mathbf{r}_i - \mathbf{r}_j) & G_{yyyyyy}(\mathbf{r}_i - \mathbf{r}_j) \end{pmatrix} \tag{2.39}$$

Once again m_{ij} diverges when $i = j$. Thus we apply a high wave vector cutoff Λ. Immediately generalizing this to the N^{th} order derivative in x and the M^{th} order derivative in y (where $N + M$ is assumed to be even) results in:

$$\begin{aligned}
G_{,NxMy}(0) &= \int_0^{2\pi}\int_0^{\Lambda} \frac{1}{(2\pi)^2}\frac{q_x^N q_y^M}{q(q^2+\lambda^2)}dqd\theta \\
&= \int_0^{2\pi}\int_0^{\Lambda} \frac{1}{(2\pi)^2}\frac{q^{N+M}(\cos(\theta))^N(\sin(\theta))^M}{q(q^2+\lambda^2)}dqd\theta \\
&= \frac{1}{(2\pi)^2}\int_0^{2\pi}(\cos(\theta))^N(\sin(\theta))^M d\theta \int_0^{\Lambda}\frac{q^{N+M-1}}{(q^2+\lambda^2)}dq \\
&= \frac{1}{(2\pi)^2}\frac{2\Gamma(\frac{N+1}{2})\Gamma(\frac{M+1}{2})}{\Gamma(\frac{N+M+2}{2})}\int_0^{\Lambda}\frac{q^{N+M-1}}{(q^2+\lambda^2)}dq \\
&= \frac{1}{(2\pi)^2}\frac{2\Gamma(\frac{N+1}{2})\Gamma(\frac{M+1}{2})}{\Gamma(\frac{N+M+2}{2})}\left(\frac{1}{2}(-\lambda^2)^{\frac{N+M}{2}-1}\log\left(1+\frac{\Lambda^2}{\lambda^2}\right)+\right. \\
&\qquad \left.\sum_{n=0}^{\frac{1}{2}N+\frac{1}{2}M-2}\frac{(-1)^n\lambda^{2n}\Lambda^{N+M-2-2n}}{N+M-2-2n}\right)
\end{aligned} \tag{2.40}$$

2.2.6 Force on the inclusions

Algorithms searching for the minimum energy state can be sped up with knowledge of the derivatives of the energy; the force. Therefore, it would be useful to derive an explicit formula for the derivatives of the energy with respect to the inclusion locations. Taking the derivative of equation (2.27) to the x coordinate of particle n yields:

$$\frac{dE}{dx_n} = \frac{1}{2}\kappa\frac{d\Lambda_\gamma}{dx_n}K_\gamma \tag{2.41}$$

Taking the derivative of equation (2.31) yields:

$$\begin{aligned}
\frac{dM_{\alpha\beta}}{dx_n}\Lambda_\beta + M_{\alpha\beta}\frac{d\Lambda_\beta}{dx_n} &= 0 \\
\frac{d\Lambda_\gamma}{dx_n} &= -M^{-1}_{\gamma\alpha}\frac{dM_{\alpha\beta}}{dx_n}\Lambda_\beta
\end{aligned} \tag{2.42}$$

Substituting this into equation (2.41) results in:

$$\begin{aligned}
\frac{dE}{dx_n} &= -\frac{1}{2}\kappa M^{-1}_{\gamma\alpha}\frac{dM_{\alpha\beta}}{dx_n}\Lambda_\beta K_\gamma \\
&= -\frac{1}{2}\kappa\left(\mathbf{M}^{-1}\frac{d\mathbf{M}}{dx_n}\mathbf{\Lambda}\right)^T\mathbf{K} \\
&= -\frac{1}{2}\kappa\mathbf{\Lambda}^T\frac{d\mathbf{M}}{dx_n}\mathbf{M}^{-1}\mathbf{K} \\
&= -\frac{1}{2}\kappa\mathbf{\Lambda}^T\frac{d\mathbf{M}}{dx_n}\mathbf{\Lambda}
\end{aligned} \tag{2.43}$$

where we used that due to the symmetry of $\mathbf{M}$, both $\mathbf{M}^{-1}$ and $\frac{d\mathbf{M}}{dx_n}$ are symmetrical. $\frac{d\mathbf{M}}{dx_n}$ can be expressed as:

$$\frac{d\mathbf{M}}{dx_n} = \begin{pmatrix} & & m^x_{1,n} & & \\ & 0 & \vdots & 0 & \\ & & m^x_{n-1,n} & & \\ m^x_{n,1} & \dots & 0 & \dots & m^x_{n,N} \\ & & m^x_{n+1,n} & & \\ & 0 & \vdots & 0 & \\ & & m^x_{N,n} & & \end{pmatrix} \tag{2.44}$$

where for $i \neq n$:

$$m^x_{in} = m^x_{ni} = \frac{dm'_{ni}}{dx_n} = \begin{pmatrix} G_{xxxxx}(\mathbf{r}_n - \mathbf{r}_i) & G_{xxxxy}(\mathbf{r}_n - \mathbf{r}_i) & G_{xxxyy}(\mathbf{r}_n - \mathbf{r}_i) \\ G_{xxxxy}(\mathbf{r}_n - \mathbf{r}_i) & G_{xxxyy}(\mathbf{r}_n - \mathbf{r}_i) & G_{xxyyy}(\mathbf{r}_n - \mathbf{r}_i) \\ G_{xxxyy}(\mathbf{r}_n - \mathbf{r}_i) & G_{xxyyy}(\mathbf{r}_n - \mathbf{r}_i) & G_{xyyyy}(\mathbf{r}_n - \mathbf{r}_i) \end{pmatrix} \tag{2.45}$$

It will only take $O(N)$ iterations to calculate $\frac{d\mathbf{M}}{dx_n}$, so once $\mathbf{\Lambda}$ has been calculated (which is necessary to calculate the energy), also calculating the force does not take a significant amount of calculation time.

2.3 The finite size particles model

In the previous section we have discussed the point particles model as described by Dommersnes and Fournier [1]. One may question the realism of this model as the point particles approximation and the wave-vector cutoff (paragraph 2.2.4) are of limited accuracy. As such, we will use the model described by Weikl et al. [2] that uses finite size particles to validate Dommersnes' model. In this section, we will follow the same approach as in section 2.2 to minimize the energy functional for the membrane shape while obeying the constraints of finite size particles as described by Weikl et al. [2].

The energy functional of equation (2.4) will be slightly modified by adding a spring term to simulate the effect of the membrane sticking to a flat surface at $u = 0$. We will explain why this is useful in paragraph 2.3.3. The energy functional becomes:

$$E = \frac{1}{2} \int \int \left(\kappa (\nabla^2 u)^2 + \sigma |\nabla u|^2 + k u^2 \right) dxdy \tag{2.46}$$

where k $\left(\mathrm{N/m^3}\right)$ is the spring constant per squared meter attaching the membrane to the surface at $u = 0$. The original energy functional is obtained by setting k to zero.

2.3.1 Boundary conditions and Lagrange multipliers

The curvature imposed by the conical particles we described in paragraph 2.1.2 can be accurately modeled by imposing a first derivative at the circle of contact between the inclusion and the membrane. In addition, the inclusions are rigid discs, so we will also impose a condition to maintain the circular shape of the inclusion. These conditions amount to the following boundary conditions [2]:

$$\begin{aligned} u|_{r_i=a} &= h_i + a\beta_i \cos(\phi_i) + a\gamma_i \sin(\phi_i) \\ \left.\frac{\delta u}{\delta r_i}\right|_{r_i=a} &= \alpha_i + \beta_i \cos(\phi_i) + \gamma_i \sin(\phi_i) \end{aligned} \tag{2.47}$$

in coordinate system (r_i, ϕ_i), which is centered at the center of the inclusion, and oriented in the same way as all other coordinate systems. Furthermore, a (m) is the inclusion radius, α_i (-) the cone angle, h_i (m) is the height of the center of the inclusion, β_i (-) is the tilt in the x direction and γ_i (-) is the tilt in the y direction (see figure 1b). Here we assumed that $\alpha_i << 1$, $\beta_i << 1$ and $\gamma_i << 1$ so that $\tan(\alpha_i) \approx \alpha_i$, $\tan(\beta_i) \approx \beta_i$ and $\tan(\gamma_i) \approx \gamma_i$.

In this problem, the membrane shape should not only minimize E while obeying these boundary conditions, but should also be minimized for h_i, γ_i and β_i. Weikl et al. [2] proposed additional boundary

conditions for zero net force and zero torque that can be used for minimizing for h_i, β_i and γ_i. However, we will be doing the last part numerically, which will make it easier to minimize the energy for these variables. The boundary conditions can be rewritten as:

$$\begin{aligned}\int\int (u(\mathbf{r}) - h_i - a\beta_i\cos(\phi_i) - a\gamma_i\sin(\phi_i))\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))dxdy = 0\\ \int\int \left(\frac{\delta u}{\delta r_i}(\mathbf{r}) - \alpha_i - \beta_i\cos(\phi_i) - \gamma_i\sin(\phi_i)\right)\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))dxdy = 0\end{aligned} \tag{2.48}$$

which has to hold for all $0 \leq \theta < 2\pi$ and where $\mathbf{s}(\theta) = \begin{pmatrix}\cos(\theta)\\ \sin(\theta)\end{pmatrix}$, $\mathbf{r} = \begin{pmatrix}x\\ y\end{pmatrix}$ and $\mathbf{r}_i = \begin{pmatrix}x_i\\ y_i\end{pmatrix}$ is the location of inclusion i. Note that ϕ_i is the angle between the line going through $\mathbf{r}$ and $\mathbf{r}_i$ and the x-axis. Thus, we can write $\cos(\phi_i) = \frac{x-x_i}{|\mathbf{r}-\mathbf{r}_i|} = \frac{x-x_i}{a}$ and $\sin(\phi_i) = \frac{y-y_i}{|\mathbf{r}-\mathbf{r}_i|} = \frac{y-y_i}{a}$, where $|\mathbf{r}-\mathbf{r}_i| = |a\mathbf{s}(\theta)| = a$ because of the delta function. Furthermore, the derivative $\frac{\delta u}{\delta r_i}(\mathbf{r})$ is in the direction away from inclusion i and can therefore be written as $\frac{\delta u}{\delta r_i}(\mathbf{r}) = \frac{\mathbf{r}-\mathbf{r}_i}{a}\cdot\nabla u(\mathbf{r})$. We then get:

$$\begin{aligned}\int\int (u(\mathbf{r}) - h_i - \beta_i(x-x_i) - \gamma_i(y-y_i))\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))dxdy = 0\\ \int\int \frac{1}{a}\left((\mathbf{r}-\mathbf{r}_i)\cdot\nabla u(\mathbf{r}) - a\alpha_i - \beta_i(x-x_i) - \gamma_i(y-y_i)\right)\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))dxdy = 0\end{aligned} \tag{2.49}$$

To minimize the energy in equation (2.46) we introduce $2N$ Lagrange multipliers $\Lambda_i^{(3)}(\theta)$ and $\Lambda_i^{(4)}(\theta)$ [1]. Adding these to the energy functional results in:

$$\begin{aligned}E^* = \frac{1}{2}\int\int \left(\kappa(\nabla^2 u)^2 + \sigma|\nabla u|^2 + ku^2\right)dxdy -\\ \int_0^{2\pi}\Lambda_i^{(3)}(\theta)\int\int \frac{1}{a}\left((\mathbf{r}-\mathbf{r}_i)\cdot\nabla u(\mathbf{r}) - a\alpha_i - \beta_i(x-x_i) - \gamma_i(y-y_i)\right)\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))dxdyd\theta -\\ \int_0^{2\pi}\Lambda_i^{(4)}(\theta)\int\int (u(\mathbf{r}) - h_i - \beta_i(x-x_i) - \gamma_i(y-y_i))\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))dxdyd\theta\end{aligned} \tag{2.50}$$

then switching the order of integration gives:

$$\begin{aligned}E^* = \frac{1}{2}\int\int \left(\kappa(\nabla^2 u)^2 + \sigma|\nabla u|^2 + ku^2\right)dxdy -\\ \int\int \frac{1}{a}\left((\mathbf{r}-\mathbf{r}_i)\cdot\nabla u(\mathbf{r}) - a\alpha_i - \beta_i(x-x_i) - \gamma_i(y-y_i)\right)\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))d\theta dxdy -\\ \int\int (u(\mathbf{r}) - h_i - \beta_i(x-x_i) - \gamma_i(y-y_i))\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i - a\mathbf{s}(\theta))d\theta dxdy\end{aligned} \tag{2.51}$$

Combining the integrals gives:

[1] There used to be a $\Lambda_i^{(1)}$ and $\Lambda_i^{(2)}$ as well, but they were deemed unnecessary and removed later on

$$
\begin{aligned}
E^* = \int\int \Bigg[& \frac{1}{2}\left(\kappa(\nabla^2 u)^2 + \sigma|\nabla u|^2 + ku^2\right) - \\
& \frac{1}{a}\left((\mathbf{r}-\mathbf{r}_i)\cdot\nabla u(\mathbf{r})\right)\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta - \\
& \frac{1}{a}\left(-a\alpha_i-\beta_i(x-x_i)-\gamma_i(y-y_i)\right)\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta - \\
& (u(\mathbf{r})-h_i-\beta_i(x-x_i)-\gamma_i(y-y_i))\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]dxdy
\end{aligned}
\tag{2.52}
$$

Integrating by parts then results in:

$$
\begin{aligned}
E^* = \int\int \Bigg[& \frac{1}{2}\left(\kappa(\nabla^2 u)^2 + \sigma|\nabla u|^2 + ku^2\right) + \\
& u(\mathbf{r})\nabla\cdot\left(\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\right) - \\
& \frac{1}{a}\left(-a\alpha_i-\beta_i(x-x_i)-\gamma_i(y-y_i)\right)\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta - \\
& (u(\mathbf{r})-h_i-\beta_i(x-x_i)-\gamma_i(y-y_i))\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]dxdy
\end{aligned}
\tag{2.53}
$$

where we have moved $\frac{1}{a}(\mathbf{r}-\mathbf{r}_i)$ into the integral and used the properties of the delta function to replace it by $\mathbf{s}(\theta)$.

2.3.2 Minimizing the energy

A small change δu in the membrane shape results in a small change δE^* in the energy:

$$
\begin{aligned}
E^*+\delta E^* = \int\int \Bigg[& \frac{1}{2}\left(\kappa(\nabla^2(u+\delta u))^2 + \sigma|\nabla(u+\delta u)|^2 + k(u+\delta u)^2\right) + \\
& (u+\delta u)\nabla\cdot\left(\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\right) - \\
& \frac{1}{a}\left(-a\alpha_i-\beta_i(x-x_i)-\gamma_i(y-y_i)\right)\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta - \\
& ((u+\delta u)-h_i-\beta_i(x-x_i)-\gamma_i(y-y_i))\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]dxdy
\end{aligned}
\tag{2.54}
$$

for small δu we may assume $\kappa(\nabla^2(u+\delta u))^2 + \sigma|\nabla(u+\delta u)|^2 + k(u+\delta u)^2 \approx \kappa(\nabla^2 u)^2 + 2\kappa\nabla^2 u\nabla^2\delta u + \sigma|\nabla u|^2 + 2\sigma\nabla u\cdot\nabla\delta u + ku^2 + 2ku\delta u$. Substituting this and then applying formula (2.53) results in:

$$
\begin{aligned}
E^*+\delta E^* = E^* + \int\int \Bigg[& \left(\kappa\nabla^2 u\nabla^2\delta u + \sigma\nabla u\cdot\nabla\delta u + ku\delta u\right) + \\
& \delta u\nabla\cdot\left(\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\right) - \\
& \delta u\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]dxdy
\end{aligned}
\tag{2.55}
$$

Using integration by parts we get:

$$\begin{aligned}\delta E^* = \int\int \Bigg[&\big(\kappa\nabla^4 u\delta u - \sigma\nabla^2 u\delta u + ku\delta u\big) + \\ \delta u\nabla\cdot\Bigg(&\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg) - \\ &\delta u\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]dxdy\end{aligned} \tag{2.56}$$

and thus:

$$\begin{aligned}\delta E^* = \int\int \Bigg[&\big(\kappa\nabla^4 u - \sigma\nabla^2 u + ku\big) + \\ \nabla\cdot\Bigg(&\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg) - \\ &\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]\delta u\ dxdy\end{aligned} \tag{2.57}$$

In the minimum energy state, δE^* has to be zero for all changes δu. Therefore, the integrand of equation (2.57) has to be zero everywhere. This yields:

$$\begin{aligned}\big(\kappa\nabla^4 u - \sigma\nabla^2 u + ku\big) = -\nabla\cdot\Bigg(&\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg) + \\ &\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\end{aligned} \tag{2.58}$$

u can then be expressed as:

$$\begin{aligned}u(\mathbf{r}') = \int G(\mathbf{r}'-\mathbf{r})\Bigg[&-\nabla\cdot\Bigg(\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg) + \\ &\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]dr^2\end{aligned} \tag{2.59}$$

where G is the Green's function of the differential equation from equation (2.58). Integrating by parts yields:

$$\begin{aligned}u(\mathbf{r}') = \int \Bigg[&-\nabla G(\mathbf{r}'-\mathbf{r})\cdot\Bigg(\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg) + \\ &G(\mathbf{r}'-\mathbf{r})\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\delta(\mathbf{r}-\mathbf{r}_i-a\mathbf{s}(\theta))d\theta\Bigg]dr^2\end{aligned} \tag{2.60}$$

where ∇ is the gradient of $\mathbf{r}$ and not $\mathbf{r}'$, so using the chain rule we get $\nabla(G(\mathbf{r}'-\mathbf{r})) = -\nabla G(\mathbf{r}'-\mathbf{r})$. Note that another minus appears due to the partial integration, making a total of three minuses with the one that was already there. Changing the order of integration and using the properties of the Dirac delta function we get:

$$u(\mathbf{r}') = \int_0^{2\pi}\Bigg[-\Lambda_i^{(3)}(\theta)\nabla G(\mathbf{r}'-\mathbf{r}_i-a\mathbf{s}(\theta))\cdot\mathbf{s}(\theta) + \Lambda_i^{(4)}(\theta)G(\mathbf{r}'-\mathbf{r}_i-a\mathbf{s}(\theta))\Bigg]d\theta \tag{2.61}$$

and thus:

$$u(\mathbf{r}') = \int_0^{2\pi} \Bigg[-\Lambda_i^{(3)}(\theta)\nabla G \cdot \mathbf{s}(\theta) + \Lambda_i^{(4)}(\theta) G \Bigg]_{(x,y)=(\mathbf{r}'-\mathbf{r}_i-a\mathbf{s}(\theta))} d\theta \tag{2.62}$$

2.3.3 Green's function

For $k = 0$ N/m^3 we get the same Green's function as the one given in equation (2.21):

$$G(\mathbf{r}) = -\frac{1}{2\pi\lambda^2}\left(K_0(\lambda r) + \log(\lambda r)\right) \tag{2.63}$$

which gives us a problem because of the $\Lambda_i^{(4)}(\theta)G$ term in equation (2.62). This results in a logarithmic term, which means that $u(\mathbf{r}')$ diverges as $\mathbf{r}' \to \infty$, unless $\Lambda_i^{(4)}(\theta)$ integrates to zero. This might cause problems because in many of the partial integration steps we assumed that $u(\mathbf{r}')$ converges to zero as $\mathbf{r}' \to \infty$. In addition, a diverging membrane is not very realistic. This problem is caused by the assumption of steady-state, which breaks down for a diverging membrane as it would take infinitely long to achieve this equilibrium membrane shape.

Several solutions have been considered and found to be inaccurate. One was to include $\int_0^{2\pi} \Lambda_i^{(4)}(\theta)d\theta = 0$ as a boundary condition for h_i, β_i and γ_i, but that would result in a suboptimal solution. Alternatively, one could try adding $\lim_{\mathbf{r}'\to\infty} u(\mathbf{r}') = 0$ as a boundary condition from the start and adding a Lagrange multiplier for it. However, as this is a boundary condition for $u(\mathbf{r}')$ and not its derivatives, it would simply add another term of the form $\Lambda_i^{(5)}(\theta)G$ in equation (2.62) that will cancel out the $\Lambda_i^{(4)}(\theta)G$ everywhere. Therefore, this will result in a problem with no solutions.

