MATH-498 Senior Research Project II

Numerical Simulation of Red Blood Cells Using a Reduced Order Modeling Approach

Final Report

by

Ayat Mohamad Alfahel

Student ID: 100059524

Supervisor:

Dr. Aymen Laadhari

Department of Mathematics

College of Computing and Mathematical Sciences

Khalifa University

Date of Submission: April 26, 2024

Abstract

This project aims to numerically simulate the deformations of red blood cell (RBC) membranes and to study their stationary shapes. Our approach simplifies the structure of the RBC membrane as a vesicle membrane due to its similarity to a phospholipid bilayer structure. We adopt the well-known Helfrich model, in which the behavior of the membrane is described by minimizing a bending energy depending on the square of the membrane curvature. This results in a highly nonlinear problem, presenting significant challenges in the field of computational fluid dynamics.

To solve this problem, we propose a relatively simplified modeling approach using tools from differential geometry. We formulate an optimization problem where the membrane minimizes its bending energy while preserving a fixed surface area and perimeter constraints. This leads to a saddle point problem, where two real Lagrange multipliers are introduced to enforce the constraints. We derive the optimality conditions and solve the resulting initial value problem using various numerical schemes, employing a shooting method.

Numerical experiments are carried out to evaluate the effectiveness of our approach and we successfully recover the characteristic biconcave shapes of red blood cells.

Contents

Chapter 1

Introduction

1.1 Motivation

Red blood cells (RBCs) are remarkable cellular entities that play a pivotal role in the circulation and transport of oxygen and carbon dioxide throughout the human body [1]. Their unique structure and composition, particularly the phospholipid biomembranes encapsulating these cells, form the foundation of their functionality and resilience within the dynamic environment of the bloodstream [2].

At the heart of a red blood cell lies its defining feature: the phospholipid bilayer membrane. It is comprised primarily of phospholipids, proteins, and cholesterol [3, 4]. Phospholipids, featuring hydrophilic heads and hydrophobic tails, form the basic building blocks of this bilayer, arranging themselves in a manner that fosters a selective and semi-permeable barrier [1, 5]. This selective permeability regulates the passage of ions, gases, and other molecules crucial for maintaining the cell's internal environment [4].

The phospholipid bilayer is not solely composed of phospholipids; integral and peripheral proteins are interspersed throughout this membrane [4]. Moreover, cholesterol molecules embedded within the phospholipid bilayer provide stability and fluidity to the red blood cell membrane. They modulate the packing of phospholipids, influencing the membrane's flexibility, crucial for the cell's ability to travel through the narrowest capillaries while maintaining its structural integrity [6].

Understanding the biology of red blood cells at the level of phospholipid biomembranes and membrane structure is fundamental not only to explain the mechanisms underlying their physiological functions, but also to comprehend various pathological conditions associated with membranes [1]. Disorders affecting these membranes, such as leukemia and sickle cell disease, emphasize the significance of maintaining the structural integrity of these membranes for the red blood cells' normal physiological functioning [4].

Vesicle membranes, similar to the phospholipid bilayer of red blood cells, represent artificially constructed lipid bilayers that mimic cellular membranes [1, 7]. These vesicles consist of amphiphilic lipid molecules arranged in bilayers, resembling the fundamental structure of biological membranes [8]. Vesicle membranes serve as powerful models for studying the properties and behaviors of natural cell membranes due to their ability to replicate membrane features, such as selective permeability and interactions with various molecules, proteins, and drugs [9, 1, 7].

In this research we aim to study vesicle membranes and their shape deformations. We try to mathematically model a phospholipid vesicle membrane and solve an optimization problem associated with minimizing its bending energy in order to understand modifications that occur to its structure and predict such occurrences.

1.2 Outline of The Present Work

The report is structured as follows. In Chapter 2, a literature review is presented. Previous work on modeling vesicle membranes and methodologies used to minimize the bending energy are discussed. Chapter 3 provides an overview of preliminary tools and fundamental concepts in differential geometry. These tools enable the formulation of the membrane problem within the framework of reduced order modeling. In Chapter 4, the mathematical description of the biological problem is discussed. The shapes of red blood cells are described through a reduced order model, obtained through a minimization problem under constraints. A saddle-point formulation is used, leading to the derivation of a simplified model for red blood cells. Chapter 5 discusses the numerical method set to solve the derived initial value problem. When the known RBC's biconcave shapes are obtained, a set of numerical tests is presented to help validate the model. Lastly, investigations are done on the model parameters and how they affect the shape obtained.

Chapter 2

Literature Review

Understanding the behavior of vesicle membranes under elastic bending energy is fundamental in biophysics and biomolecular engineering. Several studies have focused on explaining the complexities and numerical methodologies involved in minimizing bending energy within vesicle simulations.

The investigation by [10] delves deeply into the three-dimensional deformation of vesicle membranes under elastic bending energy. Their utilization of an energetic variational formulation enables an effective Eulerian description, capturing both static and dynamic deformations. Notably, their numerical experiments reveal intriguing phenomena with potential applications in real-world scenarios.

Complementary to this, [11] proposes a theory on the elasticity of lipid bilayers, emphasizing curvature as the primary elasticity governing nonspherical shapes in vesicles. Euler-Lagrange equations derived for shapes under external factors like magnetic fields and excess outside pressure expand our understanding of vesicle deformations. Additionally, suggested experimental approaches to determine elastic properties contribute to practical investigations in this field.

A different numerical methodology is introduced in [12], focusing on the modeling of biomembranes and capillary interfaces. Through a combination of a level set approach and high-order fully implicit time integration schemes, this method overcomes stability issues associated with nonlinear forces. Detailed experiments in two dimensions showcase the accuracy and efficiency of this methodology, highlighting its superiority over explicit schemes.

Addressing specific challenges in simulating red blood cells or vesicles, [13] presents a tailored numerical framework. This framework utilizes a penalty method for inextensibility constraints and higher-degree

finite elements for spatial discretization. Notably, it explores the influence of non-Newtonian rheology on system dynamics, providing insights into the behavior of vesicles in complex fluid environments.

Furthermore, [14] introduces an innovative approach employing shape optimization tools to derive a mechanical equilibrium equation for vesicle membranes under generalized elastic bending energy. This novel methodology offers a more concise derivation of the equilibrium equation compared to traditional tensorial tools, potentially streamlining analytical investigations in this domain.

One notable approach mentioned in the literature is the phase field approach, which has been applied to model the mechanical properties of RBC membranes [15, 16]. Additionally, the level set approach, implemented in both finite difference and finite element frameworks, has been utilized to study RBC deformations under different conditions [17]. Other methodologies such as the boundary integral method, immersed boundary method, and lattice Boltzmann method have also been employed to investigate the interaction between RBCs and the surrounding fluid dynamics [18, 19, 20].

However, many of these approaches suffer from limitations, particularly in addressing the coupling between the fluid and membrane dynamics. Fully explicit decoupling strategies, commonly used in classical finite element methods, often encounter numerical stability issues, especially when dealing with large interface deformations [21, 22]. Consequently, there has been a growing interest in developing implicit or semi-implicit coupling strategies to improve the stability and efficiency of numerical solvers for RBC modeling. Despite some recent advancements in this direction, such as semi-implicit strategies using the parametric finite element method [23, 24], challenges remain in achieving robust and efficient numerical solutions for the dynamics of RBCs in flow.

The collective body of literature showcases diverse methodologies, theoretical frameworks, and numerical techniques. These studies collectively contribute to advancing our understanding of vesicle deformation, offering insights crucial for biophysical and biomedical research.

Chapter 3

Preliminaries on Differential Geometry

3.1 Scalar and Vector Operators

Hereafter, we define some basic operators needed afterwards.

Gradient Operator

The gradient is an operator that represents the rate of change of a scalar or vector field. The gradient of a once-differentiable real-valued scalar function $f(x, y, z)$ is a vector given by

$$\boldsymbol{\nabla} f = (\partial_x f,\ \partial_y f,\ \partial_z f)^\top,$$

where $x, y, z \in \mathbb{R}$. The gradient of vector $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)^\top$ in $\mathbb{R}^3$ is given by:

$$\boldsymbol{\nabla} \boldsymbol{v} = \begin{pmatrix} \partial_x \boldsymbol{v}_1 & \partial_y \boldsymbol{v}_1 & \partial_z \boldsymbol{v}_1 \\ \partial_x \boldsymbol{v}_2 & \partial_y \boldsymbol{v}_2 & \partial_z \boldsymbol{v}_2 \\ \partial_x \boldsymbol{v}_3 & \partial_y \boldsymbol{v}_3 & \partial_z \boldsymbol{v}_3 \end{pmatrix}.$$

Divergence Operator

The divergence of a vector field is a vector operator that measures the rate at which the vectors of the field emanate from or converge toward a point. The divergence of a vector field $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)^\top$ in $\mathbb{R}^3$ is a scalar field given by:

$$\nabla \cdot \boldsymbol{v} = \partial_x \boldsymbol{v}_1 + \partial_y \boldsymbol{v}_2 + \partial_z \boldsymbol{v}_3,$$

where $x, y, z \in \mathbb{R}$.

Laplace Operator

The Laplace operator represents the second partial derivative of a scalar or vector function. It calculates the rate of change of that field's curvature or variation at a particular point, so it is equal to the divergence of the gradient. The Laplace operator of a twice-differentiable real-valued function $f(x, y, z)$ is a real-valued function expressed as

$$\Delta f = \nabla \cdot \boldsymbol{\nabla} f = \partial^2_{x^2} f + \partial^2_{y^2} f + \partial^2_{z^2} f,$$

where x, y and z represent the standard Cartesian coordinates of $\mathbb{R}^3$. The Laplace operator of a vector $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)^\top$ in $\mathbb{R}^3$ is computed by taking the Laplacian of each component. It writes:

$$\begin{aligned}\Delta \boldsymbol{v} &= \nabla \cdot \boldsymbol{\nabla} \boldsymbol{v} \\ &= \left(\partial^2_{x^2} \boldsymbol{v}_1 + \partial^2_{y^2} \boldsymbol{v}_1 + \partial^2_{z^2} \boldsymbol{v}_1,\ \partial^2_{x^2} \boldsymbol{v}_2 + \partial^2_{y^2} \boldsymbol{v}_2 + \partial^2_{z^2} \boldsymbol{v}_2,\ \partial^2_{x^2} \boldsymbol{v}_3 + \partial^2_{y^2} \boldsymbol{v}_3 + \partial^2_{z^2} \boldsymbol{v}_3\right)^\top.\end{aligned}$$

Curl Operator

The curl of a vector field is a vector operator that describes the rotation of a vector. A vector field that has a zero curl is called irrotational. In 3-dimensional Cartesian coordinates, the curl of a vector field $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3)^\top$ is another vector field given by:

$$\boldsymbol{\nabla} \times \boldsymbol{v} = (\partial_y \boldsymbol{v}_3 - \partial_z \boldsymbol{v}_2)\,\mathbf{i} + (\partial_z \boldsymbol{v}_1 - \partial_x \boldsymbol{v}_3)\,\mathbf{j} + (\partial_x \boldsymbol{v}_2 - \partial_y \boldsymbol{v}_1)\,\mathbf{k},$$

where $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ represent the unit vectors in the the x, y and z directions, respectively.

