Subdivision based Finite Elements for Lipid Membranes

Dissertation
zur Erlangung des Doktorgrades
des Fachbereichs Physik
der Universität Hamburg

vorgelegt von

Bastian Robert Angermann
aus Hamburg

Hamburg
2007

Gutachter der Dissertation:	Prof. Dr. Gerhard Mack Dr. Martin Meier-Schellersheim
Gutachter der Disputation:	Prof. Dr. Gerhard Mack Prof. Dr. Jan Louis
Datum der Disputation:	10. April 2006
Vorsitzender des Prüfungsausschusses:	Prof. Dr. Jochen Bartels
Vorsitzender des Promotionsausschusses:	Prof. Dr. Günter Huber
Dekan der Fakultät Mathematik, Informatik und Naturwissenschaften:	Prof. Dr. Arno Frühwald

Abstract

In this thesis we study numerical methods for simulating mechanical deformations of cell membranes. Such models are given in terms of fourth order partial differential equations. In order to enable comparisons of the models predictions to experimental results, the equations must be solved on arbitrary cell geometries. A Finite Element Method based on subdivision surfaces, which is capable of discretizing the partial differential equations, is implemented in a C++ computer program.

An integral part of cell membranes models are constraints, enforcing the conservation of the cells surface, volume and integrated mean curvature. The discretized equations can not exactly fulfill these constraints. Instead one introduces harmonic potentials of the quantities to be conserved. This allows for an approximate conservation. Several Rosenbrock methods for the solution of the resulting stiff ordinary differential equations are tested. Virtual experiments in which cells are aspirated into micropipettes are carried out as a benchmark for the performance of the simulation.

Zusammenfassung

Diese Arbeit beschäftigt sich mit numerischen Methoden zur Simulation von mechanischen Deformationen von Zellmembranen. Solche Modelle werden durch Systeme von partiellen Differentialgleichungen vierter Ordnung beschrieben. Um die Vorhersagen solcher Modelle mit biologischen Experimenten vergleichen zu können ist es notwendig, die beschreibenden Differentialgleichungen für beliebige Zellgeometrien lösen zu können. Eine Finite Elemente Methode, die auf Subdivision Surfaces beruht und in der Lage ist, die partiellen Differentialgleichungen zu diskretisieren, wurde in einem C++ Computer Programm implementiert.

Ein wesentlicher Bestandteil von Modellen von Zellmembranen sind Zwangsbedingungen, welche die Erhaltung der Zelloberfläche, des Zellvolumens und der integrierten mittleren Krümmung der Zelloberfläche erzwingen. Die diskretisierten Gleichungen können die Zwangsbedingungen nicht exakt erfüllen. Statt dessen werden harmonische Potentiale von den zu erhaltenden Grössen eingeführt, die eine näherungsweise Erhaltung erlauben. Verschiedene Rosenbrock Methoden zur Lösung der resultierenden steifen gewöhnlichen Differentialgleichungen werden getestet. Als Test für die Leistungsfähigkeit der Simulation werden virtuelle Experimente ausgeführt, in denen eine Zelle in eine Mikropipette aspiriert wird.

Contents

List of Figures

Chapter 1

Introduction

At the beginning of the 21st century cell biology is in a situation which is quite opposite to the state of high energy physics. With the depelopment of high throughput experiments, larger and larger amounts of data on new poorly understood phenomena are available, and there is a lack of a predictive quantitative framework that integrates what is accepted as fact into a concise picture.

At the same time theorists percieve a lack of data allowing them to fix free parameters in their models. The experimental data is often not collected with the intend to be used as input for quantitative methods, and thus lacks reproducible calibration.

The reason for this situation is neither a deficiency of funding nor a failing interest. The potential benefits of a predictive theory of cellular processes to human health and the related financial benefits provide ample motivation to allocate resources to the investigation of quantitative theories. Most attempts failed in developing models which would quantitatively predict the behavior of systems under physiological conditions. This is mostly due to an overabstraction of the biological processes into models which are not faithful images of the biology. The reason for the creation of these overly simplyfied models is the complexity of the studied systems and the desire to create models which are analytically solvable or at least numerically tractable. Furthermore, the experts on quantitative modeling do not share the same expertise as the experimentalists working on the same subject. Thus insights on a system tend to get lost in translation. These failings of abstract models have lead to a weariness among some experimentally working investigators towards quantitative approaches.

The problem is that the behaviour of biological entities depends strongly on their environment. This is manifest both on the experimental and conceptual level. Cells sense their chemical environment trough a variety of transmembrane receptors, and adapt their internal biochemistry in response to changes in the environment. Thus experiments utilizing cells in living tissues or organisms suffer from the lack of control of the external parameters. On the other hand, removing cells from their natural environment will affect the internal biochemistry in an unknown but possibly highly nonlinear manner. On a conceptual level the abstraction of detail in biological systems does not advance biomedical research towards it's ultimate goal. It is the change in biochemical details, which determines the difference between health and disease. Knowing the abstract input output behavior of the system alone does not enable the control of

the system. In the same way, as we need a key to a topographic map to make sense of it, we need to know how the abstract input and output maps to the biological entities, in order to assign meaning to a model and design drugs with a specific effect.

Typical biological models contain several dozens or hundreds of interacting components. Thus it is impossible to analyze the system solely by analytical means. Numerical techniques are required to gain information on the behavior of the model. The emerging field of Computational Biology applies techniques from physics, mathematics, engineering and computer science to tackle these problems and bride the gap between experimental biology and biological modeling.

This is the reference frame of this thesis. It's goal is to develop numerical tools to analyze the dynamics of deformations of cell membranes. These membranes do not only serve as barriers to maintain the integrity of the cells biochemistry, but they also serve as an interface through which cells communicate – both chemically and physically – with their environment. The outcome of these communication events affects vital processes such as the triggering of immune responses [14]. For more details on some of the mechanisms involved see Chapter 2.

The mechanical properties of cell membranes are governed by a bilayer of amphiphilic molecules. This bilayer can be viewed as a two dimensional incompressible fluid, which resists bending. Since it is impermeable to water the volume enclosed by the lipid bilayer is constant. A constraint on the difference of the areas of the two sheets of the bilayer stems from the attraction between them. This is the system which is modeled in this thesis. The fourth order partial differential equations describing the lipid membrane are discretized by a Finite Element Method based on Subdivision Surfaces. The global nonlinear constraints are approximately enforced by harmonic potentials of the conserved variables. We need this weakening of the constraints because the discretized equations can not fulfill all constraints simultaneously. We are lead to a stiff system of ordinary differential equations, which are best solved by implicit methods. Our system of equations is nonlinear. Thus we employ a Rosenbrock method for solving the ordinary differential equations to avoid the solution of a nonlinear equations for each time step. This is done by incorporating the Jacobian of the system into the update scheme, such that for each stage in the update procedure the equivalent of one Newton iteration is done [49]. Different Rosenbrock schemes are tested, and virtual experiments using an adaptive mesh are carried out.

The Simulation was implemented in the C++ programming language [101]. The need to keep track of the neighborship relations in the triangle mesh approximating the manifold, and the natural identification of points, vertices, triangles, and meshes as objects made it desirable to have language constructs, supporting complex data structures and the concept of objects. Furthermore the simulation is aimed to be executed on a variety of Unix platforms as well as on Microsoft Windows. Thus the widespread availability of reliable development tools was required. Since the simulations are on the boundary of what is computationally feasible, the overhead for supporting advanced language constructs should be small, and low level access to the data should be possible. An other requirement is the ability to access available libraries for efficient linear algebra, such as BLAS [33, 34], LAPACK [6] and PETSc [9, 8, 10] and the requirement to

make the implementation accessible from SIMMUNE [69, 71, 70].

This work should be seen in the context of the development of tools to investigate the detailed biochemical behavior of cells [71] the goal of the development of these tools is to enable biologists to develop quantitative models of biological processes on multiple scales by hiding the underlying mathematics as much as possible and thus allowing the experts on the qualitative behavior of a process to develop quantitative models.

1.1 Outline of this thesis

In chapter 2 the significance of the biophysical behavior of the plasma membrane of cells is illustrated by highlighting recent results on the interplay between a cells biochemistry and its physics. The accepted models of lipid bilayers are discussed. We introduce constraints on the surface area, volume and integrated mean curvature of a model cell as harmonic potentials in these quantities. We find the equations of motion as the functional derivative of the free energy of the bilayer. In the end of the Chapter we discuss the influence of the surrounding liquid.

Chapter 3 gives an overview on alternative numerical solution strategies for partial differential equations (PDE). The pros and cons of finite differences, level set methods and discrete differential geometry are examined.

Chapter 4 serves as an introduction to the finite element method for solving partial differential equations. A simple one dimensional example discretizing the spatial variables of the heat equation is given to acquaint the reader with the basics of finite elements. This is followed by a discussion of the choice of the proper function space on which a solution to the PDE is sought. The influence of the element shapes on the accuracy of the solution is discussed, and a method for maintaining favorable element shapes is introduced. The chapter is concluded by a discussion of local error estimates.

In Chapter 5 the concept of splines is introduced, and generalized to subdivision surfaces. Both concepts originate from the field of computer aided geometric design [42], but with the discovery of algorithms for the evaluation of subdivision surfaces at arbitrary parameter values [96, 97] they could be used as basis functions to span finite element spaces. After discussing the use of subdivision surfaces as basis functions, we discuss how to apply the finite element method to the equations of motion derived in chapter 2. A numerical quadrature scheme for the evaluation of the integrals arising in the discretization process is given.

The solution of the ordinary differential equation resulting from the finite element discretization is discussed in chapter 6. We will discuss the need for implicit solution methods, and introduce Rosenbrock methods, which avoid the solution of nonlinear equations. In the of the Chapter we apply Rosenbrock methods to the equations arising from the finite element discretization. Special care has to be taken, because the matrix multiplying the time derivative is solution dependent. The equations are transformed by interpreting the derivative of the independent variables as additional independent variables. This leads to an differential algebraic equation that can be solved with available Rosenbrock methods.

We discuss numerical experiments, which use the methods developed here, in

Chapter 7. The performance of several different Rosenbrock methods is evaluated. When a simulated membrane is close to its equilibrium shape, we observe that the relation between the allowed numerical tolerance and the largest time step compatible with these tolerances displays unexpected jumps. These jumps are artifacts of the spatial discretization. Oscillations of the constrained quantities are such artifacts. We asses the influence of the resolution of the mesh on the behavior of simulations subject to different constraints. We simulate the aspiration of cells into a micro pipette, to test the performance of the simulation in the presence of external forces and geometric constraints.

Open questions and future directions are discussed in Chapter 8. Parameters of numerical algorithms and the finite element basis functions are listed in the appendices.

Chapter 2

The Lipid Bilayer

Lipid bilayers are the fundamental component of cell membranes. They do not only maintain the biochemical integrity of cells by forming a barrier impermeable to ions and proteins, but they furthermore form the interface through which cells sense their environment and communicate. Recent research sheds light on the influence of the biophysical properties of a cell's environment on the fundamental behavior of it. In the following we give a few examples illustrating interactions between external signals, cellular biochemistry and membrane behavior.

Yeung et al. [111] showed that fibroblasts[1] and endothelial cells[2] upregulate[3] $\alpha 5$ integrin in response to changes in the stiffness of the substrate to which the cells adhere. $\alpha 5$ integrin is a protein that regulates the attachment of a cell to the extracellular matrix[4] and is involoved in signal transduction processes related to the adhesion of cells [4]. More recent findings of Weaver et. al [109] indicate similar behavior in human mammary epithelial cells[5], which change their expression profiles towards profiles similar to those of carcinoma, if the are embedded in an extracellular matrix, which is stiffer than in healthy tissues. This clearly illustrates the deep interconnection between the biophysical environment and the expression of genes in a cell. Even if we lack a detailed understanding of the mechanisms involved in the aforementioned regulation of gene expression, we can expect to see an influence of the cell membrane on this regulatory system. The cell membrane will not nessecarily be a part in the mechanism sensing the stiffness of the environment, but it will give rise to background forces and therefore indirectly affect the forces sensed by other parts of the cell.

The protein domain BAR is the key component of a known mechanism of biochemical sensing and regulation of membrane curvature. BAR is a banana shaped dimer which binds electrostaticly to the cytosolic leaflet of the bilayer [113]. Depending on the spontaneous curvature and bending rigidity of the membrane it can either bend the membrane into a cylindrical shape with a radius

[1] Fibroblasts are cells in the connective tissues, which secrete an extracellular matrix rich in collagen. They have a major role in wound healing as they migrate into the injured tissue and secrete collageneous matrix, helping to repair the tissue[4]

[2] Endothelial cells form a single cell layer lining the interior surface of blood vessels [4].

[3] Regulatory process in which cells increase the expression of a protein.

[4] A extracellular matrix is a meshwork of macromolecules in the extracellular space. It is the major component of connetive tissues[4].

[5] Epithelial cells line the surfaces of an organism[4].

of ≈10 nm, or, if the affinity is too low for effective binding and deformation, sense the curvature of the membrane because of its curvature dependent affinity.

The curvature elasticity of the plasma membrane affects the communication of cells. Antigen Presenting Cells (APC) take up proteins from their environment and break them into peptides. These are presented on the surface of the APC, where they are bound to proteins that are encoded in the major histocompatibility complex (MHC). Encountering a T-Cell[6], the two cells make close contact. Now the MHC and the presented peptide can bind to the T-cell receptor (TCR), which can lead to the activation of the T-Cell and a subsequent adaptive immune response [55]. Furthermore, the APC also possess adhesion molecules ICAM-1, which bind to the LFA-1 protein on the T-cell. The resulting protein complexes differ in size. Complexes formed by ICAM-1 and LFA-1 have a size of 42 nm whereas the size of the TCR-pMHC complex is only 15 nm (Fig. 2.1) [81, 14]. This difference in complex sizes makes it energetically unfavor-

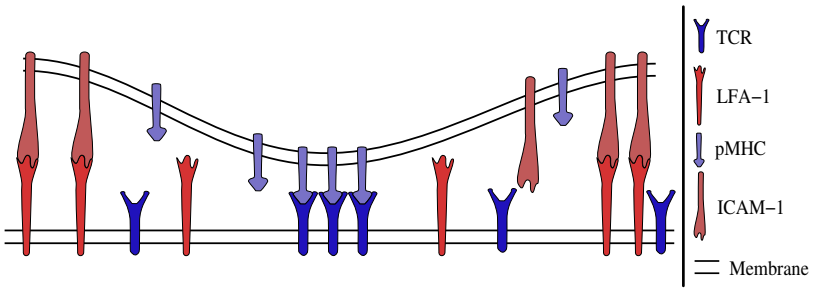

Figure 2.1: Detailed view of a T-Cell APC contact

able to bring TCR-pMHC and LFA-1-ICAM-1 complexes into close proximity, since the membrane resists bending [14]. This can lead to a self assembly of a central cluster of TCR-pMHC complexes and a surrounding ring of adhesion complexes. The degree of clustering of the TCR-pMHC complexes is known to be indicative for whether the T-Cell will be activated as a result of this signaling process [14]. This is an example of the fundamental importance of membrane physics for the understanding of signal transduction processes, but so far only qualitative insight has been gained. Even though attempts have been made to model the T-Cell APC contact mathematically [81, 64, 65, 63], current models of this process lack constraints on the mean curvature and enclosed volume of the cell membrane which are crucial features of the accepted models [77, 92] of lipid membranes.

2.1 Amphiphilic Molecules

The basic constituent of a cell membrane is a self assembled bilayer of amphiphilic molecules. They typically consist of a hydrophilic head group, and two hydrophobic, possibly partially unsaturated, hydrocarbon chains. The length of

[6] T-Cells form a part of the adaptive immune system. In an adaptive immune response the main classes of T-cells kill infected cells (cytotoxic T-Cells) or help activating other immune cells (helper T-Cells) [4].

the hydrocarbon chains ranges from fourteen to twenty CH_2 groups, yielding a thickness of the bilayer of 4-5 nm [15]. Most head groups are phosphate groups $-P-OOH-O-R$. Under physiological conditions the head group is not only polar but ionized. The exact chemical composition of lipid membranes in cells is complex. There are typically hundreds of different constituents, which have a highly asymmetric transversal distribution, regulated by a number of poorly characterized proteins [56]. We will absorb the chemical details into the material constants of the continuum model. A lateral organization has been observed in artificial giant unilamellar vesicles [12]. The existence and physiological relevance of these domains of different lipid composition (lipid rafts) in cells is still debated [74].