Instead, we add the spring term to our energy functional to penalize deviations from equilibrium. This serves both as an alternative and a check for the $k = 0$ case. To derive the Green's function related to equation (2.58) we have to solve:

$$(\nabla^4 - \lambda^2\nabla^2 + \bar{k}^2)G(\mathbf{r}) = \delta(\mathbf{r}) \tag{2.64}$$

where $\bar{k} = \sqrt{k/\kappa}$ (N/m^2) and $\lambda = \sqrt{\sigma/\kappa}$ (Nm^{-1}). Taking the Fourier transform of both sides yields:

$$(q^4 + \lambda^2 q^2 + \bar{k}^2)F\{G\} = 1 \tag{2.65}$$

$$G(\mathbf{r}) = \frac{1}{(2\pi)^2}\int \frac{e^{i\mathbf{q}\cdot\mathbf{r}}}{(q^2+a^2)(q^2+b^2)} d^2q \tag{2.66}$$

where

$$a^2 = \frac{\lambda^2}{2} - \sqrt{\left(\frac{\lambda^2}{2}\right)^2 - \bar{k}^2} \tag{2.67}$$

$$b^2 = \frac{\lambda^2}{2} + \sqrt{\left(\frac{\lambda^2}{2}\right)^2 - \bar{k}^2} \tag{2.68}$$

Note that this only holds for $4\bar{k}^2 < \lambda^4$. We are mostly interested in the problem for small k, so this is not a limitation in practical cases. Using partial fraction decomposition we obtain:

$$\begin{aligned} G(\mathbf{r}) &= \frac{1}{(2\pi)^2}\int \frac{1}{b^2-a^2}\left(-\frac{1}{q^2+b^2} + \frac{1}{q^2+a^2}\right) e^{i\mathbf{q}\cdot\mathbf{r}} d^2q \\ &= \frac{1}{b^2-a^2}\left(\frac{1}{(2\pi)^2}\int \frac{e^{i\mathbf{q}\cdot\mathbf{r}}}{-q^2-b^2} d^2q - \frac{1}{(2\pi)^2}\int \frac{e^{i\mathbf{q}\cdot\mathbf{r}}}{-q^2-a^2} d^2q\right) \end{aligned} \tag{2.69}$$

which can be rewritten as:

$$G(\mathbf{r}) = \frac{1}{b^2 - a^2}(G_1 - G_2) \tag{2.70}$$

where G_1 and G_2 are the solutions to the differential equations:

$$\begin{aligned}(\nabla^2 - b^2)G_1(\mathbf{r}) &= \delta(\mathbf{r}) \\ (\nabla^2 - a^2)G_2(\mathbf{r}) &= \delta(\mathbf{r})\end{aligned} \tag{2.71}$$

These are known to have solutions:

$$\begin{aligned}G_1(\mathbf{r}) &= -\frac{1}{2\pi}K_0(br) \\ G_2(\mathbf{r}) &= -\frac{1}{2\pi}K_0(ar)\end{aligned} \tag{2.72}$$

where the other linearly independent solution, $I_0(\lambda r)$, has been left out as it diverges as $r \to \infty$. Thus we get:

$$G(\mathbf{r}) = \frac{1}{2\pi(b^2 - a^2)}\left(K_0(ar) - K_0(br)\right) \tag{2.73}$$

which nicely goes to zero as $r \to \infty$.

2.3.4 Numerical application of the boundary conditions

We can find the Lagrange multipliers by applying the boundary conditions. Combining equation (2.47) and (2.62) gives:

$$\begin{aligned}u(\mathbf{r}_j + a\mathbf{s}(\phi)) &= \int_0^{2\pi} \Bigg[-\Lambda_i^{(3)}(\theta)\nabla G \cdot \mathbf{s}(\theta) + \Lambda_i^{(4)}(\theta)G \Bigg]_{(x,y)=(\mathbf{r}_j + a\mathbf{s}(\phi) - \mathbf{r}_i - a\mathbf{s}(\theta))} d\theta \\ &= h_j + a\beta_j\cos(\phi) + a\gamma_j\sin(\phi)\end{aligned} \tag{2.74}$$

and

$$\begin{aligned}\frac{\delta u}{\delta r}(\mathbf{r}_j + a\mathbf{s}(\phi)) &= \mathbf{s}(\phi)\cdot\nabla u(\mathbf{r}_j + a\mathbf{s}(\phi)) \\ &= \int_0^{2\pi} \Bigg[\mathbf{s}(\phi)\cdot\nabla\left(-\Lambda_i^{(3)}(\theta)\nabla G\cdot\mathbf{s}(\theta) + \Lambda_i^{(4)}(\theta)G\right) \Bigg]_{(x,y)=(\mathbf{r}_j + a\mathbf{s}(\phi) - \mathbf{r}_i - a\mathbf{s}(\theta))} d\theta \\ &= \alpha_j + \beta_j\cos(\phi) + \gamma_j\sin(\phi)\end{aligned} \tag{2.75}$$

which has to be solved for $\Lambda_i^{(3)}(\theta)$ and $\Lambda_i^{(4)}(\theta)$ so that both equations hold for all ϕ. The Green's function of either equation (2.63) or (2.73) can be inserted in these equations. Unfortunately, we are not able to solve this system of integral equations analytically. Therefore, we will solve it numerically.

The integrals of equations (2.74) and (2.75) can be approximated as sums over n values of θ:

$$\begin{aligned}&\frac{2\pi}{n}\sum_{p=1}^{n}\Bigg[-\Lambda_i^{(3)}\left(\frac{2p\pi - \pi}{n}\right)\nabla G\cdot\mathbf{s}\left(\frac{2p\pi-\pi}{n}\right) + \\ &\qquad \Lambda_i^{(4)}\left(\frac{2p\pi-\pi}{n}\right)G\Bigg]_{(x,y)=\left(\mathbf{r}_j + a\mathbf{s}\left(\frac{2q\pi - 3\pi/2}{n}\right) - \mathbf{r}_i - a\mathbf{s}\left(\frac{2p\pi-\pi}{n}\right)\right)} \\ &= h_j + a\beta_j\cos\left(\frac{2q\pi - 3\pi/2}{n}\right) + a\gamma_j\sin\left(\frac{2q\pi - 3\pi/2}{n}\right)\end{aligned} \tag{2.76}$$

and

$$\begin{aligned}\frac{2\pi}{n}\sum_{p=1}^{n}\Bigg[\mathbf{s}\left(\frac{2q\pi-3\pi/2}{n}\right)\cdot\nabla\Bigg(-\Lambda_i^{(3)}\left(\frac{2p\pi-\pi}{n}\right)\nabla G\cdot\mathbf{s}\left(\frac{2p\pi-\pi}{n}\right)+\\ \Lambda_i^{(4)}\left(\frac{2p\pi-\pi}{n}\right)G\Bigg)\Bigg]_{(x,y)=\left(\mathbf{r}_j+a\mathbf{s}\left(\frac{2q\pi-3\pi/2}{n}\right)-\mathbf{r}_i-a\mathbf{s}\left(\frac{2p\pi-\pi}{n}\right)\right)}\\ =\alpha_j+\beta_j\cos\left(\frac{2q\pi-3\pi/2}{n}\right)+\gamma_j\sin\left(\frac{2q\pi-3\pi/2}{n}\right)\end{aligned}\tag{2.77}$$

which must hold for all q, and where we replaced the integral by a sum and replaced all θ by $\left(\frac{2q\pi-3\pi/2}{n}\right)$ and all ϕ by $\left(\frac{2q\pi-3\pi/2}{n}\right)$.

Because of the replacement of the continuous variable θ by a set of discrete values, $\Lambda_i^{(3)}$ and $\Lambda_i^{(4)}$ have now essentially become a finite set of variables instead of functions. Therefore, it is no longer possible to find a solution that holds for all ϕ as there are only a finite set of unknowns while there is an infinite set of values of ϕ. As such, we also replaced ϕ by a set of n discrete values. Notice that the values of ϕ and θ are different. This is necessary because if $\phi=\theta$, the second derivatives of G (the first term in the sum of equation (2.77)) diverge when $\mathbf{r}_i=\mathbf{r}_j$.

Equations (2.76) and (2.77) are a set of $2nN$ equations with $2nN$ unknowns. Furthermore, all equations are linear for all unknowns.Therefore, this system can be solved by a simple matrix inversion to obtain the for $2nN$ values of $\Lambda_i^{(3)}$ and $\Lambda_i^{(4)}$. Once $\Lambda_i^{(3)}$ and $\Lambda_i^{(4)}$ have been calculated, the energy of the system can be very simply obtained by rewriting the energy functional of equation (2.46). Integrating by parts results in:

$$E=\frac{1}{2}\int\int\left(\kappa\nabla^4u-\sigma\nabla^2u+ku\right)u dxdy\tag{2.78}$$

Substituting equation (2.58) into this expression and integrating by parts results in:

$$E=\frac{1}{2}\int_0^{2\pi}\int\int\left(\left(\Lambda_i^{(3)}(\theta)\mathbf{s}(\theta)\cdot(\nabla u)(\mathbf{r}_i+a\mathbf{s}(\theta))d\theta\right)+\int_0^{2\pi}\Lambda_i^{(4)}(\theta)u(\mathbf{r}_i+a\mathbf{s}(\theta))\right)dxdyd\theta\tag{2.79}$$

Finally using the boundary conditions we get:

$$\begin{aligned}E&=\int_0^{2\pi}\Lambda_i^{(3)}(\theta)\left[\alpha_i+\beta_i\cos(\theta)+\gamma_i\sin(\theta)\right]d\theta+\int_0^{2\pi}\Lambda_i^{(4)}(\theta)\left[h_i+a\beta_i\cos(\theta)+a\gamma_i\sin(\theta)\right]d\theta\\ &\approx\frac{2\pi}{n}\sum_{p=1}^{n}\Lambda_i^{(3)}\left(\frac{2q\pi-3\pi/2}{n}\right)\left[\alpha_i+\beta_i\cos\left(\frac{2q\pi-3\pi/2}{n}\right)+\gamma_i\sin\left(\frac{2q\pi-3\pi/2}{n}\right)\right]+\\ &\frac{2\pi}{n}\sum_{p=1}^{n}\Lambda_i^{(4)}\left(\frac{2q\pi-3\pi/2}{n}\right)\left[h_i+a\beta_i\cos\left(\frac{2q\pi-3\pi/2}{n}\right)+a\gamma_i\sin\left(\frac{2q\pi-3\pi/2}{n}\right)\right]\end{aligned}\tag{2.80}$$

3 Numerical methods for finding inclusion patterns at equilibrium

Using equation (2.30) or equation (2.80), it is possible to calculate the membrane energy when given the locations and the prescribed curvatures of any number of inclusions. Our goal is to use this to find the inclusion configuration for which the energy is at its global minimum. Unfortunately, due to the high dimensionality of the problem, it is not easy to find the global minimum of the energy for systems with large numbers of inclusions. This is especially the case for the finite size particles model of equation (2.80), because calculating even one data point is already considerably computationally expensive. Therefore, we will only be applying the point particles model to systems with a large number of inclusions. In this chapter we will discuss three numerical methods that we will use to look for global minima in these systems: the Metropolis algorithm, the gradient descent method and the Brownian motion method.

3.1 Metropolis algorithm

The most common method that is applied to a high dimensional minimization problem like this one is the Metropolis algorithm. We implemented this method by moving a random inclusion in a random direction at every timestep. Then, the energy of the new state is calculated and the move is accepted if the energy decreases. If the energy increases, the move is accepted with a chance depending on the magnitude of the energy increase.

For this method, a step size of 0.1 times the size of the inclusions was used for all simulations. Higher accuracy was never necessary as we were only interested in the pattern that gives the global minimum. On the other hand, we empirically determined that the algorithm has difficulties converging to a minimum when working with lower accuracy.

We also need to choose the function relating the magnitude of energy increase to the chance of accepting the move. In most literature, like for example [1], as well as in the work of both my predecessors [17,18], this was implemented according to the standard Metropolis algorithm [19]. In this algorithm, the chance of accepting a move is given by:

$$P = e^{-\beta \Delta E} \tag{3.1}$$

where $\beta = \frac{1}{k_B T}$ (1/J), k_B (J/K) the Boltzmann constant, T (K) the temperature, and ΔE (J) is the increase in energy of the proposed move. $k_B T$ and the bending modulus κ are typically related by $\kappa = 20 k_B T$ [20].

This algorithm has several advantages. Firstly, it is a rather accurate reflection of reality because it mimics the Boltzmann distribution. Secondly, this algorithm is step size independent: an energy increasing move from A to B has an equal chance of being accepted whether it is taken in one step, or in two smaller steps (assuming both these steps are energy increasing). Thirdly, it can be applied to any system that is either bounded or has diverging energy as it approaches infinity. Unfortunately, our system meets neither of those criteria. However, this can be simply solved by introducing either periodic or random boundary conditions, i.e. draw a box around the inclusions and reintroduce any inclusion that leaves the box either on the opposite side or at a random location along the edge. Therefore, this algorithm can be quite effectively applied to our problem, as long as the global minimum does not involve one of the inclusions being ejected to infinity.

Yet, despite these advantages of the Metropolis algorithm we will not be using it. After all, my predecessor, Rachel Los [17], has already applied the Metropolis algorithm to this problem, and found that it is incapable of solving the system with 10 inclusions within 10^7 time steps. As such, there is no point in retrying this using the same method. So instead, we will be using the following formula:

$$P = \frac{E_0 - E_{\min}}{50\Delta E + E_0 - E_{\min}} \frac{n_{\text{Rej}}}{n_{\text{Acc}}} \tag{3.2}$$

where $E_0 - E_{\min}$ is the difference between the initial energy E_0 (J) and the lowest energy $E_{\min}$ (J) obtained so far, ΔE (J) is the increase in energy of the proposed move, and n_{Rej} (-) and n_{Acc} (-) are

respectively the number of rejected and accepted energy increasing moves so far.

An advantage of using this formula is that the chance of accepting a move is dependent on the rejection/acceptance ratio of energy increasing moves so far. This prevents the simulation from getting stuck in local minima. In addition, due to dependency on $E_0 - E_{\min}$, the simulation will work well even without a priori knowledge of the magnitude of the variations in the energy for different inclusion configurations. Furthermore, small increases in energy are more heavily penalized than in the Metropolis algorithm. As a result, it is no longer necessary to keep the particles contained using boundary conditions; the inclusions will not diverge unless their minimum energy state is at infinity.

The downside of this algorithm is that it is not step size independent as it is not exponential. This means that we cannot simply decrease the step size to get the same results with higher accuracy at the cost of longer calculation time. However, since we are keeping the step size fixed at 0.1 anyway, this will not be an issue. In addition, a non-exponential algorithm like equation (3.2) has much more difficulty overcoming wide energy barriers, although it is better at overcoming narrow ones.

3.2 Gradient descent method

The Metropolis algorithm does not make use of the force on the inclusions, even though they can be simply calculated (see paragraph 2.2.6). Therefore, we will develop a new method that does use this force. Finding an equilibrium for a system of N inclusions on a membrane is equivalent to finding a root for all $2N$ equations of the force on these inclusions. Unfortunately, most root finding algorithms rely either on knowledge of the topology of the function or the derivative, both of which are hard to come by for more than a few inclusions.

Thus, instead of an efficient, effective, fancy method we will use the very crude gradient descent method. At every timestep every inclusion will simply move one stepsize in the direction of the force. Note that all inclusions move the same distance every timestep. We have tried to scale the stepsize of each inclusion with the magnitude of the force it experiences. However, inclusions oscillating near an equilibrium experience much greater force than inclusions that are some distance away from other inclusions. This results in isolated inclusions moving towards the other inclusions very slowly, greatly increasing computation time.

At some point, the algorithm will be approaching equilibrium, after which it should decrease the stepsize. We have tried two different methods for determining whether the algorithm was oscillating near an equilibrium. The first one tests whether all of the inclusions moved in opposite directions two sequential steps ($\mathbf{F}_n^{(i)} \cdot \mathbf{F}_{n-1}^{(i)} < 0$) at least 60 times during 100 timesteps. When this happens, the step size is halved and the process repeats. This continues until the stepsize is below a certain threshold stepsize (which will be the accuracy of the result).

The second method, instead of looking at the characteristic oscillation of inclusions near equilibrium, looks at whether there has been significant movement during the past 100 timetsteps. That is to say, it checks whether any inclusion has moved more than 2 stepsizes during these 100 timesteps. We have empirically determined that the oscillation based method works slightly better.

The gradient descent method has the advantage that it very quickly converges towards a minimum. However, there is no guarantee that this will be the global minimum, as it would similarly get stuck in any stable local minimum. Therefore, this method only works well when there are no local minima in the system. Even with local minima, a good estimation of the equilibrium can still be obtained by running the algorithm repeatedly for many different initial configurations and looking for the lowest minimum that can be found this way.

3.3 Brownian Motion method

In order to reduce the local minimum sensitivity of the gradient descent method, noise can be added to the system. This will effectively make it a Brownian Motion simulation. For this problem, the Brownian

Motion method will be implemented by simply adding a noise term to the movement of the inclusions at every timestep of the gradient descent method. The displacement $\Delta \mathbf{r}_t^{(i)}$ (m) of particle i at timestep t is given by:

$$\Delta \mathbf{r}_t^{(i)} = ds \cdot \frac{\mathbf{F}^{(i)}}{|\mathbf{F}^{(i)}|} + ds \cdot q\mathbf{N} \tag{3.3}$$

where ds (m) is the stepsize, $\frac{\mathbf{F}^{(i)}}{|\mathbf{F}^{(i)}|}$ (-) is the normalized force on inclusion i, $\mathbf{N} =$ (-) is a vector of two random variables uniformly distributed between 0 and 1, and q (-) is a constant that determines the magnitude of the noise.

When the algorithm determines that it is close to an equilibrium, both the stepsize ds and the noise q are halved. As a result the magnitude of $ds \cdot q\mathbf{N}$ decreases by a factor four. The initial magnitude of the noise will be labeled q_0 (-).

4 Results

Armed with the theory described in the previous two chapters, we can now set out to obtain actual results. We programmed an implementation of all three numerical methods of the point particles model in Matlab and the finite size model in Mathematica. Unless otherwise mentioned, all results use the following values for the variables: cone angle $\alpha_i = \pm\frac{\pi}{12}$ for all inclusions, $\lambda a = 0.4$, high wave-vector cutoff $\Lambda a = 4$ (when using the point particles model) and $\bar{k}a^2 = 0.02$ (when it is not 0).

4.1 Validation of the models using results for small systems

4.1.1 Membrane shape

An obvious sanity check for the formalism described in chapter 2 is to plot the membrane shape and see if it makes sense. In the point particles model, we can use equation (2.28) to plot this membrane shape for any configuration of inclusions. For a single negatively oriented inclusion (i.e. point down, $\alpha < 0$) the membrane shape has been plotted in figure 2a. Strikingly enough, near 0 (the location of the inclusion) the membrane diverges, so obviously the second derivatives of u diverge as well, meaning that there is no way it can meet the boundary conditions of equation (2.6).

So how is it possible that we used the boundary conditions to solve for Λ in equation (2.27)? In fact, we did encounter a divergence in paragraph 2.2.4. We solved this problem by applying a high wave-vector cutoff to the self-interaction term. This essentially means that every inclusion exists on its own unique membrane where its own influence on the membrane has received this wave-vector cutoff while all others have not. This unique membrane is the only membrane where the inclusion meets its boundary conditions. In figure 2b we plotted the membrane shape for a single inclusion where the high wave-vector cutoff has been applied. The figure shows that the membrane no longer diverges at 0 but instead has a nice peak with a well-defined second derivative. This membrane does meet the boundary conditions, but it is actually not a solution to the differential equation of equation (2.13).

As a result of the high wave-vector cutoff, this membrane shows some strange oscillations further away from the center (see figure 2b). This is caused by the low frequencies of the membrane shape, whose oscillations are normally compensated by the high frequencies that have been removed by the cutoff. Another way to understand this is as follows: the application of the high wave vector cutoff is equivalent to applying a low pass filter. In other words, the membrane shape is convoluted with a sinc function. As a result, the oscillations in the sinc function are carried over into the membrane shape.