In curvilinear coordinates, we remind of certain known vector-calculus identities associated with the curl operator, which will be used later on.

- The curl of the gradient of any scalar field always results in the zero vector field; i.e. for any scalar field f,

$$\boldsymbol{\nabla} \times (\boldsymbol{\nabla} f) = \mathbf{0}.$$

- The divergence of the curl of any vector field is always equal to zero; i.e. for any vector field $\boldsymbol{v}$,

$$\nabla \cdot (\boldsymbol{\nabla} \times \boldsymbol{v}) = 0.$$

3.2 Curves in 2-Dimensional Space ($\mathbb{R}^2$)

A curve is a term used to describe the path of a continuously moving point. Such a path is usually generated by an equation of one dimension. A curve can always be expressed implicitly in terms of two spatial coordinates x and y in the form:

$$f(x, y) = 0,$$

where $x, y \in \mathbb{R}$. However, an explicit description for a curve cannot always be found. We write an explicit equation for a curve if we can isolate one variable and express it in terms of the other, in the form

$$y = f(x),$$

where f is here a function of x.

Unit Normal Vector

The unit normal vector to a curve is a unit vector perpendicular to the curve at a given point. The unit normal vector at x_0 on a curve $y = f(x)$ is given by

$$\mathbf{n} = \frac{(\partial_x f(x_0),\ -1)^\top}{\sqrt{(\partial_x f(x_0))^2 + 1}},$$

where $x, y, x_0 \in \mathbb{R}$.

Curvature of a Curve

The curvature of a curve is a measure of how much the curve deviates from being a straight line at a particular point on the curve. It quantifies how quickly the direction of the tangent vector to the curve changes as you move along the curve. In other words, it describes the rate of change of the curve's tangent vector. Let a curve be given by $\mathbf{s}(t) = (x(t), y(t))^\top$, then the curvature κ is defined by

$$\kappa = \left\| \frac{d\mathbf{T}}{d\mathbf{s}} \right\| = \frac{\left\| \frac{d\mathbf{T}}{dt} \right\|}{\left\| \frac{d\mathbf{s}}{dt} \right\|},$$

where $\mathbf{T}$ is the unit tangent vector and $\mathbf{s}$ is the arc length vector. The unit tangent vector is given by

$$\mathbf{T}(t) = \frac{s'(t)}{\|s'(t)\|}.$$

In two dimensions, the formula is simplified to:

$$\kappa = \frac{|x'y'' - y'x''|}{(x'^2 + y'^2)^{3/2}}.$$

3.3 Surfaces in 3-Dimensional Space ($\mathbb{R}^3$)

A surface denotes a topological space of dimension two. This means it can be implicitly expressed in terms of three variables, $x, y, z \in \mathbb{R}$, in the form

$$f(x, y, z) = 0,$$

where f is a function of x, y, and z. Similarly to curves, surfaces cannot always be expressed explicitly. If we can solve for z in the implicit equation of a surface, we can express it explicitly in the form

$$z = f(x, y),$$

where f is a function of x and y.

Unit Normal Vector

The unit normal vector to a surface is a vector which is perpendicular to the surface at a given point. The unit normal vector at a point (x_0, y_0) on a surface $z = f(x, y)$ is given by

$$\mathbf{n} = \frac{(\partial_x f(x_0, y_0),\ \partial_y f(x_0, y_0),\ -1)^\top}{\sqrt{(\partial_x f(x_0, y_0))^2 + (\partial_y f(x_0, y_0))^2 + 1}},$$

where $x, y, z, x_0, y_0 \in \mathbb{R}$.

Normal Curvature

The normal curvature of a surface at a given point is a measure of how much the surface curves in the direction of its unit normal vector at that point. It describes the curvature of the surface in the direction perpendicular to the tangent plane at that point. It is expressed as the dot product of the surface's unit

normal vector and the derivative of the unit tangent vector with respect to arc length. That is,

$$\kappa_n = \mathbf{n} \cdot \frac{d\mathbf{T}}{d\mathbf{s}},$$

where κ_n is the normal curvature and $\mathbf{n}$ is the unit normal vector to the surface at that point.

We will introduce the principal curvatures, which are two distinct curvatures associated with the normal curvatures of a surface at a specific point. The first principal curvature, denoted as κ_1, corresponds to the maximum normal curvature of the surface at that point, while the second principal curvature, denoted as κ_2, represents the minimum normal curvature.

Mean Curvature

The mean curvature, H, is a measure of the sum of the principal curvatures of a surface at a particular point. It is defined by

$$H = \kappa_1 + \kappa_2.$$

The mean curvature of a surface denoted by

$$\Sigma = \{(x, y, z) \in \mathbb{R}^3 : \phi(x, y, z) = 0\}$$

is given by the divergence of the unit normal vector. Consider

$$\mathbf{n} = (n_1,\ n_2,\ n_3)^\top.$$

Hence, the mean curvature writes

$$H = \nabla \cdot \mathbf{n} = \partial_x n_1 + \partial_y n_2 + \partial_z n_3.$$

We introduce the gradient of $\mathbf{n}$ which writes

$$\boldsymbol{\nabla}\mathbf{n} = \begin{pmatrix} \partial_x n_1 & \partial_y n_1 & \partial_z n_1 \\ \partial_x n_2 & \partial_y n_2 & \partial_z n_2 \\ \partial_x n_3 & \partial_y n_3 & \partial_z n_3 \end{pmatrix}.$$

This tensor can be written in the compact form $(\boldsymbol{\nabla}\mathbf{n})_{ij} = \partial_j n_i$, with $1 \leq i \leq 3$ and $1 \leq j \leq 3$.

Let us now introduce ϕ as a signed distance to the surface Σ and denote it by δ. Accordingly, we can

express

$$\Sigma = \{\mathbf{x} \in \mathbb{R}^3 : \delta(\mathbf{x}) = 0\}.$$

Here, Ω denotes the interior domain, and Σ corresponds to its boundary, so that $\Sigma = \partial\Omega$. The signed distance function is formulated as follows:

$$\delta : \begin{cases} \mathbb{R}^3 \to \mathbb{R} \\ \mathbf{x} \mapsto \begin{cases} \min_{y \in \Sigma} \quad |\mathbf{x} - \mathbf{y}|, & \text{if } \mathbf{x} \in \mathbb{R}^3 \backslash \Omega \\ -\min_{y \in \Sigma} \quad |\mathbf{x} - \mathbf{y}|, & \text{if } \mathbf{x} \in \Omega. \end{cases} \end{cases}$$

It follows that $||\boldsymbol{\nabla}\delta|| = 1$, see e.g. [25]. Therefore, we obtain $\mathbf{n} = \dfrac{\boldsymbol{\nabla}\delta}{||\boldsymbol{\nabla}\delta||} = \boldsymbol{\nabla}\delta$, and $\boldsymbol{\nabla}\mathbf{n} = \boldsymbol{\nabla}\boldsymbol{\nabla}\delta$.

Now, consider vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ in $\mathbb{R}^3$. The cross product of $\boldsymbol{u}$ and $\boldsymbol{v}$ is denoted as $\boldsymbol{u} \times \boldsymbol{v}$, while their dot product is represented by $\boldsymbol{u} \cdot \boldsymbol{v}$. Assuming sufficient regularity, the subsequent identity is valid for all vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ (see for example [26]):

$$\boldsymbol{\nabla}(\boldsymbol{u} \cdot \boldsymbol{v}) = (\boldsymbol{u} \cdot \boldsymbol{\nabla})\boldsymbol{v} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{u} + \boldsymbol{u} \times (\boldsymbol{\nabla} \times \boldsymbol{v}) + \boldsymbol{v} \times (\boldsymbol{\nabla} \times \boldsymbol{u}).$$

We will set $\boldsymbol{u} = \boldsymbol{v} = \mathbf{n}$. Then,

$$\boldsymbol{\nabla}(\mathbf{n} \cdot \mathbf{n}) = 2(\mathbf{n} \cdot \boldsymbol{\nabla})\mathbf{n} + 2\mathbf{n} \times (\boldsymbol{\nabla} \times \mathbf{n})$$
$$\boldsymbol{\nabla}||\mathbf{n}||^2 = 2(\boldsymbol{\nabla}\mathbf{n})\mathbf{n} + 2\mathbf{n} \times (\boldsymbol{\nabla} \times \boldsymbol{\nabla}\delta).$$

We know that $\boldsymbol{\nabla} \times \boldsymbol{\nabla}\delta = \mathbf{0}$ since δ is a scalar function, so the equation becomes

$$\mathbf{0} = 2(\boldsymbol{\nabla}\mathbf{n})\mathbf{n}$$
$$\mathbf{0} = \boldsymbol{\nabla}\mathbf{n} \cdot \mathbf{n}.$$

The relationship $\boldsymbol{\nabla}\mathbf{n} \cdot \mathbf{n} = \mathbf{0}$ shows us that $\boldsymbol{\nabla}\mathbf{n}$ is a matrix with eigenvalue $\lambda = 0$ corresponding to an eigenvector $\mathbf{n}$ according to the following relationship between eigenvalues and eigenvectors:
For a square matrix A, if there exists a vector $\boldsymbol{v}$ and a scalar λ such that

$$A\boldsymbol{v} = \lambda\boldsymbol{v},$$

where $\boldsymbol{v}$ is a non-zero vector, then $\boldsymbol{v}$ is an eigenvector of A, and λ is the corresponding eigenvalue.

Let us go back to our case where $\boldsymbol{\nabla}\mathbf{n} = \boldsymbol{\nabla}\boldsymbol{\nabla}\delta$. This means that

$$\boldsymbol{\nabla}\mathbf{n} = \boldsymbol{\nabla}\left(\partial_x\delta,\ \partial_y\delta,\ \partial_z\delta\right)^\top, \quad \text{so} \quad \boldsymbol{\nabla}\mathbf{n} = \begin{pmatrix} \partial^2_{x^2}\delta & \partial^2_{yx}\delta & \partial^2_{zx}\delta \\ \partial^2_{xy}\delta & \partial^2_{y^2}\delta & \partial^2_{zy}\delta \\ \partial^2_{xz}\delta & \partial^2_{yz}\delta & \partial^2_{z^2}\delta \end{pmatrix}.$$

We can see that $\boldsymbol{\nabla}\mathbf{n}$ is a real symmetric matrix; i.e. $\partial_i\partial_j\delta = \partial_j\partial_i\delta \quad \forall i, j$. This means that $\boldsymbol{\nabla}\mathbf{n} = (\boldsymbol{\nabla}\mathbf{n})^\top$. Hence, $\boldsymbol{\nabla}\mathbf{n}$ has only real eigenvalues and can be diagonalized in the form $\boldsymbol{\nabla}\mathbf{n} = QXQ^\top$ where $QQ^\top = Q^TQ = I$, because Q is orthogonal [27]. X is a diagonal matrix with the eigenvalues, Q is a matrix with the eigenvectors as its columns, and I is the identity matrix.