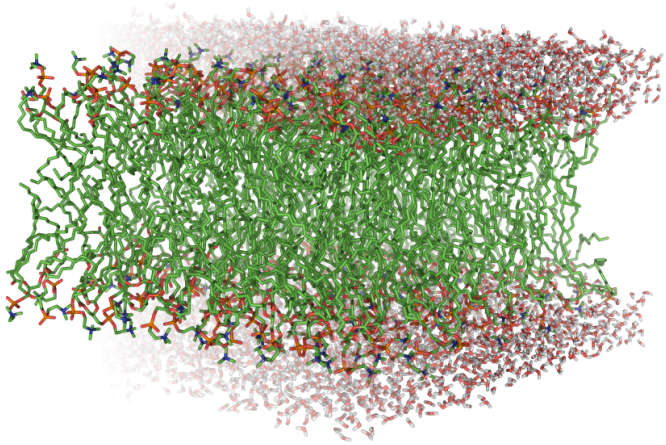

Figure 2.2: Visualization of a lipid bilayer surrounded by water molecules using PyMOL [29] and molecular dynamics simulation data from Heller et al. [52]. The different colors indicate the type of chemical elements in the molecules. White indicates hydrogen, red indicates oxygen, green indicates carbon, blue indicates nitrogen and orange indicates phosphorus. Note that the hydrogen atoms in the lipid molecules are not displayed to increase the clarity of the Figure.

2.2 A continuum view of the bilayer

The need of a continuum description stems from the lack of relevance of the molecular details for most biological applications and the need for a computationally tractable model.

Several different continuum models for lipid bilayers have been discussed in the literature [94, 39, 79, 103, 104, 108, 92]. The general consent is that the free

energy of a lipid bilayer depends on it's extrinsic curvature. The models differ in their assumptions on constraints such as the conservation of the integrated mean curvature. We will introduce a simple universally accepted model without constraints first, and discuss the different constraints later on.

The requirement for a physically meaningful model of the lipid bilayer is the invariance of the bilayer with respect to euclidian transformations i.e translations and rotations. Given a transversal extension of 4-5 nm and a lateral scale on the order of μm we can treat lipid bilayers as two dimensional [92]. Furthermore we note that the bilayer behaves like a fluid, there are no in–plane stresses in response to locally shearing or stressing the bilayer. Thus we require the model to be invariant with respect to reparametrizations of the bilayer.

Let Ω be our model membrane, we assume it is a sufficiently differentiable 2-Manifold embedded in $\mathbb{R}^3$. It is defined by

$$\Omega = \left\{\vec{r}(v,w) : (v,w) \in U \subset \mathbb{R}^2\right\} \qquad \text{with} \quad \vec{r} = \begin{pmatrix} r_1(v,w) \\ r_2(v,w) \\ r_3(v,w) \end{pmatrix}.$$

We use the symbol u_i to denote either v or w. In the following we will use the comma notation for derivatives of quantities x

$$x_{,i} = \frac{\partial x}{\partial u_i}.$$

Multiple indices following the comma indicate higher derivatives w.r.t. these indices. Setting one parameter u_i constant we define tangent vectors at the point (v_0, w_0)

$$\vec{r}_{,v} = \frac{\partial}{\partial v}\vec{r}(v,w_0)\bigg|_{v_0} = \vec{t}_1$$
$$\vec{r}_{,w} = \frac{\partial}{\partial w}\vec{r}(v_0,w)\bigg|_{w_0} = \vec{t}_2$$

In our definition of the tangent vectors we already made use of the embedding of Ω in $\mathbb{R}^3$, we can go further and define a[7] normal vector as

$$\vec{n} = \frac{\vec{t}_1 \times \vec{t}_2}{|\vec{t}_1 \times \vec{t}_2|}$$

The components g_{ij} of the metric tensor g are

$$g_{ij} = \vec{t}_i \cdot \vec{t}_j$$

The inverse of g_{ij} will be g^{ij}. We define the second fundamental form l through its components l_{ij}

$$l_{ij} = \vec{n} \cdot \vec{t}_{i,j} = -\vec{t}_i \cdot \vec{n}_{,j}\,. \tag{2.1}$$

Both definitions of l_{ij} are related by partial integration.

Geometrically the second fundamental form at a point (v_0, w_0) defines a quadric. It measures the deviation of Ω from the tangent plane at (v_0, w_0) in a small neighborhood around the considered point. If we diagonalize the quadric

[7] We will later choose an orientation such that the mean curvature on a sphere is negative.

l we can read off the local behavior, if l is positive or negative definite the whole surface in the neighborhood is on one side of the surface. In the indefinite case the local surface lies on both sides of the tangent plane. If l is semidefinite Ω lies in the plane in one direction, and curves to one side of the tangent plane in the perpendicular direction, both directions are given by eigenvectors of l.

The second fundamental form measures the curvature $k_n(s)|_{s=0}$ of a curve, naturally parameterized by s through (v_0, w_0), which is given by the intersection of Ω and the plane spanned by the normal and a tangent of Ω at (v_0, w_0). We have the curve $C = \{y|y(s) = \vec{r}(v(s), w(s))\}$. By the definition of the tangent we have $y'(s) = u_i'(s)\vec{r}_{,i}$ and thus for the second derivative

$$y''(s) = u_i''(s)\vec{r}_{,i} + u_i'(s)u_j'(s)\vec{r}_{,ij}\,. \tag{2.2}$$

Since the parameter s is the arclength of C we have

$$k_n(s) = ||y''(s)||.$$

From the theory of curves [99] we know that y'' is perpendicular to y' . Since C lies in the plane spanned by y' and $\vec{n}$, y'' is parallel to $\vec{n}$. Thus we can evaluate the curvature (2.2) by multiplying the second derivative (2.2) by $\vec{n}$

$$k_n(s) = u_i'(s)u_j'(s)\left(\vec{r}_{,ij}\cdot\vec{n}\right),$$

Here we recovered the second fundamental form.

Now we can ask for which directions the normal curvature becomes extremal. Thus we want to extremize the expression $u'^T l u'$, which is subject to the constraint $u'^T g u' = 1$ enforcing C to be parameterized by its arclength. Introducing a Lagrange multiplier λ and differentiating for u' we arrive at the generalized eigenvalue problem

$$lu' - \lambda g u' = 0$$

The eigenvalues κ_1 and κ_2 are called principal curvatures. Using these we define the surface properties central to this thesis:

Definition 1. *The mean curvature H of a surface is the mean of the principal curvatures*

$$H = \frac{\kappa_1 + \kappa_2}{2} = \frac{g^{ij}l_{ij}}{2det(g)} \tag{2.3}$$

The Gaussian curvature K is the product of the principal curvatures

$$K = \kappa_1\kappa_2 = \frac{det(l)}{det(g)} \tag{2.4}$$

Note that some authors define the mean curvature H not as the mean of the principal curvatures, but instead as the trace of the second fundamental form. Thus these definitions differ by a factor 2.

Let us express the derivative of the tangent vectors in terms of the threebein $\vec{r}_{,i}$ and $\vec{n}$

$$\vec{r}_{,ij} = \Gamma_{ij}^k\vec{r}_{,k} + L_{ij}\vec{n}. \tag{2.5}$$

Multiplying with $\vec{n}$ we recover equation (2.1), and therefore $L_{ij} = l_{ij}$. We can derive an expression for the Levi-Civita connection Γ by multiplying equation

(2.5) by $\vec{r}_{,l}$. The resulting terms $\vec{r}_{,ij} \cdot \vec{r}_{,l}$ can be written as linear combinations of derivatives of components of the metric [31]

$$g_{ij,k} = \vec{r}_{,ik} \cdot \vec{r}_{,j} + \vec{r}_{,i} \cdot \vec{r}_{,jk} \,,$$

leading to

$$\Gamma^k_{ij} = \frac{1}{2} \left(g_{kj,i} + g_{ik,j} - g_{ij,k} \right) g^{kl}.$$

Now we have the necessary tools to define the Hamiltonian of a closed vesicle Ω, which is named after Canham [19] and Helfrich [51].

Definition 2. *The Canham–Helfrich Hamiltonian of a vesicle Ω is*

$$F = \int_\Omega \left(\kappa (H - C_0)^2 + \bar{\kappa} K \right) dA \tag{2.6}$$

Here κ and $\bar{\kappa}$ are material constants measuring the rigidity of the lipid bilayer towards bending. The constant C_0 is a spontaneous curvature, to allow for membranes with a non flat relaxed shape.

The integral of the Gaussian curvature K is a topological invariant [31]. Since we are only interested in closed vesicles, which do not change their topology, we drop the term involving K. Note that models taking account of domains of different lipid composition leading to different $\bar{\kappa}$ would have to take into account boundary terms arising from the Gaussian curvature integral.

The electrostatic forces between the amphiphilic molecules lead to a strong resistance of the bilayer towards stretching; an increase of the average distance between the hydrophilic head groups would expose the hydrophobic tails to water. Shrinking of the membrane is resisted by the repulsive forces between the head groups [56]. Thus we have to add the constraint

$$\int_\Omega dA = A_0 \tag{2.7}$$

of conserved area A_0 to the Canham–Helfrich Hamiltonian. The volume of a cell does not change on the timescales covered in our simulations. We can use Stoke's theorem to express the conserved volume in terms of a surface integral

$$\int_\Omega \frac{1}{3} \vec{r} \cdot \vec{n} dA = V_0. \tag{2.8}$$

The Canham–Helfrich Hamiltonian, together with the two constraints (2.7) and (2.8) gives us the simplest models of a lipid bilayer. With the constraints enforced by Lagrange multipliers we have

$$F = \int_\Omega \left(\kappa (H - C_0)^2 + \lambda + \Delta p \frac{1}{3} \vec{r} \cdot \vec{n} \right) dA$$

The Lagrange multiplier λ has the interpretation of a tensile stress, whereas Δp is a pressure difference [112].

2.2.1 The Bilayer Couple Hypothesis

On a microscopic level a lipid bilayer is composed of two sheets. The finite distance between these sheets causes the areas of the two sheets to differ. Early on it was realized that interactions between the two sheets of the lipid bilayer could affect its behavior [94, 39]. The distribution of phospholipids and membrane proteins [94, 7] may be changed asymmetrically across the bilayer, causing the areas of the two leaflets change relative to each other. Since both halves are coupled by van der Waales interactions, changes of the ratio of the areas will induce deformations of the membrane. Different shapes of a membrane will have different ratios of areas because the finite thickness of the membrane will translate its mean curvature into different areas of the bilayers sheets.

A mathematical model of the bilayer couple hypothesis [103] has been built upon the Canham–Helfrich Hamiltonian. An additional nonlocal term is introduced which enforces a constant area difference ΔA_0 between the two layers of the membrane

$$d \int_\Omega H dA = \Delta A_0.$$

Here d measures the distance between the sheets of the bilayer. Experiments inducing a change of area and volume of vesicles by heating, thus inducing shape transformation, showed behavior which was either consistent with a constrained or with an unconstrained area difference. This motivated the investigation of a model interpolating between both existing models. The constraint is relaxed and replaced with a harmonic potential penalizing an area difference, which deviates from the relaxed area difference [72]

$$F_A = \frac{\kappa' \pi}{2Ad^2} \left(\Delta A - \Delta A_0\right)^2.$$

In this thesis we will make use of this so called area difference elasticity (ADE) model. Furthermore we will relax the area and volume constraints. This has the technical reason that in the numerical treatment of the equations of motion the terms defining the Lagrange multipliers will result in poorly conditioned matrices[8], which deteriorate the convergence of the numerical solution.

2.3 The Equations of Motion

Given the Canham–Helfrich Hamiltonian we will derive the normal stresses f acting on the lipid bilayer. Since we consider the bilayer as a fluid, and constructed a Hamiltonian which is invariant under reparametrizations, we can ignore in–plane stresses. Of course, as soon as we would allow membranes that are not closed, in plane stresses would add nontrivially to the boundary conditions [21]. The normal stresses are given by the variation of Ω in the direction

[8] in most cases the matrices were singular up to machine precision

of the surface normals

$$f = \delta\Bigg(\int_\Omega \left(\kappa(H-C_0)^2 dA\right) + \frac{\kappa'\pi}{2Ad^2}\left(\Delta A - \Delta A_0\right)^2 + a\left(A-A_0\right)^2 + v\left(V-V_0\right)^2 \Bigg). \tag{2.9}$$

The following derivations are based on [112, 31]. Other approaches to derive the stresses acting on a lipid bilayer can be found in [22, 21, 20, 5, 105, 106]. The variation of $\vec{r}$ in the direction of the surface normal $\vec{n}$ is given by

$$\vec{r}_n(v,w) := \vec{r}(v,w) + \delta\vec{r}(v,w) = \vec{r} + \psi(v,w)\vec{n}(v,w)$$

with a sufficiently small smooth function ψ. We find the variation of the tangents as

$$\begin{aligned}
\delta\vec{r}_{,i} &= (\vec{r}_n - \vec{r})_{,i} \\
&= \vec{r}_{,i} + (\delta\vec{r})_{,i} - \vec{r}_{,i} \\
&= (\delta\vec{r})_{,i} \\
&= \psi_{,i}\,\vec{n} + \psi\vec{n}_{,i}
\end{aligned}$$

We proceed with the variation of the components of the metric

$$\begin{aligned}
\delta g_{ij} &= g_{ij}(\vec{r}_n) - g_{ij}(\vec{r}) \\
&= \vec{r}_{n,i}\cdot\vec{r}_{n,i} - \vec{r}_{,i}\cdot\vec{r}_{,j} \\
&= \delta\vec{r}_{,i}\cdot\vec{r}_{,j} + \vec{r}_{,i}\cdot\delta\vec{r}_{,j} + \delta\vec{r}_{,i}\cdot\delta\vec{r}_{,j}
\end{aligned}$$

Since we choose ψ to be small and to have small derivatives, we drop all terms not linear in ψ and it's derivatives

$$\delta g_{ij} = (\psi_{,i}\,\vec{n} + \psi\vec{n}_{,i}\,)\cdot\vec{r}_{,j} + \vec{r}_{,i}\cdot(\psi_{,j}\,\vec{n} + \psi\vec{n}_{,j}\,).$$

Recalling the definition of the second fundamental form, we are led to

$$\delta g_{ij} = -2\psi l_{ij}.$$

The variations of other quantities are found in a similar manner [112, 31]

$$\begin{aligned}
\delta\det(g_{ij}) &= \delta\det(g) = -4\psi H g \\
\delta\sqrt{g} &= -2\psi H\sqrt{g} \\
\delta l_{ij} &= \psi_{,ij} + \psi(K g_{ij} - 2H l_{ij}) - \Gamma^k_{ij}\psi_{,k} \\
\delta H &= \psi(2H^2 - K) + \frac{1}{2}g^{ij}\nabla_i\psi_{,j} \\
\delta V &= \int_\Omega \psi dA,
\end{aligned}$$

The first term in equation (2.9) becomes

$$\begin{aligned}\delta \int_\Omega \left(\kappa(H-C_0)^2 dA\right) &= \kappa \int_\Omega \left(2(H-C_0)\delta H dA + (H-C_0)^2 \delta dA\right) \\ &= \kappa \int_\Omega \Bigg(2(H-C_0)[\psi(2H^2-K) + \frac{1}{2} g^{ij}\nabla_i \psi_{,j}] \\ &\quad - 2(H-C_0)^2 \psi H\Bigg) dA\end{aligned}$$

After rearranging, partially integrating, and making use of the definition of the Laplacian as the trace of the covariant derivative[31]

$$\Delta = g^{ij}\nabla_i \nabla_j, \tag{2.10}$$

we have

$$\begin{aligned}&\delta \int_\Omega \left(\kappa(H-C_0)^2 dA\right) \\ &\qquad = \kappa \int_\Omega \Big(2(H-C_0)(H^2-K-C_0) + \Delta H\psi\Big) dA. \end{aligned} \tag{2.11}$$

Note that the expressions reported in [112, 92] differ due to differences of a factor 2 in the definition of the mean curvature (2.11), which can be absorbed into the bending rigidity κ.