So, the model actually describes two different membrane shapes. The first one, depicted in figure 2a, is a solution to equation (2.13), but it does not meet the boundary conditions. The second one, depicted in figure 2b, is not a solution to equation (2.13), but it does meet the boundary conditions. Unfortunately, equation (2.30), that we use to calculate the membrane energy, is derived assuming that the membrane shape is both a solution to equation (2.13) and meets the boundary conditions.

Similarly, we can plot the membrane shape for the finite size particles model using equation (2.62). Figure 2c shows the membrane for a single negatively oriented inclusion for $\bar{k}a^2 = 0$ and similarly in figure 2d for $ka^2 = 0.02$. Notice that the shape of both plots is almost identical, but that all values of u are lower for the system with $ka^2 = 0.02$. This is to be expected, since the harmonic coupling makes it more difficult for the membrane to leave the $u = 0$ plane.

Comparing figure 2a and 2b with figure 2c and 2d, we notice that close to $x = y = 0$, the finite size model resembles the membrane with high wave-vector cutoff of figure 2b. On the other hand, further away from the origin, the finite size model resembles the membrane without cutoff of figure 2a more. This is quite interesting as it gives a hint that perhaps using the wave-vector cutoff of paragraph 2.2.4 will not give us problems at the end. After all, the point particles model basically assumes that the membrane looks like figure 2b near an inclusion while it looks like figure 2a far away from one. This assumption has now been confirmed by the finite size model through the plots of figure 2c and 2d.

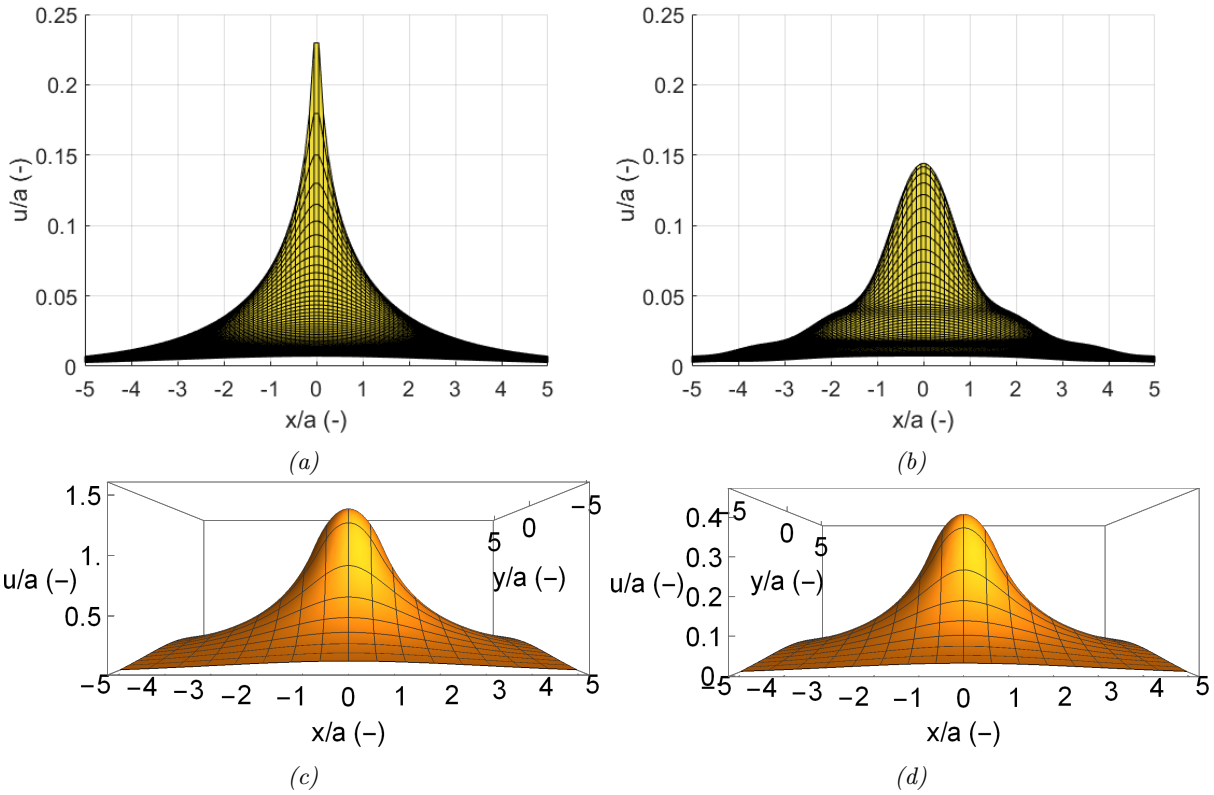


Figure 2: Side view of the membrane shape with one conical inclusion with $\lambda a = 0.4$. All plots are rotationally symmetric. Note that the z-axis is not on the same scale as the other axes. (a) and (b) were calculated using the point particles model, (a) without a wave-vector cutoff and (b) with a wave-vector cutoff. (a) is a solution to equation (2.13), but it does not meet the boundary conditions as it diverges at $x = y = 0$. (b) does meet the boundary conditions, but it is not a solution of equation (2.13). (c) and (d) were calculated using the finite size particles model with $\lambda a = 0.4$ and (c) $\bar{k}a^2 = 0$ and (d) $\bar{k}a^2 = 0.02$.

4.1.2 Two conical inclusions

Figure 3 shows four graphs that all show the membrane energy as a function of the distance between two inclusions on a membrane. We used the finite size particles model for figure 3a and figure 3d and the point particles model for figure 3b and figure 3e. Results of both models were plotted together in figure 3c.

Figure 3a shows the finite size particles model plot of the energy as a function of the distance between two inclusions for two positively oriented inclusions, for one positively and one negatively oriented inclusion, and for $\bar{k}a^2 = 0$ and $\bar{k}a^2 = 0.02$. All four curves diverge at $R/a = 2$ where the inclusions start touching.

For the systems with two identical inclusions, the graphs appear to be strictly decreasing, indicating a purely repulsive interaction. Upon closer scrutiny, we discover that this is not the case for one of them. In figure 3d, we made a rescaling of the y-axis for the system with two identical inclusions and $\bar{k}a^2 = 0.02$. Clearly, this system actually does have a very shallow minimum at $R/a = 26.4$ after which the energy starts increasing for increasing distance. This can be understood as follows. Two identical inclusions deform the membrane in the same direction. If these membrane deviations overlap, the total deviation from zero is reduced. Because of the harmonic coupling to the surface $u = 0$, the reduced total deformation becomes energetically favorable. This leads to an attractive force for large values of R/a until the curvature starts dominating for smaller values of R/a.

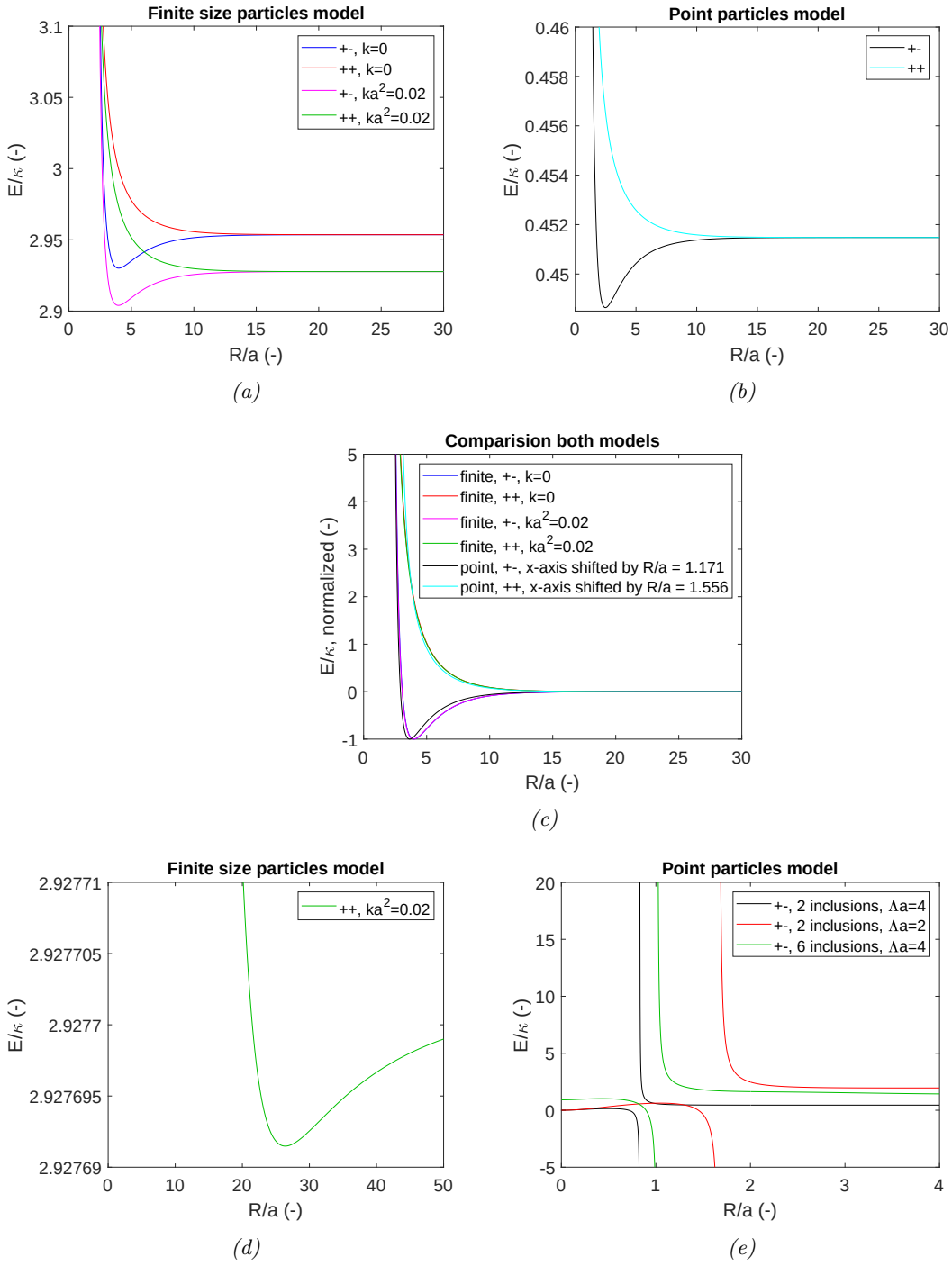


Figure 3: Membrane energy as a function of the distance between two inclusions on a membrane. ++ marks systems with two positive (point up), and +- marks systems with one positive and one negative inclusion. All curves in (a) diverge at $R/a = 2$ while the curves in (b) diverge at $R/a = 0.829$ for +- and $R/a = 0.444$ for ++. To compare the shapes of the curves, in figure (c) we shifted the $x-axis$ for the curves in (b) so that their location of divergence is at $R/a = 2$. Next we normalized the energy by shifting and rescaling the y-axis for the curves of (a) and (b), so that they go to zero at infinity and that the minimum of the +- curves is at $y = -1$. All curves with two identical inclusions (++) are strictly decreasing, except $\bar{k}a^2 = 0.02$ in (a), which has a very shallow minimum at $R/a = 26.4$. This becomes apparent after rescaling the y-axis as depicted in (d). The black line in (c) and (e) is a system with a positive in inclusion at $(-R/2a, 0)$ and a negative inclusion at $(R/2a, 0)$ and diverges at $R/a = 0.829$. The red line is the same system, but with $\Lambda a = 2$, it diverges at $R/a = 1.67$. The green line is the same system in the presence of 4 other inclusions located at (0.9,1.2), (-0.9,1.2), (0.9,-1.2) and (-0.9,-1.2) and it diverges at $R/a = 1.02$.

Figure 3b shows the point particles model plot of the energy as a function of the distance between two inclusions for two positively oriented inclusions and for one positively and one negatively oriented inclusion. For one positive and one negative inclusion, the energy diverges at $R/a = 0.829$, while for two positive inclusions, the energy diverges at $R/a = 0.444$.

To compare the shapes of the curves in figure 3a and figure 3b, we plotted them together in figure 3c. In this plot, the two curves of the point particles model were shifted so that their location of divergence was at $R/a = 2$. Next, we normalized the energy. First, we shifted the y-axis of all six curves so that they converge to zero at infinity. Next, we rescaled the y-axis so that the minimum of the +- curves was at $y = -1$. Note that the two curves from figure 3a overlap so well that they are not distinguishable. As shown by the figure, the shape of the point particles curves are a good match for the curves obtained using the finite size particles model. In addition, they also agree well with the findings of many other researchers, see for example [1, 2, 4, 5].

These results highlight an important limitation of the point particles model: distances are not well defined. The location of divergence is usually interpreted as the diameter of the particles. However, in the point particles model, the location of divergence is different for a system with two positive inclusions and for a system with one negative and one positive inclusion.

In the point particles model, inclusions do not have a radius and the definition of the length scale is therefore reliant on the high wave-vector cutoff Λ. As expected, the location of divergence is mostly linearly dependent on $1/\Lambda$. This is demonstrated by the red and black lines in figure 3e where halving Λ results in doubling of the location of divergence. Therefore, it seems like we can properly renormalize the length scale by choosing the value of Λ. However, the fact that two positive inclusions have a different location of divergence than a positive and a negative inclusion makes this normalization difficult. Furthermore, things become even more problematic as we discover that the location of divergence changes when we put it in the presence of other inclusions. The green line in figure 3e depicts the membrane energy as a function of the distance between two inclusions when there are four other inclusions present. Clearly, the location of divergence has shifted to the right.

4.1.3 Three conical inclusions

When two positively oriented inclusions are placed on a membrane together with a negatively oriented inclusion, they do not form a straight line as one might expect. Instead, they form a trianglular formation depicted in figure 4. This somewhat counter intuitive result is a consequence of the fact that membrane mediated interactions do not add up linearly, i.e. the interaction between two inclusions changes in the presence of a third inclusion.

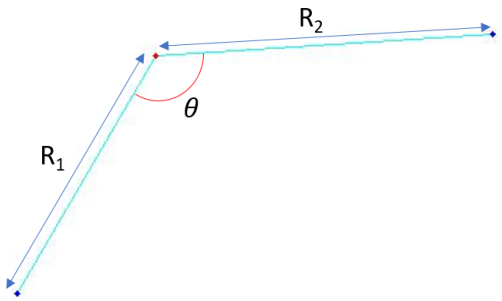


Figure 4: Schematic drawing of the minimum energy configuration for 3 inclusions where one inclusion (middle) has a negative orientation (point down) and the other two (sides) have a positive orientation. For most purposes we assume $R_1 = R_2 \equiv R$

To demonstrate this, we have plotted the energy landscape for three inclusions calculated using the point particles model in figure 5e,f, the finite size particles model and $\bar{k}a^2 = 0$ in figure 5a,b and the finite size particles model and $\bar{k}a^2 = 0.02$ in figure 5c,d. All plots show a similar shape, and strikingly, the finite particle model predicts an energy minimum at $(R/a, \theta) = (3.9, 1.88)$ for both $\bar{k}a^2 = 0$ and $\bar{k}a^2 = 0.02$. On the other hand, the point particle model predicts a minimum at $(R/a, \theta) = (2.33, 1.90)$. The angles θ match really well for both models, but the distance R/a at equilibrium differs a bit more due to the faulty length scale of the point particles model, as explained in paragraph 4.1.2.

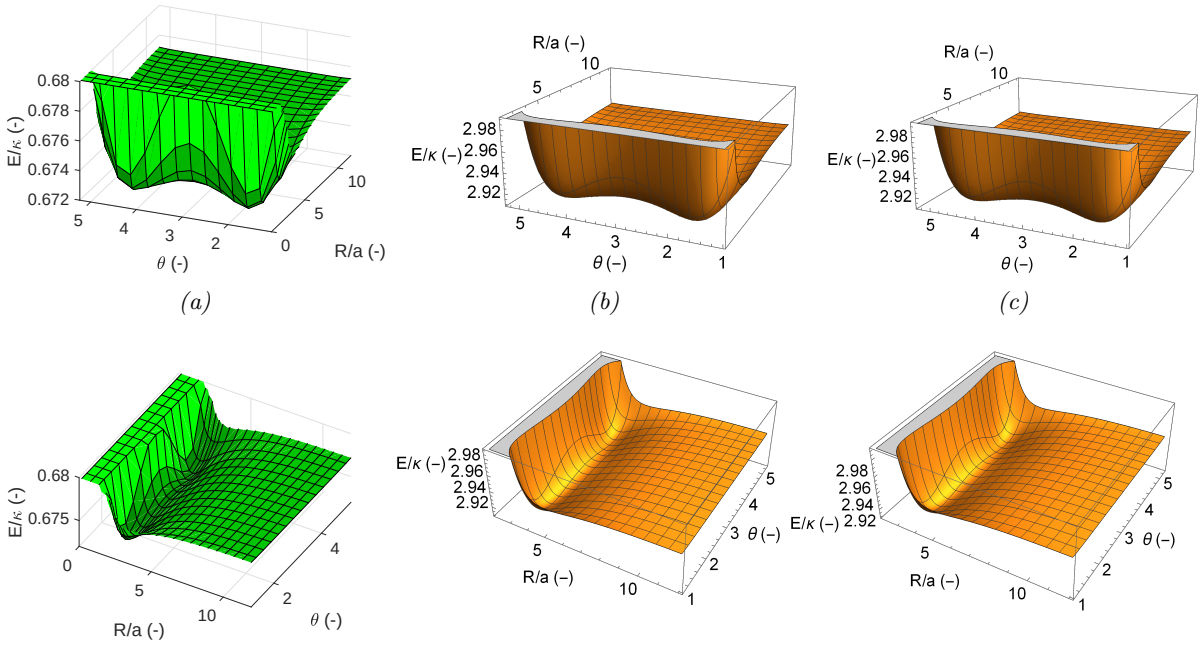


Figure 5: Two different views of the membrane energy plot for two identical and one opposite inclusion as a function of the distance $R = R_1 = R_2$ and the angle θ as depicted in figure 4. Here we used $\lambda a = 0.4$ and (a),(d) the point particles model, (b),(e) the finite size particles model with $\bar{k}a^2 = 0$ and (c),(f) the finite size particles model with $\bar{k}a^2 = 0.02$. For the point particle model, the global minimum energy is at $R/a = 2.33$ and $\theta = 1.90$. For both the finite size particle models, the minimum energy is at $R/a = 3.9$ and $\theta = 1.88$.

We can calculate the location of the equilibrium for different values of λ. In figure 6a and figure 6b we plotted the angle θ and the distance R at equilibrium as calculated using 3 different numerical methods in the point particles model. The Metropolis algorithm and Gradient descent method are as described in chapter 3, and the $R-\theta$ grid method looks for the minimum by simply calculating the energy for a large grid of $R = R1 = R2$ and θ values. The kink at approximately $\lambda = 0.5$ is the result of the inclusions being kept at a minimum distance of $R/a = 2$ to prevent them from passing the point where the energy diverges. At $\lambda = 0.5$ the inclusions touch, after which they start pushing against each other, changing their behavior. Note that the inclusions were not kept apart for the gradient descent method as this was not necessary.

The Gradient descent method seems to perform better as it is more stable for extreme values of λ. This is because numerical errors are less of a problem when looking for the location where the force is zero than when trying to calculate the difference between two values of the energy. After all, the energy is usually several orders of magnitude higher than its difference between two neighboring points, thus resulting in large errors when subtracting.

Figure 6c and figure 6d show the minimum energy configuration for the finite size model for both $\bar{k} = 0$ and $\bar{k} = \lambda^2/8$. These values were obtained using a Metropolis algorithm with zero temperature, i.e. reject all moves that increase the energy. This works well, because we already know from figure 5 that the energy has no local minima and smoothly leads to the global minimum. The minimum energy configuration for the point particles model obtained with the gradient descent method has been added for reference.

The curves for $\bar{k} = 0$ and $\bar{k} = \lambda^2/8$ of the finite size model almost perfectly overlap, meaning that the harmonic coupling has little or no impact on the equilibrium state. In addition, the angle θ at equilibrium is very close to what was obtained with the point particles model. Also, the distance R for the finite size model is consistently around a factor 1.416 above the point particles model, until it flattens out

near $R/a = 2$. One can try to renormalize the point particles model by multiplying all distances by this factor of 1.416. Unfortunately, we saw in figure 3a and 3a that the location of divergence in the point particles model is a factor $2/0.829 = 2.41$ or $2/0.444 = 4.50$ away from that of the finite size particles model. Therefore, the renormalization is limited in its ability to align definitions of distance in both models.