We know that one of the eigenvalues of $\boldsymbol{\nabla}\mathbf{n}$, corresponding to eigenvector $\mathbf{n}$, is equal to zero. The two other eigenvalues are the principal curvatures κ_1 and κ_2. This means that X and Q can be expressed as follows:

$$X = \begin{pmatrix} \kappa_1 & 0 & 0 \\ 0 & \kappa_2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad Q = \begin{pmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \mathbf{n} \end{pmatrix}.$$

$\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are the eigenvectors corresponding to the principal curvatures. We can see that the mean curvature of the surface is equal to the trace of the gradient of the unit normal vector. That is

$$H = tr(\boldsymbol{\nabla}\mathbf{n}) = \kappa_1 + \kappa_2.$$

Example Let us consider the example where Γ is a circle of radius R centered around the origin. That is $\Gamma = \{(x, y) \in \mathbb{R}^2 \mid \phi(x, y) = \sqrt{x^2 + y^2} - R = 0\}$. The unit normal vector and mean curvature of the curve would be as follows:

1. $\mathbf{n} \equiv \dfrac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||} = \left(\dfrac{x}{\sqrt{x^2+y^2}}, \dfrac{y}{\sqrt{x^2+y^2}}\right) = \left(\dfrac{x}{R},\ \dfrac{y}{R}\right).$
2. $H = \nabla \cdot \mathbf{n} = \partial_x\left(\dfrac{x}{\sqrt{x^2+y^2}}\right) + \partial_y\left(\dfrac{y}{\sqrt{x^2+y^2}}\right) = \dfrac{1}{\sqrt{x^2+y^2}} = \dfrac{1}{R}.$

We assume that $H > 0$ for a curve if the curve is convex and $H < 0$ if it is concave.

Gaussian Curvature

The Gaussian curvature, denoted as K, also describes the intrinsic curvature of a surface at a specific point. It is determined by the product of the principal curvatures, κ_1 and κ_2, defined as follows:

$$K = \kappa_1 \kappa_2$$

The relationship between the principal curvatures κ, mean curvature H, and Gaussian curvature K is represented by the equation:

$$\kappa^2 - H\kappa + K = 0.$$

The Gaussian curvature does not influence the membrane dynamics in the two-dimensional case and will not be considered for the remainder of this study, see e.g. [12].

Chapter 4

Mathematical Modeling

4.1 Membrane Energy

For an accurate representation of biomembrane mechanics, we turn to a renowned work based on W. Helfrich's model [28] and subsequent studies by Evans [29]. Building on continuum theory, numerous experiments have been carried out to characterize the mechanical response of lipid membranes, including the impact of polymers or proteins. A comprehensive overview of the main experimental results and methodologies used over the past decades is provided in [30].

Let $\Gamma \in \mathbb{R}^2$ represent the cell membrane. The bending energy, which is dependent on the mean curvature of the membrane, can be expressed as:

$$E = \frac{k}{2} \int_{\Gamma} (H - H_0)^2 \, \mathrm{d}s,$$

where k denotes the bending rigidity constant. In this work, we consider k to have a unit value. The spontaneous curvature H_0 is utilized to depict the asymmetry effect of the membrane and its surrounding environment, and in the two-dimensional scenario, it does not influence the cell's shape, as explained in [12]. According to the Helfrich's model [28], the membrane must minimize its bending energy.

The RBC's shape is shown to depend on a dimensionless parameter, referred to as the circularity number χ, see e.g. [31]. It described the ratio between the actual membrane's area A with the area of a circle having the same perimeter P as the vesicle. It is expressed as:

$$\chi \equiv \frac{A}{\pi \times \left(\frac{P}{2\pi}\right)^2} = \frac{4\pi A}{P^2}.$$

4.2 Problem Formulation: Minimization Under Constraints

Consider a curve that represents the vesicle's membrane in $\mathbb{R}^2$. It is described in an implicit way as follows:

$$\Gamma = \{(x,\ y) \in \mathbb{R}^2 : \phi(x, y) = h(x) - y = 0\}.$$

This is known as the level set representation in the field of computational fluid dynamics [32, 33]. We only consider the part of the curve present in the first quadrant because we assume shape symmetry with respect to the x-axis and y-axis. That is, we set $x \in [0,\ x_m]$, where $x_m \in \mathbb{R}^+$ represents the maximum stretching in the x direction. We assume $h \in C^7([0,\ x_m])$. Due to shape symmetry, we also know that $h'(0) = 0$ and $h'(x_m) = \infty$. Therefore, the membrane represents the boundary of the interior domain denoted by Ω so that $\Gamma = \partial\Omega$. See Fig. 4.1.

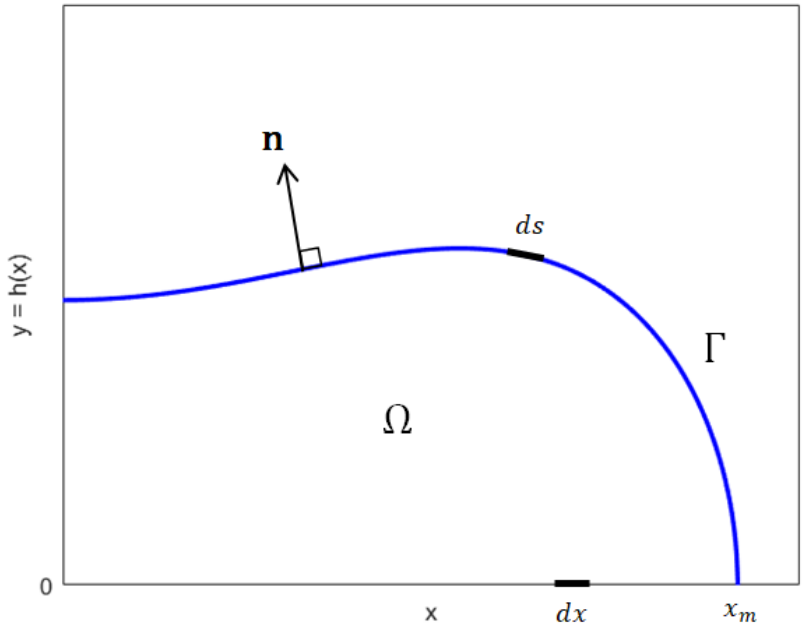


Figure 4.1: Sketch for one-fourth of the curve representing the vesicle's membrane.

We can express the following geometric quantities and fields:

1. Length $L \equiv |\Gamma| = 4\int_0^{x_m} \sqrt{1 + (h'(x))^2}\, dx.$
2. Area $A = 4\iint_\Omega dL = 4\int_0^{x_m} h(x)\, dx.$
3. The unit normal vector $\mathbf{n} = \dfrac{\boldsymbol{\nabla}\phi}{||\boldsymbol{\nabla}\phi||} = \dfrac{(h'(x), 1)}{\sqrt{(h'(x))^2 + 1}}.$
4. The mean curvature $H = \nabla \cdot \mathbf{n} = \partial_x \left(\dfrac{h'(x)}{\sqrt{(h'(x))^2 + 1}} \right) + \partial_y \left(\dfrac{1}{\sqrt{(h'(x))^2 + 1}} \right) = \dfrac{h''(x)}{((h'(x))^2 + 1)^{3/2}}.$

For simplification, we consider a first change of variable $h'(x) = w(x)$, so that $h''(x) = w'(x)$. Hence,

$$H = \frac{w'(x)}{((w(x))^2 + 1)^{3/2}}$$

and the bending energy can be formulated as

$$\begin{aligned} E(h) &= \frac{1}{2}\int_\Gamma (H - H_0)^2 \, ds \\ &= \frac{1}{2}\int_0^{x_m} \left(\frac{h''(x)}{((h'(x))^2 + 1)^{3/2}} - H_0 \right)^2 \sqrt{(h'(x))^2 + 1}\, dx \\ &= \frac{1}{2}\int_0^{x_m} \left(\frac{w'(x)}{((w(x))^2 + 1)^{3/2}} - H_0 \right)^2 \sqrt{(w(x))^2 + 1}\, dx, \end{aligned}$$

Circular membrane Assume that Γ is a circle of radius R centered around the origin. That is,

$$\Gamma = \{(x, y) \in \mathbb{R}^2 : \phi(x, y) = \sqrt{x^2 + y^2} - R = 0\}.$$

The area of the inner domain and membrane perimeter can be expressed as:

1. $L \equiv |\Gamma| = 4\int_0^R \sqrt{1 + \left(\frac{dy}{dx}\right)^2}\, dx = 4\int_0^R \sqrt{1 + \left(-\frac{x}{\sqrt{R^2 - x^2}}\right)^2}\, dx$
2. $A \equiv |\Omega| = 4\int_0^R \sqrt{R^2 - x^2}\, dx$

4.3 Derivation of The Reduced Order Model

The optimization problem writes:

$$\begin{aligned} \min_h \quad & \frac{1}{2}\int_\Gamma (H - H_0)^2 \, ds \\ \text{s.t.} \quad & L = L_0, \\ & A = A_0, \end{aligned}$$

where L_0 and A_0 are the length and area evaluated at the initial time. We refer to the section Appendix: Optimization with Equality Constraints for additional information regarding the approach to address equations involving a minimization problem under constraints.

The Lagrangian function associated with this constrained optimization problem is given by

$$\begin{aligned}
\mathcal{L}(h,\lambda,p) &= \frac{1}{2}\int_{\Gamma}(H-H_0)^2\,ds + \lambda(L-L_0) + p(A-A_0) \\
&= \frac{1}{2}\int_0^{x_m}\left(\frac{h''(x)}{((h'(x))^2+1)^{3/2}} - H_0\right)^2\sqrt{(h'(x))^2+1}\,dx \\
&\quad + \lambda\left(4\int_0^{x_m}\sqrt{(h'(x))^2+1}\,dx - L_0\right) + p\left(4\int_0^{x_m}h(x)\,dx - A_0\right) \\
&= \frac{1}{2}\int_0^{x_m}\left(\frac{w'(x)}{((w(x))^2+1)^{3/2}} - H_0\right)^2\sqrt{(w(x))^2+1}\,dx \\
&\quad + \lambda\left(4\int_0^{x_m}\sqrt{(w(x))^2+1}\,dx - L_0\right) + p\left(-4\int_0^{x_m}xw(x)\,dx - A_0\right).
\end{aligned}$$

To solve the optimization problem, the optimality conditions must be verified. Accordingly, the derivatives of the Lagrangian functions with respect to w, λ, and p must be equal to 0. To impose that, we will first introduce the directional derivative.

The directional derivative of a function $g(x)$ in the direction of a function ψ at a point x is a concept from calculus used in the context of Gateaux derivatives. Let $g : \mathbb{R}^n \to \mathbb{R}$ be a real-valued function of a variable x and ψ be a function in the same space. The directional derivative of $g(x)$ in the direction of ψ at the point x is denoted by $D_\psi g(x)$. It is defined as:

$$D_\psi g(x) = \lim_{t\to 0}\frac{g(x+t\psi)-g(x)}{t}.$$

This derivative measures the rate of change of the function g at the point x in the direction of ψ. It represents the slope of the function along the direction specified by ψ at the point x.