We proceed with the derivative of the potential of the mean curvature differences

$$\delta\left(\frac{\kappa'\pi}{2Ad^2}\left(\Delta A - \Delta A_0\right)^2\right) = \frac{\kappa'\pi}{2Ad}\left(\Delta A - \Delta A_0\right)\int_\Omega \delta(H dA).$$

In the variation of the integral over the mean curvature all terms proportional H cancel. Since the Laplacian of ψ only contributes a boundary term we find

$$\delta\left(\frac{\kappa'\pi}{2Ad^2}\left(\Delta A - \Delta A_0\right)^2\right) = -\frac{\kappa'\pi}{2Ad}\left(\Delta A - \Delta A_0\right)\int_\Omega K dA$$

The variation of the term penalizing area changes evaluates to

$$a\delta\left(A - A_0\right)^2 = 2a\left(A - A_0\right)\int_\Omega -2H\psi dA,$$

For the volume term we obtain

$$v\delta\left(V - V_0\right)^2 = 2v\left(V - V_0\right)\int_\Omega \psi dA.$$

Putting everything together the normal stress of a lipid bilayer is given by

$$\begin{aligned} f = \int_\Omega \Big(&\kappa(2(H-C_0)(H^2-K-C_0)\psi + H\Delta\psi) \\ &- \frac{\kappa'\pi}{2Ad}\left(\Delta A - \Delta A_0\right) K\psi \\ &- 4a\left(A - A_0\right) H\psi \\ &+ 2v\left(V - V_0\right)\psi\Big) dA. \end{aligned}$$

Note that the terms ΔA, A and V are integrals over the whole surface Ω, and thus render the stresses non-local.

When we formulate the equations of motion, we have to consider the water surrounding the membrane. At the typical dimension of a cell of some micron, and typical velocities of microns per minute, the Reynolds number of the fluid is so small that inertial forces can be neglected [50]. In principle this would lead to a an incompressible Stokes equation [92]

$$\begin{aligned} \nabla p(x) - \eta \nabla^2 v(x) &= K(x), \\ \nabla \cdot v &= 0 \end{aligned}$$

with pressure p, velocity field $v(x)$ and viscosity η. Forces external to the liquid are given by $K(x)$. In a realistic model these would not only be those that stem from the moving boundary, which is formed by the cell membrane, but also contain unknown viscoelastic terms from the cytoskeleton. Here we will take a different approach and simply assume that the lipid bilayer moves freely, but all motion is strongly dampened. Thus we drop the inertial term in the equations of motion.

To estimate the damping λ we solve the Stokes equation formally by Fourier transformation [92]. The velocity field can then be expressed as

$$v(x) = \int d^3x' \mathcal{O}(x, x') K(x') \tag{2.12}$$

Where the entries of the Oseen Tensor are given by

$$\mathcal{O}_{ij}(x, x') = \frac{1}{8\pi\eta|x - x'|} \left(\delta_{ij} + \frac{(x_i - x'_i)(x_j - x'_j)}{|x - x'|^2} \right).$$

Since the lipid bilayer in impermeable to the surrounding fluid, we can identify the normal velocity of the fluid at the membrane with its deformation. For a sphere with uniform normal forces the integral (2.12) evaluates to $\frac{\pi}{8\eta}$. Neglecting the $1/x$ coupling introduced by the fluid we use this as our damping term, and arrive at

$$\begin{aligned} \int_\Omega \frac{\partial \vec{r}}{\partial t} \psi = \frac{\pi}{8\eta} \vec{n} \int_\Omega \Big(& \kappa (2(H - C_0)(H^2 - K - C_0)\psi + H \Delta \psi) \\ & - \frac{\kappa' \pi}{2Ad} (\Delta A - \Delta A_0) K \psi \\ & - 4a (A - A_0) H \psi \\ & + 2v (V - V_0) \psi \Big) dA \end{aligned} \tag{2.13}$$

This is the equation of motion that we will solve in this thesis.

Chapter 3

Solution Strategies for PDEs

In most cases numerical solutions to time–dependend PDEs are obtained in two steps. First the PDE is discretized into a system of ODEs, then the ODEs are integrated separately. This thesis will make use of the Finite Element Method (cf. Chapter 4).In this chapter we will review alternative methods of discretizing a PDE, and give reasons why these methods were rejected for this thesis.

3.1 Finite differences

The basic idea of finite differences is to replace derivatives with difference quotients

$$\frac{\partial f(x,\dots)}{\partial x} = \frac{f(x+h,\dots) - f(x,\dots)}{h}$$

turning the differential equation into a difference equation. Since the evaluation of derivatives requires knowledge of the values of f at neighboring points we overlay the domain, Ω on which the problem is posed, with a grid with grid constant h. On the boundary of Ω we change the value of h in order to evaluate f on the boundary. The evaluation of the difference equation on all points of the grid leads to a system of n algebraic equations, where n is the number of grid points.

We can apply this approach to (2.13) posed on a patch of membrane, such that Ω is in a Monge representation , i.e the graph of a function $(v, w) \mapsto h(v, w)$. Now some limitations of the finite difference method become apparent. The finite difference operator breaks the symmetries of the original system, by singling out two directions, and additionally it breakes the parity by the asymmetry of the finite difference. Moving towards an arbitrarily shaped membrane reveals further problems. Since one now needs more than one chart. Consider two overlapping coordinate patches. Since both charts have to be compatible on their overlap, we have to adjust the grid in at least one of the charts. We are not totally free in moving points in the image of one chart without affecting the other. A point of our discretization can be in the image of an other chart, therefore we force our discretization from the domain for our problem onto it's image. Numerical tests with an early prototype,which was developed for this thesis, indicated the need for very fine grained discretizations both in space and time to ensure stability. The need for a small grid constant h reflects results by

Stokely and Wu [98], who analyzed the systematic error of curvature estimates on three dimensional grids. Their conclusion was that in order to correctly measure the curvature a neighborhood containing up to one hundred surface voxels[1] would have to be sampled. With high resolution and small timesteps one can get a better estimate of the curvatures as a spatio–temporal average without modifying the Laplacian to sample larger neighborhoods. Still, the computational cost associated with highly resolved grids and small timesteps make this method unsuited for the modeling of membranes.

3.2 Level Set Methods

In the previous section we saw that the naive finite differences approach of discretizing coordinate charts led to serious numerical difficulties. Some of these can be alleviated if instead of charts in monge representation we use an implicit function to describe the surface we are interested in

$$\Omega(t) = \{\vec{r} | \phi(\vec{r}, t) = 0\}.$$

This idea was initially conceived by Osher and Sethian and became known as Level Set Methods [76]. The unit normals of $\Omega(t)$ are given by the restriction of

$$\vec{n} = -\nabla \frac{\phi(\vec{r}, t)}{|\phi(\vec{r}, t)|}$$

to $\Omega(t)$ [75]. This has the advantage that by making use of the extension of the normals into the bulk , we can improve the aforementioned systematic errors for curvature estimates for finite difference schemes . The motion of $\Omega(t)$ can be analyzed as the convection of $\phi(\vec{r}, t)$ with a normal velocity field $v_n = \vec{v} \cdot \vec{n}$ [75, 93]

$$\frac{\partial \phi(\vec{r}, t)}{\partial t} + v_n |\phi(\vec{r}, t)| = 0. \tag{3.1}$$

In our case the velocity field would be a function of the geometry of Ω. The performance of the level set method depends on different factors. We need a suitable choice of the implicit function ψ defining the level set. Too shallow or too steep gradients could negatively affect numerical precision. An accepted choice for ψ is setting it to the signed distance $d(\vec{r}, t)$ to the initial level set $\phi = 0$. By convention one defines the distance $d(\vec{r}, t)$ to be positive if $\vec{r}$ is inside the volume that is enclosed by the level set. Efficient algorithms for constructing ψ have been developed [78, 102]. The basic idea is to make use of the fact that $|\nabla d| = 1$ and solve for the steady state of eq. (3.1)

$$\frac{\partial \psi(\vec{r}, t)}{\partial t} + sign(\phi)(|\nabla d| - 1) = 0$$

with the initial condition

$$\psi(\vec{r}, 0) = \phi(\vec{r}, t).$$

We can extend properties $p(\vec{r}, t)$ that are only defined on the evolving surface into properties $q(\vec{r}, t)$ defined in the the bulk in a similar manner. Such an

[1] A voxel is the n-dimensional generalization of a pixel.

extension is necessary, if the velocity field in (3.1) depends on surface properties. Again we solve for the steady state of a convection equation for the new bulk properties [24].

$$\frac{\partial q(\vec{r},t)}{\partial t} + sign(\phi)(-\vec{n}_\phi \cdot \nabla q) = 0,$$

with the initial condition

$$q(\vec{r},0) = p(\vec{r},t).$$

This enforces the gradient of q to be parallel to the surface, therefore the change of q along $\vec{n}$ will be as small as possible in a neighborhood around the surface.

A major issue for level set methods is the discretization of the whole embedding space. Therefore the storage complexity and time complexity of level set algorithms with linear extension n samples in each of the d-directions scales as $O(n^d)$. Assuming two double precision values per sample, one for the implicit function ϕ and one for an auxiliary property, the memory needed to store a single time step with an extension of 512 samples is 2 GB. At the time of writing, the total amount of random acess memory on commodity hardware is on the order of 2 GB. Therefore more economical algorithms are required. A number of data structures have been devised to achieve better storage complexity[2] , based on storing only a narrow band around the level set $\phi = 0$ [2], making use of hierarchical space partitions [67], or combinations of both [54]. The drawback of all these schemes is that the random access of values of ϕ is not of time complexity[3] $O(1)$. Efficient algorithms are still a matter of active research.

An approach to model the behavior of lipid vesicles using a variant of level set methods has been made by [35] and [18]. Their models are capable of simulating the dynamics of vesicles including topological changes such as fusion of membranes. Yet the limitations of the level set methods apply here as well. Simulations in 3 dimensions with more than 64 samples per direction were not reported in [35] or [18]. If a level set method is to be coupled to the biochemistry of a cell, the use of a structured grid leads to stronger restrictions on the resolution. This is due to the cost for the storage of concentrations of the membrane bound molecule complexes in the simulated reaction network. For realistic networks the number of chemical species tracked in a simulation can be several hundred or even more. Simulations of the biochemistry could be executed on a different discretization, but additional effort is needed to couple the different discretization schemes.

3.3 Discrete Differential Geometry

Instead of discretizing the derivatives of functions, one could start with a manifold, which is intrinsically discrete and formulate the differential equation in terms of the discrete analogs of differential forms [53, 30]. This approach has the advantage that, by construction, local invariants of the modeled system are

[2]Storage complexity measures how the amount of memory that is needed to carry out an algorithm asymptotically scales with the size of its input, or more generally how it scales with a parameter derived from the input. In this case the relevant parameter is the linear resolution or the related number of samples n in each direction of the grid that is used to discretize the level set equations.

[3]The time complexity of an algorithm is an asymptotic expression for the time it takes to execute an algorithm, as a function of the length of the input.

preserved in the discrete setting. A violation of these leads to spurious oscillations and other features of solutions, which are only artifacts of an improper discretization [16, 30]. The basic ingredient is a triangular or tetrahedral mesh (in the 2-d resp. 3-d case), i.e. a simplicial complex, serving as a discrete analog of the differential manifold in the continuous setting.

This introduction to the ideas of discrete differential geometry follows [30], most of the concepts introduced here can also be found in [73]. A k-simplex σ_k is the nondegenerate convex hull of k geometrically distinct points

$$\sigma_k = \{v_1, \dots v_n\} = \{x \in \mathbb{R}^n | x = \sum_{i=0}^{k} \alpha^i v_i \,\text{with}\, \alpha^u \leq 0 \,\text{and}\, \sum_{i=0}^{k} a^i = 1\}.$$

The v_i are points in $\mathbb{R}^n$, which we call the vertices of σ_k. The order of the vertices induces an orientation of a simplex, all even permutations of the set $\{v_i\}$ give rise to the same orientation. We can form $(k-1)$-simplices, called faces, from a k-simplex by removing one vertex. The formal sum

$$\partial\{v_1, \dots, v_n\} = \sum_{j=0}^{k} (-1)^j \{v_1, \dots, \hat{v_j}, \dots, v_n\},$$

where $\hat{v}_j$ indicates the omission of v_j, defines the boundary of a k-simplex. We now define a simplicial complex as a set $\mathcal{K}$, such that each face of a simplex in $\mathcal{K}$ is in $\mathcal{K}$ as well, and the intersection of two simplices is either empty or a face. We can now define a discrete n-manifold as a simplicial complex, such that a face of a n-simplex has one or two incident n-simplices. The n-simplex is considered on the boundary of $\mathcal{K}$ in the case of one incident n-simplex.

A k-chain c on a simplicial complex $\mathcal{K}$ is a linear combination of all k-simplices in $\mathcal{K}$

$$c = \sum_{\sigma \in \mathcal{K}} c(\sigma)\sigma$$

Requiring linearity we can extend the boundary operator to chains.

Now we define differential forms on the discrete manifold by identifying the duals of chains (cochains) with differential forms. Thus a differential k-form is a map ω taking a k-chain $c \in \mathcal{C}_k$ to the real numbers

$$\begin{aligned} \omega : c \in \mathcal{C}_k &\to \mathbb{R} \\ c &\mapsto \omega(c) \end{aligned}$$

Thus we integrate a discrete form ω over a chain c as

$$\int_c w = \sum_{\sigma \in \mathcal{K}} c(\sigma)\omega(\sigma)$$

Further geometric concepts can be applied to the discrete setting in a similar spirit [53, 30].

While the notion of discrete differential geometry was successfully applied to different problems [16, 38, 87], discrete differential geometry is a field of active research, and the convergence of discrete systems to their continuous analogs is not yet fully understood [87].

Because of the drawbacks of the algorithms used here, we employ the Finite Element method, which we will discuss in the next Chapter. The mathematical fundamentals are well understood [25] and efficient algorithms for solving finite element problems are known [17].

Chapter 4

The Finite Element Method

The Finite Element Method (FEM) is tailored to solve PDE's on irregularly shaped domains. It shares its underlying principles with the Rayleigh-Ritz method for solving variational problems. The function space on which the problem is posed, is approximated by a finite dimensional subspace. This transforms the initial variational or weak problem into an algebraic one. The approximated solution will converge to the exact solution in the limit of a countably infinite dimensional subspace.

The FEM and Rayleigh-Ritz methods differ in the way how these series of subspaces are constructed. For practical considerations it is important to keep computational time manageable, in addition to ensuring convergence. Good approximations of the exact solution should be achieved with a relatively low dimensional subspace. On domains with certain symmetries one can find basis functions, which are orthogonal to any but a few others and respect the boundary conditions. Then a subset of the basis functions is used to span the finite dimensional space, on which an approximation of the problem can be easily solved.

For domains with arbitrary shape this is not possible in practice, but neither is dealing with hundreds of millions of pairs of functions with non vanishing scalar product. The way out of this dilemma is to restrict the support of the basis functions to disjoint subsets that cover the domain. The FEM generally uses polynomial functions on polygonal or polyhedral domains –the elements – to yield a numerically tractable system of algebraic equations. Recommendable introductory texts on the finite element method are the lecture notes by Giladi [45] and a textbook by Braess [17].