Even though the point particles model has some problems with both the prediction of the membrane shape and the energy curves of two inclusion systems as we demonstrated in paragraphs 4.1.1 and 4.1.2, the results for three inclusions are a very good match for those of the finite size particles model. As such, we can conclude that the point particles model should be able to accurately predict the equilibrium configurations of inclusions on a membrane. In the end, what matters is not that the membrane shape is correctly predicted, or that the energy curves have the shape we want, but only that the equilibrium configuration is predicted correctly. Therefore, from this point on, all results will be produced using the point particles model.

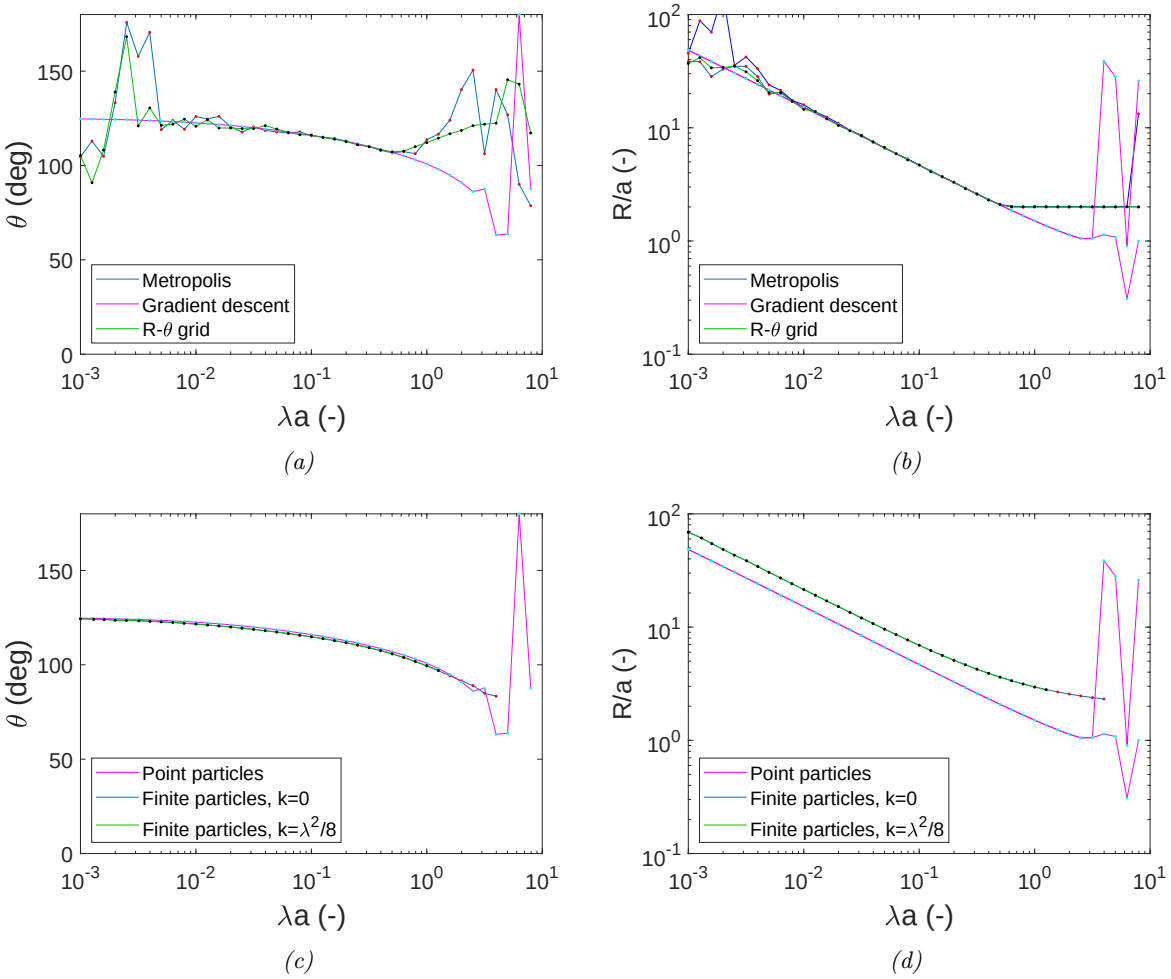


Figure 6: The minimum energy configuration for two identical inclusions and one opposite inclusion on a membrane. Notice the logarithmic scales. See figure 4 for an explanation of θ and $R = R_1 = R_2$. (a) and (b) depict the results for three different numerical methods of finding the minimum energy using the point particles model. (c) and (d) show the results for the finite size particles model for $\bar{k} = 0$ and $\bar{k} = \lambda^2/8$ together with the gradient descent method for the point particles model. The finite size particle model was plagued by large numerical errors for higher λ than plotted here, producing no useful results.

4.2 Systems with one negative and multiple positive inclusions

In paragraph 4.1.3, we saw that 3 inclusions on a membrane have a rather interesting behavior: they form triangular structures instead of the straight line we had expected. In this section, we will use the point particles model to look at some similar systems, to see how they behave. First, we will vary the curvature imposed by the negatively oriented inclusion while the two positively oriented inclusions remain the same. Next, we will add more positive inclusions to the system to see how it behaves with one negatively and three, four or five positively oriented inclusions.

4.2.1 Four conical inclusions imposing different curvatures

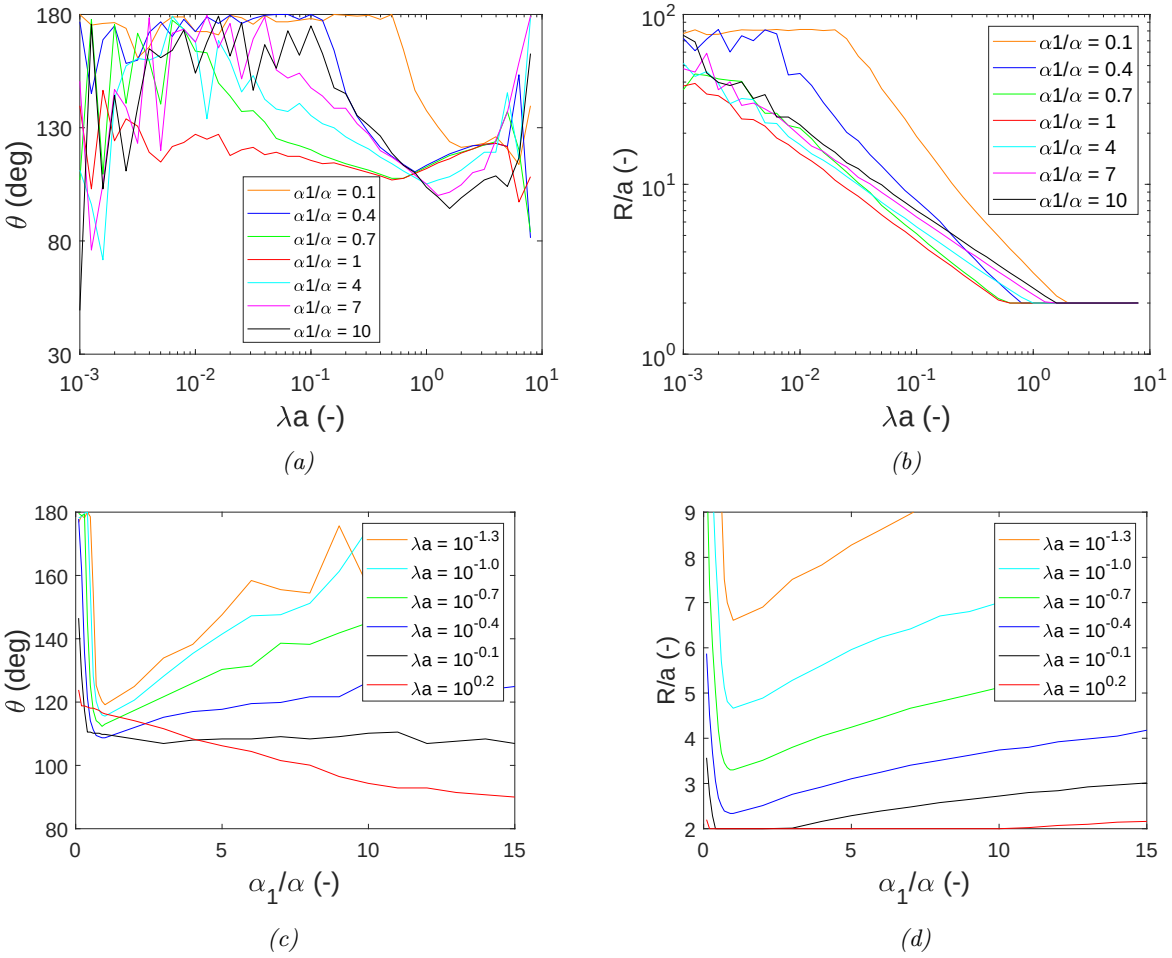


Figure 7: Angle θ and distance R/a at equilibrium for three inclusions (see figure 4) as a function of $\lambda a = a\sqrt{\sigma/\kappa}$ and relative curvature α_1/α (see figure 1) imposed by the negatively oriented inclusion.

By changing some of the values of the imposed curvatures α_i of the curvature vector $\mathbf{K}$ of equation (2.7), we can decrease or increase the curvature imposed by some inclusions. For the same system with three inclusions described in paragraph 4.1.3, we varied the curvature α_1 of the one negatively oriented inclusion while keeping the curvature of the two positive oriented inclusions fixed at $\alpha = \pi/12$. We used the Metropolis algorithm to find the equilibrium configuration as a function of λa for seven different values of α_1/α, as shown in figures 7a and 7b.

Most remarkable is that the $\alpha_1/\alpha = 1$ curve is consistently at the bottom for both the equilibrium values of θ and R/a. The inclusions move further apart both for increasing and for decreasing values of α_1. In addition, the angle between the inclusions also increases. This means that not only do the pos-

itive particles drift further away from the negative inclusion, they also drift further away from each other.

To further study this phenomenon, we also plotted both the angle θ and the distance R/a as functions of α_1/α in figures 7c and 7d. As expected, the minimum of both θ and R/a is consistently at $\alpha_1/\alpha = 1$, as long as λa is low enough that the inclusions do not touch.

4.2.2 Four,five and six conical inclusions

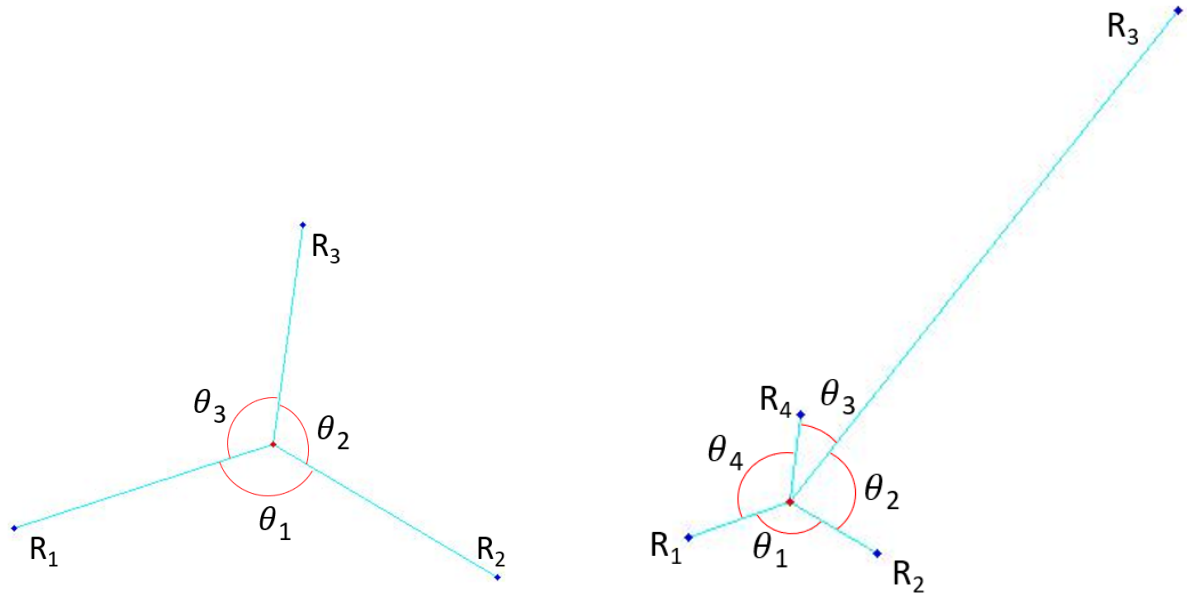


Figure 8: Schematic drawing of the minimum energy configuration for (a) 4 inclusions and (b) 5 inclusions where one inclusion (middle) has a negative orientation (point down) and the others (sides) have a positive orientation

Instead of changing the curvature of the negative inclusion, we can also introduce more positively oriented inclusions to the system. A schematic for the system with four and five inclusions is shown in figures 8a and 8b respectively. Definitions of distances and angles are similar for a system with six or more inclusions.

Figure 9 shows the angles θ_i and distances R_i at equilibrium for four, five and six inclusions, as calculated using the Metropolis algorithm. All three systems eject all positive inclusions except for two for $\lambda a < 10^{-1.4} \approx 0.0398$, after which all particles suddenly get close to the negative inclusion. Furthermore, at this point, one of the angles is very large, meaning that all positive inclusions are very close to each other, at one side of the negative inclusion. This continues until approximately $\lambda a = 10^{-0.6} \approx 0.251$. At this point, the positive inclusions start to be distributed more evenly around the negative inclusion. Finally, at approximately $\lambda a = 10^{0.1} \approx 1.25$, the positive inclusions are almost completely evenly distributed around the negative one. At this point, the positive inclusions have started touching and are pushing against the negative inclusion, thus changing the behavior of the system. This is a result of the point particles model's necessity to keep the inclusions apart and is probably not a reflection of reality.

The main conclusions we can draw from this is that we can distinguish three types of inclusion patterns, depending on the value of λa:

1. For low values of λa, the negative inclusion will keep only 2 positive inclusions close while ejecting all others.

2. For medium values of λa, all positive inclusions are close to the negative one, and are surprisingly all on the same side of the negative inclusion.

3. For high values of λa, the positive inclusions touch the negative inclusion, resulting in them surrounding the negative inclusion evenly.

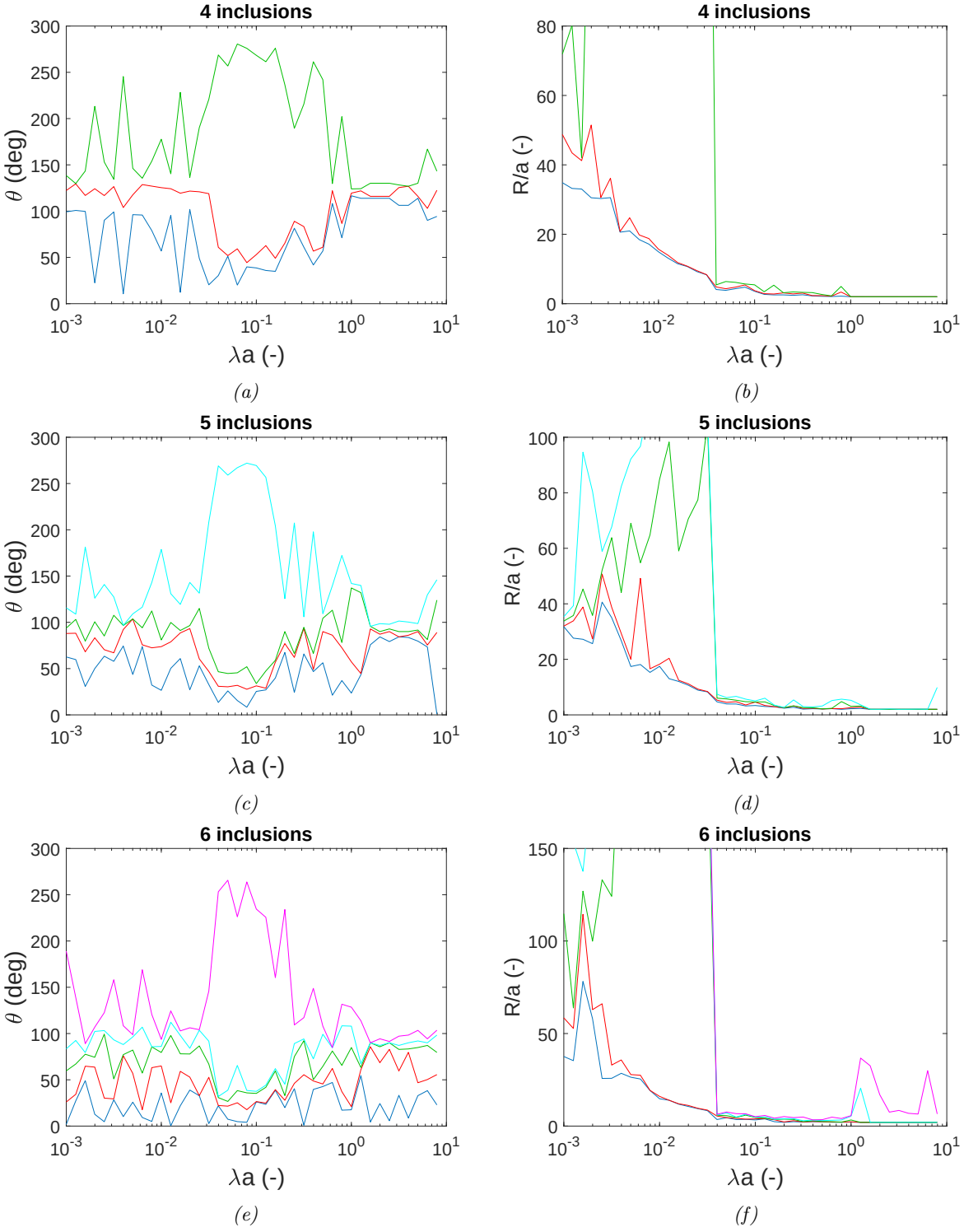


Figure 9: Angles θ_i and distance R_i/a at equilibrium for four, five and six inclusions (see figure 8) as a function of λa.

4.3 Differently shaped inclusions

So far, we have only looked at cone shaped inclusions. Unfortunately, not all inclusions are cone shaped, in fact, most are not. As such, it would be useful if our models were capable of describing a greater variety of inclusion shapes. In this section, we will discuss two kinds of non-conical inclusions in the point particles model: inclusions with 180 degrees rotational symmetry and ones with higher orders of rotational symmetry.

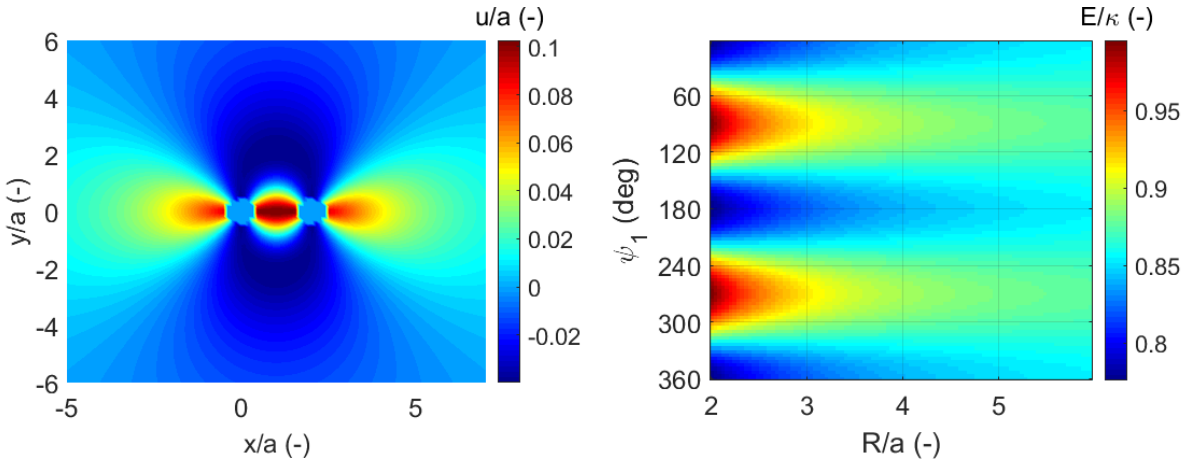
4.3.1 180 degrees rotational symmetry

The easiest way to manipulate the shape of the inclusions in the point particles model is by changing the vector of prescribed curvature $\mathbf{K}$ of equation (2.7). Here we will describe saddle points which can be modeled by:

$$\mathbf{K} = \begin{pmatrix} \alpha_1/a * \cos(\psi_1) \\ \alpha_1/a * \sin(\psi_1) \\ -\alpha_1/a * \cos(\psi_1) \\ \vdots \end{pmatrix} \tag{4.1}$$

where ψ_i is the counterclockwise rotation in the xy-plane of inclusion i. ψ_i is defined so that when $\psi_i = 0$, a peak of the saddle shape (hill) is aligned with the x-axis. A membrane plot for two inclusions with $\psi_1 = \psi_2 = 0$ is shown in figure 10a. Both inclusions have orientation $\psi_1 = \psi_2 = 0$ and therefore there is a hill for both directions along the x-axis while there is a valley (lowest point of the saddle shape) for both directions along the y-axis. For increasing ψ_i the hill will rotate counter clockwise in the direction of the y-axis. When $\psi_i = \pi/2$ the hill will be on the y-axis.