Let us consider a space of admissible functions w and ψ

$$\mathbb{V}(\alpha,\beta) = \left\{v \in C^2([0,x_m]) : v(0)=\alpha; v(x_m)=\beta\right\}.$$

We know that $w(0)=0$ and $w(x_m)=\infty$. Now let us consider ψ to be a test function. We can define w and ψ as follows:

$$\begin{aligned}
&w \in \mathbb{V}(0,\infty) \text{ and} \\
&\psi \in \mathbb{V}(0,0).
\end{aligned}$$

Using the definition of directional derivatives, we find that

$$\begin{aligned} D_\psi(w') &= \lim_{t\to 0} \frac{(w+t\psi)' - w'}{t} = \psi', \\ D_\psi(w^2+1) &= \lim_{t\to 0} \frac{(w+t\psi)^2 + 1 - (w^2+1)}{t} = 2w\psi, \text{ and} \\ D_\psi(w) &= \lim_{t\to 0} \frac{w+t\psi - w}{t} = \psi. \end{aligned}$$

We can now differentiate the directional derivative of $\mathcal{L}(w,\lambda,p)$, with respect to w, in the direction of the test function ψ, denoted as $\partial_w \mathcal{L}\,[\psi]$.

$$\begin{aligned} \partial_w \mathcal{L}[\psi] = \frac{1}{2}\int_0^{x_m} &\left[2\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{\psi'(w^2+1)^{3/2} - 3ww'\psi(w^2+1)^{1/2}}{(w^2+1)^{5/2}}\right)\right. \\ &\left. + \left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)^2\left(\frac{w\psi}{(w^2+1)^{1/2}}\right)\right] dx + 4\lambda\int_0^{x_m} \frac{w\psi}{(w^2+1)^{1/2}}\,dx \\ &- 4p\int_0^{x_m} x\psi\,dx \\ \partial_w \mathcal{L}\,[\psi] = &\int_0^{x_m}\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{\psi'}{w^2+1}\right)dx \\ &- \int_0^{x_m}\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{3ww'\psi}{(w^2+1)^2}\right)dx \\ &+ \int_0^{x_m}\left[\frac{1}{2}\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)^2 + 4\lambda\right]\left(\frac{w\psi}{(w^2+1)^{1/2}}\right)dx - 4p\int_0^{x_m} x\psi\,dx. \end{aligned}$$

Let $u = \left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{1}{w^2+1}\right)$ and $dv = \psi'$. Then by integration by parts,

$$\begin{aligned} \int_0^{x_m}\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{\psi'}{w^2+1}\right)dx = &\left[\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{\psi}{w^2+1}\right)\right]_0^{x_m} + \\ \int_0^{x_m}\psi\left[\frac{w''(w^2+1)^{3/2} - 3ww'^2(w^2+1)^{1/2}}{(w^2+1)^4} - \frac{2ww'}{(w^2+1)^2}\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\right]dx. \end{aligned}$$

We know that

$$\left[\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{\psi}{w^2+1}\right)\right]_0^{x_m} = 0,$$

so the expression for $\partial_w \mathcal{L}\,[\psi]$ becomes

$$\begin{aligned} \partial_w \mathcal{L}[\psi] = \int_0^{x_m}\psi\left[\left(\frac{2ww'}{(w^2+1)^2}\right)\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\right. \\ - \left(\frac{w''(w^2+1)^{3/2} - 3ww'^2(w^2+1)^{1/2}}{(w^2+1)^3}\right)\left(\frac{1}{w^2+1}\right) \\ - \left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)\left(\frac{3ww'}{(w^2+1)^2}\right) + \\ \left.\left(\frac{1}{2}\left(\frac{w'}{(w^2+1)^{3/2}} - H_0\right)^2 + 4\lambda\right)\left(\frac{w}{(w^2+1)^{1/2}}\right) - 4px\right]dx. \end{aligned}$$

Equating $\partial_w \mathcal{L}[\psi]$ to 0 will lead us to the ordinary differential equation (ODE)

$$\left(\frac{2ww'}{(w^2+1)^2}\right)\left(\frac{w'}{(w^2+1)^{3/2}}-H_0\right)-\left(\frac{w''(w^2+1)^{3/2}-3ww'^2(w^2+1)^{1/2}}{(w^2+1)^3}\right)\left(\frac{1}{w^2+1}\right)$$
$$-\left(\frac{w'}{(w^2+1)^{3/2}}-H_0\right)\left(\frac{3ww'}{(w^2+1)^2}\right)+\left(\frac{1}{2}\left(\frac{w'}{(w^2+1)^{3/2}}-H_0\right)^2+4\lambda\right)\left(\frac{w}{(w^2+1)^{1/2}}\right)$$
$$-4px=0.$$

This equation simplifies to

$$\frac{w''}{(w^2+1)^{5/2}}-\frac{5ww'^2}{2(w^2+1)^{7/2}}-\frac{(H_0^2+8\lambda)(w)}{2(w^2+1)^{1/2}}+4px=0. \tag{4.1}$$

To further simplify the equation, we will use the expression that appears in the third term of the ODE 4.1 to introduce a second change of variables

$$\eta=\frac{w}{(w^2+1)^{1/2}}.$$

It follows that

$$\eta'=\frac{w'}{(w^2+1)^{3/2}} \text{ and}$$
$$\eta''=\frac{w''}{(w^2+1)^{3/2}}-\frac{3ww'^2}{(w^2+1)^{5/2}}.$$

Note that η corresponds to the first component of the unit normal vector $\mathbf{n}$ of the function $h(x)$, and η' corresponds to divergence of this unit normal vector, or the mean curvature H, of the same function. We obtain the second term of the ODE 4.1 through the following:

$$\frac{5ww'^2}{2(w^2+1)^{7/2}}=\frac{5}{2}\left(\frac{w}{(w^2+1)^{1/2}}\right)\left(\frac{w'}{(w^2+1)^{3/2}}\right)^2$$
$$\frac{5ww'^2}{2(w^2+1)^{7/2}}=\frac{5}{2}\eta\eta'^2.$$

We then express w in terms of η to find the first term.

$$\eta=\frac{w}{(w^2+1)^{1/2}}, \text{ so}$$
$$w=\frac{\pm\eta}{\sqrt{1-\eta^2}}.$$

Based on the definition of η, we know that η and w have the same sign, so, we eliminate the negative expression for w and keep

$$w=\frac{\eta}{\sqrt{1-\eta^2}}. \tag{4.2}$$

The first term of the ODE 4.1 is then given by

$$\begin{aligned}
\eta'' &= \frac{w''}{(w^2+1)^{3/2}} - \frac{3ww'^2}{(w^2+1)^{5/2}} \\
\frac{w''}{(w^2+1)^{3/2}} &= \eta'' + \frac{3ww'^2}{(w^2+1)^{5/2}} \\
\frac{w''}{(w^2+1)^{5/2}} &= \frac{\eta''}{w^2+1} + \frac{3ww'^2}{(w^2+1)^{7/2}} \\
\frac{w''}{(w^2+1)^{5/2}} &= \frac{\eta''}{\left(\frac{\eta}{\sqrt{1-\eta^2}}\right)^2+1} + 3\eta\eta'^2 \\
\frac{w''}{(w^2+1)^{5/2}} &= \eta''\left(1-\eta^2\right) + 3\eta\eta'^2.
\end{aligned}$$

Lastly, the ODE 4.1 becomes

$$\eta''\left(1-\eta^2\right) + \frac{1}{2}\eta\eta'^2 - \frac{H_0^2+8\lambda}{2}\eta + 4px = 0$$

or after multiplying by 2,

$$2\eta''\left(1-\eta^2\right) + \eta\eta'^2 - \left(H_0^2+8\lambda\right)\eta + 8px = 0. \tag{4.3}$$

Given that $w(0) = 0$ and $w'(0) = \gamma$, we find that $\eta(0) = 0$ and $\eta'(0) = \gamma$. This gives us the following 2nd order initial value problem (IVP):

$$2\eta''\left(1-\eta^2\right) + \eta\eta'^2 - \left(H_0^2+8\lambda\right)\eta + 8px = 0, \quad 0 \le x \le x_m, \quad \eta(0) = 0, \quad \eta'(0) = \gamma. \tag{4.4}$$

We can transform this IVP into a system from the 1st order. First, we solve the ODE 4.3 for η''. This yields to

$$\eta'' = \frac{(H_0^2+8\lambda)\eta - \eta\eta'^2 - 8px}{2(1-\eta^2)}.$$

Then, we define a new vector $\boldsymbol{\xi} \equiv (\xi_1,\ \xi_2)^\top = (\eta,\ \eta')^\top$. It follows that

$$\boldsymbol{\xi}' = (\eta',\ \eta'')^\top = \left(\eta',\ \frac{(H_0^2+8\lambda)\eta - \eta\eta'^2 - 8px}{2(1-\eta^2)}\right)^\top = \left(\xi_2,\ \frac{(H_0^2+8\lambda)\xi_1 - \xi_1\xi_2^2 - 8px}{2(1-\xi_1^2)}\right)^\top = f\left(x,\ \boldsymbol{\xi}\right).$$

Then the IVP 4.4 becomes

$$\boldsymbol{\xi}' = \left(\xi_2,\ \frac{(H_0^2+8\lambda)\xi_1 - \xi_1\xi_2^2 - 8px}{2(1-\xi_1^2)}\right)^\top, \quad 0 \le x \le x_m, \quad \boldsymbol{\xi}(0) = (0,\ \gamma)^\top. \tag{4.5}$$

This is a system of two 1st order ODEs that includes parameters H_0, λ, p, and γ.

Chapter 5

Numerical Approximation

5.1 Numerical Strategy

Hereafter, we will introduce the numerical methods used for approximating the solutions to the devised initial value problem. Consider the following compact form of the initial value problem:

$$\boldsymbol{\xi}' = f\left(x,\, \boldsymbol{\xi}\right), \quad 0 \leq x \leq x_m, \quad \boldsymbol{\xi}(0) = (0,\, \gamma)^\top,$$

where

$$f\left(x, \boldsymbol{\xi}\right) = \left(\xi_2,\, \frac{(H_0^2 + 8\lambda)\xi_1 - \xi_1\xi_2^2 - 8px}{2(1-\xi_1^2)}\right)^\top.$$

In the following, we will introduce several numerical methods characterized by distinct convergence behaviors, each designed to address the initial value problem mentioned earlier. Subsequently, we will present the overall algorithm detailing the entire implemented numerical strategy.

Euler's Method

Euler's Method is a first-order method that uses linear approximation to estimate the solution. The method involves taking small steps along the slope of the derivative at each point. The algorithm for the method is as follows:

1. Start with an initial values for the independent variable x and the dependent variable $\boldsymbol{\xi}$, 0 and $\boldsymbol{\xi}(0)$, respectively.
2. Choose a step size, Δx, which determines how far along the x-axis to move in each iteration.

3. Update the values of $\boldsymbol{\xi}$ and x using the formula:

$$\boldsymbol{\xi}_{n+1} = \boldsymbol{\xi}_n + \Delta x f(x_n,\ \boldsymbol{\xi}_n)$$

$$x_{n+1} = x_n + \Delta x$$

4. Repeat steps 3-5 until you reach the desired endpoint.

The error in the Euler method is directly proportional with the step size (Δx). It is a first-order method, where the error can be expressed in $O(\Delta x)$, indicating that it diminishes linearly as the step size decreases. Consequently, greater accuracy in the approximations can be achieved by reducing the step size when using the Euler method. The error equation is expressed as follows:

$$|\boldsymbol{\xi}(x_n) - \boldsymbol{\xi}_n| \leq C \cdot \Delta x,$$

where C is a constant that depends on the maximum value of the derivative of the function f over the interval $0 \leq x \leq x_m$.