4.1 A one dimensional example

We now illustrate how to find the weak solution of a one dimensional heat equation for a medium with an arbritrary smooth, position dependent diffusion constant $\kappa(x)$. We are searching for a function $T(x)$ such that:

$$\frac{\partial}{\partial t}T(t,x) = \frac{\partial}{\partial x}\left(\kappa(x)\frac{\partial}{\partial x}T(t,x)\right)$$

with the Neumann boundary conditions

$$\frac{\partial}{\partial x}T(t,0) = \frac{\partial}{\partial x}T(t,1) = 0.$$

First we express $T(t,x)$ in terms of basis functions $\phi_i(x)$ with time dependent coefficients $a_i(t)$. We then construct the weak formulation using $\phi_j(x)$ as test functions

$$\int_0^1 dx \frac{\partial}{\partial t} \sum_i a_i(t)\phi_i(x)\phi_j(x) = \int_0^1 dx \frac{\partial}{\partial x}\left(\kappa(x)\frac{\partial}{\partial x}\sum_i a_i(t)\phi_i(x)\right)\phi_j(x).$$

Integrating the rhs by parts and making use of the boundary conditions we have

$$\sum_i \frac{\partial a_i(t)}{\partial t}\int_0^1 dx\phi_i(x)\phi_j(x) = -\sum_i a_i(t)\int_0^1 dx\kappa(x)\frac{\partial\phi_i(x)}{\partial x}\frac{\partial\phi_j(x)}{\partial x}.$$

This is a system of ordinary differential equations. We can rewrite the summation as a multiplication of a vector $\vec{a}(t)$ with components $a_i(t)$ and matrices M and S with

$$M_{ij} = \int_0^1 dx\phi_i(x)\phi_j(x)$$

and

$$S_{ij} = -\int_0^1 \frac{\partial\phi_i(x)}{\partial x}\frac{\partial\phi_j(x)}{\partial x}\kappa(x)dx.$$

The system of ODEs now reads

$$\frac{\partial\vec{a}(t)}{\partial t}M = \vec{a}(t)S.$$

So far we did not make use of any technique particular to the FEM. The advantages of finite elements will become apparent, when we explicitly construct the mass matrix M and stiffness[1] matrix S. Assume the interval $I = [0,1]$ is divided into n intervals $n_i = (i/n, (i+1)/n)$ with $i \in 0 \ldots n-1$. Two linear functions $\psi_i^1(x)$ and $\psi_i^2(x)$ are defined on each interval n_i, such that each equals 1 evaluated on one end of the interval and 0 on the other.

$$\psi_i^1(x) = \begin{cases} -nx + 1 + i & \text{if } x \in n_i \\ 0 & \text{else} \end{cases}$$

$$\psi_i^2(x) = \begin{cases} nx - i & \text{if } x \in n_i \\ 0 & \text{else} \end{cases}$$

Requiring continuity of the solution across element boundaries we have $n+1$ degrees of freedom[2]. We now construct a set of basis functions $\phi_j(x)$ $j \in \{0 \ldots n\}$ which is associated with the element boundaries (nodes) by setting

$$\begin{aligned} \phi_0(x) &= \psi_0^1(x) \\ \phi_j(x) &= \psi_{j-1}^2(x) + \psi_j^1(x) \qquad j \in \{1 \ldots n-1\} \\ \phi_n(x) &= \psi_{n-1}^2(x). \end{aligned}$$

[1] The designations of M and S as mass resp. stiffness matrix stem from their interpretation in solid mechanics.

[2] The coefficients of the two basis function at an element boundary have to be the same.

Now we have one basis function for each degree of freedom. Evaluating the Mass– and Stiffness– matrices we notice that only basis functions associated with neighboring nodes have overlapping support. In order to evaluate the integrals analytically, we now introduce the additional assumption $\kappa(x) = 1$.

$$M = \frac{1}{n}\begin{pmatrix} \frac{1}{3} & \frac{1}{6} & & & & & & \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & \frac{1}{6} & \frac{2}{3} & \frac{1}{6} & \\ & & & & & \frac{1}{6} & \frac{1}{3} & \end{pmatrix}$$

$$S = \frac{1}{n}\begin{pmatrix} 1 & -1 & & & & & & \\ -1 & 2 & -1 & & & & & \\ & \ddots & \ddots & \ddots & & & & \\ & & -1 & 2 & -1 & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & -1 & 2 & -1 & \\ & & & & & 1 & -1 & \end{pmatrix}.$$

The matrices are sparse, which will hold for higher dimensional domains as well, thus the resulting system of algebraic equations can efficiently be solved numerically. Furthermore, we observe that the 1-dimensional finite difference discretization of the Laplacian being recovered by our discretization, this is of course only the case for this simple geometry, more complex examples don't have finite difference equivalents.

4.2 Mathematical Background

This section gives a short outline of the theory of Finite Element Methods. It is not meant to substitute introductory texts, but it fixes notation and motivates the choice of the set of basis functions that we use here. The function space V on which we search our solution can not be arbitrarily chosen. Assume we were able to formulate our differential equation as a variational problem, by requiring the action of the system in question to be stationary. Thus we search the minimum of a functional J of functions v on some open set $\Omega \in \mathbb{R}^n$ given by

$$J(v) = \int_\Omega a(v,v) - l(v) \qquad v \in V, \tag{4.1}$$

where a is a symmetric positive bilinear form on V and l is a linear functional on V. One can prove [17] that $J(v)$ attains its minimum at u if and only if

$$a(u,v) = l(v) \quad \text{for all}\, v \in V. \tag{4.2}$$

We have to ensure two properties, the integral defining the functional has to exist for all functions in V, and we need a meaningful notion of convergence towards a solution, and thus completeness of the function-space wrt. a suitable norm.

Given a contains no derivatives higher than order k, these requirements are fulfilled by the Sobolev space $H^k(\Omega)$ [17], which is the space of all square integrable functions $u \in L_2(\Omega)$ which possess weak derivatives $\partial^\alpha u$ for all $|\alpha| \leq k$. The square integrability of u ensures the existence of the integral in eq. (4.1). One can show that $H^k(\Omega)$ is closed with respect to $||u||_k = (u,u)_k$, the norm induced by the scalar product

$$(u,v)_k = \sum_{|\alpha|\leq k} \int_\Omega \partial^\alpha u(x) \partial^\alpha v(x) dx.$$

Thus H^k is a Hilbert space. We also define the semi-norm

$$|u|_m = \sqrt{\sum_{|\alpha|=k} \int_\Omega \partial^\alpha u(x) \partial^\alpha u(x) dx}.$$

We have to discuss how to implement the boundary conditions. Given a Dirichlet boundary condition $u(x) = g$ on $\partial\Omega$, and a function u_0, which fulfills the boundary condition and for which a(u,u) exists, we can translate the problem by $-u_0$ to yield a homogeneous problem with a translated variable $w = u - u_0$ and $l_1(v) = v - a(u_0, v)$ [17]. A homogeneous Dirichlet problem is solved by considering the subspace $H_0^k(\Omega) \subset H^k(\Omega)$, where $H_0^k(\Omega)$ is the completion of $C_0^\infty(\Omega) \cap H^k(\Omega)$ with respect to the norm $|| \cdot ||_k$. An more intuitve way of looking at this construction is that we dispose of all functions that will violate our boundary conditions, and make sure that the resulting function space is a Hilbert space. $C_0^\infty(\Omega)$ is the space of smooth functions with compact support in Ω. In other words we choose our function space, on which we seek the solution, such that the Dirichlet boundary conditions are enforced. In contrast Neumann boundary conditions

$$\frac{\partial u}{\partial n} = n \cdot \nabla u = g \qquad \text{on } \partial\Omega$$

are implicitly enforced, and thus called natural boundary conditions. We make use of Greens theorem, and amend the rhs. of the variational problem by an integral over the boundary term g

$$(g,v) = \int_{\partial\Omega} gv.$$

Do not have to take care of Neumann boundary terms, therefore we do not give any more details here, and refer for a proof to Chapter II., §3 in the book by Braess [17] instead.

In the nonlinear case the requirement of an existing L_2-norm is not sufficient anymore. The functional $J(v)$ can now contain fixed but arbitrary powers p of v or its derivatives thus we need to generalize the definition of H^k to L_p-norms. These Sobolev spaces are denoted $W^{k,p}$, but they are not Hilbert spaces, which renders most of the theory developed for linear problems useless in the nonlinear case.

In our case one can show that the principal curvatures are in L_p, where $p > 2$ depends on the number of elements incident on a node [91]. Furthermore we can linearize the equations of motion at the latest time of the solution. Assume the second variation $\delta^2 H$ of the Canham–Helfrich Hamiltonian (cf. equation (2.6))

is positive. Then the problem of finding the solution of the equations of motion at later times can be viewed as solving an elliptic problem with the state of the membrane at the current time entering as a boundary condition [110]. Given we started with a positive second variation, we can make use of the continuity of the functional derivative, and infer that there is a small neighborhood around the current shape of the membrane, on which the second variation of H is positive as well. Of course, if the simulation is approaching an unstable point, we will experience Zenos Paradox of Achilles and the tortoise. We are forced to take smaller and smaller time steps to ensure the linearized problem is elliptic.

Since the second variation of H involves second derivatives of the functions v it is acting on, we have to require $v \in H^2$. Generally the construction of finite elements whose basis functions lie in H^2 involves the prescription of not only function values as coefficients, but also first and second derivatives as well as compatibility constraints on the element boundaries [17, 25]. A Method which only relies on function values is beneficial, since the coefficients have a direct interpretation. This makes techniques modifying and adapting the finite element mesh more efficient, since a mesh with new vertex positions naturally translates into H^2 basis functions without the need to solve constraint equations to determine derivatives. Such a method, based on subdivision surfaces, was recently suggested by Cirak et al. [26].

So far we did not explicitly discuss the discretization of H^2. in the numerical implementation we can only use a finite number of basis functions; thus we operate on a subspace $S_h \subset H^2$, where h is a parameter of the discretization, such as the largest radius of any element used. Now two questions arise: The first one is whether the discrete solutions converge towards the exact solution in the limit $h \to 0$. The second question is about an estimate of the error of a discrete solution.

In answering the the first question we follow [17]. First we need to specify the discretization of the domain Ω. We call a partition $\mathcal{T} = \{T_1, \ldots, T_j\}$ of Ω admissible if the elements exactly cover the closure of the domain $\bar{\Omega} = \bigcup_{i=1}^{j} T_i$, and the intersection of two elements is either one vertex of both elements, or one edge of both elements. Furthermore we define the following regularity criterion

Definition 3. *A family of triangulations $\mathcal{T}_h$ of a domain Ω is called shape regular, if for all h there exists a shape parameter $\kappa > 0$ such that for the radius of the in–circle ρ_T of each triangle T of radius h_T*

$$\rho_T \leq h_T/\kappa$$

holds. This forbids triangles which have angles approaching 0, and would become infinitely thin.

Given a family of shape regular triangulations $\mathcal{T}_h$ of the domain of the problem, we can now discuss the convergence of a finite element scheme. Assume we have polynomial basis functions of degree $t-1$, with $t > 2$. We define a interpolation operator $I_h : H^2 \to S_h$. The error of the interpolation $I_h u$ of a function u can now be estimated as

$$\sqrt{\sum_{T_i \in \mathcal{T}} ||u - I_h u||_m} \leq c(\Omega, \kappa, t) h^{t-m} |u|_t \qquad \text{for } u \in H^2(\Omega) \tag{4.3}$$

The parameter $c(\Omega, \kappa, t) > 0$ can be regarded as meassuring the suitability of a triangulation to discretize a given function space. We can relate the interpolation error to the approximation error of the finite element method with the help of Cea's lemma [17].

Lemma 1. *Let u_h be the solution to the variational problem (4.2) with V-elliptic, continuous bilinear form a*

$$a(v,v) \geq \alpha ||v||_{,m} \qquad \textit{and} \qquad |a(u,v)| \leq C||u||_m||v||_m \qquad \textit{for all } u, v \in V.$$

Then the error of the discrete solution is bounded by

$$||u - u_h|| \leq \frac{C}{\alpha} \inf_{v_h \in S_h} ||u - v_h||_m.$$

For an optimal interpolation operator this bound is just the error of the interpolation. An important observation is the dependence of the constant c in (4.3) on the shape parameter κ. Assume κ is given for a reference triangle T_r. Then we can find a bound c_I on κ for affine images T_I of T_r. This bound can be stated in term of the ratio of the circum radii $r_{I,r}$ and in–radii $\rho_{I,r}$ of T_I and T_r

$$c_I \leq \frac{r_r r_I}{\rho_r \rho_I}.$$

Now it becomes apparent, why we required the triangulation to be shape regular, the r.h.s. of (4.2) is large for very thin triangles. This is not only a weakness of our estimate, but also a feature in actual calculations [17]. Thus it makes sense to adapt the geometry of a mesh to avoid thin triangles [27]. A similar idea is the adaptation of the mesh to the shape of the solution. Regions of the solution displaying high gradients should be covered by smaller elements to adequately resolve the gradient in the solution [60]. Based on this notion many algorithms for the adaptation of the mesh geometry do not use the mesh geometry itself to determine how to change the discretization, but use local error estimates of the discretized solution. The gradient of these local error estimates with respect to the geometry of the mesh is used to adapt the mesh by moving along a flow defined by that gradient [60, 11]. We have to adapt our mesh, since simulations performed without iteratively improving the mesh quality, typically leads to very thin triangles and numerical instabilities (cf. Section 7.3). Cristini et al. [27] discussed the use of an approach solely based on the mesh geometry, which is independent of the differential equations describing the dynamics the the model under consideration. The basic idea is to treat the mesh as a network of springs and let the restoring forces drive the mesh into a better shape. We define a mesh energy as

$$E = \frac{1}{2} \sum_{e \in \text{edges}} \left(l(e) - l_0 \right)^2, \tag{4.4}$$

Where l_0 is resting length of the springs, while their actual length is given by $l(e)$. Minimizing the energy (4.4) obviously minimizes the root mean square deviation from the resting length l_0. In order to preserve the shape of the mesh we use an iterative scheme and project the resulting force on each vertex v to the tangent plane at the vertex

$$\dot{x} = P \sum_{i=1}^{\text{order}(v)} \left(l(e_i) - l_0 \right) \vec{e}_i, \tag{4.5}$$

$\vec{e}_i$ is a unit vector pointing away form the vertex in direction of the edge e_i; the projection to the tangent plane is given by $P = \mathbb{1} - \vec{n}\vec{n}^T$ where $\vec{n}$, is the unit normal vector of the tangent plane. We now solve the equations of motion of the vertices (4.5) and take five steps to relax the mesh after each time integration step. Since long wave perturbations of the node density relax slowly Cristini suggests to circumvent this by changing the mesh topology. Given that in our case the approximation error depends on the order of the vertices (cf. Section 5.2) we want to avoid those changes.

In an implementation of a finite element scheme we will not always be able to ensure that the basis functions used are in $H^k(\Omega)$. Consider numerical quadrature schemes, which will generally not be exact. The map from a reference triangle to a curvilinear element will have a Jacobian that is not constant, thus the function we are integrating is not a polynomial. But we can still ensure convergence. To achive that, the order of a complete polynomial, to which the quadrature is exact, has to be high enough, furthermore the map does not yield elements with angles of 0 or π and the deviation of the edges from straight lines is of order $O(h^2)$ or less [100]. The last condition is fulfilled by our choice of subdivision surfaces as finite elements, since the subdivision process is linear in the positions of vertices (cf. Section 5.2). The occurrence of degenerate elements with angles 0 or π can be avoided by moving vertices within the tangent plane and was discussed in the previous paragraph. We saw that we only need to ensure convergence in the semi norm $|\cdot|_m$. This will hold if the difference of the exact bilinear form and its numerical approximation is of order h [100]. Thus, given polynomials of order $t-1$, we need a quadrature scheme which is accurate to order $t-m$.

Chapter 5

Subdivision Surfaces and FEM

This chapter serves as an introduction into subdivision surfaces, first we will discuss a one dimensional analog and motivate the use of subdivision surfaces. Then we will discuss loop subdivision surfaces and the connection between the vertices of a triangular mesh and the basis functions associated with a subdivision surface. We will use these functions to construct basis functions for a finite element method.

5.1 The Basics

The underlying principle of all subdivision schemes is an algorithm for repeatedly subdividing a piecewise linear curve (surface) which will converge to a smooth[1] limit surface. The following figures illustrate this notion. A smooth

Figure 5.1: Subdividing a Bezier curve [42]

limit surface alone does not make subdivision schemes a worthy field of study, but it has been shown that these schemes have properties which make them valuable tools in Computer Aided Geometric Design (CAGD). The most noteworthy property is the possibility to define subdivision schemes on meshes of almost arbitrary topology with the only restriction, that the mesh gives rise to a surface which is a manifold. Other methods to define smooth surfaces in CAGD suffer from the additional restriction to regular meshes, in which every vertex is restricted to have a certain numbers of neighbors. In the common cases this leads to the restriction to the Euler characteristic $\chi = 0$ for closed orientable

[1] The limit curve (surface) is generally not C^∞ but sufficiently smooth for the application in question.