Figure 10b shows a plot of the energy as a function of the orientation ψ_1 of the second inclusion and the distance R/a between the inclusions ($\psi_2 = 0$). The minimum energy state for the system is when $\psi_1 = 0$, at which point two hills are aligned (see figure 10a).



(a) *(b)*

Figure 10: Two saddle shaped inclusions on a membrane. (a) Plot of the membrane shape for the case $\psi_1 = \psi_2 = 0$ and $R/a = 2$. Note that two hills are aligned. (b) Energy for distance R between the two inclusions and rotation angle ψ_1 of the first saddle point (the rotation ψ_2 of the second saddle point is fixed at 0). Notice that due to the symmetry of the saddle points, $\psi_1 = 0$ is the same as $\psi_1 = 180$.

4.3.2 Higher orders of rotational symmetry

In paragraph 2.2.5, we have shown how the point particles model can be extended to allow higher order derivatives to act as constraints to model inclusions. We will apply this to both inclusions with 120

degree rotational symmetry, described by third order constraints, and to inclusions with 90 degrees of rotational symmetry, described by fourth order constraints. These are essentially the third and fourth order analogues of the saddle shaped inclusions described in the previous paragraph. The third order shapes are called monkey saddles and we will call the fourth order shape fourth order monkey saddles. The vector of prescribed curvature for the third order monkey saddles that models these inclusions is given by:

$$\mathbf{K} = \begin{pmatrix} \alpha_1/a * \cos(\psi_1) \\ \alpha_1/a * \sin(\psi_1) \\ -\alpha_1/a * \cos(\psi_1) \\ -\alpha_1/a * \sin(\psi_1) \\ \vdots \end{pmatrix} \tag{4.2}$$

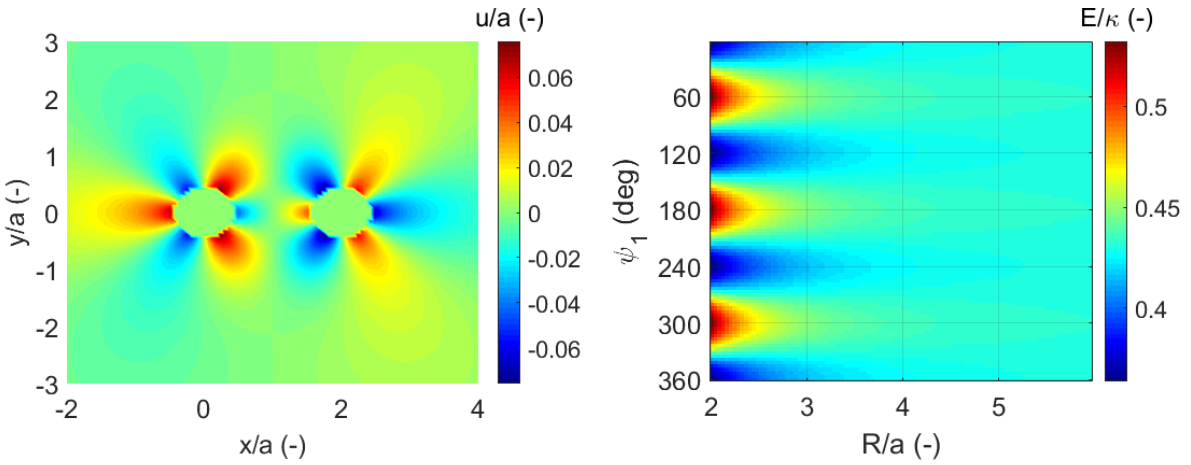
For fourth order monkey saddles the vector of prescribed curvature is given by:

$$\mathbf{K} = \begin{pmatrix} \alpha_1/a * \cos(\psi_1) \\ \alpha_1/a * \sin(\psi_1) \\ -\alpha_1/a * \cos(\psi_1) \\ -\alpha_1/a * \sin(\psi_1) \\ \alpha_1/a * \cos(\psi_1) \\ \vdots \end{pmatrix} \tag{4.3}$$

We can now make the same energy calculations as in figure 10b for third and fourth order symmetrical inclusions, as shown in figure 11b and 12b. Notice that in both figure 10b and 12b $\psi = 0$ and $R/a = 2$ corresponds to the lowest energy state, and that this state corresponds to two hills being aligned (see figure 10a and 12a). On the other hand, in figure 11b the lowest energy state is also $\psi = 0$ and $R/a = 2$, however this state corresponds to the alignment of a hill and a valley (see figure 11a).

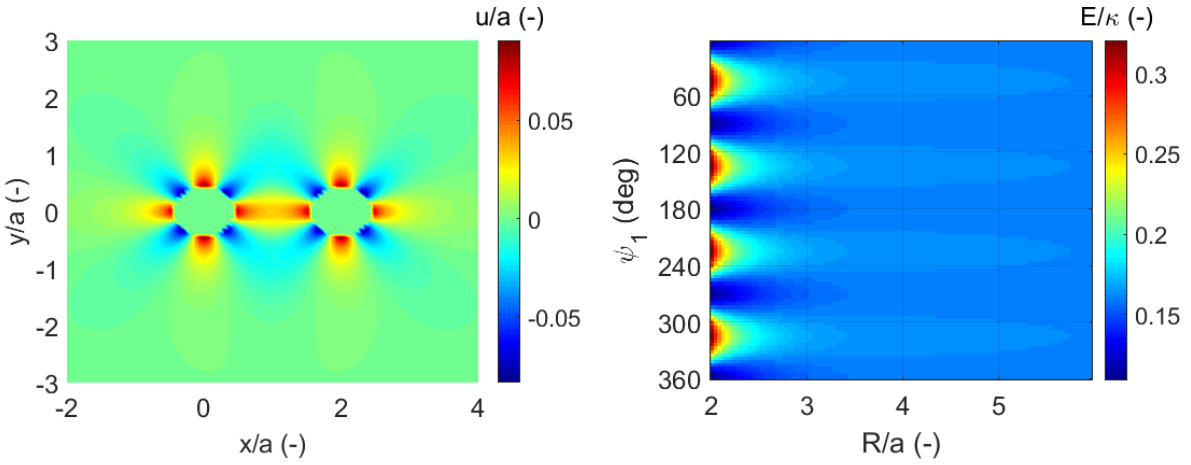
This seemingly contradicting outcome is caused by the difference in nature between derivatives of an odd order and those of an even order. Rather than getting two hills or two valleys to align, the minimum energy configuration for two derivatives is of the same orientation to align. Due to the symmetry of even order polynomials this means that two hills align when an even order derivative is imposed. On the other hand, uneven order polynomials are antisymmetric, thus when an uneven order derivative is imposed, a valley and a hill will align.

The third order monkey saddles do not behave like we would expect from the results of saddle shaped inclusions. The fundamental difference in nature between second order and higher order polynomials means that the third order derivatives are not capable of accurately modeling the inclusions they are supposed to describe. As such, we recommend abandoning this direction of research.



(a) (b)

Figure 11: Two monkey saddles on a membrane. (a) Plot of the membrane shape for the case $\psi_1 = \psi_2 = 0$ and $R = 2$. Note that a hill and a valley are aligned. (b) Energy for distance R/a between the two inclusions and rotation angle ψ_1 of the first inclusion (the rotation ψ_2 of the second inclusion is fixed at 0). Notice that due to the symmetry of the monkey saddles, $\psi_1 = 0$ is the same as $\psi_1 = 120$ and $\psi_1 = 240$.



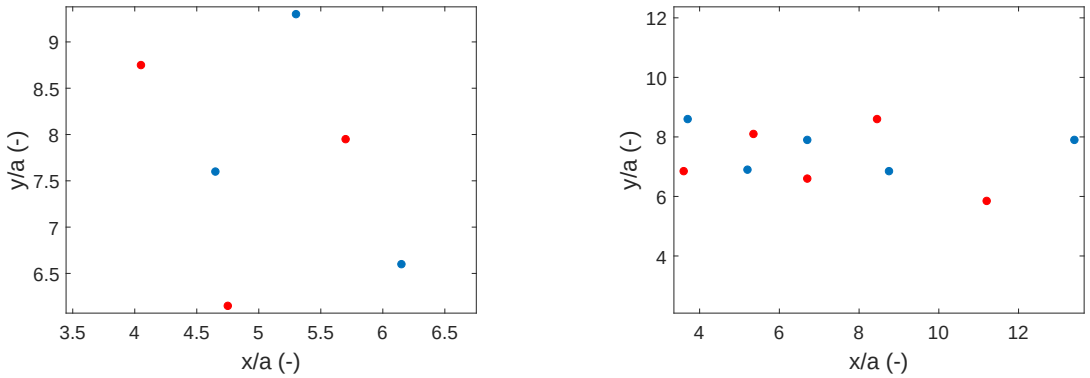
(a) (b)

Figure 12: Two fourth order monkey saddles on a membrane. (a) Plot of the membrane shape for the case $\psi_1 = \psi_2 = 0$ and $R = 2$. Note that two hills are aligned. (b) Energy for distance R/a between the two inclusions and rotation angle ψ_1 of the first inclusion (the rotation ψ_2 of the second inclusion is fixed at 0). Notice that due to the symmetry of the fourth order monkey saddles, $\psi_1 = 0$ is the same as $\psi_1 = 90$, $\psi_1 = 180$ and $\psi_1 = 270$.

4.4 Systems with N/2 positive and N/2 negative inclusions

In this section, we will be discussing large systems with an equal number of positive and negative inclusions. We will first show the equilibrium configuration for 6 inclusions as calculated using a Metropolis simulation, after which we will analyze its limitations for larger systems. Next, we will use the gradient descent method to find the equilibrium for systems with six and ten inclusions and compare its speed to the Metropolis algorithm for six inclusions. Also, we will evaluate the results of the Gradient descent method for systems of eight or more inclusions. Lastly, we will apply the Brownian motion method and evaluate its ability to find the equilibrium for large systems of inclusions.

4.4.1 Metropolis simulations



(a) *(b)*

Figure 13: Lowest energy states for (a) 6 and (b) 10 inclusions that were found by a Metropolis simulation with 10^6 and 10^7 timesteps respectively. The blue dots represent negative inclusions and the red dots represent positive inclusions. (a) was found after 81428 timesteps and has energy $E/\kappa = 1.32631$. (b) was found after 738711 timesteps and has energy $E/\kappa = 2.21776$.

Figure 13a shows the lowest energy state for 6 inclusions that was found by a Metropolis simulation with 10^6 timesteps. The butterfly-like shape matches the lowest energy state found by [17]. This shape was obtained after 81428 timesteps, but additional steps are still necessary to ascertain (to a certain extent) that no lower energy state is attainable.

Figure 13b shows the lowest energy state for 10 inclusions that was found by a Metropolis simulation with 10^7 timesteps. This shape was obtained after only 738711 timesteps (7.4% of the total). The pattern has a nice rectangle containing 8 inclusions, but the other two inclusions seem to be positioned rather randomly. As such, we expect that this is not actually the globally minimal energy state. In the next paragraph we will show several configurations with a lower energy to confirm this.

This clearly sub-optimal configuration was obtained with only 738711 timesteps, so it seems logical that the algorithm simulation should be able to find lower energy states if it was kept running for more timesteps. However, in the more than $9 \cdot 10^6$ timesteps that followed, no lower energy state was found. As such, it appears to be infeasible to find the global minimum for 10 inclusions using the Metropolis algorithm.

The system with 10 inclusions required almost 10 times more timesteps to find the minimum compared to the system with 6 inclusions. Even then, the minimum obtained was not the global one. Obviously, the timesteps required to find the global minimum scales either with a very high order polynomial, or, more likely, exponentially with the number of inclusions. Therefore, finding the global minimum may still be somewhat feasible for 10 inclusions, but it will very quickly become intractable for problems with larger numbers of inclusions.

4.4.2 Gradient descent simulations

Figure 14 shows the equilibrium configuration for 6 inclusions that results from a gradient descent simulation. Every time the gradient descent method is applied to this system with 6 inclusions, the simulation converges to this exact same pattern. The butterfly-like shape is a good match for what we found using the Metropolis algorithm (figure 13a). This pattern was obtained after 800 timesteps. However, the approximate butterfly shape was actually already attained after 200 timesteps, after which the algorithm used 600 timesteps to ascertain that it was close to the minimum and to converge closer to the actual minimum by gradually reducing the stepsize. The resulting high precision of $\Delta r < 0.001$ is the reason that the configuration has an energy $E/\kappa = 1.32293$, which is somewhat lower than the energy $E/\kappa = 1.32631$ that was obtained with the Metropolis algorithm in the previous paragraph.

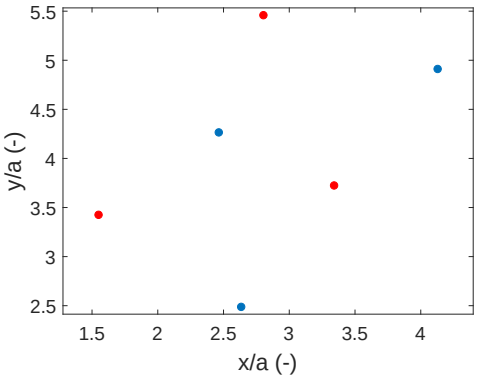


Figure 14: Equilibrium configuration for 6 inclusions calculated using the gradient descent method. It was calculated using 800 timesteps and has energy $E/\kappa = 1.32293$. *The blue dots represent negative inclusions and the red dots represent positive inclusions.*

In summary, the gradient descent method needed only 200 timesteps to find the approximate butterfly shape which is approximately 400 times faster than the Metropolis algorithm. In addition, after another 600 timesteps, the algorithm not only ascertained that it was actually at the minimum but it also converged considerably closer to this minimum than the Metropolis algorithm could.

In figures 15, 16, 17 and 18 we will give several examples of equilibrium configurations for systems with more than 6 inclusions. For each of these cases, between 20 and 30 simulations have been run to estimate the frequency at which some of the patterns appear.

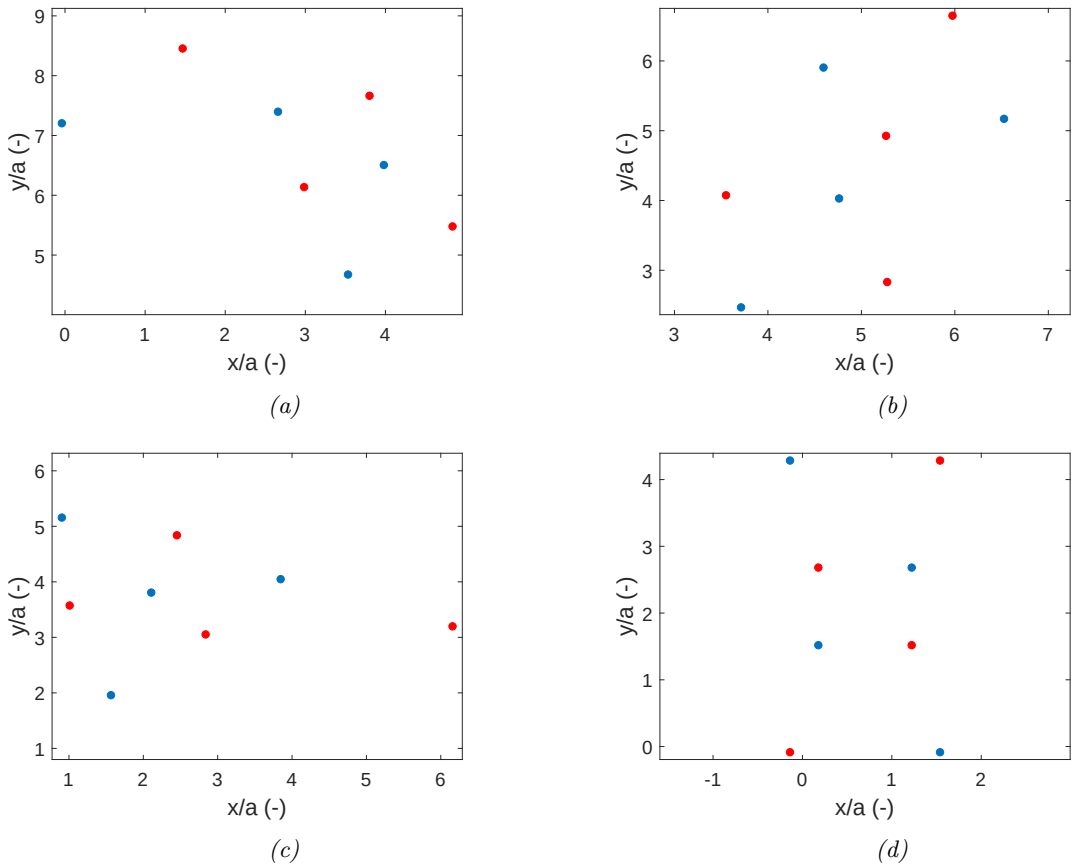


Figure 15: Several equilibrium configurations found with the gradient descent method for a system of 8 inclusions. The blue dots represent negative inclusions and the red dots represent positive inclusions. (a) Lowest energy state of the pattern found in 59% of the cases, with $E/\kappa = 1.76861$. This pattern is characterized by one group of six in the butterfly shape from figure 14 and a group of two inclusions attached at one corner of this butterfly. (b) Lowest energy state of the pattern found in 32% of the cases, with $E/\kappa = 1.76511$. This pattern is characterized by two groups of four organized in squares, which are linked together at an angle to form a bent rectangle. (c) Lowest energy state of the pattern found in 9% of the cases, with $E/\kappa = 1.76817$. This pattern is characterized by 7 inclusions in a triangle formation and one attached to one corner. (d) Pattern found when starting with the inclusions in an evenly spaced 2 by 4 rectangular structure. This pattern has energy $E/\kappa = 1.76666$, which is lower than found in (a) and (c), but higher than the energy of (b). It is therefore not the global minimum for 8 inclusions.

Figure 15 shows four equilibrium configurations found with the gradient descent method for a system of 8 inclusions. Most of these simulations take 1000-5000 steps to converge. Figures 15a, 15b and 15c were obtained from random initial conditions. Of the three, figure 15c has the lowest energy and is therefore our best guess at the global minimum.

From the results of 6 inclusions in figure 14, we may speculate that the equilibrium for 8 inclusions is a similar rectangle. We can test this by simply setting the initial configuration to a 2 by 4 rectangle and subsequently letting the gradient descent algorithm converge. Figure 15d shows the result, which is a 2 by 4 rectangle, implying that this structure is at least stable. However, the energy $E/\kappa = 1.76666$ of this structure is slightly higher than that of the structure of figure 15b, and it is therefore just a local minimum. Thus, our best guess at the global minimum is the configuration in figure 15b, but it is likely that the true global minimum continues to elude us.