Modified Euler Method (RK2)

The modified Euler method is a second-order numerical method. It improves upon Euler's method by taking a more accurate approximation of the slope within each step. Below are the steps to implement this method:

1. Start with the initial values 0 and $\boldsymbol{\xi}(0)$.
2. Choose a step size Δx.
3. At each step:
   (a) Calculate the slope K_1 at the current point:

$$K_1 = f(x_n,\ \boldsymbol{\xi}_n)$$

   (b) Calculate the slope K_2 at x_{n+1} using an estimation of the value of $\boldsymbol{\xi}$ at x_{n+1} equal to $\boldsymbol{\xi}_n + K_1 \Delta x$:

$$K_2 = f(x_{n+1},\ \boldsymbol{\xi}_n + K_1 \Delta x)$$

(c) Update $\boldsymbol{\xi}$ using the average of the slopes:

$$\boldsymbol{\xi}_{n+1} = \boldsymbol{\xi}_n + \frac{\Delta x}{2}(K_1 + K_2)$$

4. Repeat steps 3 for each step until the desired endpoint is reached.

Based on the previous steps,

$$\boldsymbol{\xi}_{n+1} = \boldsymbol{\xi}_n + \frac{\Delta x}{2}\left[f(x_n,\, \boldsymbol{\xi}_n) + f(x_{n+1},\, \boldsymbol{\xi}_n + \Delta x f(x_n, \boldsymbol{\xi}_n))\right].$$

The modified Euler method improves upon the Euler method. This improvement results in a smaller error in the order of $O(\Delta x^2)$. The method exhibits second-order accuracy, meaning that the error decreases quadratically with the step size. This indicates a higher speed in providing an accurate approximation to the solution compared to the Euler method. The error equation for the modified Euler method writes as:

$$|\boldsymbol{\xi}(x_n) - \boldsymbol{\xi}_n| \leq C \cdot \Delta x^2,$$

where C is a constant that depends on the maximum value of the second derivative of the function f over the interval $0 \leq x \leq x_m$.

Fourth-Order Runge-Kutta Method (RK4)

RK4 is a higher-order numerical method for solving ordinary differential equations. It is one of the most widely used numerical schemes due to its balance between accuracy and computational efficiency. The following steps outline the algorithm for this method:

1. Start with the initial values 0 and $\boldsymbol{\xi}(0)$.
2. Choose a step size Δx.
3. At each step:
   (a) Calculate the slope K_1 at the current point:

$$K_1 = f(x_n,\, \boldsymbol{\xi}_n)$$

(b) Calculate the slope K_2 at $x_n + \frac{\Delta x}{2}$ using $\boldsymbol{\xi}_n + \frac{\Delta x}{2} K_1$:

$$K_2 = f\left(x_n + \frac{\Delta x}{2},\ \boldsymbol{\xi}_n + \frac{\Delta x}{2} K_1\right)$$

(c) Calculate the slope K_3 at $x_n + \frac{\Delta x}{2}$ using $\boldsymbol{\xi}_n + \frac{\Delta x}{2} K_2$:

$$K_3 = f\left(x_n + \frac{\Delta x}{2},\ \boldsymbol{\xi}_n + \frac{\Delta x}{2} K_2\right)$$

(d) Calculate the slope K_4 at x_{n+1} using $\boldsymbol{\xi}_n + K_3 \Delta x$:

$$K_4 = f(x_{n+1},\ \boldsymbol{\xi}_n + K_3 \Delta x)$$

(e) Update $\boldsymbol{\xi}$ using the weighted average of the slopes:

$$\boldsymbol{\xi}_{n+1} = \boldsymbol{\xi}_n + \frac{\Delta x}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

4. Repeat steps 3 for each step until the desired endpoint is reached.

The previous steps give us the following evaluation for $\boldsymbol{\xi}_{n+1}$:

$$\begin{aligned}\boldsymbol{\xi}_{n+1} = \boldsymbol{\xi}_n + \frac{\Delta x}{6}\Bigg[& f(x_n,\ \boldsymbol{\xi}_n) + 2f\left(x_n + \frac{\Delta x}{2},\ \boldsymbol{\xi}_n + \frac{\Delta x}{2} f(x_n,\ \boldsymbol{\xi}_n)\right) \\ & + 2f\left(x_n + \frac{\Delta x}{2},\ \boldsymbol{\xi}_n + \frac{\Delta x}{2} f\left(x_n + \frac{\Delta x}{2},\ \boldsymbol{\xi}_n + \frac{\Delta x}{2} f(x_n,\ \boldsymbol{\xi}_n)\right)\right) \\ & + f\left(x_{n+1},\ \boldsymbol{\xi}_n + \Delta x f\left(x_n + \frac{\Delta x}{2},\ \boldsymbol{\xi}_n + \frac{\Delta x}{2} f\left(x_n + \frac{\Delta x}{2},\ \boldsymbol{\xi}_n + \frac{\Delta x}{2} f(x_n,\ \boldsymbol{\xi}_n)\right)\right)\right)\Bigg].\end{aligned}$$

The RK4 method is a fourth-order numerical method. Its error is typically in the order of $O(\Delta x^4)$, making it faster than both the Euler and the modified Euler methods in approximating the solution accurately. This is due to the fact that its error decreases quartically with the step size. Its error equation is:

$$|\boldsymbol{\xi}(x_n) - \boldsymbol{\xi}_n| \leq C \cdot \Delta x^4,$$

where C is a constant that depends on the maximum value of the fourth derivative of the function f over the interval $0 \leq x \leq x_m$.

Error Analysis

Consider an IVP that we can compute the solution for, analytically.

$$y' = 1 - y^2, \quad 0 \leq x \leq 50, \quad y(0) = \frac{e-1}{e+1}.$$

The exact solution for this problem writes as:

$$y = \frac{e^{2x+1} - 1}{e^{2x+1} + 1}.$$

To test the orders of the three methods we discussed, we compute the error produced by their approximations with respect to the exact solution. Plotted in the logarithmic scale, the errors for different values of N are shown in Fig. 5.1, where N is the number of points in the solution.

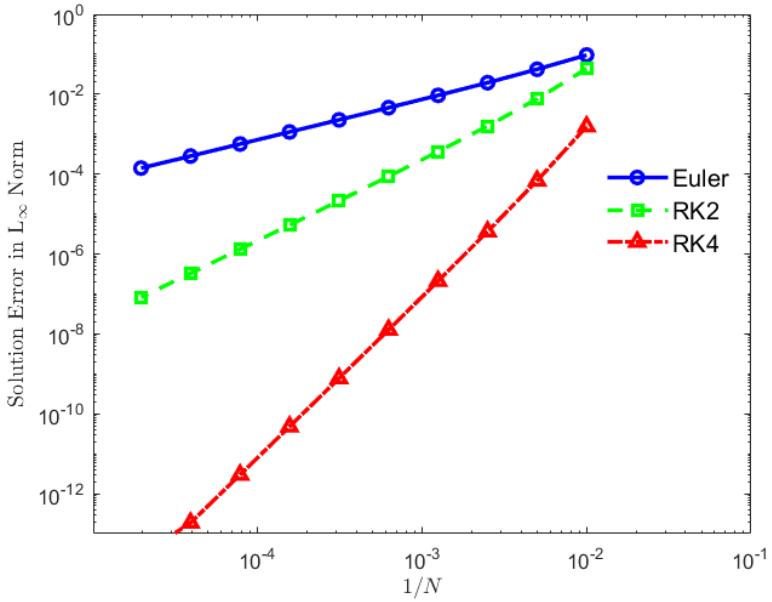


Figure 5.1: Plot of errors produced by different numerical methods vs. $1/N$ in the logarithmic scale.

We observe that the error decreases as we increase the number of points, N, for all methods. However, they do not all exhibit the same speed. Notice that the RK4 method is the fastest one to converge to an almost zero error, then the RK2 method, and lastly, the forward Euler method.

The slope of each curve represents the order of convergence for the method. Hence, we compute the slopes of the curves. The results are displayed in table 5.1.

Table 5.1: Slopes of error curves produced by different numerical methods

Numerical method	Slope
Euler	1.035
RK2	2.0859
RK4	4.0102

The Shooting Method

The shooting method is a numerical technique used to solve boundary value problems (BVPs). In our case, we transform the IVP 4.5 into a BVP by considering the value of $\boldsymbol{\xi}(x_m)$, where x_m is unknown. Notice that $w(x_m) = \infty$, and via 4.2, we know that when $\xi_1^2 \equiv \eta^2 = 1$, $w = \infty$. By iteratively adjusting a guessed parameter representing the value of x_m, numerical methods are employed to solve the IVP until the solution satisfies the forementioned condition. Through this process, the shooting method effectively converges upon a solution for the IVP, allowing for the determination of the value of x_m. Here's a brief outline of how the shooting method can be used in our context:

1. Start by guessing an initial value for x_m.
2. Solve the initial value problem using the guessed x_m as the upper limit by the forementioned numerical techniques.
3. Check if the computed value of $\boldsymbol{\xi}(x_m)$ satisfies the desired boundary condition.
4. If the boundary condition is not satisfied, adjust the guessed value of x_m and repeat steps 2 and 3 until the boundary condition is met.
5. Once the boundary condition is satisfied, the computed solution $\boldsymbol{\xi}(x)$ over the interval $[0, x_m]$ and the value of x_m are considered sufficiently close to the true values.

Numerical Integration

After computing the solution $\boldsymbol{\xi}(x)$, and hence $\eta(x)$, we can find the values of $w(x)$ using the relationship 4.2. Lastly, knowing that $h'(x) = w(x)$, we will employ the trapezoidal rule to compute the values of $h(x)$. The trapezoidal rule is a numerical integration technique used to approximate the definite integral of a function. It approximates the area under a curve by dividing it into trapezoids and summing up their areas. Its formula writes as:

$$h(x_{i+1}) = h(x_i) + \frac{\Delta x}{2}(w(x_i) + w(x_{i+1})).$$

However, since we only know the final value of the desired function $(h(x_m) = 0)$, we manipulate the formula to become:

$$h(x_i) = h(x_{i+1}) - \frac{\Delta x}{2}(w(x_i) + w(x_{i+1})),$$

and iterate over the interval $[0, x_m]$ backwards from the upper bound to the lower bound.

Geometric Quantities

We are particularly interested in the perimeters, areas, and circularity ratios of the shapes produced, so we compute their values for each solution. The formulas for for the length and area of curve made of discrete points write as:

$$\text{Length } L = \sum_{i=1}^{k-1} \sqrt{(x_{i+1} - x_i)^2 + (h_{i+1} - h_i)^2},$$

$$\text{Area } A = \frac{1}{2} \sum_{i=1}^{k-1} (x_{i+1} - x_i)(h_i + h_{i+1}),$$

where k is the number of x values.

Numerical Algorithm

We started by solving the system using the built-in solver on MATLAB, ode45. This solver uses a variable-step Runge-Kutta method to approximate the solution of ODEs. Algorithm 1 displays the program written to solve the IVP 4.5, find the values of w, and as a result, the values of h, determine the exact value of x_m, and lastly, compute the geometric quantities associated with the generated shape.