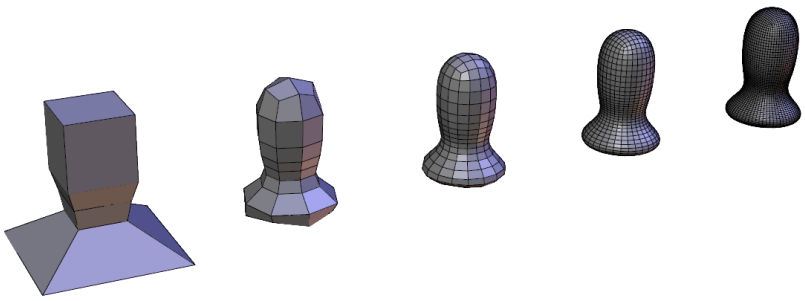

Figure 5.2: Subdividing a surface using the scheme of Catmull and Clark[23]

surfaces. The ability to define a subdivision scheme on arbitrary meshes furthermore allows for localized changes in the connectivity of the mesh in order to add detail to a surface. An other important property of common subdivision schemes is that they allow for a local definition of the rules for the generation of refined surfaces. This results in efficient calculation of surface attributes. Furthermore, the rules are invariant under affine transformations. In recent years much attention was paid to the wavelet properties of subdivision surfaces. A wavelet is an L_2 function which can be decomposed into translated and dilated copies of itself [107]. This allows for an efficient decomposition of a subdivision surface in terms of surfaces with increasingly fine detail and thus spurred the development of a host of multiresolution algorithms for subdivision surfaces [37, 114, 13].

5.1.1 Splines

As we have seen in Chapter 4, where we introduced the FEM, the smoothness properties at the element boundaries are not only crucial for the convergence of the solution but also for an acceptably small discretization error. Maintaining the required C^2 continuity on the element boundaries poses a difficult problem in schemes in which polynomial functions of the elements are defined as sums of monomial basis functions. In the following we will discuss this difficulty for the case of curves and construct a basis which enforces arbitrary smoothness at element boundaries.

For now we will leave the notion of subdivision schemes and focus on curves approximating or interpolating a set of control-points. Consider a curve given by $C(t) = \sum_i P_i B(t-i)$ with control–points $P_i \in \mathbb{R}^n$, $i = 0 \dots m > 0$ and a choice of the function $B : \mathbb{R} \to \mathbb{R}$ in such a way that the curve has arbitrary but fixed smoothness and control-points have a local influence. Note that we do not specify any tangents or higher derivatives to define the curve. To ensure smoothness the function B has to be continuously differentiable up the required order. The local influence of the control-points requires a compact support of

B. For practical reasons we want the curve to approximate the control-points, so we require $0 \in \text{supp}(B)$ which ensures the control-point closest to a segment determines the shape of that segment. This is of course just a matter of notation. Changing the parametrization of the curve can be undone by changing the index set of the control-points, which we will use in the next paragraph. A last property we require is that the $B(t-i)$ form a partition of unity, to insure affine invariance of the curve. Consider a translation b of the curve $C(t) \to C(t) + b$. Thus we have

$$C(t) + b = (\sum_i P_i B(t-i)) + b.$$

But with $\sum_i B(t-i) = 1$ we have

$$C(t) + b = (\sum_i P_i B(t-i)) + b \sum_i B(t-i)$$

and thus

$$C(t) + b = \sum_i (P_i + b) B(t-i).$$

Given these requirements we will now construct the spline basis functions by repeated convolution and use some properties of convolutions to reintroduce to notion of subdivision schemes. Let us start with a piecewise constant curve, which is obviously given by a piecewise constant basis function

$$B_0(t) = \begin{cases} 1 & \text{if } 0 \leq t < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Convoluting B_0 with itself defines a new basis function

$$B_1(t) = (B_0 \otimes B_0)(t) = \int B_0(s) B_0(t-s) ds = \begin{cases} t & \text{if } 0 \leq t < 1 \\ 2-t & \text{if } 1 \leq t < 2 \\ 0 & \text{otherwise.} \end{cases}$$

This is (up to a reparametrization of the curve) equivalent to the piecewise linear basis function we used at the beginning of this section. We now define splines of degree l by

$$B_l(t) = (B_0 \otimes B_{l-1})(t) \tag{5.1}$$

The basis function $B_l(t)$ is C^{l-1} continuous [114].

5.1.2 Refinement

An important property of the spline basis functions is that they obey a refinement equation i.e. a function $B_l(t)$ can be written as a sum of translated and dilated copies of itself. This draws the connection to subdivision schemes. Figure 5.3 illustrates this for the case $B_1(t)$.

To derive a general refinement equation for $B_l(t)$ we note that we can write

$$B_0(t) = B_0(2t) + B_0(2t-1) \tag{5.2}$$

recalling the definition of B_l (5.1) we substitute (5.2)

$$B_l(t) = (B_0(2t) + B_0(2t-1)) \otimes B_{l-1}(t) = \bigotimes_{i=0}^{l} (B_0(2t) + B_0(2t-1)).$$

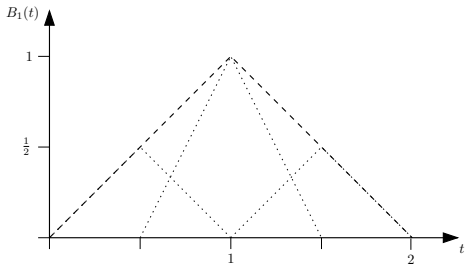

Figure 5.3: The refinement equation for $B_1(t)$, which is illustrated by the dashed line. Three copies of $B_1(t)$, as shown by the dotted lines, add up to $B_1(t) = 1/2B_1(2t) + B_1(2t-1) + 1/2B_1(2t-2)$.

Making use of the linearity, shift and scaling properties of convolutions we arrive at the refinement equation

$$B_l(t) = \frac{1}{2^l} \sum_{i=0}^{l+1} \binom{l+1}{i} B_l(2t-i). \tag{5.3}$$

Since we require exactly one basis function for every control point, the refinement equation gives rise to new control-points on the level of spline curves. Recall that a spline curve of degree l is defined as

$$C(t) = \sum_i P_i B_l^i(t) \quad \text{with } B_l^i(t) = B_l(t-i)$$

Now consider P_i and $B_l^i(t)$ as components of column or row vectors $\mathbf{P}$ and $\mathbf{B}_l(t)$, respectively. The curve $C(t)$ is now denoted as $\mathbf{B}_l(t)\mathbf{P}$. Applying the refinement equation (5.3) to

$$\mathbf{B}_l(t) = [\ldots \quad B_l(t+1) \quad B_l(t) \quad B_l(t-1) \quad \ldots]$$

yields a vector

$$\mathbf{B}_l(2t) = [\ldots \quad B_l(2t+2) \quad B_l(2t+1) \quad B_l(2t) \quad B_l(2t-1) \quad B_l(2t-2) \quad \ldots].$$

We can interpret the application of the refinement equation as the multiplication with a subdivision matrix S whose columns are given by the weights of the refinement equation

$$\mathbf{B}(t) = \mathbf{B}(2t)S \qquad \text{with } S_{2i+k,i} = \frac{1}{2^l}\binom{l+1}{k}.$$

Thus we are able to rewrite the curve $C(t)$ as

$$C(t) = \mathbf{B}(2t)S\mathbf{P}, \tag{5.4}$$

which is the same curve as before, but we replaced the basis functions with a new basis of dilated, twice as dense functions. Instead of considering the new basis as a decomposition of the old one, we can turn this argument around, and just consider the new basis, forgetting how we found it. Now looking at the

action of the subdivision matrix on the control-points we realize that we end up with twice the number of control-points. This is our subdivision scheme. All what remains to be done now is to actually show that a linear interpolation of the control–points generated by repeated application of the subdivision matrix converges to the curve C, which can be found in [114].

5.2 Loop Subdivision Surfaces

Let's turn back to subdivision surfaces. We will focus on the scheme developed by Loop [66]. This scheme subdivides triangles by quadrisection (see Fig. 5.4). Each subdivision step assigns new positions to existing vertices by calculating a weighted mean of it's position and the average position of it's neighbors. Edges are subdivided by generating new vertices based on the positions of the vertices of the two triangles sharing the edge. The weights of the vertices entering the calculation of the new vertex positions are given in Figure 5.4. The choice of the weights $\beta(N)$ for the new positions of existing vertices with N neighbors is constrained by the requirements of convergence of the subdivision surface to a limit surface with well defined normals, see [66] for details. Loop's choice of

$$\beta(N) = \frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{N}\right)^2$$

not only ensures a limit surface with well defined normals, but well defined curvatures on almost every point of the limit surface [66]. Schröder and Reif [91] proved the principal curvatures to be in L_p where $p > 2$ depends on the order (number of neighbors) of the vertex. For increasing order of a vertex the limit surface shows an increasing number of radial ripples and an increasingly sharp crease over the triangles incident on the vertex [114]. We want to avoid these artifacts and limit the order of vertices to five and six.

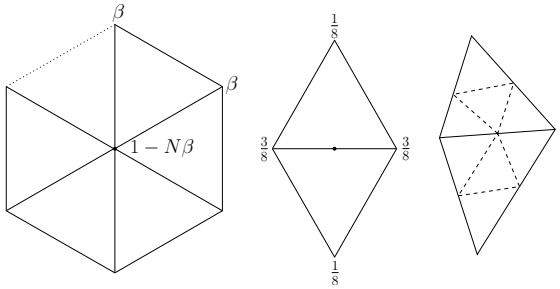

Figure 5.4: Subdivision stencil. Left & middle: weights for the calculation of new vertex positions. Right: illustration of the quadrisection of two triangles

The splitting of edges ensures the new vertices will have six neighbors: two vertices given by the edge which was subdivided, and two in each of the triangles being subdivided, stemming from connecting the new vertices (cf. Figure 5.4(Right)). Only vertices that are irregular will give rise to irregular vertices in the subdivided mesh.

Definition 4. *A vertex is regular, iff it has six neighbors, otherwise it is irregular. A triangle is regular, iff it has only regular vertices,otherwise it is irregular.*

Let us stress the importance of Loop subdivision only adding regular vertices. For parts of meshes consisting only of regular triangles, a polynomial description of the limit surface can be given. Which allows for the efficient evaluation of the limit surfaces.

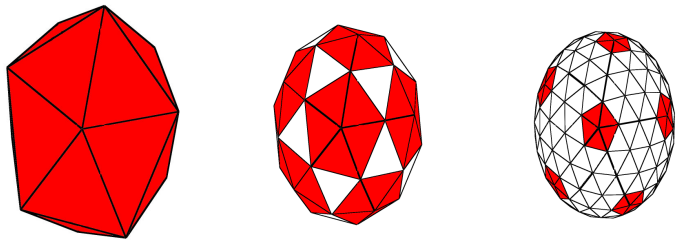

Figure 5.5: Repeated subdivision of an icosahedron. The irregular triangles are shown in red.

On regular triangles polynomial expressions for the basis functions are known as the three directional box spline [42, 114, 66], but for triangles with at least one irregular vertex there is no closed expression for all points of the triangle. We evaluate irregular triangles by repeated subdivision. The subdivision rules generate no new irregular triangles, after one subdivision any triangle has either one or zero irregular vertices. A subdivision will always split a triangle with k irregular vertices into k triangles with one irregular vertex and $4-k$ triangles with regular vertices (cf. Figure 5.5). Therefore we do not have to treat triangles with more than one irregular vertex in a special manner.Hence we will follow [96], and only discus the case of one irregular vertex per triangle.

Subdividing an irregular triangle will give rise to four new triangles, three of which will be regular (cf. Figure 5.6). Thus each subdivision step will allow to evaluate three quarters of the domain for which no closed expression can be found. The part of the domain which cannot be evaluated is the new triangle containing the irregular vertex. Applying the subdivision rules repeatedly will thus allow to evaluate points arbitrarily close to the irregular vertex.

5.2.1 Evaluation

Loop subdivision surfaces in the neighborhood of regular vertices are equivalent to [42, 114, 66] tridirectional box splines. The box spline over a triangle is determined by the twelve vertices in the one–ring around it.

Definition 5. *An one-ring around a triangle t[2] is the union of t and all triangles sharing at least one vertex with t. A n-ring around t is the union the one-ring of all triangles in the (n-1)-ring around t (cf. Figure 5.9 and 5.10).*

A one–neighborhood of a vertex v is the set of all triangles that contain the vertex v. An n-neighborhood of a vertex v is the union of all one–neighborhoods of all vertices in the $(n-1)$–neighborhood of the vertex v.

[2] From now on t can also refer to a triangle, the meaning should be obvious from the context in which the symbol occurs.

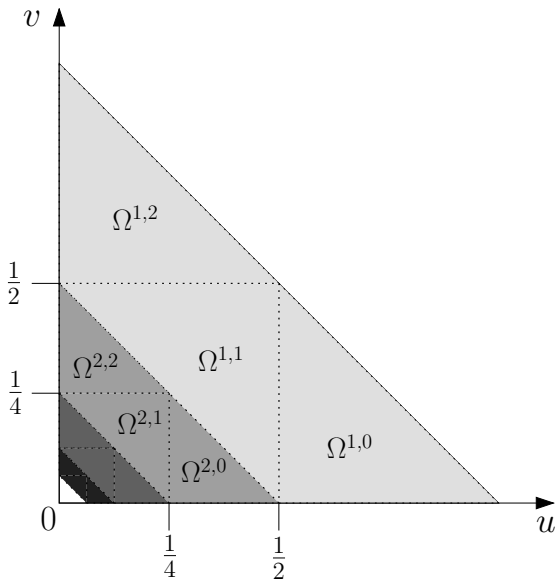

Figure 5.6: Repeated subdivision of an irregular triangle, the irregular vertex is at the origin of the parameterization.

The basis functions on a triangle are given by Stam [96] in terms of barycentric coordinates, since the FEM solver will make use of the canonical coordinates v, w in $\mathbb{R}^2$ restricted to $(v > 0) \wedge (w > 0) \wedge (v + w \leq 1)$ we rewrite them in canonical coordinates (see Appendix A)

We will now formalize the repeated subdivision procedure as described by Stam [96]. Given a vector P of vertices in the one–ring of an irregular triangle, we can generate a new vector P' as $P' = SP$ with S being a $(N+6) \times (N+6)$ subdivision matrix of the loop scheme, where N is the order of the irregular vertex. The subdivision matrix is chosen such that the refinement is centered around the irregular vertex. Thus repeated multiplication of the resulting vector P' will generate new control-points closer to the irregular vertex. We repeat the multiplication until the point, at which we want to evaluate the basis function, is in a regular triangle. The last subdivision step has to be modified. Since S only generates vertices centered around the irregular vertex, but the evaluation of points in the newly created regular triangles requires six additional control-points which are not generated by S. Thus the final subdivision is carried out by multiplying with an amended $(N+6) \times (N+12)$ matrix S' (see Fig. 5.7). The matrices S and S' are constructed by applying the subdivision stencil. Each row contains one stencil. The columns in which the stencil appears depend on the ordering of vertices in P, whereas the rows of S and S' are sorted such that the local ordering scheme of vertices around the irregular vertex is conserved in the subdivision step.

After multiplication with S', the resulting vector P'' contains $(N+12)$ vertices, only $(N+6)$ of which are needed to evaluate the box spline. The correct subset of vertices is chosen by mapping the indices of the $(N+12)$ vertices to the 12 vertices in the one–ring around a regular triangle (cf. Fig. 5.9). Thus

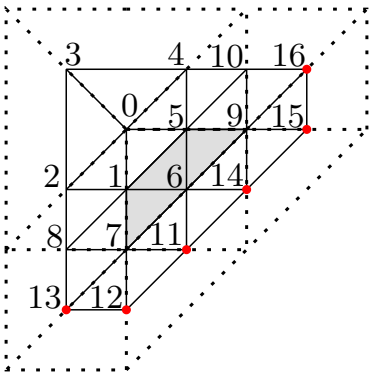

Figure 5.7: Numbering of the vertices generated by applying the subdivision matrix to a triangle with one vertex of order five. The coarser Triangles are indicated by the dotted lines. Vertices marked with a red dot are only generated by the amended subdivision matrix. Triangles shaded grey are the parts of the coarse triangle which can be evaluated (cf. Fig. 5.6).