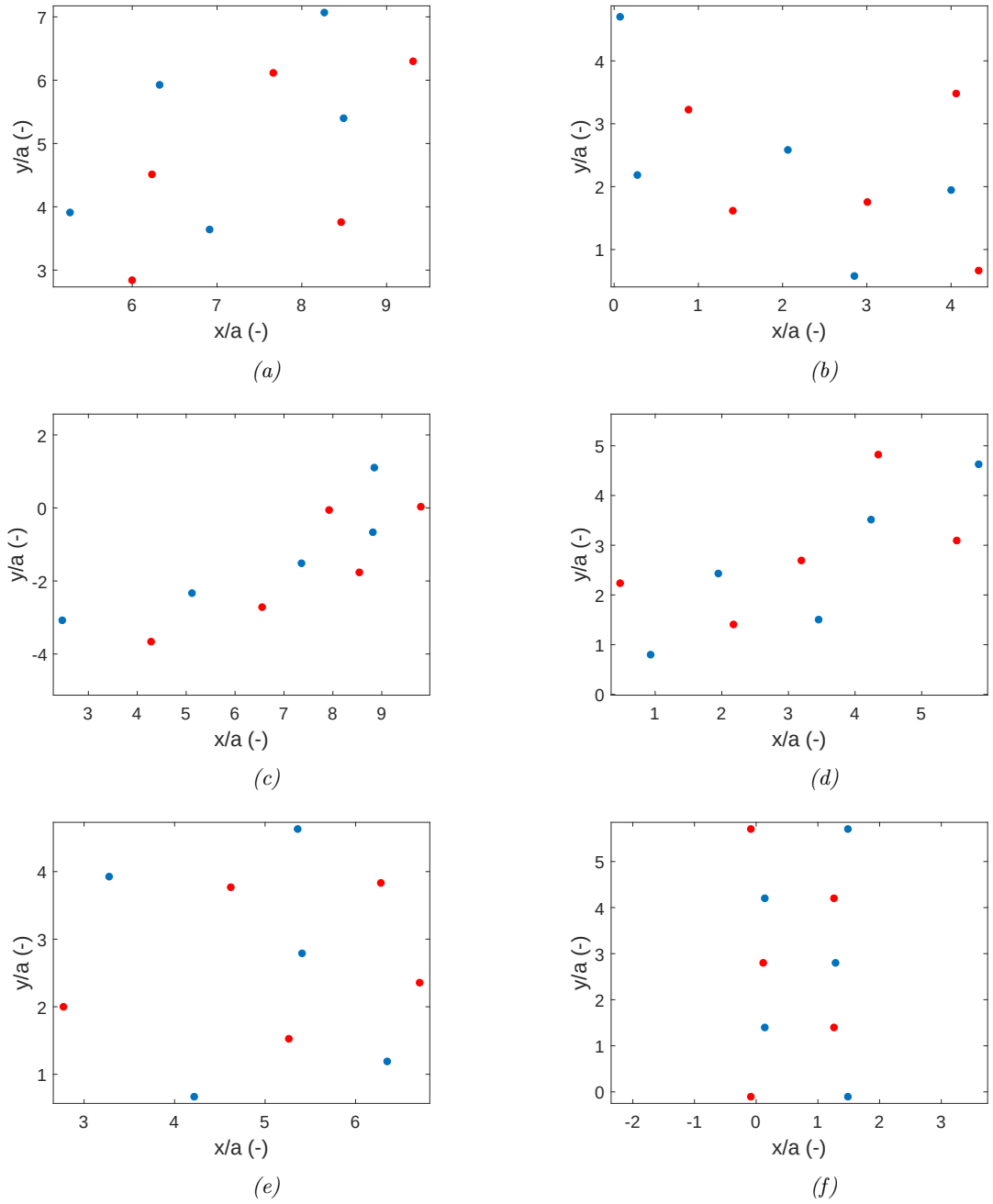


Figure 16: Several equilibrium configurations found with the gradient descent method for a system of 10 inclusions. Each image shows the lowest energy case if the pattern was found multiple times. The blue dots represent negative inclusions and the red dots represent positive inclusions. The pattern with the lowest energy we obtained for 10 inclusions is shown in (a). It was found only once in 24 simulations, and has energy $E/\kappa = 2.20389$*. The pattern in (b) was obtained 17% of the simulations, has energy* $E/\kappa = 2.21005$*, and is characterized by a single rectangle of 8 inclusions, while the remaining two seem randomly attached to one of the corners. The pattern in (c) was obtained 17% of the simulations, has energy* $E/\kappa = 2.21232$*, and is characterized by 6 inclusions forming a butterfly shape, with the remaining 4 attached at one corner in a zigzag pattern. The pattern in (d) was obtained 13% of the simulations, has energy* $E/\kappa = 2.20671$*, and is characterized by 6 inclusions forming a butterfly shape, with the remaining 4 attached at one corner in a square shape. The pattern in (e) was obtained 8% of the cases, has energy* $E/\kappa = 2.21355$*, and always looks exactly like the one in the figure. The pattern in (f) is found both in 17% of the simulations, and when starting with the inclusions in an evenly spaced 2 by 5 rectangular structure. It has energy* $E/\kappa = 2.20570$*, which is higher than the energy of (a). It is therefore not the global minimum for 10 inclusions.*

Figure 16 shows six equilibrium configurations for the gradient descent method for a system of 10 inclusions. Most of these simulations take 1000-8000 steps to converge. Figures 16a-e were obtained from random initial conditions. Of the five, figure 16a has the lowest energy and is therefore our best guess at the global minimum.

Figure 16f shows the pattern obtained both in 17% of the simulations from random initial configuration, and also when the simulation is started from an evenly spaced 2 by 5 rectangle. The configuration in the figure was obtained by starting from an evenly spaced rectangle, and it has energy $E/\kappa = 2.2057$. This energy is higher than the energy of the configuration in figure 16a, and it is therefore not the global minimum.

In 13% of the simulations, eight inclusions formed one of the patterns in figure 15, while the other two inclusions were ejected. The remaining 13% of the simulations found unique patterns that did not resemble any of the others.

All the patterns obtained here have a considerably lower energy than the one obtained with the Metropolis algorithm in figure 13b. Not only are the results better than those obtained with Metropolis algorithm, the simulations are also much faster. The algorithm needs on average about 3000 timesteps to find one of these configurations. That is approximately 250 times less timesteps than the Metropolis algorithm needed to find the suboptimal configuration shown in figure 13b.

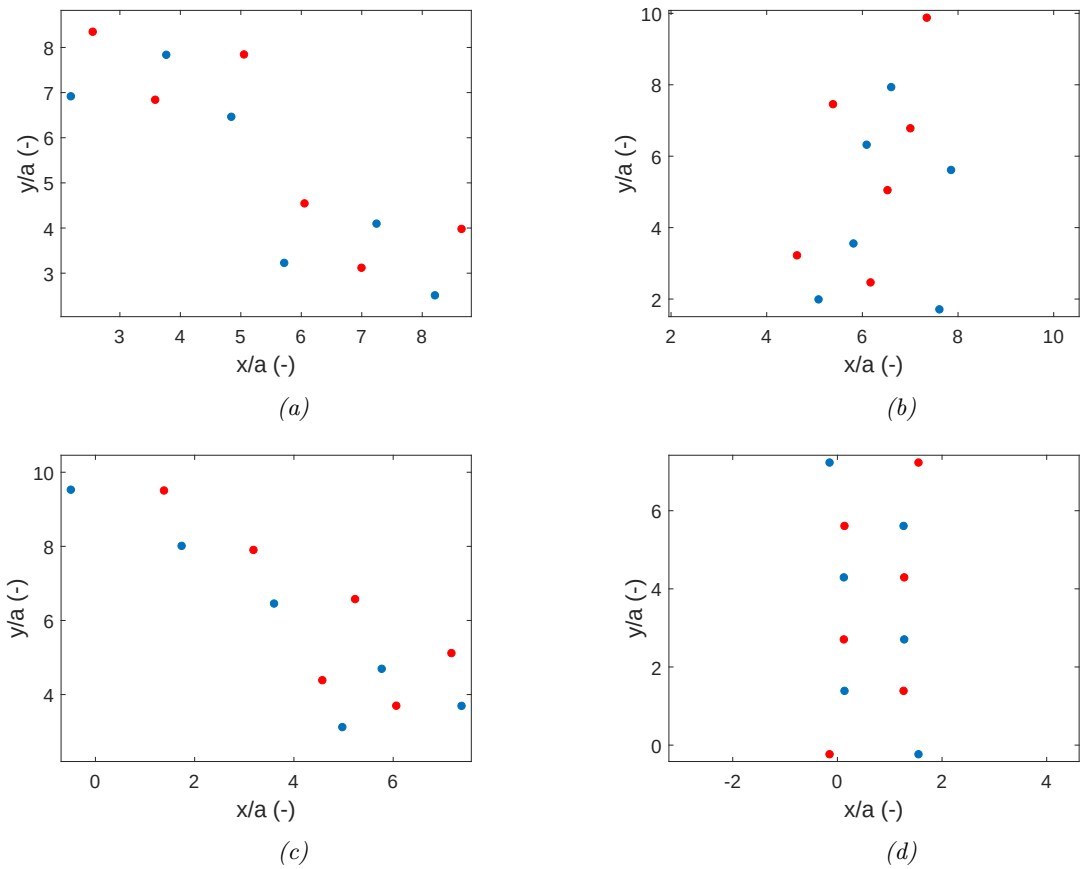


Figure 17: Several equilibrium configurations found with the gradient descent method for a system of 12 inclusions. The blue dots represent negative inclusions and the red dots represent positive inclusions. (a) Pattern found in 36% of the simulations. This configuration has energy $E/\kappa = 2.64399$, which is the lowest of any of the configurations found with this pattern. This pattern is characterized by two groups of six in the butterfly shape from figure 14. The other 64% of the simulations resulted in greatly varying patterns at equilibrium, all with higher energies than the configurations with a pattern similar to (a). Two examples are shown in (b) with $E/\kappa = 2.65290$ and in (c) with $E/\kappa = 2.66001$. (d) Pattern found when starting with the inclusions in an evenly spaced 2 by 6 rectangular structure. It has energy $E/\kappa = 2.64526$, which lower than found in (b) and (c), but higher than the energy found in (a). It therefore not the global minimum for 12 inclusions.

Figure 17 shows four equilibrium configurations for the gradient descent method for a system of 12 inclusions. Most of these simulations take 1000-8000 steps to converge. Figures 17a, 17b and 17c were obtained from random initial conditions. Of the three, figure 17a has the lowest energy and is therefore our best guess at the global minimum.

Figure 17d shows the pattern obtained when the simulation is started from an evenly spaced 2 by 6 rectangle. This configuration has energy $E/\kappa = 2.64526$. This energy is higher than the energy of the configuration in figure 16a, and it is therefore not the global minimum.

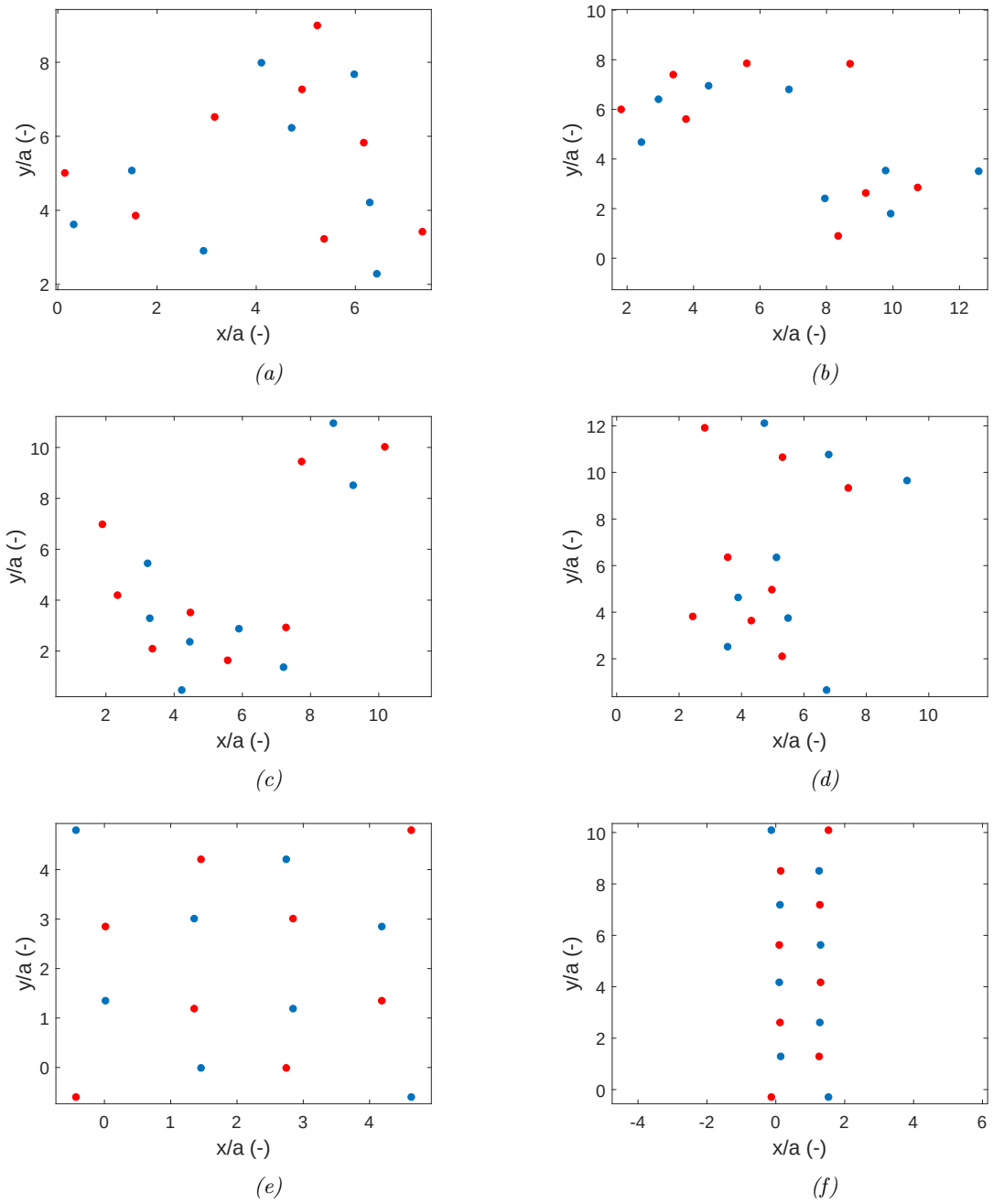


Figure 18: Several equilibrium configurations found with the gradient descent method for a system of 16 inclusions. The blue dots represent negative inclusions and the red dots represent positive inclusions. When starting with a random initial configuration, the resulting equilibrium patterns greatly vary. (a) Lowest energy state found from random initial configuration, $E/\kappa = 3.53230$. (b), (c) and (d) show three example equilibrium configurations with $E/\kappa = 3.53903$, $E/\kappa = 3.54460$ and $E/\kappa = 3.55121$ respectively. (e) Pattern found when starting with an evenly spaced 4 by 4 square structure. It has energy $E/\kappa = 3.53453$, which is better than (b), (c) and (d), but worse than that of (a). (f) Pattern found when starting starting with the inclusions in an evenly spaced 2 by 8 rectangular structure. It has energy $E/\kappa = 3.52454$ and is our best guess for the global minimum for 16 inclusions.

In all simulations of a system with 16 inclusions, the equilibrium configuration formed a unique pattern that did not resemble the pattern of any other simulation. Figure 18 shows six equilibrium configurations. Most of these simulations take 1000-8000 steps to converge.

Figure 18a shows a configuration with energy $E/\kappa = 3.53230$, which is the lowest energy found with the gradient descent method for a system with 16 inclusions when starting from a random initial configuration. Note that this pattern consists of six inclusions forming a butterfly shape at the top, and two sets of 5 inclusions (a square with one inclusion attached to one side) on the left and right. The butterfly shape is extremely commonly seen in the equilibrium configurations of these systems. This is not surprising as we already saw it quite regularly in the earlier simulations with less inclusions. On the other hand, we did not see the 5 inclusion structure in any of the earlier simulations (except maybe figure 16a depending on how you interpret it). However, it actually appeared in 24% of these simulations.

Figures 18b,18c and 18d show three example equilibrium configurations with $E/\kappa = 3.53903$, $E/\kappa = 3.54460$ and $E/\kappa = 3.55121$ respectively. The 6-inclusion butterfly shapes of figure 18b and the 4-inclusion square shapes of figure 18c were both common in these simulations. On the other hand, figure 18d is an example of an equilibrium configuration that does not have any structures resembling something found by another simulation.

Figure 18e shows the pattern obtained when the simulation is started from an evenly spaced 4 by 4 square. This configuration has energy $E/\kappa = 3.53453$, which is worse than that of the configuration shown in figure 18a.

Figure 18f shows the pattern obtained when the simulation is started from an evenly spaced 2 by 8 rectangle. This configuration has energy $E/\kappa = 3.52454$, which is the lowest energy we found for a system with 16 inclusions. Thus this is our best guess at the global minimum for this system.

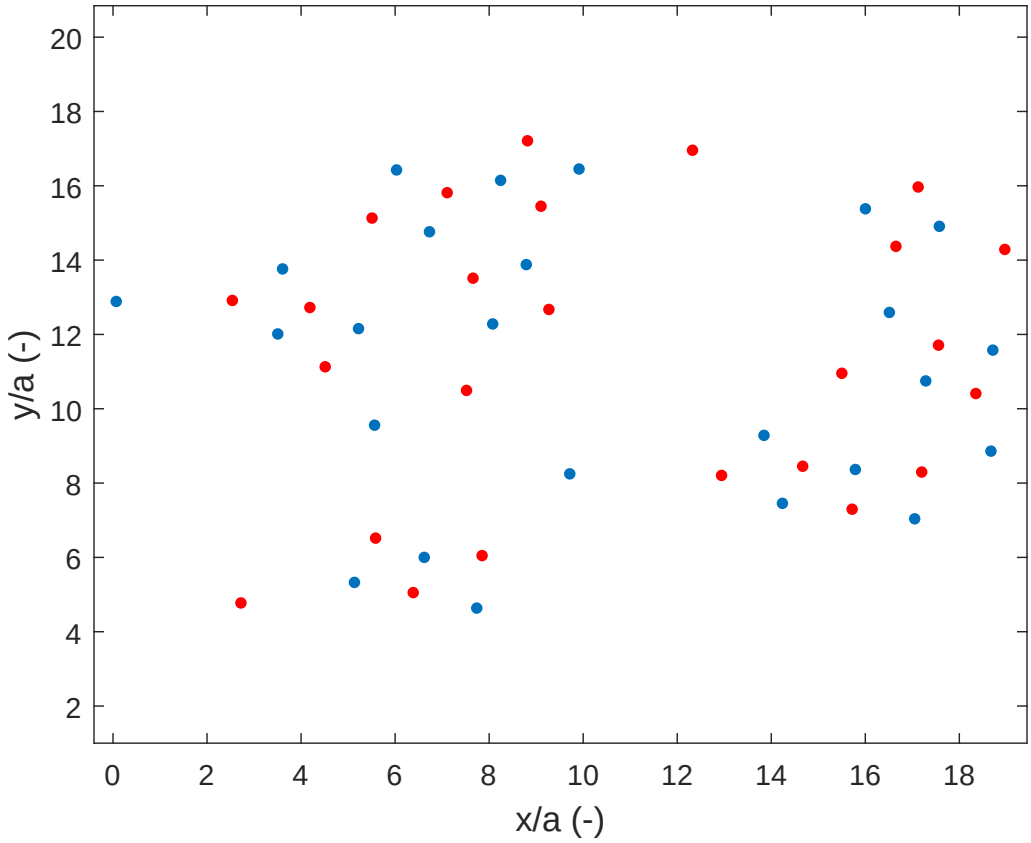


Figure 19: Equilibrium configuration for the gradient descent method for a system of 50 inclusions, calculated using 27400 timesteps. The blue dots represent negative inclusions and the red dots represent positive inclusions. Notice that there are several 6-inclusion butterfly structures like the one in figure 14, as well as several 4-inclusion squares.

We ran a single simulation for a system of 50 inclusions. Even though it only took 27400 timesteps to find an equilibrium, the simulation still took several hours due to the sheer size of the matrix that needed to be inverted at every timestep. The resulting configuration is shown in figure 19. Obviously, this is not the global minimum configuration. However, nature is usually not in the global minimum either, thus it might still be possible to see some of the structures that we see in this figure.

There are two structures that appear multiple times in this configuration: the 6-inclusion butterfly-shape and the 4-inclusion square-shape. We also saw these structures regularly in the earlier simulations with less inclusions. Obviously both structures are very stable, and it is therefore quite likely that they appear in nature also.