Algorithm 1 Solver for the membrane problem using ODE45 in MATLAB

1: **function** RBCsolver($\gamma,\ H_0,\ p,\ \lambda,\ x_m,\ N$)
2: Pack initial conditions into a vector: $\xi_0 \leftarrow [0;\gamma]$
3: Define the ODE system: $f \leftarrow @(x,\xi)[\xi(2);((H_0^2+8\lambda)\xi(1)-\xi(1)\xi(2)^2-8px)/(2(1-\xi(1)^2))]$
4: Call ode45 to solve the ODE system:
5: $[x\_values, \xi\_values] \leftarrow$ ode45$(f,$ linspace$(0, x_m, N), \xi_0,$ odeset$('RelTol', 1e-13))$
6: Stopping criterion:
$elements\_to\_remove \leftarrow (\xi\_values(:,1))^2 \geq 1$
$\xi\_values(elements\_to\_remove,:) \leftarrow [\,]$
$x\_values(elements\_to\_remove) \leftarrow [\,]$
7: Find the exact value of x_m: $exact_x_m \leftarrow x_values(end)$
8: Find the values of w:
$w\_values \leftarrow$ zeros$(1,$ length$(x_values))$
9: **for** $i = 1 :$ length(x_values) **do**
10: $w\_values(i) \leftarrow \xi\_values(i,1)/\sqrt{1-\xi\_values(i,1)^2}$
11: **end for**
12: Find the values of h using trapezoidal rule:
$h_values($length$(x_values)) \leftarrow 0$
$i \leftarrow$ length$(x_values) - 1$
13: **while** $(i \geq 1)$ **do**
14: $h\_values(i) \leftarrow h\_values(i+1) - 0.5 \times (x\_values(i+1) - x\_values(i)) \times (w\_values(i) + w\_values(i+1))$
$i \leftarrow i - 1$
15: **end while**
16: Compute the length of the curve: $L \leftarrow 0$
17: **for** $i = 1 :$ length$(x_values) - 1$ **do**
18: $d \leftarrow \sqrt{(x\_values(i+1) - x\_values(i))^2 + (h\_values(i+1) - h\_values(i))^2}$
$L \leftarrow L + d$
19: **end for**
20: Compute the area under the curve: $A \leftarrow$ trapz(x_values, h_values)
21: Compute the circularity ratio: $R \leftarrow 4\pi A/L^2$
22: **return** $L,\ A,\ R,\ exact\_x_m,\ x\_values,\ h\_values$
23: **end function**

The algorithm will be straightforwardly adjusted to implement the Euler, modified Euler, and fourth-order Runge-Kutta methods. These adaptations will be made in lines 4 and 5 of the algorithm.

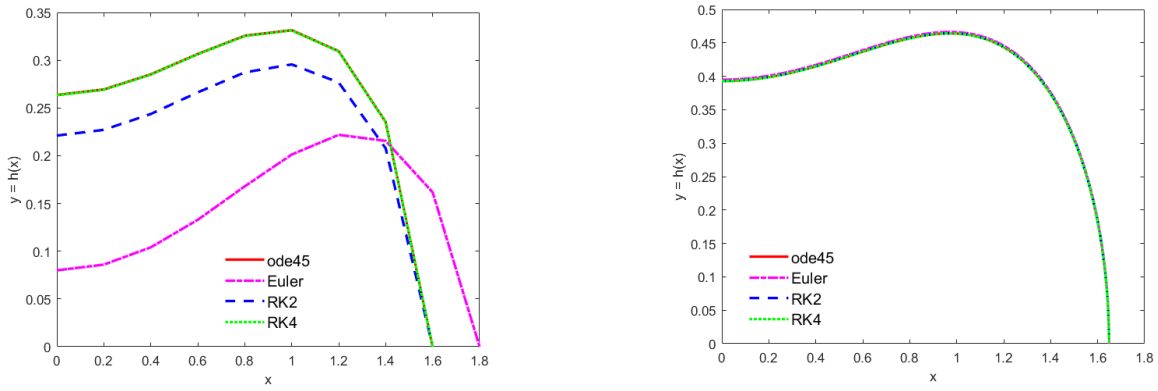
5.2 Tests and Validation

This section will focus on exploring the various characteristics of the solver, as well as examining the significance of the model in terms of its biological meaning.

Biconcave Shapes

For the following set of inputs, the plots shown in figure 5.2 were generated.

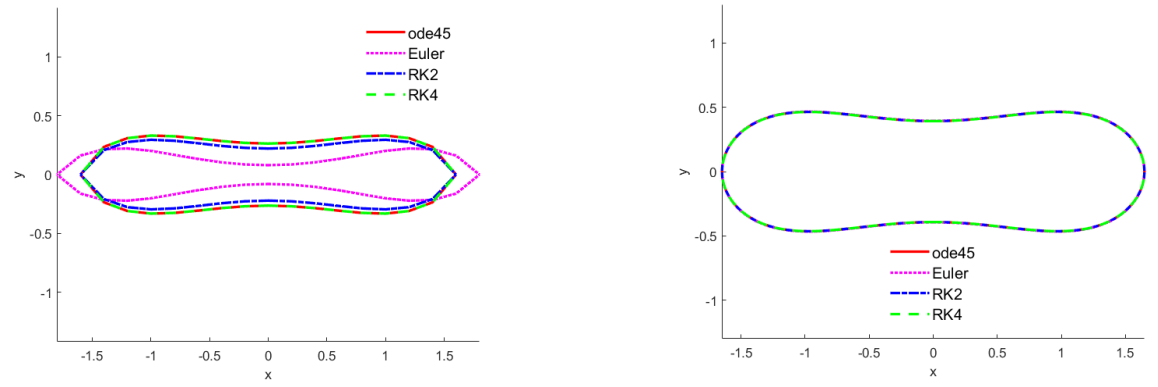
$$\gamma = 0.3, \quad H_0 = 0, \quad p = 0.5, \quad \lambda = 0.125, \quad x_m = 1.8$$



(a) $N = 10$ **(b)** $N = 10^5$

Figure 5.2: Plots of one-fourth of the curve h(x) obtained using different numerical methods and number of points, N.

Because the shape of the membrane is symmetrical, we reflected the values of x and $h(x)$ over the x-axis, y-axis, and origin to obtain the full shape of the curve. See Fig. 5.3.



(a) $N = 10$ **(b)** $N = 10^5$

Figure 5.3: Plots of the full curve h(x) obtained using different numerical methods and number of points, N.

We observe that the methods produce more precise and smoother plots with a larger number of points

(see Fig. 5.2). In Fig. 5.3, it appears that the assumption $h'(x_m) = \infty$ is not achieved when $N = 10$, which requires us to test the methods with a larger number of points to converge to the solution.

We then multiplied the output length and area by four to compute the perimeter and area of the full shape, and recalculated the circularity ratio. The output results are shown in table 5.2.

Table 5.2: Special values of the curve obtained using different numerical methods and number of points.

(a) $N = 10$

	ode45	Euler	RK2	RK4
Exact x_m	1.6	1.8	1.6	1.6
Perimeter, P	6.9048	7.4969	6.8166	6.904
Area, A	1.7559	1.0649	1.5319	1.7542
Circularity ratio, χ	0.4628	0.23809	0.41431	0.46247

(b) $N = 10^5$

	ode45	Euler	RK2	RK4
Exact x_m	1.6489	1.6489	1.6489	1.6489
Perimeter, P	7.5535	7.5621	7.5544	7.5544
Area, A	2.6661	2.6801	2.6676	2.6676
Circularity ratio, χ	0.58721	0.58896	0.5874	0.5874

The table shows that when $N = 10$, methods of higher order like ode45 and RK4 produce precise results compared to each other. The Euler and RK2 methods on the other hand, generate different geometric values. For $N = 10^5$, RK2 and RK4 methods produce equal values, which are very close to those of ode45. Euler being the method of the least order, generates slightly different approximations. Additionally, all four methods approximate the same value of x_m.

Sensitivity to The Spatial Discretization

Next we proceeded to set different values for the number of points, N, when passed as an input to the Euler, RK2, and RK4 functions, in order to test the convergence of each method.

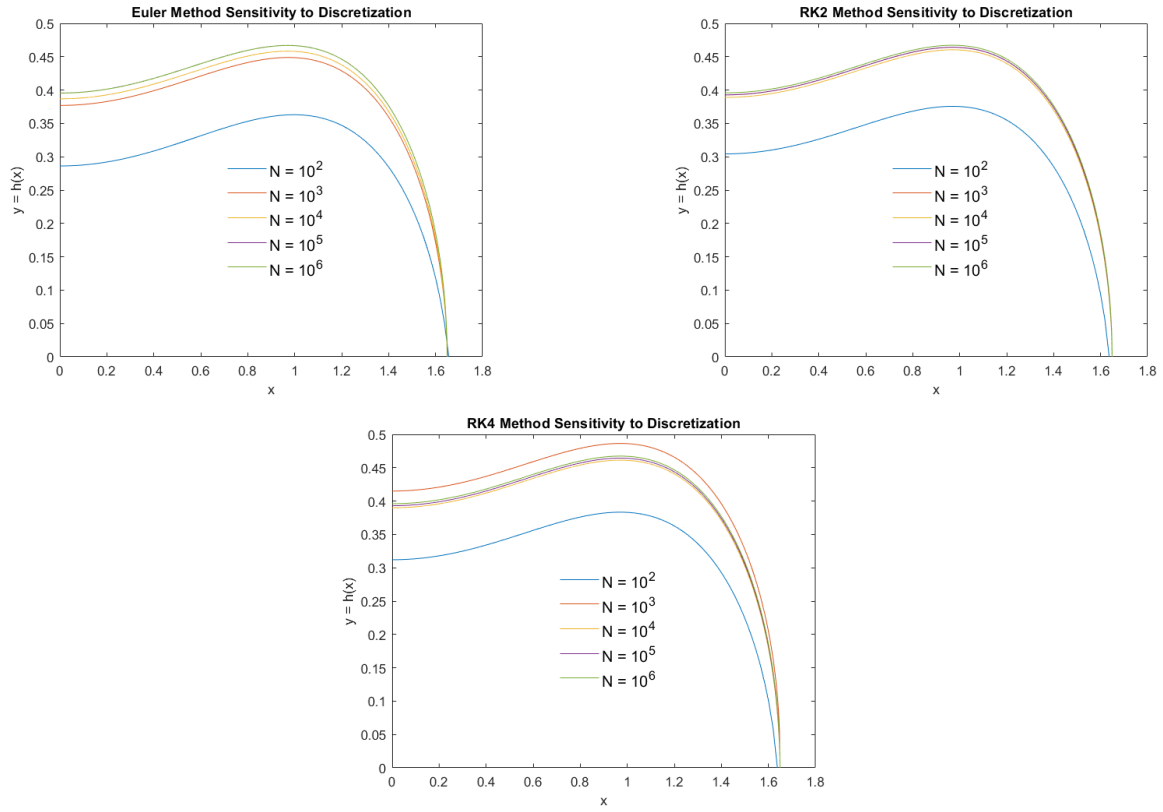


Figure 5.4: Sensitivity to discretization for all three methods.

Fig. 5.4 shows that the solutions obtained by by RK2 and RK4 methods converge to the exact solution for all $N > 10^3$. The difference between the solutions obtained using different number of points becomes negligible after $N = 10^3$. The Euler method on the other hand, requires $N \geq 10^5$ to approximate the solution correctly. This becomes clearer when we provide a closer look at the x-intercepts of the plots, as shown in Fig. 5.5.