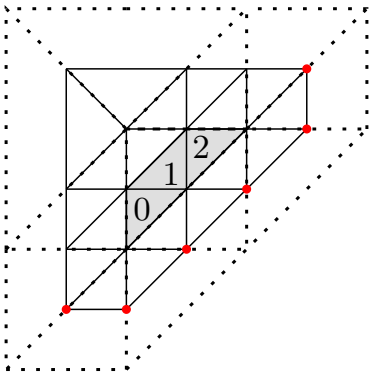

Figure 5.8: Numbering of the regular sub triangles, assuming the numbering of the vertices as in Fig. 5.7

the vertices needed to evaluate a point (v, w) over the regular sub-triangle l (cf. Fig. 5.8) with $(1/2)^n < v + w < (1/2)^{n-1}$ are given by

$$P_{eval}^{n,l} = C_l S' S^n P.$$

The matrix C_l, picking the proper vertices is given by

$$C_{lij} = \begin{cases} 1 & \text{if index } j \in P'' \text{ is mapped to } i \in P_{eval} \\ 0 & \text{otherwise} \end{cases}$$

In contrast to Stam [96] we choose to store pre-calculated matrices $C_l S' S^n$ instead of diagonalizing S in order to speed up numerical calculations of the powers of S, which saves 12 floating point multiplications at each evaluation of a point. Since the points at which triangles are evaluated and the order of the

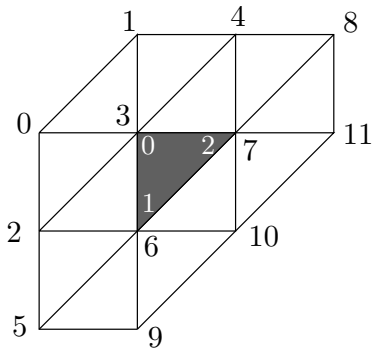

Figure 5.9: Numbering of the vertices (black) and association with basis functions in the one ring around a regular triangle (shaded grey). The white numbers indicate the enumeration of the triangles vertices.

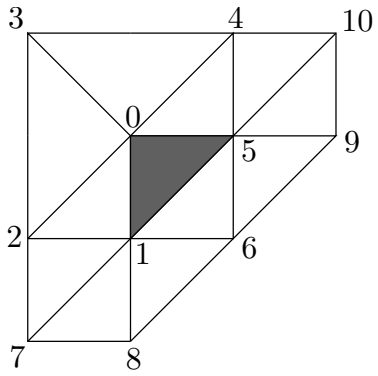

Figure 5.10: Numbering of the vertices and association with basis functions in the one–ring around an irregular triangle with one vertex of order five. Note that the numbering scheme differs significantly from the scheme for regular triangles. This scheme was chosen for consistency with the existing literature.

irregular vertex are a priori known, the number of matrices which need to be stored is small.

The entries in $P_{eval}^{n,l}$ are the weights of the basis–functions $\vec{b}(v,w)$ of the box spline, thus a surface patch $s(v,w)$ can be written as

$$s(v,w) = {P_{eval}^{n,l}}^T \vec{b}(v',w')$$

As discussed earlier, each subdivision step amounts to a reparametrization of the surface patch. In order to evaluate irregular patches in the same coordinates as regular patches, we have to define the correct transformation $t^{n,l}$ from the unit patch Ω to the subdivided patch $\Omega^{n,l}$

$$\begin{aligned} t^{n,0}(v,w) &= (2^n v' - 1, 2^n w') \\ t^{n,1}(v,w) &= (1 - 2^n v', 1 - 2^n w') \\ t^{n,2}(v,w) &= (2^n v', 2^n w' - 1) \end{aligned}$$

The surface patch is now given as

$$s(v,w) = {P^{n,l}_{eval}}^T \vec{b}(t^{n,l}(v,w)).$$

The first and second derivatives are

$$s(v,w)_{,\cdot} = (-1)^l 2^n {P^{n,l}_{eval}}^T \vec{b}_{,\cdot}\,(t^{n,l}(v,w))$$
$$s(v,w)_{,\cdot\cdot} = 4^n {P^{n,l}_{eval}}^T \vec{b}_{,\cdot\cdot}\,(t^{n,l}(v,w)).$$

5.2.2 Nodal Basis Functions

Analogous to the one dimensional example, elements share degrees of freedom (DOF). In the case of regular subdivision surfaces each DOF is shared not only by the adjacent 6 elements, but by 24 elements in a two–neighborhood around the DOF. To construct the nodal basis functions we sum basis functions over all elements[3] which have the vertex associated with the DOF in their one-ring (cf. Figure 5.9 and 5.10). Each term in this sum is the basis function – over the element we are looking at – which is associated with the DOF at the vertex. For subdivision surfaces with irregular vertices the procedure for this change of reference is essentially the same. Only now we don't have an analytic expression for the basis functions over irregular elements.

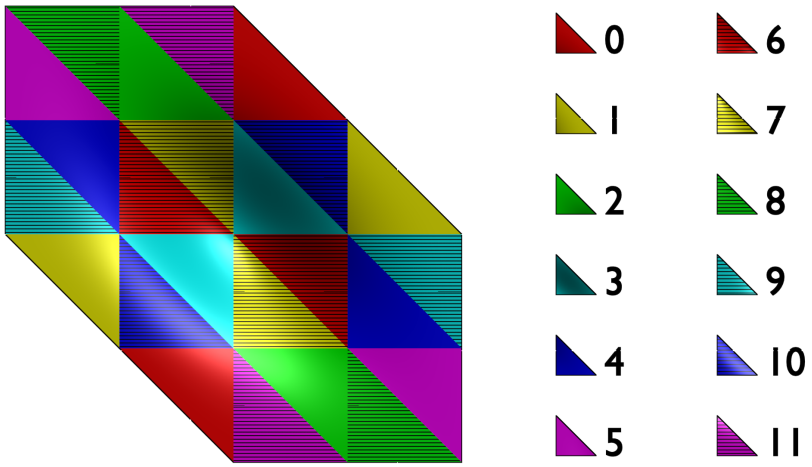

Figure 5.11: A nodal basis function on a regular mesh.
This is the tri–directional box spline assembled from basis functions over individual elements. The numbers of the triangles are the numbers of the central vertex with respect to the local numbering of vertices in the neighborhood of each triangle, cf. Figure 5.9.

[3] An element is the domain, on which we define our basis functions. Here an element is a triangle.

5.3 Assembling the FEM-System using Subdivision Surfaces

This section will discuss how to apply the FEM discretization using subdivision surfaces to the equation of motion of the lipid bilayer (2.13). Analogous to the example in section 4.1, carrying out all spatial integrations will lead to a system of ordinary differential equations

$$Mu,t = f(u). \tag{5.5}$$

Note that every vertex is associated with three degrees of freedom, one for each spatial coordinate.

In order to carry out the integrations to assemble the FEM-system using subdivision surfaces, we need a map M from elements e and global indices i of the nodal basis functions to the local index l of the vertices. Given the local vertex indices, we know which basis functions of an element contribute to the integrand:

$$M(e,i) \mapsto l \in \{1,\ldots,n\},$$

where $n = 11$ if e is irregular, otherwise $n = 12$. For a regular mesh the construction of Mcan be illustrated with the help of Fig. 5.11. The vertex in the center has the global index i, finding the element e we choose the local index l such that we reproduce the box spline around e.

There is no analytic expression for the basis functions on irregular triangles, thus we need to evaluate the integrals arising from the FEM numerically. The numeric integration is called quadrature – sometimes cubature in the multivariate case – in the FEM literature. We will use a Gaussian quadrature rule to integrate a function $f(\alpha,\beta,\gamma)$ over a triangle t with barycentric coordinates and area A

$$\int_A f(\alpha,\beta,\gamma)dA = A\sum_{i=1}^{n} w_i f(\alpha_i,\beta_i,\gamma_i) \tag{5.6}$$

The weights w_i and quadrature points $(\alpha_i,\beta_i,\gamma_i)$ are given by [36]. Assuming a complete polynomial function f of degree p and

$$n = \frac{(p+1)(p+2)}{6}$$

quadrature points, the numerical integration is exact as indicated in (5.6)[4]. Since we are using isoparametic elements, we always map a reference triangle to the actual element. In our case the reference triangle is given by $t = \{(u,w) \in \mathbb{R} \| (u>0) \wedge (v>0) \wedge (u+v \leq 1)\}$ thus we can express f and the quadrature points in canonical coordinates (cf. B).

Integrating the l.h.s. of the equation of motion (2.13) $\vec{r}_{,t}$ formally yields the same result as in the one dimensional example. $Mu_{,t}$ with 3×3 block matrices:

$$M_{ij} = \mathbb{1}_3 \int_\Omega \phi_i\phi_j.$$

[4]Of course, there will be round off errors on real world computer architectures such as implementations of IEEE 754-1985 [1]

In order to evaluate this integral for the nodal basis function ϕ_i we only need to integrate over elements e in the intersection of the two–rings $\circledast_i$ around the vertices i and j. Thus

$$M_{ij} = \sum_{e \in (\circledast_i \cap \circledast_j)} \int_e \psi_{M(e,i)} \psi_{M(e,j)}.$$

The integration is carried out by numerically integrating over the reference triangle in the canonical coordinates discussed in equation (5.6)

$$M_{ij} = \frac{1}{2} \sum_{e \in (\circledast_i \cap \circledast_j)} \sum_{k=1}^{n} w_k \psi_{M(e,i)}(\alpha_k, \beta_k) \psi_{M(e,j)}(\alpha_k, \beta_k) dA(\alpha_k, \beta_k). \tag{5.7}$$

We can evaluate the r.h.s. of the equation of motion (2.13) in a similar manner. We integrate the test function ψ over each two–ring $\circledast_i$ around vertex i to evaluate the components $f(u)_i$. Note that the Laplacian of ψ is expressed in terms of partial derivatives of nodal basis functions. Each component $f(u)_i$ is a 3 vector of the three components of the force at the vertex i.

$$f(u)_i = \frac{1}{2} \vec{n}_i \sum_{e \in \circledast_i} \sum_{k=1}^{n} w_k \left(\psi_{M(e,i)} \mathcal{A} + \kappa H(\alpha_k, \beta_k) \mathcal{B} \right) (\alpha_k, \beta_k) dA(\alpha_k, \beta_k) \tag{5.8}$$

with

$$\begin{aligned} \mathcal{A} = {} & \kappa (2(H(\alpha_k, \beta_k) - C_0)(H(\alpha_k, \beta_k)^2 - K(\alpha_k, \beta_k) - C_0) \\ & - \frac{\kappa' \pi}{2Ad} \left(\Delta A - \Delta A_0 \right) K(\alpha_k, \beta_k) \\ & - 4a \left(A - A_0 \right) H(\alpha_k, \beta_k) \\ & + 2v \left(V - V_0 \right) \end{aligned}$$

and

$$\begin{aligned} \mathcal{B} = {} & g^{lm}_{,l}(\alpha_k, \beta_k)(\det g(\alpha_k, \beta_k))^{\frac{1}{2}} \psi_{M(e,i),m} \\ & + g^{lm}(\alpha_k, \beta_k)(\det g(\alpha_k, \beta_k))^{\frac{1}{2}}_{,l} \psi_{M(e,i),m} \\ & + g^{lm}(\alpha_k, \beta_k) \psi_{M(e,i),lm}(\alpha_k, \beta_k)(\det g(\alpha_k, \beta_k))^{\frac{1}{2}} \end{aligned}$$

The summation over the two neighborhood of each vertex is done by summing over all elements which have the vertex in their one neighborhood. This leads to the same result since the union of all elements which have a vertex in their one neighborhood, is exactly the two neighborhood of the vertex. The reason for this change of reference is that we need to keep track of ordered one neighborhoods of elements in order to evaluate the limit surface over the element, which is a sum over the basis functions weighted with the positions of the associated vertices. The ordering of the vertices encodes the association of vertices and basis functions (cf. Chapter 5.2 and Figures 5.11, 5.9 and 5.10). Figure 5.12 illustrates the integration process.

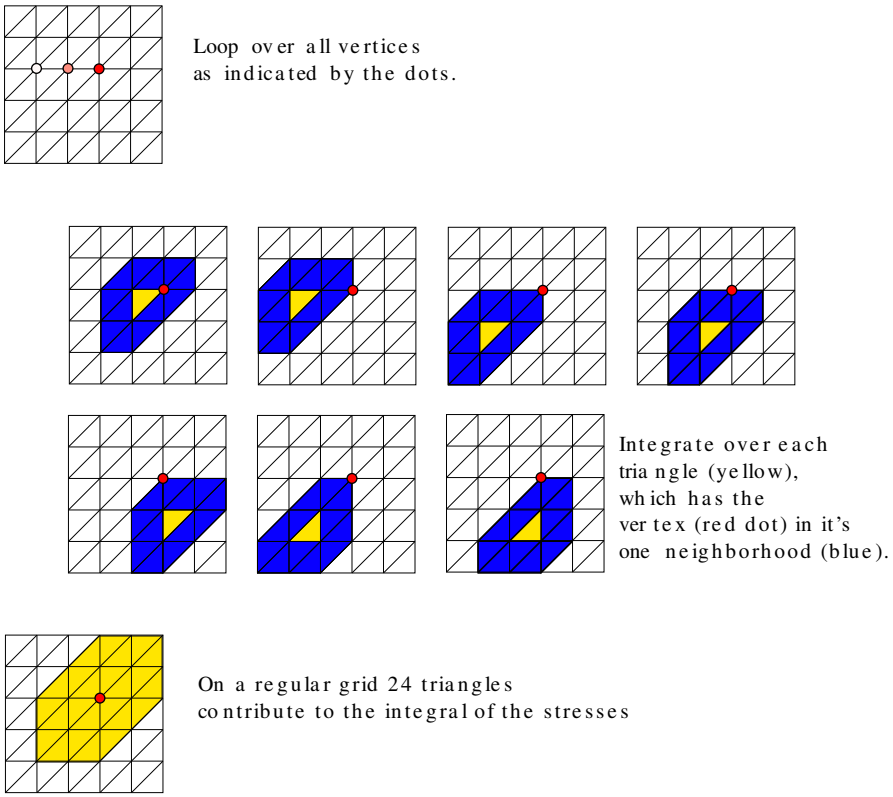

Figure 5.12: Illustration of the integration process.

Chapter 6

Timestepping

In this chapter we will discuss different strategies for the integration of systems of ordinary differential equations. There is not a single algorithm which performs strictly better than all others [49], and not only the differential equations, but also the initial conditions influence the performance to a degree which can render an algorithm useless for the problem at hand.