While the algorithm stopped here, this is not actually a minimum. This is because the energy will decrease if some of the smaller groups move closer to each other. However, the force they experience that pulls them towards each other is several orders of magnitude smaller than the force the individual inclusions experience from being slightly out of equilibrium within their group. As such, the inclusions oscillate around their equilibrium, with virtually no net movement towards the other group.

4.4.3 Brownian motion simulations

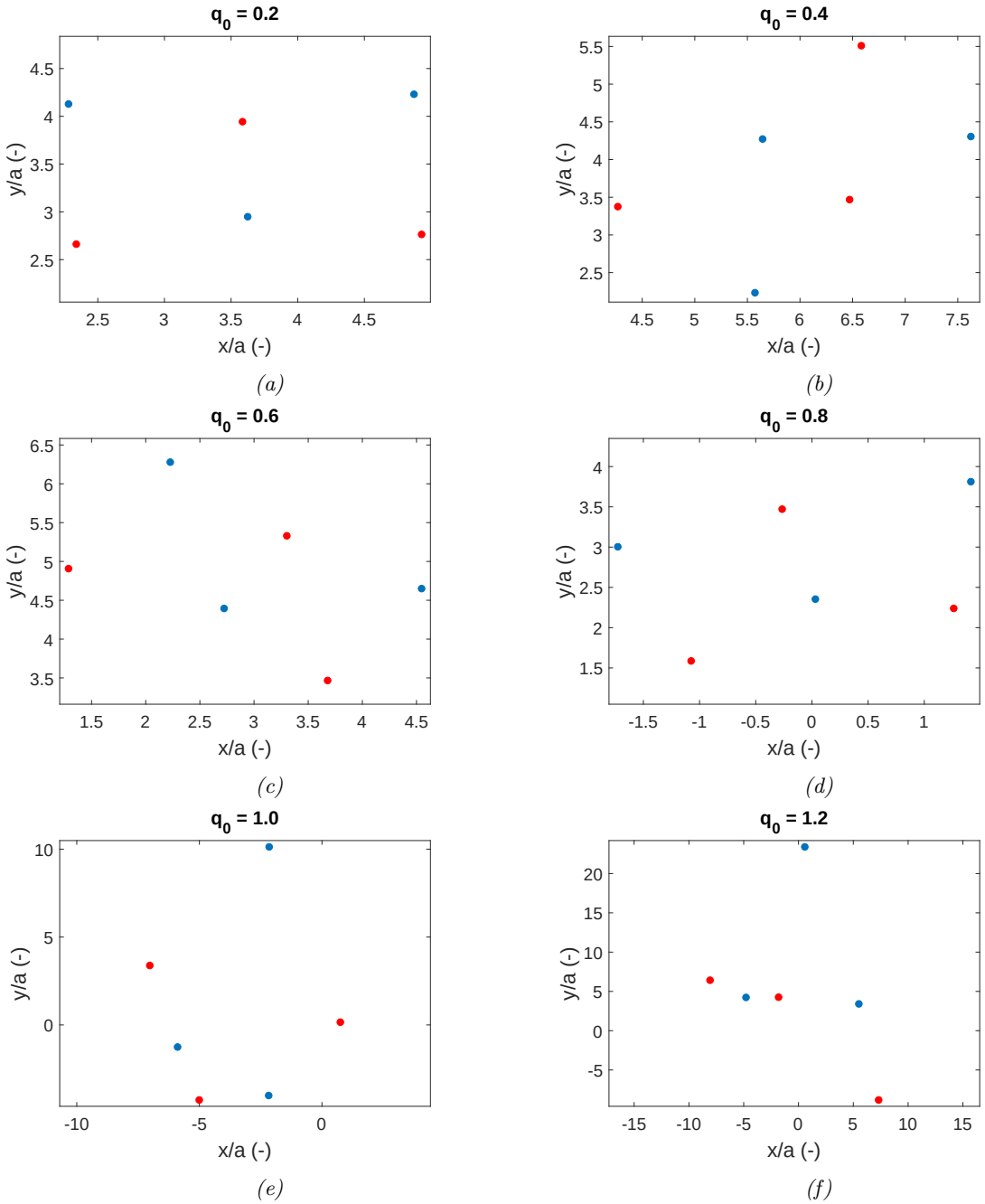


Figure 20: Several equilibrium configurations found with the Brownian motion method for a system of 6 inclusions. The blue dots represent negative inclusions and the red dots represent positive inclusions. The noise q_0 in the system was (a) 0.2, (b) 0.4, (c) 0.6, (d) 0.8, (e) 1.0 and (f) 1.2. The configurations have energy (a) $E/\kappa = 1.32227$, (b) $E/\kappa = 1.32648$, (c) $E/\kappa = 1.32519$, (d) $E/\kappa = 1.32652$, (e) $E/\kappa = 1.34803$ and (f) $E/\kappa = 1.3504$.

Figure 20 shows the configurations obtained for 6 inclusions obtained with the Brownian motion method for six different values of the noise q_0 (defined in paragraph 3.3). Figures 20a-d are a good match for the butterfly shapes we obtained with the Metropolis algorithm and the gradient descent method in figures 13a and 14. As expected, the energies of these configurations are close to those found with the other two methods.

On the other hand, figure 20e and especially figure 20f do not resemble these butterfly shapes, and correspondingly have a considerably higher energy than the configuration with the butterfly shape. We can conclude that Brownian motion method only yields good results when the noise $q_0 \leq 0.8$. For noise $q_0 \geq 1.0$, the simulation becomes so noisy that the algorithm is unable to converge to a minimum.

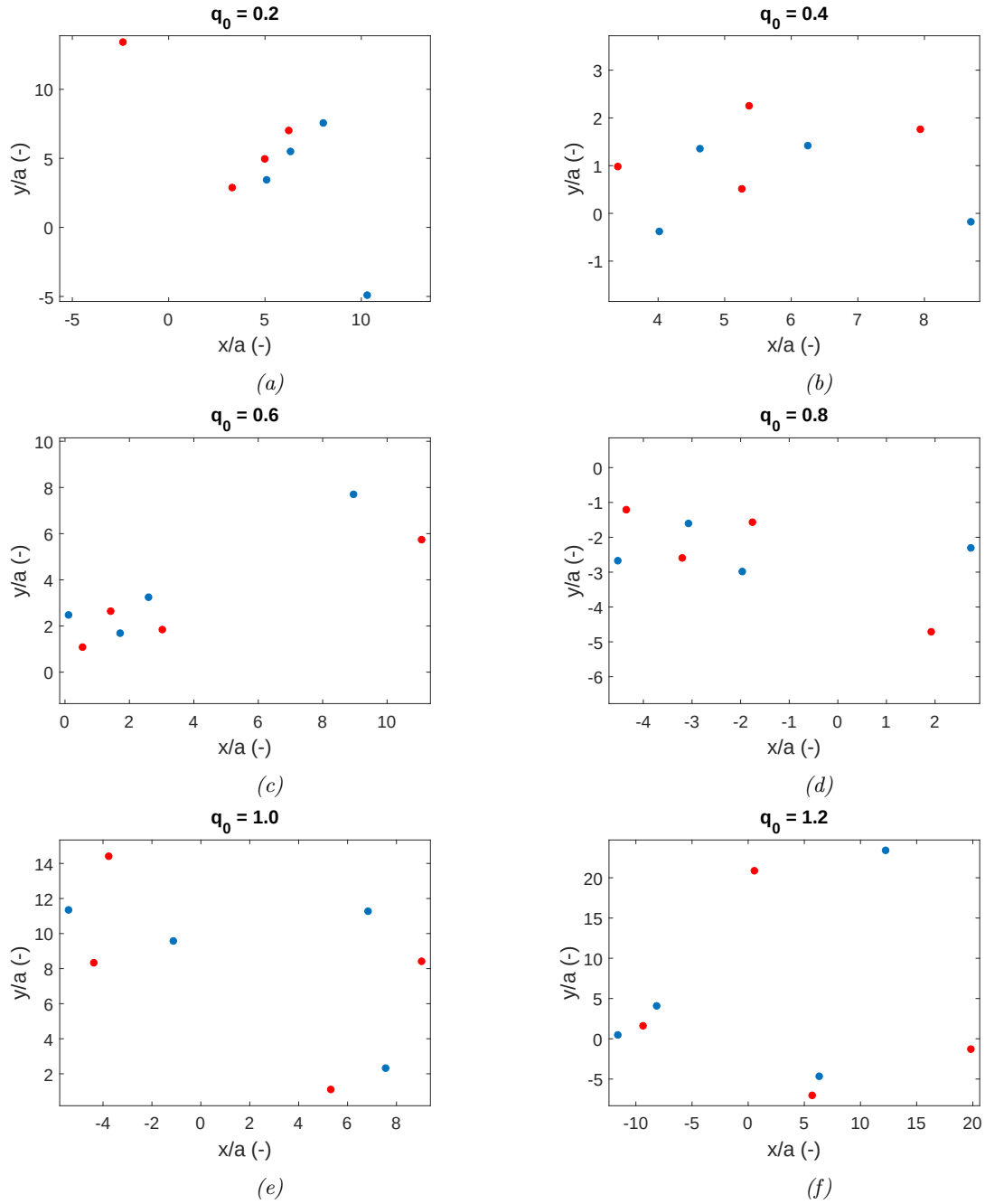


Figure 21: Several equilibrium configurations found with the Brownian motion method for a system of 8 inclusions. The blue dots represent negative inclusions and the red dots represent positive inclusions. The noise in the system was (a) 0.2, (b) 0.4, (c) 0.6, (d) 0.8, (e) 1.0 and (f) 1.2. The configurations have energy (a) $E/\kappa = 1.78925$, (b) $E/\kappa = 1.76853$, (c) $E/\kappa = 1.77119$, (d) $E/\kappa = 1.77022$, (e) $E/\kappa = 1.79512$ and (f) $E/\kappa = 1.79857$

Figure 21 shows the configurations obtained for 8 inclusions with the Brownian motion method for six different values of the noise q_0. The configurations shown in figures 21b-d are similar to the patterns we obtained with the gradient descent method in figure 15. As a result, the energies of these configurations are close to many of the configurations found with the gradient descent method. 21a has a rather strange pattern that we did not encounter with the gradient descent method and the energy $E/\kappa = 1.78925$ is considerably higher than the configurations obtained there. This is likely a coincidence where the algorithm got stuck in a local minimum somehow.

On the other hand, figure 21e and especially figure 21f seem to be almost pure chaos. As a result, the energies of these systems, $E/\kappa = 1.79512$ and $E/\kappa = 1.79857$, are even higher than that of the configuration in figure 21a. Just as we concluded before, The simulations are so noisy for $q_0 \geq 1.0$ that the algorithm is unable to converge to a minimum.

However, even with $q_0 \leq 0.8$, the Brownian motion method is not able to find any lower energy configurations than the gradient descent method could. This is because these systems are not noisy enough to pull the simulation out of local minima, unless they are very shallow. As a result, the patterns obtained with this method are somewhat more uniform than those obtained with the gradient descent method, but they are not much better. In addition, these simulations took 100000 timesteps to produce, instead of the 1000-8000 timesteps of the gradient descent method. As such, better results can be obtained by simply running multiple gradient descent simulations.

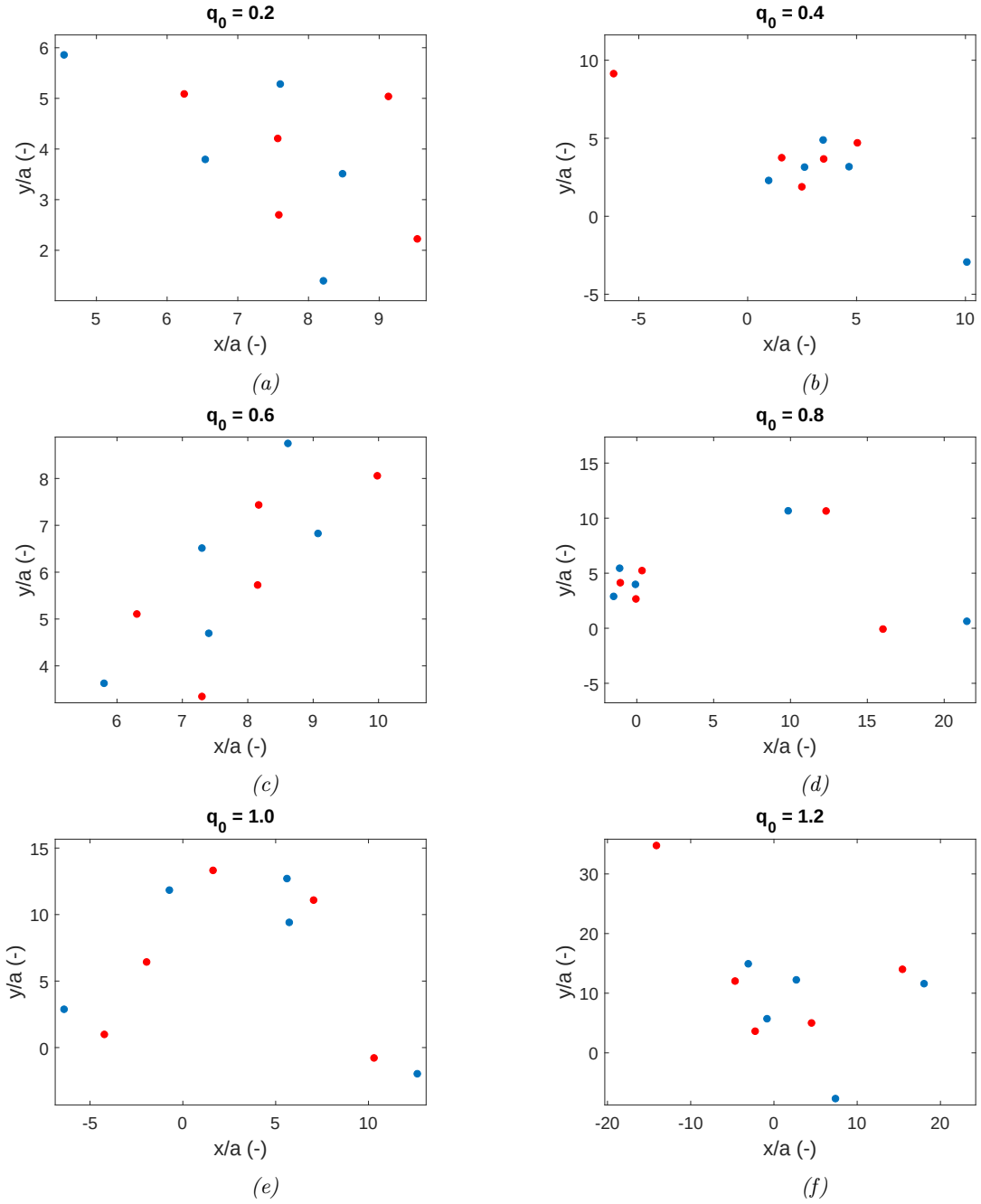


Figure 22: Several equilibrium configurations found with the Brownian motion method for a system of 10 inclusions. The blue dots represent negative inclusions and the red dots represent positive inclusions. The noise in the system was (a) 0.2, (b) 0.4, (c) 0.6, (d) 0.8, (e) 1.0 and (f) 1.2. The configurations have energy (a) $E/\kappa = 2.2111$, (b) $E/\kappa = 2.21639$, (c) $E/\kappa = 2.20532$, (d) $E/\kappa = 2.22160$, (e) $E/\kappa = 2.24258$ and (f) $E/\kappa = 2.24967$

Figure 22 shows the configurations obtained for 10 inclusions with the Brownian motion method for six different values of the noise q_0. The configurations shown in figures 22a-d are similar to the patterns we obtained with the gradient descent method in figure 16 and as result, their energies are also similar. The configuration in figure 22c is actually the rectangular shape that was the pattern with the second lowest energy that we obtained in only 17% of the gradient descent simulations. However, the simulations needed 100000 timesteps to produce these results, so it is still more effective to run the gradient descent method multiple times as it only needs 1000-8000 timesteps.

Again, for $q_0 \geq 1.0$, the configurations obtained seem to be mostly chaos, as shown in figure 21e and figure 21f. As a result, the energies of these systems, $E/\kappa = 2.24258$ and $E/\kappa = 2.24967$, are even higher than that of the configuration in figure 21a. Obviously, simulations with $q_0 \geq 1.0$ are too noisy to obtain useful results. As such, we will not be trying this again for the system with 16 inclusions.

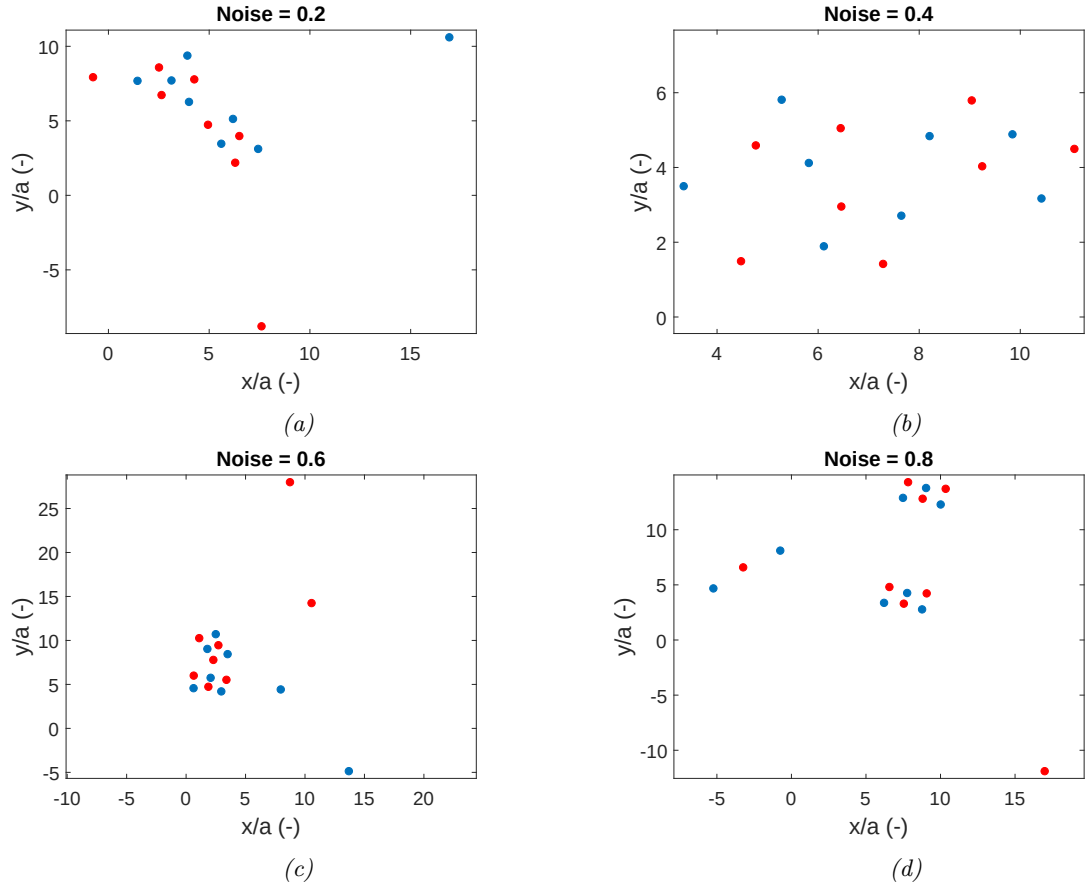


Figure 23: Several equilibrium configurations found with the Brownian motion method for a system of 16 inclusions. The blue dots represent negative inclusions and the red dots represent positive inclusions. The noise in the system was (a) 0.2, (b) 0.4, (c) 0.6, (d) 0.8. The configurations have energy (a) $E/\kappa = 3.54008$, (b) $E/\kappa = 3.52718$, (c) $E/\kappa = 3.54798$, (d) $E/\kappa = 3.54339$

Figure 23 shows the configurations obtained for 16 inclusions with the Brownian motion method for four different values of the noise q_0. All four configurations are similar to the patterns we obtained with the gradient descent method in figure 18 and as result, their energies are also similar. In addition, we once again used 100000 timesteps to obtain these configurations, way more than the 1000-8000 required for the gradient descent method. Therefore, we can conclude that the Brownian motion method is in no way better at obtaining low energy states for the 16 inclusion system than the gradient descent method.