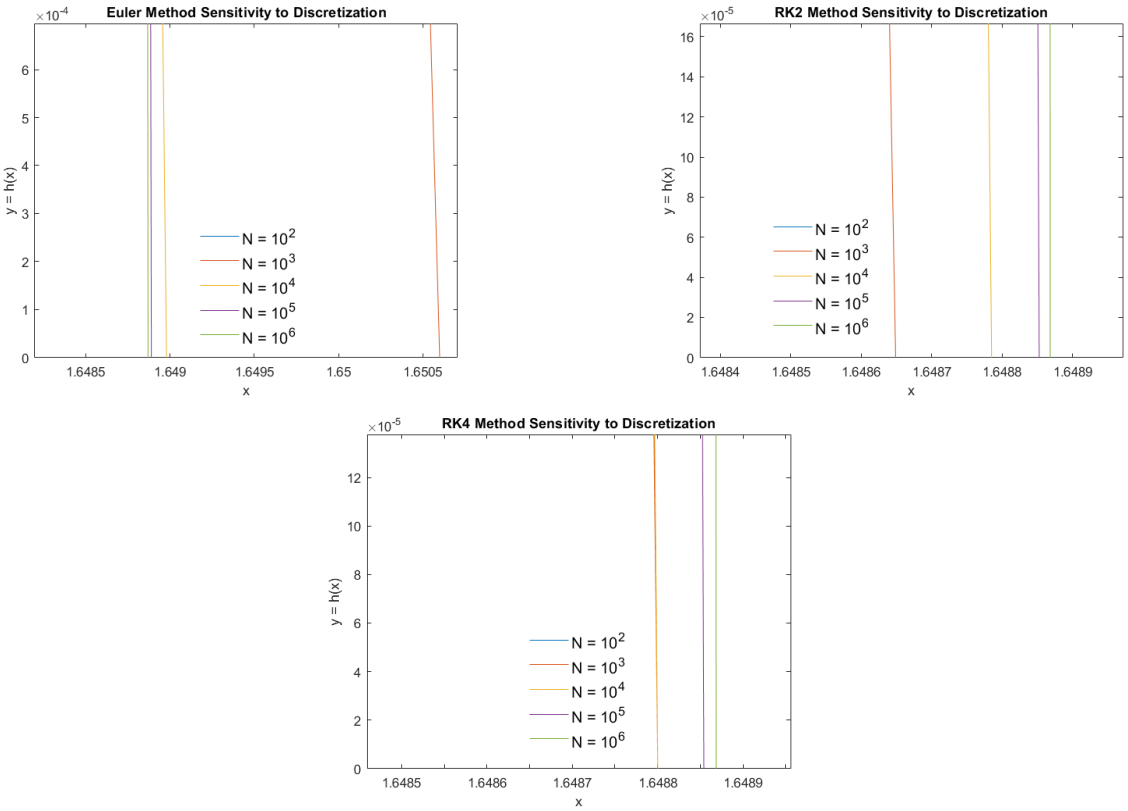


Figure 5.5: Zoomed in version of Fig. 5.4

Numerical Investigation of Shape Variations with Respect to Model Parameters

We proceed to study how the shape changes when we change the input parameters γ, λ, and p. The circularity ratio informs us about the roundness of the shape, so we compute χ for all the different shapes generated. In Fig. 5.6, we vary the value of γ between -1.5 and 0.5, and keep the values of the other parameters fixed as follows:

$$H_0 = 0, \quad p = 0.5, \quad \lambda = 0.125.$$

We run computations using different parameters of the model and we reconstruct the set of membrane shapes, which will be dimensionalized and provided with the corresponding circularity parameters.

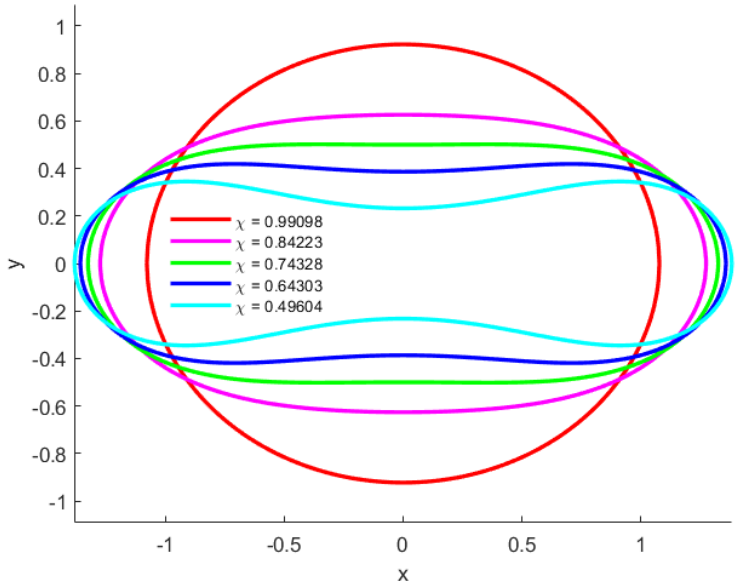


Figure 5.6: Shapes of the reconstructed membranes, as well as their corresponding circularity ratios (changing the value of γ).

Next, we alter the values of λ in a range between -1 and 1.5, and fix the following values for the rest of the parameters (see Fig. 5.7).

$$\gamma = 0.2, \quad H_0 = 0, \quad p = 0.5.$$

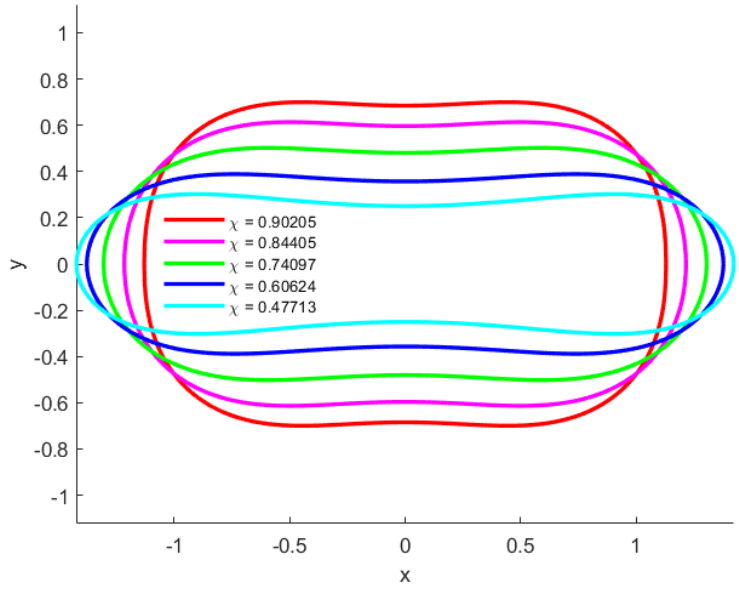


Figure 5.7: Shapes of the reconstructed membranes, as well as their corresponding circularity ratios (changing the value of λ).

Lastly, we study the variations of the shape with respect to different values of p (see Fig. 5.8), ranging between 0 and 100, while holding the following values of the rest of the parameters:

$$\gamma = -0.3, \quad H_0 = 0, \quad \lambda = -0.2.$$

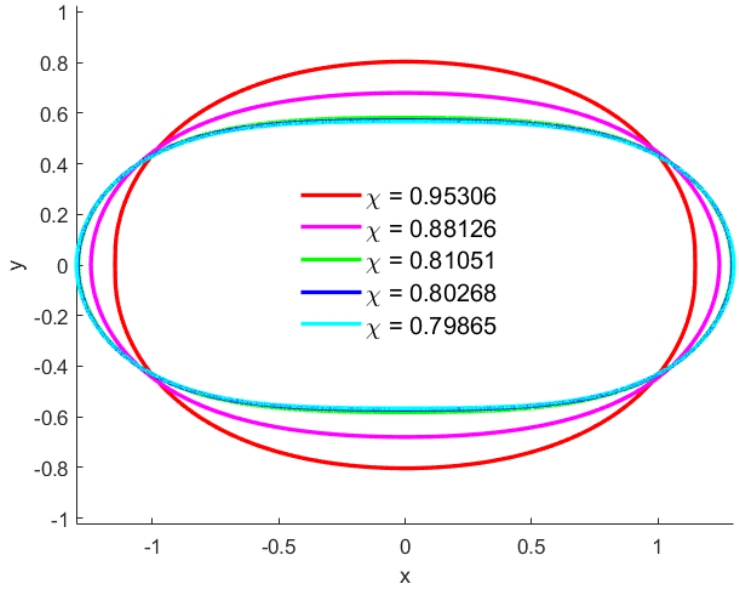


Figure 5.8: Shapes of the reconstructed membranes, as well as their corresponding circularity ratios (changing the value of p).

Based on the changes observed in the shape and its circularity ratio for each parameter, we take the

following values for the input parameters in order to obtain a circularity ratio very close to 1:

$$\gamma = -0.7, \quad H_0 = 0, \quad p = 0.5, \quad , \lambda = -0.6.$$

Fig. 5.9 is produced. We notice that when $\chi \approx 1$, the shape of the curve becomes very close to full-rounded circle.

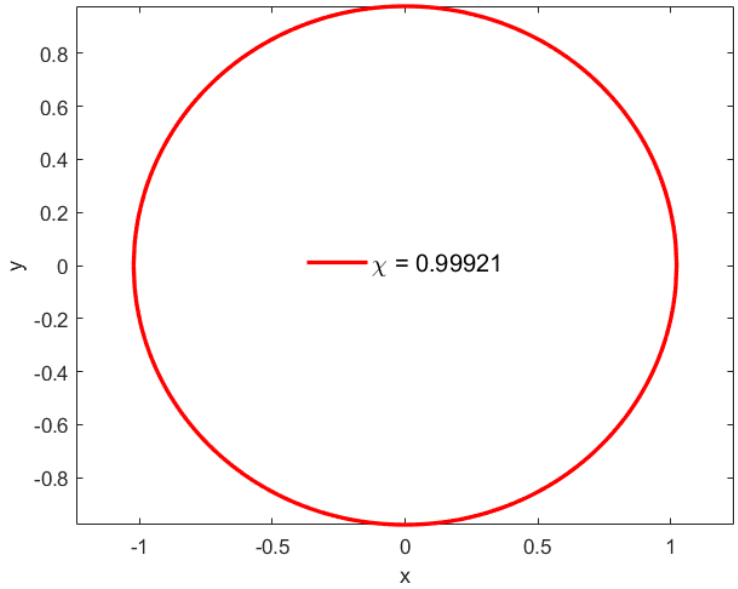


Figure 5.9: Plot of the reconstructed circular membrane with $\chi \approx 1$.

Conclusions and Future Work

The current project presents a simple yet accurate reduced-order model for simulating the membrane deformations of two-dimensional red blood cells. The model is derived through the minimization of a bending energy function, describing the membrane mechanics, and is formulated as an initial value problem via a saddle point approach. The resulting second-order ordinary differential equation is reformulated into a system of first-order ODEs, simplified through a change of variables, and numerically solved using Euler, RK2, and RK4 schemes. A shooting method employed to determine the maximal membrane length. The known biconcave RBC shapes are found numerically. A set of numerical tests is presented for validation, and investigations are done to study model parameters.

This work can be extended to consider three-dimensional red blood cell dynamics and incorporate partial differential equations (PDEs) to capture spatial variations in membrane properties and fluid dynamics. The model can be applied to study the effect of pathological conditions on red blood cell morphology and deformability.