In most cases the ODEs are not solved directly over the whole domain of interest. Instead an approximate solution is found by a stepwise extrapolation of the known part of the solution starting from the initial condition. Besides the obvious limit for the step size, which is given by the required accuracy of the solution, there is a second – in many cases much smaller – limit stemming from the fact that all but the first values of the dependent variables are only approximate. Discretization errors can grow without bound, the solution method is deemed unstable. Based on [49] we introduce a simple example of this phenomenon. Consider the system of ODEs

$$y'(x) = f(x, y) \tag{6.1}$$

which is solved by $\phi(x)$. We can expand $f(x, y)$ around ϕ and rewrite (6.1) as

$$y'(x) = f(x, \phi(x)) + \left.\frac{\partial f(x, y)}{\partial y}\right|_{\phi(x)} (y(x) - \phi(x)) + \dots$$

now set $p(x) = y(x) - \phi(x)$. Recalling $\phi(x)$ is a solution of (6.1) we can express perturbations $p(x)$ up to first order as

$$\begin{aligned} p'(x) &= \left.\frac{\partial f(x, y)}{\partial y}\right|_{\phi(x)} p(x) \\ &= J(x)p(x). \end{aligned}$$

Now we apply the explicit Euler's method and approximate the derivative by finite differences. Furthermore we assume $f(x, y)$ to be linear, i.e. $J(x)$ to be constant

$$\frac{p_{m+1} - p_m}{h} = Jp_m. \tag{6.2}$$

Thus the perturbation at timestep $m + 1$ can be expressed as

$$p_{m+1} = R(hJ)p_m$$

with the stability function

$$R(z) = 1 + z.$$

in order to study the behavior of a perturbation for $m \to \infty$ we need to evaluate powers of $R(z)$. For $z \in \mathbb{C}$ it is obvious that $R(z)^m$ remains bounded for $|R(z)| \leq 1$ or if z lies in a circle with unit radius around $(-1, 0i)$. To apply this to the matrix valued argument J, we assume J to be diagonizable with eigenvalues λ_i and eigenvectors v_i. Expressing p_0 in terms of the eigenvectors as

$$p_0 = \sum_i a_i v_i$$

we arrive at

$$p_{m+1} = \sum_i R(h\lambda_i)^m a_i v_i.$$

Now we can see the connection between the eigenvalues of the Jacobian of the ODE and the step size h necessary for a bounded error (stability)

$$|h\lambda_i + 1| \leq 1$$

Let's turn back to the discretization of the derivative. Replacing the rhs. of (6.2) with Jp_{m+1}, we employ the implicit Euler method. Now we have to solve a linear equation to find the next step

$$p_m = (\mathbb{1} - hJ)p_{m+1}. \tag{6.3}$$

Given the formal solution $p_{m+1} = (\mathbb{1} - hJ)^{-1}p_m$ we can follow the previous analysis. The stability criterion now reads

$$|(1 - h\lambda_i)^{-1}| \leq 1$$

Thus the stability domain is the complex plane except a unit circle around $(1, 0i)$. Therefore the problem is stable for a much larger range of products of eigenvalues and step sizes. This improvement of the stability comes at the price of solving the linear equation (6.3).

Here we see that the step size needed to obtain a stable method depend on the eigenvalues of the Jacobian. If the dynamics of a linear system span multiple scales, the ratio of the largest and the smallest eigenvalue is large. High frequency components, i.e the modes of the largest eigenvalues, will often be dampened quickly and thus will not affect the accuracy of the solution, even if the step size is too small to sample highest frequency modes. But if an explicit method such as the explicit Euler method is used, those modes still determine the step size due to the stability requirement. Such systems are called stiff.

For nonlinear systems the notion of stiffness can not be reduced to the ratio of eigenvalues of a Jacobian. Several definitions of stiffness have been developed [3, 28, 49]. The general notion is for stiff problems implicit solvers tend to have better stability properties. Or in other words, the step size of an explicit solver would be limited by stability and not by accuracy.

Our case lends itself to the use of implicit schemes. First of all the FEM discretization of the PDE results in an implicit ODE with a solution dependent matrix on the lhs. Thus an explicit scheme would have to perform exceedingly better than an implicit method in order to justify the numerical cost and difficulties involved in the repeated inversion of the matrix on the lhs. Furthermore

we anticipate a stiff system because of the penalty terms enforcing the global constrains.

Consider a membrane which is a slightly perturbed sphere and set the spontaneous curvature C_0 close to the inverse of the sphere's radius. The stresses stemming from the deformation of the membrane out of it's resting shape are vanishingly small. In contrast, any deformation will change the area and volume, thus the stresses due to the constraints can be large, even though only slight deformations will occur. In this situation a given accuracy would allow for a large step size because the membrane does not significantly alter its shape, but the large stresses will introduce faster timescales and the step size will be governed by stability requirements.

Another classification of solution methods of ODEs is the distinction between single– and multi–step methods. Single–step methods only use one step to calculate a new point of the solution, whereas multi–step methods use multiple known steps to advance the solution. In many cases multi–step methods like BDF [49] perform better than single–step methods, especially for stringent tolerances [68]. On the other hand, the multi–step character can be a disadvantage if the system of ODEs stems from a discretization of some PDE. In this case adapting the discretization to the changing solution can increase the accuracy or decrease the number of DOF. But previous steps of the solution become unusable to the multi–step algorithm, which enforces smaller time steps and abates benefits from an adaptive discretization [60].

In our case the term $f(x, y)$ is nonlinear, thus in principle each new step is the solution of a system of nonlinear equations. Numerical algorithms solving nonlinear equations generally require iterative approaches. Furthermore the convergence of the iterative process can not be guaranteed [80]. Rosenbrock Methods avoid this difficulty by linearizing the nonlinear map around the current step of the solution [49]. Thereby, for each stage of the update scheme, one Newton iteration of the solution of the nonlinear equation is included into the integration scheme of the ODE. Because of these advantages we will use Rosenbrock methods in our simulation.

6.1 Basic Concepts

In the previous section we introduced the notion of stability to motivate the use of Rosenbrock schemes, but we made a number of simplifying assumptions, which are not valid for the problem at hand. Now we will have a more thorough look at the properties of solution methods for differential equations following [48, 49].

We introduced the stability function $R(z)$ to asses the long term behavior of perturbations of the solution of an ODE. We formally define it as follows:

Definition 6. *The stability function $R(z)$ of an integration method for ODEs is the result of one step of the numerical solution of the Dahlquist test equation $y' = \lambda y$ with the initial condition $y_0 = 1$ and $z = h\lambda$. The set*

$$S = z \in \mathbb{C}; |R(z)| \leq 1$$

is the stability domain of the method [49].

Since the stability domain of the exact solution of the Dahlquist test equation coincides with the negative half plane $\mathbb{C}^- = \{s \in \mathbb{C}; \mathrm{Re}(s) \leq 0\}$, methods which have the negative half plane as their stability domain, and thus preserve the stability properties of the exact solution seem to be optimal.

If the stability domain of a method was a subset of the negative half plane, modes that decay in the exact solution would grow exponentially. Acknowledging this as unwanted we define

Definition 7. *A method is A-stable iff the negative half plane is a subset of its stability domain $S \supset \mathbb{C}^-$.*

Yet in certain situations an exact coincidence of the stability domain with $\mathbb{C}^-$ is not desirable. For stiff problems the product of the integration step h and λ has a large absolute value. Thus we have to consider the behavior for $R(z)$ in the limit $z \to \infty$. We have $\lim_{z=iy, y\to\infty} R(z) = \lim_{z\to-\infty} R(z)$ and if $S = \mathbb{C}^-$ holds, $|R(z)| = 1$ for $\mathrm{Re}(z) = 0$. Thus for large $h\lambda$ the stability function $R(z)$ is close to one, high frequency transient perturbations are dampened very slowly. In contrast methods, which dampen transient perturbations quickly can be desirable thus we define

Definition 8. *A method is called L-stable, if it is A-stable and if*

$$\lim_{z\to\infty} R(z) = 0$$

On the other hand, if the stability domain extends into the positive half plane we experience a damping of modes which grow in the exact solution. We end our discussion of the stability observing that there can not be a single method that performs well in all situations.

Leaving the long term behavior of perturbations, we turn our attention to the local behavior of the error. We introduce the concept of the order of a method:

Definition 9. *A method is of order p, if for an approximate solution y_1 at $x_0 + h$*

$$\|y(x_0 + h) - y_1\| < Kh^{p+1}$$

for some $K>0$ holds.

To derive conditions for the order of a method we expand the exact solution and the expression for the next step in a Taylor series around x_0

$$y(x_0 + h) = y(x_0) + hy'(x_0) + h^2 y''(x_0) + \dots \quad .$$

Every occurrence of y' is replaced by f

$$y(x_0 + h) = y(x_0) + hf(y_0) + h^2 f_{,y}(y_0) f(y_0) + \dots \quad .$$

As an example, compare this to the expansion of y_1 assuming an explicit Euler scheme

$$y_1 = y_0 + hf(y_0).$$

Since the explicit Euler scheme uses exactly the first term of a Taylor expansion, we note that the explicit Euler scheme is of first order. Using the same strategy of comparing the terms in the Taylor expansions, we can determine the order of other schemes.

6.2 Rosenbrock Methods

In this section we will lay out the construction and implementation of Rosenbrock Methods following [49]. We want to solve a differential equation $y'(x) = f(y)$. A diagonally implicit Runge-Kutta method [49] with s stages would calculate the next y_1 step from the current step y_0 by

$$\begin{aligned} k_i &= hf(y_0 + \sum_{j=1}^{i-1} a_{ij}k_j + a_{ii}k_i) \\ y_1 &= y_0 + \sum_{i=1}^{s} b_i k_i \end{aligned} \tag{6.4}$$

Thus the next step is a weighted sum over estimates k_i of the derivative of $y(x)$ at points found using the previous estimates. Since each estimate is given implicitly, each of the s stages of the step requires the computationally expensive solution of a nonlinear equation. Linearizing (6.4) around $y_0 + k_{i-1}$ results in

$$\begin{aligned} k_i &= hf(g_i) + hf'(g_i)a_{ii}k_i \\ g_i &= y_0 + \sum_{j=1}^{i-1} a_{ij}k_j. \end{aligned}$$

Now each stage only requires the solution of a linear equation and the evaluation of the Jacobian. Even more computations could be saved if the Jacobian would only be evaluated once. Also, if the matrix defining the linear equation would be the same for all stages only one LU-decomposition would be needed per step. Pushing even further in this direction leads us to W-Methods, in which approximations to the Jacobian are used. In particular reusing the Jacobian from previous steps saves further LU-decompositions [49]:

Definition 10. *Given a system of ordinary differential equations $y'(x) = f(y)$, an s-stage Rosenbrock-Method is given by:*

$$k_i = hf\left(y_0 + \sum_{j=1}^{i-1} a_{ij}k_j\right) + hJ\sum_{j=1}^{i} \gamma_{ij}k_j \tag{6.5}$$

$$y_1 = y_0 + \sum_{j=1}^{s} b_j k_j \tag{6.6}$$

with $y_0 = y(x_0)$, $y_1 \approx y(x_0 + h)$, $\gamma_{11} = \cdots = \gamma_{ii} = \gamma \neq 0$ and $J = f'(y_0)$. The coefficients a_{ij},γ,γ_{ij} and b_i determine the method.

The stability function of a Rosenbrock method is

$$R(z) = 1 + zb^T(\mathbb{1} - z\mathcal{B})^{-1}(1, \ldots, 1)^T, \tag{6.7}$$

with $b^T = (b_1, \ldots, b_s)^T$ and $\mathcal{B}_{ij} = \alpha_{ij} + \gamma_{ij}$. For a suitable choice of coefficients a Rosenbrock method can be A or L–stable.

Coefficients for several different methods have been published [61, 85, 49, 68, 88, 86, 32] and benchmarked for certain problems [90, 85, 49].

This definition is still insufficient for the problem at hand. We seek a method to solve

$$M(y)y'(x) = f(y). \tag{6.8}$$

If $M(y)$ was constant we could formally multiply with M^{-1} and arrive at a method which does not require matrix inversions [49]. Instead we introduce a new variable $z = y'$ and rewrite (6.8)

$$y' = z \tag{6.9}$$
$$0 = f(y) - M(y)z, \tag{6.10}$$

in this new system the matrix $\tilde{M} = \mathrm{diag}(\mathbb{1}, \mathbb{0})$ is constant, but singular [68]. Such systems are known as differential algebraic systems since eqn. (6.10) is formally an algebraic equation.

6.2.1 Differential Algebraic Equations

Differential algebraic equations (DAE) are differential equations with an algebraic constraint. They can also be considered as stiff differential equations in the limit of infinite stiffness [49]. Consider the system of equations

$$y'(x) = f(y, z) \tag{6.11}$$
$$\epsilon z'(x) = g(y, z) \tag{6.12}$$

in the limit $\epsilon \to 0$ eq. 6.12 turns into an algebraic equation. Let us now apply a Rosenbrock Method to (6.12):

$$\begin{pmatrix} k_i \\ \epsilon l_i \end{pmatrix} = h \begin{pmatrix} f(v_i, w_i) \\ g(v_i, w_i) \end{pmatrix} + hJ \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}. \tag{6.13}$$

With

$$\begin{pmatrix} v_i \\ w_i \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + \sum_{j=1}^{i-1} a_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}, \qquad J = \begin{pmatrix} f_{,y} & f_{,z} \\ g_{,y} & g_{,z} \end{pmatrix}\Big|_{y_0, z_0} \tag{6.14}$$

we have

$$\begin{pmatrix} y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + \sum_{j=1}^{s} b_j \begin{pmatrix} k_j \\ l_j \end{pmatrix}. \tag{6.15}$$

In the limit $\epsilon \to \infty$ we have to solve a linear equation with matrix

$$\begin{pmatrix} \mathbb{1} & 0 \\ 0 & 0 \end{pmatrix} - h\gamma \begin{pmatrix} f_{,y} & f_{,z} \\ g_{,y} & g_{,z} \end{pmatrix}.$$

Assuming an invertible $g_{,z}$ the inverse exists for small h and the Rosenbrock method can be applied. The requirement of an invertible $g_{,z}$ can also be interpreted as a regularity requirement on the function $0 = g(y, z)$, which by virtue of the implicit function theorem leads to the differential equation $y' = f(y, G(y))$, with a local solution of the algebraic equation $z = G(y)$. The importance of

the regularity of $g(y,z)$ gives rise to the definition of the index of a DAE [49]. Several definitions of the index can be found in the literature [62, 44, 49], we will define the index of a DAE $F(y(t), y'(t)) = 0$ as the number of differentiations with respect to t, needed to yield an ODE. By construction the index of (6.10) is 1 [68]. Most algorithms fail on problems with higher index.

A Rosenbrock method defines the next solution step as a function of the previous step and the step size h . We can calculate a Taylor expansion in h of the Method around the previous stepx_0 to derive conditions for the desired order. These conditions will be algebraic equations in the coefficients of the method. This will not be demonstrated here instead we refer to [48, 49, 88, 61, 85]. Since we are operating in the limit of infinite stiffness, the question of the behavior of the stability function $R(z)$ in the limit $z \to \infty$ becomes even more important. We have to require $R(\infty) < 1$ to retain a convergent scheme [49]. This requirement introduces additional constraints on the coefficients of the method.

6.3 Implementation

In the previous section we learned how to integrate an implicit system of ODEs with solution dependent matrix multiplying the r.h.s.

$$M(y)y' = f(y).$$

We will now transform this system into a form suitable for an efficient implementation following ideas from [68, 49]. Applying a Rosenbrock scheme for DAEs (6.13) to the transformed system of ODEs with constant M (6.10) we get

$$\begin{pmatrix} k_i \\ 0 \end{pmatrix} = h \begin{pmatrix} w_i \\ f(v_i) - M(v_i)w_i \end{pmatrix} + hJ \sum_{j=1}^{i} \gamma_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix} \tag{6.16}$$

with

$$\begin{pmatrix} v_i \\ w_i \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + \sum_{j=1}^{i-1} a_{ij} \begin{pmatrix} k_j \\ l_j \end{pmatrix}, \tag{6.17}$$

$$J = \begin{pmatrix} 0 & 1 \\ (f(y) - M(y)z)_{,y} & -M(y) \end{pmatrix}\Big|_{y_0, z_0}. \tag{6.18}$$

We still have

$$\begin{pmatrix} y_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} y_0 \\ z_0 \end{pmatrix} + \sum_{j=1}^{s} b_j \begin{pmatrix} k_j \\ l_j \end{pmatrix} \tag{6.19}$$

If we carry out the matrix multiplication in (6.16), we can eliminate l_i,

$$0 = f(v_i) - M(v_i)w_i + (f(y) - M(y)z)_{,y}\,|_{y_0,z_0} \sum_{j=1}^{i} \gamma_{ij} k_j - \frac{1}{h} M(y_0)(k_i - h w_i). \tag{6.20}$$

Let us now transform (6.20). The goal of this transformation is twofold. First we want to collect the terms k_i to be able to use available solvers which generally expect linear equations of the form $Ax = b$. Secondl we avoid the vector matrix multiplications Jk_j. Recall that we assumed $\gamma_{ii} = \gamma \neq 0$ to avoid LU-decompositions and evaluations of the Jacobian. Since the k_i do not depend on future stages, the matrix $\Gamma = (\gamma_{ij})$ is of lower triangular form, and is therefore invertible. We set

$$u_i = \sum_{j=1}^{i} \gamma_{ij} k_j \qquad \text{with i=1...s}$$

and

$$k_i = \frac{u_i}{\gamma} - \sum_{j=1}^{i-1} c_{ij} u_j \qquad \text{with } C = (c_{ij}) = diag(\gamma^{-1}) - \Gamma^{-1}$$

and arrive at

$$\left(\frac{M(y_0)}{h\gamma} - (f(y) - M(y)z)_{,y} \,|_{y_0,z_0} \right) u_i \tag{6.21}$$

$$= f(v_i) - M(v_i)w_i - M(y_0) \left(w_i - \sum_{j=1}^{i-1} c_{ij} u_j \right). \tag{6.22}$$

This form of the Rosenbrock method is suitable for an implementation on computers.