In none of the systems with 6, 8, 10 or 16 inclusions did the Brownian motion method find any configuration that had a lower energy than what we found with the gradient descent method. When the noise $q_0 \geq 1.0$, the simulations became to noisy to converge to any minimum. When the noise $q_0 \leq 0.8$, the resulting configurations were not very different from those obtained with the gradient descent method. Most patterns that were found with the Brownian motion method were also found with the gradient descent method. Those that were not either had similar or considerably higher energy compared to the configurations found with the gradient descent method. In addition, the Brownian motion simulations needed many more timesteps to converge to an equilibrium due to the noise. We conclude that the Brownian motion method does not add any real value when compared to the gradient descent method.

5 Discussion

5.1 Point particles model

The point particles model has the large advantage that it is not computationally intensive. Simulations up to 10^7 time steps are easily feasible. However, this advantage comes at a price. Due to the point particle approximation as well as the wave-vector cutoff we applied in paragraph 2.2.4, the model has some strange characteristics.

Firstly, the point particles model is supposed to approximate the influence of a cone-shaped inclusion on the membrane shape. The influence of such an inclusion is characterized by the imposition of an angle on the membrane along the contour of the inclusion (basically what Weikl et al. [2] described). The point particles model approximates this by a second derivative imposed at the center of the inclusion. However, the membrane flattens out so quickly (as can be seen in figure 2a) that the first derivative is several orders of magnitude smaller than it should be at the location of the inclusion's edge.

Secondly, the membrane diverges at the location of any inclusion (see figure 2a). This is not only unrealistic, but it also causes problems for the boundary conditions imposed by the inclusions. After all, the second derivative is not well-defined at the location of a divergence. As a result, we did not actually solve for the Lagrange multipliers by applying the boundary conditions to the expression of the membrane shape itself as the divergences would have caused problems.

Instead, each boundary condition was applied to the expression of the membrane shape that has a high wave-vector cutoff applied to the term corresponding to that boundary condition. The wave-vector cutoff prevents the divergence and results in a solvable set of equations. As a result, however, the membrane shape as calculated with the point particles model does not actually match the boundary conditions imposed by the inclusions.

Thirdly, the energy of two inclusions on a membrane diverges at some non-zero distance (see figures 3b and 3e). This distance can be interpreted as an analogue for the diameter of the particles, which would make a divergence at that point logical. Furthermore, we found that the location of divergence is mostly linearly dependent on the wave-vector cutoff $1/\Lambda$, which serves as the length scale in this model. However, a system with two positive inclusions has a different location of divergence compared to a system with one positive and one negative inclusion. In addition, the location of this divergence also changes when two inclusions are in the presence of another inclusion (see figure 3e). These problems makes the interpretation of the divergence location being the diameter of the particles difficult. In addition, it greatly complicates simulations for large numbers of inclusions because it is difficult to say with certainty that a configuration is not within the ill-defined regime beyond the divergence. This area usually has considerably lower, and often strongly negative energies, and is therefore likely to be picked out as being the global minimum.

The cause of this divergence at some non-zero distance has to do with the high wave-vector cutoff. Normally, the self-interaction (the m_{ii} terms on the diagonal of the matrix in equation (2.31)) should always be able to compensate for outside influences. This allows the membrane to meet the boundary condition of each inclusion under any circumstance. However, due to the wave-vector cutoff, this ability of the self-interaction becomes limited. As a result of the cutoff, the self-interaction can only have finite influence on the membrane shape at the location of the inclusion. On the other hand, another inclusion can have unbounded influence, as the membrane diverges near it. This divergence means that if this other inclusion gets close enough, its influence on the first inclusion can get larger than the first inclusion's influence on itself. At this point, the solutions quickly diverge.

And yet, despite these problems, the patterns formed at equilibrium may still be correctly predicted by the point particles model. For that reason we employed the finite size particles model to validate the results of point particles model for three inclusions. And indeed, we saw that the finite size model predicts the same triangle structure with the same characterizing angle as the point particles model for three inclusions. This can be understood as follows: the influence of the wave vector cutoff only becomes

problematic at short range, as can be seen by divergence of the energy. Though this still influences distances between inclusions at equilibrium, relative distances and angles are clearly not affected. As such, we can conclude that the point particles model, despite its faults, is likely still able to correctly predict the equilibrium patterns formed by inclusions on the flat membrane.

5.2 Finite size particles model

The finite size particles model describes the influence of inclusions a lot more accurately than Dommersnes' model. As such, the results obtained with this model should be more accurate and are therefore a good way to validate the results obtained with Dommersnes' model. However, the formalism is correspondingly a lot more difficult to solve. We were not able to solve the membrane shape for three inclusions fully analytically and were therefore forced to numerically approximate the Lagrange multipliers.

As a result, the numerical part of the finite size particles model is a lot more computationally expensive than the point particles model. In our simulations, we approximated the circular contour of an inclusions with n points ($n = 21$ for most of our simulations). We subsequently had to solve a linear system involving a $2nN$ by $2nN$ matrix instead of the $3N$ by $3N$ matrix of the point particles model. For $n = 21$ this is approximately $14^3 = 2744$ times slower for a naïve implementation (though it may be somewhat fast if a better algoritm is used).

As a result, it is infeasible to do simulations with large numbers of inclusions and many time steps with the finite size particles model. However, it is still possible to use this model to calculate the membrane energy for a small set of inclusion configurations. As such, it can be used as a control for the point particles model by calculating the energy for several inclusion configurations around a predicted minimum to check if it really is a minimum in the finite size particles model also. However, to do so, one would need to overcome the problem of the differing length scales first.

Another application of the finite size particles model are for simulations involving inclusions with more complicated shapes. The point particles model can only be used to simulate inclusions with second order symmetries (like the saddle points we briefly discussed in paragraph 4.3.1). We have sought to extend the model accommodate higher order symmetrical inclusions (monkey saddles), but we have come to conclude that they do not describe the effects they were intended to (see paragraph 4.3.2). As such, the point particles model is very limited in the variety of shapes it can describe.

The finite size particle model, on the other hand, can describe virtually any shape of inclusion. In this project we only worked with conical inclusions. However, this model can describe differently shaped inclusions by simply changing the boundary conditions of equation (2.47). In the current application of the model, the boundary conditions impose a constant value for the magnitude of the gradient (though it changes direction) over a circle. An inclusion which is not rotationally symmetrical, like a saddle-shaped inclusion, can be modeled by simply making the magnitude of the derivative constraint a function of the location on the inclusion. Similarly, an inclusion that is not circular can be modeled by imposing the constraints around some other shape that is not a circle. Of course, the applications will be rather limited as systems with large numbers of inclusions require terrifying computation power to solve.

When applying the finite size model one may want to use a better numerical method than the Metropolis algorithm. Before it is possible to use the gradient descent method, however, one will first need to find an expression for the derivative of the energy in equation (2.80). This may prove somewhat more difficult than it was for the point particles method. To make it easier, one can try to neglect the influence of h_i, β_i and γ_i, so $\frac{d}{dx_n}h_i \approx \frac{d}{dx_n}\beta_i \approx \frac{d}{dx_n}\gamma_i \approx 0$. However this may have a considerable influence on the simulations. Alternatively, one can use the Metropolis algorithm with zero temperature, i.e. reject all moves that increase the energy, like we did for a system with three inclusions. After all, we saw that for many systems, there was a smooth path to the global minimum without any local minima in between. As such, there is no need for any noise.

5.3 Numerical methods

One of the challenges for future research is to predict the patterns formed by systems with a large number of inclusions. For these systems, only the point particles model is feasible, given the current calculation power. Even then, a good numerical method to minimize the membrane energy for inclusion configurations is essential. In this project, we have investigated three numerical methods: the standard Metropolis algorithm, the gradient descent method and Brownian motion method.

5.3.1 Metropolis algorithm

The widely used Metropolis algorithm has the obvious advantage of being applicable to practically any system. However, the number of states (a finite value due to numerical precision) scales exponentially with the number of inclusions. In addition, in the limit of infinitely many timesteps, the chance of being in globally minimal state at any given timestep is inversely proportional to the number of states [19]. As a result, the steps required to find the global minimum also scales exponentially with the number of inclusions. Using gradual reduction of temperature might make the method sub-exponential, but it is still definitely some high order polynomial. It is still able to find the global minimum for six inclusions. However, it already gets stumped on 10 inclusions, even with 10^7 timesteps. So obviously, using this method to find the global minimum quickly becomes infeasible for even larger number of inclusions.

Interestingly, the lowest energy configuration that was obtained in a Metropolis simulation, was found after only 7.3% of the 10^7 timesteps in the simulation for 10 inclusions. In other words, the last 92.7% of the simulation was not able to find any better configurations, even though there are a great variety of configurations with a lower energy, as we discovered with the gradient descent method. A cause of its inability to find better configurations might be the requirement to keep the inclusions a certain minimum distance apart. This makes certain tightly-packed configurations difficult to reach, because it requires inclusions move to the exact spot between two others. Any small deviation from this spot is not allowed because it would come too close to another inclusion.

In addition, for some low-energy patterns, the inclusions actually lie closer together than the minimum allowed distance, even though the energy is not within the divergence region (see figure 3b). As a result, the Metropolis algorithm is not able to find them. However, decreasing the minimum required distance will not necessarily work, because other configurations with the same or even higher inclusions distances might be within the divergence region (see figure 3e). Therefore, the Metropolis algorithm might not even be able to reach the global minimum energy configuration.

5.3.2 Gradient descent

We developed the gradient descent method for this study. It is several hundred times faster than the Metropolis algorithm. In addition, it does not have much trouble with inclusions getting too close. After all, the force diverges as the energy does, and therefore the inclusions are repelled very strongly before being able to get past this divergence. Thus, if the step size is small enough, the region within the divergence cannot cause problems.

Another advantage is that for very large systems, the gradient descent method is still able to predict structures that will form. For 50 inclusions, it takes approximately 27400 timesteps to find a local minimum. Though this minimum is undoubtedly not the global one, it still demonstrates some small patterns that are very stable, like the four inclusion square-shaped structure and the six inclusion butterfly-shaped structure. These patterns appear in most simulations with many inclusions, and it might therefore be possible to find such patterns in nature also.

The gradient descent method is currently still limited in that it is unable to find the global minimum for systems with 8 inclusions or more. For these systems, the algorithm quickly gets stuck in local minima or in configurations with several smaller groups. In the last case, the system is not actually in a minimum. However, the force applied to an inclusion for being slightly out of equilibrium within its own group is far larger than the force that pulls the groups together (typically as much as 20 orders of magnitude larger). As a result, the inclusions will quickly oscillate within their groups while the center of mass of

each group stays mostly in place. There are two ways that could potentially resolve this issue.

The first way is by grouping inclusions together and keeping the internal distances fixed while moving the group as a whole. This way, the strong forces resulting from slight deviations from equilibrium for the individual inclusions in a group will no longer cause problems as the net force on the whole group is used to move them. To implement this, one would need a good way to group inclusions together. A simple method would be single-linkage hierarchical clustering where one cuts at a distance around the typical inclusion distance in a stable group.

In the second method one should move all inclusions normally using gradient descent during two timesteps. Next, calculate the net displacement during these two time steps for all inclusions. If all inclusions are in stable groups, the total net displacement will be small. Rescale this movement so that the total displacement is comparable to that over one normal time step and repeat the process. This essentially filters out all 2-step oscillations, functioning as a low pass filter.

5.3.3 Brownian motion

To counter the local minima problem of the gradient descent method, a logical choice would be to add noise to the system. We have tried this in the form of the Brownian motion method. However, as we saw in paragraph 4.4.3, Brownian motion simulations do not add any real value when compared to the gradient descent method. For low values of the noise, the force dominated and it was therefore similarly unable to get out of local minima. For high values of the noise, the simulations did not get stuck in local minima as much, but were instead so noisy that they were unable to find any minima at all.

This was made even more difficult by the taboo divergence regions. For stochastic simulations like Brownian motion and Metropolis, it would be most effective to keep track of the minimum energy found so far during the simulation. However, in this system, it is very difficult to discern whether the low energy is the result of two inclusions being too close, i.e. within the area beyond the divergence of figure 3b. In the Metropolis algorithm this can still be somewhat resolved by keeping the inclusions a minimum distance apart. However, this does not work well with the Brownian motion method, because it is a dynamic simulation were all inclusions move at once. If an inclusion is fixed in place due to it wanting to move onto another inclusion, it disrupts the dynamics of the system and introduces certain biases (i.e. moving apart is more favored than it should be). As a result, we can only hope that the simulation will at some point converge to the global minimum after which we stop the simulation. This, however, greatly limits the methods performance

So instead of the Brownian motion methods as described here, it might be more effective to apply another method to reduce local minima sensitivity. For example, instead of consistent noise, it will probably be more effective to apply short periods of high noise once the gradient descent method has converged to a minimum. This way we can avoid the problem of simulations becoming so noisy that no minima can be found, while simultaneously ensuring it can't get stuck in a local minimum. In addition, there is no need to keep track of the minimum energy during the noisy periods, while the gradient descent method is not very sensitive to inclusions getting too close.

5.4 Alternative models

In addition to the two models we discussed in this thesis, there are several other models that describe interactions between inclusions on the membrane. We briefly looked at two of these models before determining that they were not suitable for our applications.

5.4.1 Kahraman's model

Kahraman et al. (2016) [21] developed a model to predict the spontaneous pattern formation by bacterial mechanosensitive channel of large conductance. This model describes inclusions that change the thickness of the membrane by separating or compressing the two layers of lipid molecules. It models

these inclusions by imposing the membrane thickness along a closed line that can be any shape.

This model has the advantage that it can be applied to inclusions of any closed shape. However, their model describes inclusions that change the thickness of the membrane. Our models, on the other hand, seek to describe inclusions that bend the membrane. Therefore, the model of Kahraman et al. is not a suitable alternative for our models.

5.4.2 Evans' model

Evans et al. [5] described a method of modeling inclusions by applying a force field to the membrane. The energy of the membrane is then given by:

$$E = \int d^2r \left[\frac{\kappa}{2}[\nabla^2 u(\mathbf{r})]^2 + \frac{\gamma}{2}|\nabla u(\mathbf{r})|^2 - f(\mathbf{r})u(\mathbf{r})\right] \tag{5.1}$$

where $f(\mathbf{r})$ is a force field that attempts to mimic the effect of a set of inclusions. The problem with this model for our applications is that there are no fixed boundary conditions. This means that inclusions can only interact through the energy functional. In addition, the highest order term in the energy functional is quadratic. This means that in this model, any curvature that is already present on the membrane has no influence on the interaction between two inclusions. This becomes especially clear when following the same methodology as Evans et al. to solve the system for three inclusions. One will see that there is an interaction term for any two inclusions, but not three-point interaction between all three. Thus, this model has been deemed unsuitable for our applications.

6 Conclusion

In this thesis, we investigated interactions between conical inclusions on a lipid bilayer membrane and made predictions about the patterns they form. We applied the formalism developed for the point particles model by Dommersnes and Fournier [1] to the finite size particles model developed by Weikl et al. [2]. We compare the results of both models for a system of three inclusions, to validate the point particles model's ability to accurately predict equilibrium patterns for conical inclusions. We attempted to describe a greater variety of non-conical inclusions by extending the point particles model to higher order constraints, but this proved inadequate, leaving only the computationally intensive finite size particles model to be used for these more complex inclusions.

The gradient descent method, a new numerical method for finding equilibrium patterns that we developed, turned out to be several hundred times faster than the standard Metropolis algorithm. Applying the gradient descent method to the point particles model, we found that four-inclusion square-shaped structures and six-inclusion butterfly-shaped structures are favored in all systems with more than six inclusions. For these large systems of inclusions, the new method is very sensitive to local minima and has difficulties merging small groups. The addition of noise in the Brownian motion method proved to be unable to resolve the local minima sensitivity. In future research, the local minima sensitivity can potentially be resolved by applying small bursts of high noise, instead of the consistent noise in the Brownian motion simulations. The difficulties of merging small groups can be resolved using a pattern recognition to recognize such groups, then moving the group as a whole, instead of the individual particles separately.

References

[1] 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*, vol. 83, pp. 2898–2905, dec 2002.

[2] T. R. Weikl, M. M. Kozlov, and W. Helfrich, "Interaction of conical membrane inclusions: Effect of lateral tension," vol. 57, no. 6, pp. 6988–6995, 1998.

[3] J. Darnell, H. Lodish, and D. Baltimore, *Molecular cell biology*. Scientific Amer Inc, may 1986.

[4] M. Goulian, R. Bruinsma, and P. Pincus, "Long-Range Forces in Heterogeneous Fluid Membranes," *Europhysics Letters (EPL)*, vol. 22, pp. 145–150, apr 1993.

[5] A. R. Evans, M. S. Turner, and P. Sens, "Interactions between proteins bound to biomembranes," *Physical Review E*, vol. 67, p. 041907, apr 2003.

[6] A. Vahid, A. Šarić, and T. Idema, "Curvature variation controls particle aggregation on fluid vesicles," *Soft Matter*, vol. 13, pp. 4924–4930, mar 2017.

[7] S. E. Feller, "Molecular dynamics simulations of lipid bilayers," *Current Opinion in Colloid & Interface Science*, vol. 5, no. 3-4, pp. 217–223, 2000.

[8] D. Tieleman, S. Marrink, and H. Berendsen, "A computer perspective of membranes: molecular dynamics studies of lipid bilayer systems," *Biochimica et Biophysica Acta (BBA) - Reviews on Biomembranes*, vol. 1331, no. 3, pp. 235–270, 1997.

[9] L. Saiz, S. Bandyopadhyay, and M. L. Klein, "Towards an Understanding of Complex Biological Membranes from Atomistic Molecular Dynamics Simulations," *Bioscience Reports*, vol. 22, no. 2, pp. 151–173, 2002.

[10] H. Agrawal, L. Liu, and P. Sharma, "Revisiting the curvature-mediated interactions between proteins in biological membranes," *Soft Matter*, vol. 12, no. 43, pp. 8907–8918, 2016.

[11] A. Srivastava and G. A. Voth, "Hybrid Approach for Highly Coarse-Grained Lipid Bilayer Models," *Journal of Chemical Theory and Computation*, vol. 9, pp. 750–765, jan 2013.

[12] B. J. Reynwar, G. Illya, V. A. Harmandaris, M. M. Müller, K. Kremer, and M. Deserno, "Aggregation and vesiculation of membrane proteins by curvature-mediated interactions," *Nature*, vol. 447, no. 7143, pp. 461–464, 2007.

[13] A. Vahid and T. Idema, "Pointlike Inclusion Interactions in Tubular Membranes," *Physical Review Letters*, vol. 117, no. 13, pp. 1–5, 2016.

[14] P. Sens and M. S. Turner, "Theoretical Model for the Formation of Caveolae and Similar Membrane Invaginations," *Biophysical Journal*, vol. 86, no. 4, pp. 2049–2057, 2004.

[15] W. Helfrich, "Elastic Properties of Lipid Bilayers: Theory and Possible Experiments," *Zeitschrift fur Naturforschung - Section C Journal of Biosciences*, vol. 28, pp. 693–703, jan 1973.

[16] J. Z. Y. Chen, "Swollen-to-Globular Transition of a Self-Avoiding Polymer Confined in a Soft Tube," *Physical Review Letters*, vol. 98, p. 088302, feb 2007.

[17] R. Los, *Self-Assembly and Pattern Formation of Oppositely Oriented Inclusions on a Membrane*. Bsc thesis, TU Delft, 2016.

[18] J. van Lopik, *Membrane mediated interactions between non-uniform inclusions*. Msc internship thesis, TU Delft, 2017.

[19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, vol. 21, pp. 1087–1092, jun 1953.

[20] E. Evans and D. Needham, "Physical properties of surfactant bilayer membranes: Thermal transitions, elasticity, rigidity, cohesion, and colloidal interactions," *Journal of Physical Chemistry*, vol. 91, no. 16, pp. 4219–4228, 1987.

[21] O. Kahraman, P. D. Koch, W. S. Klug, and C. A. Haselwandter, "Architecture and function of mechanosensitive membrane protein lattices," *Scientific Reports*, vol. 6, no. January, p. 19214, 2016.