Appendix: Optimization with Equality Constraints

Optimization Using Lagrange Multipliers

Consider the following optimization problem

$$\begin{aligned} \min_{\mathbf{x} \in \mathbb{R}^n} \quad & f(\mathbf{x}) \\ \text{such that} \quad & g_i(\mathbf{x}) = 0, \quad 1 \leq i \leq m, \end{aligned}$$

where $f, g_i : \mathbb{R}^n \to \mathbb{R}$ are convex and continuously differentiable for all $1 \leq i \leq m$, and $m > 0$ is an integer. A common way to solve such a problem is the Lagrange multiplier method. The Lagrange multiplier method introduces Lagrange multipliers associated with the equality constraints to transform the constrained optimization problem into an unconstrained problem. The Lagrangian, a function that combines the objective function and the constraints with the Lagrange multipliers, is defined as follows:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \sum_{i=1}^{m} \lambda_i \cdot g_i(\mathbf{x}),$$

where $\boldsymbol{\lambda}$ is a vector in $\mathbb{R}^m$. The components of the vector, λ_i, are the Lagrange multipliers associated with the equality constraints and $\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})$ is the Lagrangian function. The solution to the optimization problem is then found by setting the gradient of the Lagrangian with respect to the decision variables vector $\mathbf{x}$ and the Lagrange multipliers vector $\boldsymbol{\lambda}$ equal to zero. This creates optimality conditions that ensure the objective function is optimized while respecting the equality constraints.

Example

Let us consider the following problem:

$$\begin{aligned} \min_{\mathbf{x}\in\mathbb{R}^2} \quad & f(\mathbf{x}) = x_1^2 + x_2^2 \\ \text{s.t.} \quad & \begin{cases} g_1(\mathbf{x}) = x_1 + x_2 - 1 = 0 \\ g_2(\mathbf{x}) = x_1 - x_2 = 0. \end{cases} \end{aligned}$$

In this problem:

- $f(\mathbf{x})$ is the objective function that we want to minimize.
- $\mathbf{x} = (x_1, x_2)^\top$ is the vector of decision variables.
- $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ represent the two equality constraints.

The Lagrangian function is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f(\mathbf{x}) + \lambda_1 g_1(\mathbf{x}) + \lambda_2 g_2(\mathbf{x}) = x_1^2 + x_2^2 + \lambda_1(x_1 + x_2 - 1) + \lambda_2(x_1 - x_2),$$

where $\boldsymbol{\lambda} = (\lambda_1, \lambda_2)^\top$. To find the optimal solution, we set the gradient of the Lagrangian with respect to the variables x_1 and x_2 and the Lagrange multipliers λ_1 and λ_2 to zero. We obtain:

$$\begin{cases} \partial_{x_1}\mathcal{L} = 2x_1 + \lambda_1 + \lambda_2 = 0, \\ \partial_{x_2}\mathcal{L} = 2x_2 + \lambda_1 - \lambda_2 = 0, \\ \partial_{\lambda_1}\mathcal{L} = x_1 + x_2 - 1 = 0, \\ \partial_{\lambda_2}\mathcal{L} = x_1 - x_2 = 0. \end{cases}$$

Solving this system of equations, we find that

$$\begin{cases} x_1 = \dfrac{1}{2}, \\ x_2 = \dfrac{1}{2}, \\ \lambda_1 = 1, \\ \lambda_2 = 0. \end{cases}$$

So, the optimal solution to the minimization problem is $\mathbf{x}^* = \left(\dfrac{1}{2},\ \dfrac{1}{2}\right)^\top$ with $f(\mathbf{x}^*) = \dfrac{1}{4}$, and $\boldsymbol{\lambda} = (1,\ 0)^\top$.

Bibliography

[1] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, P. Walter, Molecular Biology of the Cell, 6th Edition, Garland Science, 2014.

[2] W. Choi, J. Yi, Y. Kim, Fluctuations of red blood cell membranes: The role of the cytoskeleton, Phys. Rev. E 92 (2015) 012717. doi:10.1103/PhysRevE.92.012717.

[3] S. A. Safran, Statistical Thermodynamics of Surfaces, Interfaces and Membranes, Frontier in Physics Vol 90, Addison-Wesley Publishing Company, Reading, Massachusetts, 1994.

[4] H. Lodish, A. Berk, S. L. Zipursky, P. Matsudaira, D. Baltimore, J. Darnell, Molecular Cell Biology, 7th Edition, W. H. Freeman, 2012.

[5] A. Verkleij, R. Zwaal, B. Roelofsen, P. Comfurius, D. Kastelijn, L. van Deenen, The asymmetric distribution of phospholipids in the human red cell membrane. a combined study using phospholipases and freeze-etch electron microscopy, Biochimica et Biophysica Acta (BBA) - Biomembranes 323 (2) (1973) 178–193. doi:https://doi.org/10.1016/0005-2736(73)90143-0.

[6] K. Simons, E. Ikonen, Functional rafts in cell membranes, Nature 387 (6633) (1997) 569–572. doi:10.1038/42408.

[7] R. Lipowsky, The Conformation of Membranes: Biological Membranes - Structure and Dynamics, Elsevier Science, 2014.

[8] E. A. Disalvo, M. C. Garcia (Eds.), Liposomes: Methods and Protocols, Vol. 606, Humana Press, 2011.

[9] M.-A. Mader, V. Vitkova, M. Abkarian, A. Viallat, T. Podgorski, Dynamics of viscous vesicles in shear flow, Eur. Phys. J. E: Soft Matter Biol. Phys. 19 (2006) 389–397. doi:10.1103/PhysRevE.72.011901.

[10] Q. Du, C. Liu, X. Wang, Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions, Journal of Computational Physics 212 (2006) 757–777. doi:10.1016/j.jcp.2005.07.020.

[11] W. Helfrich, Elastic properties of lipid bilayers: Theory and possible experiments, Zeitschrift für Naturforschung C 28 (11-12) (1973) 693–703. doi:10.1515/znc-1973-11-1209.

[12] A. Laadhari, P. Saramito, C. Misbah, G. Székely, Fully implicit methodology for the dynamics of biomembranes and capillary interfaces by combining the level set and newton methods, Journal of Computational Physics 343 (2017) 271–299. doi:10.1016/j.jcp.2017.04.019.

[13] A. Laadhari, A. Deeb, B. Kaoui, Hydrodynamics simulation of red blood cells: Employing a penalty method with double jump composition of lower order time integrator, Mathematical Methods in the Applied Sciences 46 (18) (2023) 19035–19061. doi:10.1002/mma.9607.

[14] A. Laadhari, C. Misbah, P. Saramito, On the equilibrium equation for a generalized biological membrane energy by using a shape optimization approach, Physica D: Nonlinear Phenomena 239 (16) (2010) 1567–1572. doi:10.1016/j.physd.2010.04.001.

[15] Q. Du, C. Liu, X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, Journal of Computational Physics 198 (2) (2004) 450–468. doi:https://doi.org/10.1016/j.jcp.2004.01.029.

[16] D. Jamet, C. Misbah, Toward a thermodynamically consistent picture of the phase-field model of vesicles: Curvature energy, Physical Review E 78 (2008) 031902. doi:10.1103/PhysRevE.78.031902.

[17] Cottet, Georges-Henri, Maitre, Emmanuel, Milcent, Thomas, Eulerian formulation and level set models for incompressible fluid-structure interaction, ESAIM: M2AN 42 (3) (2008) 471–492. doi:10.1051/m2an:2008013.

[18] C. Pozrikidis, Numerical simulation of the flow-induced deformation of red blood cells, Annals of Biomedical Engineering 31 (10) (2003) 1194–1205. doi:10.1114/1.1617985.

[19] A. Rahimian, S. K. Veerapaneni, G. Biros, Dynamic simulation of locally inextensible vesicles suspended in an arbitrary two-dimensional domain, a boundary integral method, Journal of Computational Physics 229 (18) (2010) 6466–6484. doi:https://doi.org/10.1016/j.jcp.2010.05.006.

[20] B. Kaoui, J. Harting, C. Misbah, Two-dimensional vesicle dynamics under shear flow: Effect of confinement, Phys. Rev. E 83 (2011) 066319. doi:10.1103/PhysRevE.83.066319.

[21] D. Salac, M. J. Miksis, Reynolds number effects on lipid vesicles, Journal of Fluid Mechanics 711 (2012) 122–146. doi:10.1017/jfm.2012.380.

[22] V. Doyeux, Y. Guyot, V. Chabannes, C. Prud'homme, M. Ismail, Simulation of two-fluid flows using a finite element/level set method. application to bubbles and vesicle dynamics, Journal of Computational and Applied Mathematics 246 (2013) 251–259, fifth International Conference on

Advanced COmputational Methods in ENgineering (ACOMEN 2011). doi:https://doi.org/10.1016/j.cam.2012.05.004.

[23] J. W. Barrett, H. Garcke, R. Nürnberg, Numerical computations of the dynamics of fluidic membranes and vesicles, Phys. Rev. E 92 (2015) 052704. doi:10.1103/PhysRevE.92.052704.

[24] Bonito, A., Nochetto, R.H., Pauletti, M.S., Dynamics of biomembranes: Effect of the bulk fluid, Math. Model. Nat. Phenom. 6 (5) (2011) 25–43. doi:10.1051/mmnp/20116502.

[25] D. Gilbarg, N. Trudinger, Elliptic partial differential equations of second order, 2nd Edition, Vol. 224 of Classics in Mathematics, Springer, 1983.

[26] M. Spiegel, S. Lipschutz, J. Liu, Schaum's Outline of Mathematical Handbook of Formulas and Tables, Fifth Edition, McGraw Hill LLC, 2017.

[27] R. A. Horn, C. R. Johnson, Matrix Analysis, 2nd Edition, Cambridge University Press, 2012.

[28] W. Helfrich, Elastic properties of lipid bilayers: theory and possible experiments, Z Naturforsch C. 28 (11) (1973) 693–703. doi:10.1515/znc-1973-11-1209.

[29] E. A. Evans, Bending resistance and chemically induced moments in membrane bilayers, Biophys. J. 14 (12) (1974) 923–931. doi:10.1016/S0006-3495(74)85959-X.

[30] P. Bassereau, B. Sorre, A. Lévy, Bending lipid membranes: Experiments after w. helfrich's model, Advances in Colloid and Interface Science 208 (2014) 47–57, special issue in honour of Wolfgang Helfrich. doi:10.1016/j.cis.2014.02.002.

[31] A. Laadhari, P. Saramito, C. Misbah, Computing the dynamics of biomembranes by combining conservative level set and adaptive finite element methods, Journal of Computational Physics 263 (2014) 328–352. doi:https://doi.org/10.1016/j.jcp.2013.12.032.

[32] A. Laadhari, An operator splitting strategy for fluid-structure interaction problems with thin elastic structures in an incompressible Newtonian flow, Applied Mathematics Letters 81 (2018) 35–43. doi:https://doi.org/10.1016/j.aml.2018.01.001.

[33] A. Laadhari, R. Ruiz-Baier, A. Quarteroni, Fully Eulerian finite element approximation of a fluid-structure interaction problem in cardiac cells, International Journal for Numerical Methods in Engineering 96 (11) (2013) 712–738. doi:10.1002/nme.4582.