So far we only mentioned the dependence of the accuracy and stability of the solution on the step size h. What is lacking is a way to choose h as large as possible and as small as needed to fulfill given a accuracy requirement [48]. Since the exact solution is not known, we can only estimate the error. Given a method of order p and a second method of order $q > p$, we can compute solutions y_1 and $\hat{y}_1$. An estimate of the local error is given by $|y_1 - \hat{y_1}|$. Now we require the components of the local error estimate to be smaller than a threshold sc_i, which is given in terms of two tolerances. One tolerance is relative to the scale of the variable *Rtol*, the other sets an absolute scale of the allowable error *Atol*

$$sc_i = Atol + Rtol \max(|y_{0i}|, |y_{1_i}|).$$

We define the error measure as

$$err_y = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{y_{1i} - \hat{y}_{1i}}{sc_i} \right)^2}.$$

Since we treat the derivatives $y' = z$ as independent variables, we have to control the errors of z as well. Following [68] we scale the error of z with the step size.

$$err_z = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{hz_{1i} - h\hat{z}_{1i}}{sc_i} \right)^2}.$$

The total error err is $err = err_y + err_z$. For an optimally chosen step size h_o, the error err is equal to one. Given the order p of the method, we know the error scales as $err \sim Ch^{p+1}$ with unknown C, but our definition of err gives us

$1 \sim Ch_o^{p+1}$. Thus we estimate the optimal step size to be $h_o = h(1/err)^{1/(p+1)}$. If the error estimate for a step is larger than one the step is rejected, and the step is attempted again with the new estimate for the optimal step size. We multiply our estimate of the optimal step size by a safety factor 0.9 to avoid costly step rejections in case the estimate is slightly to large. Furthermore we want to limit changes of the step size, enlarging the step size much too fast will lead to costly rejections whereas a too small step size leads to unnecessary computational work. Thus we set

$$h_{new} = \min(5., 0.9\max(0.2, err^{-1/p+1}))$$

How do we calculate two solutions to our system without creating too much overhead? The answer lies in what is called an embedded method. We use a second method, which evaluates the same stages, but we choose the coefficients $\hat{b}_i$ such that the order of the of the new method is smaller than the order of the original method. This way most of the calculations can be reused. Instead of using the lower order result of the next step, we will leave the concept of error estimates and continue integrating with the higher order result and consider the error estiamte only for the choice of the timestep. The accepted notation for the order of a Rosenbrock method is $p(q)$, where p is the order of the method, and q is the order of the embedded error estimator.

Chapter 7

Numerical Experiments

Here we will discuss some numerical experiments using the algorithms developed in this thesis.

7.1 Performance of Rosenbrock schemes

Here we analyze the performance of different Rosenbrock schemes solving equation (5.5), (5.7) and, (5.8). The first scheme, which we will study in more detail, is called ROWDAIND2 [68]. The method is of order 3(2), has four stages, and order conditions that are valid of index two problems.

We choose the initial conditions for all tests, such that the typical scale of the DOF is 1. A sphere is perturbed randomly such that the largest curvature radii are of the order 10. It takes about 1 time unit to relax to a stable shape. Our model membrane consists of 320 elements with 162 vertices, giving rise to 486 DOF. All simulations were done on a single core of an Intel CoreDuo U2500 CPU running at a clock speed of 1.2 GHz.

In Fig 7.1(a) we see that the time needed to integrate the model depends only weakly on the chosen initial conditions, as long as the scale of the problem does not change significantly. Each model was integrated for 0.001 time units, thus these simulations only represent the period in which the model is far from its relaxed state and the geometries differ.

Figure 7.1(b) shows the time it took to integrate a model for 0.005 time units with varying tolerances. The starting time step was chosen such that the error estimate is close to 1. In all simulations we set $Atol = Rtol$. Looking at tolerances of $3 \cdot 10^{-3}$ and smaller, we note that the run time of the simulation increases inversely proportional to the required tolerance. This is not the behavior we would expect from an algorithm of order 3(2) (cf. Figure (d)).

Figure 7.1(c) gives us a detailed view of the progress of the simulations used for the timings in (b). We see that for tolerances 10^{-2} and larger the simulations need only a few steps. This means the timings are not reliable because the last step towards $t = 0.005$ is much smaller than the step size which would have been chosen if the end of the integration interval was not encountered. For tolerances of 0.0032 and smaller we note a growing step size over the course of the simulation. This is an expected behavior if the step size is dominated by accuracy requirements, since the system is relaxing and thus the forces on the

membrane are decreasing.

The last figure shows the optimal step size depending on the required tolerance $Rtol = Atol$ (solid line) and the expected behaviors of the step size for methods of order 3 and order 1. The step size does not scale as expected, for small tolerances the order of the method is reduced to 1. This phenomenon of order reduction is well known [59]. Its reason lies in the fact that the underlying equation describing the dynamics is a PDE. Powers of the functional derivative of the right hand side of a equation of the form $u_{,t} = F(t, u)$ will be present in the local truncation error, and thus the order as derived from the Taylor expansion of the ODE and the Rosenbrock scheme are not valid without modification.

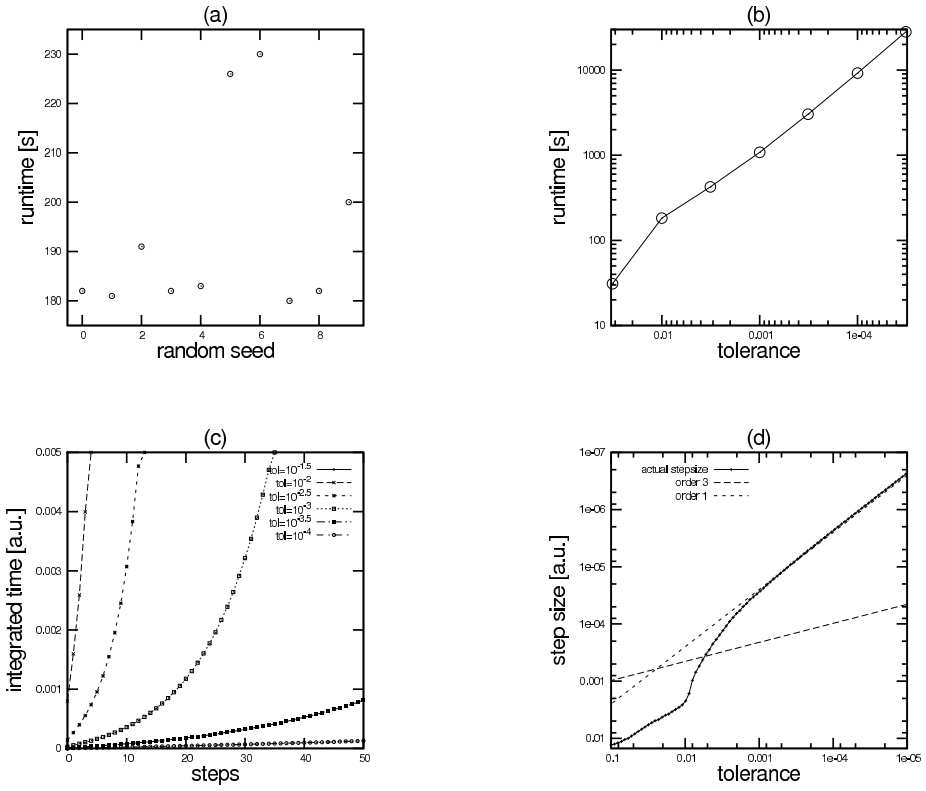

Figure 7.1: Performance of the ROWDAIND2 scheme. (a) Comparison of the total time needed to integrate test problems with different initial conditions. (b)Time needed to integrate test problem with different tolerances. (c) Progress of the integration for different tolerances. (d) Optimal step size for different tolerances.

We now compare the performance of different Rosenbrock schemes. The same initial conditions as in the test of the ROWDAIND2 scheme are used. The results are displayed in Figure 7.2. Part (a) shows the optimal step size for different tolerances of the Rosenbrock schemes as indicated in the figure legend. The step sizes were found at the beginning of a simulation; thus the dominant stresses stem from the bending rigidity of the membrane. In part (b) the optimal

step size is shown at a time at which the membrane is close to its relaxed shape. The stresses stemming from bending rigidity are balanced against those form the constraints on area, volume and area difference. To get an idea to the relative weights of the different contributions we take weighted averages over the 2-norms of the forces acting on the vertices at the different stages of the ROSI2P1 (cf. Appendix C) scheme, while setting either the bending rigidity or the forces from the constraints to zero. The step size was set to 0.001. The weights for the average are taken from the weights b_i of the stages of the Rosenbrock scheme (cf. Equation(6.13)). At the beginning of the simulation we have $||f||_{\kappa=0} = 0.00011815$ and $||f||_{\text{Constraints}=0} = 4.0003$ thus the forces acting on the vertices are dominated by the bending rigidity. For the membrane configuration used in figure 7.2(b), we have $||f||_{\kappa=0} = 0.43104$ and $||f||_{\text{Constraints}=0} = 0.38842$. Here the forces are approximately balanced against each other.

All methods use 4 stages, with one evaluation of the Jacobian. Given that the major cost of an integration step is the evaluation of the Jacobian and subsequent matrix factorization, all methods have comparable cost per step. Formally the methods are of order 3(2). In the bending rigidity dominated regime (part (a)) we note that all methods suffer from order reduction. The method ROWDAIND2 performs best, the other methods have step sizes of up to a factor 10 less. The fast decrease of step sizes at tolerances around a tolerance of 0.01 becomes much more prominent in part (b). Here we see two sharp decreases in step size around tolerances of 0.03 and 0.005. These stem from the coupling of the spatial and temporal discretization error. The solution of the continuous equation of motion would relax towards a shape which balances stresses from bending rigidity and constraints. The discretized membrane is only an approximation of the continuous solution. Therefore residual forces will occur. These residual forces lead to a motion of the membrane around a small neighborhood of the relaxed state of the continuous model. For small tolerances we trace this trajectory around the relaxed state. If we accept larger tolerances, we don't have to follow the model circling around the minimum and can take much larger step in the time integration.

7.2 The number of degrees of freedom and global constraints

To get a better idea of the behavior of simulations near the relaxed state of a membrane, we compare simulations with the same initial conditions and vary the number of degrees of freedom and the strength of the penalty term enforcing the conservation of the total mean curvature. We use the same initial conditions as in the previous experiments. Each simulation is run with 80, 320 and 1280 elements, which correspond to 42, 162 and 642 vertices. All simulations were run for two different values for the penalty against mean curvature differences, the penalties differ by a factor of 100.

To assess the change in volume V and total mean curvature M, we introduce the reduced quantities

$$v_0 = \frac{V}{\frac{4}{3}\pi r_0^3}$$

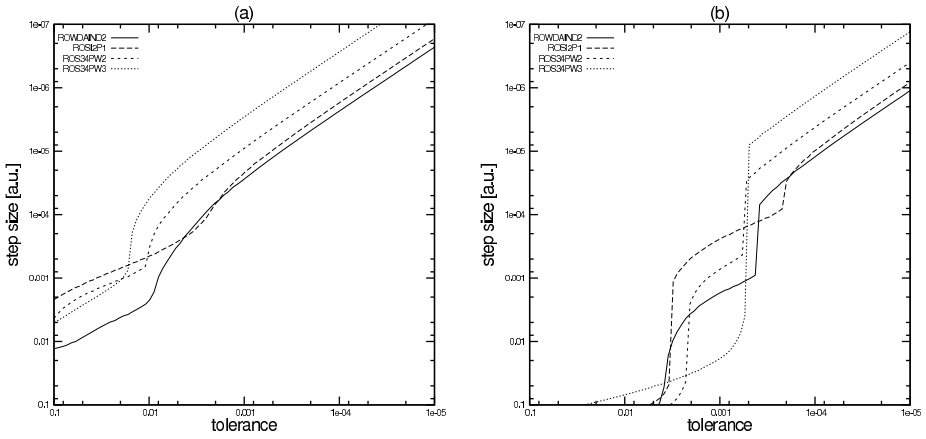

Figure 7.2: Comparison of the performance of different Rosenbrock schemes. Both plots show the optimal step size for different tolerances: (a) the dynamics of the system is governed by the bending rigidity of the membrane, (b) bending rigidity and constraints are balanced.

and

$$m_0 = \frac{M}{4\pi r_0}$$

with

$$r_0 = \sqrt{\frac{A}{4\pi}},$$

which measure the volume and total mean curvature relative to a sphere with the same surface area A [72, 77].

If we compare the range of reduced mean curvatures in the simulations (cf. Figure 7.3) we see that – as expected – increasing the penalty term for mean curvature differences decreases the mean curvature differences occurring during the simulation. In the previous section we discussed that the exact solution of the equation of motion can not be expressed by the discretized model and we saw the effect on the step size. Here we can see a more direct evidence for these oscillations: the reduced volume and curvature oscillate between different values. One would expect that the amplitude of the oscillations decreases with a better approximation of the exact solution. We can observe this behavior in Figure 7.3(b), and less pronounced in (a) as well. The amplitude of the oscillations in (a) is smaller because of the smaller penalty on differences in the mean curvature, and therefore smaller residual forces.

7.3 Adaptive Mesh Geometry

Now we discuss the influence of the moving mesh algorithm for improving the mesh quality. Two simulations with the same initial conditions are compared. A randomly deformed surface relaxes to its resting shape. All time integration was done using the ROWDAIND2 scheme. In the first simulation no attempts are made to improve the shapes of the elements. After 400 steps some triangles

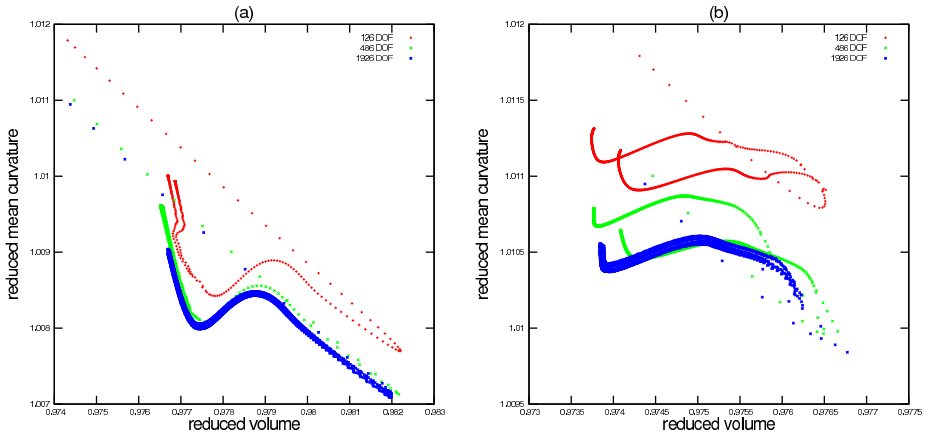

Figure 7.3: Reduced mean curvature and volume over the course of simulations with different numbers of DOF. The penalty on mean curvature differences in (a) is 100 times smaller than in (b). The curvature and volume values at the beginning of the simulation are those in the upper left of (a) and in the center and upper left of (b). The differences of the starting values between the simulations with different numbers of DOF are a result of the inexact numerical integration (cf. Section 5.3). The initial conditions are the same.

become very thin, the step size of the time integration decreases (cf. Fig. 7.4). After 500 steps the elements invert, the surface intersects with itself and the simulation becomes unstable. The step size starts to oscillate and decreases by 2 orders of magnitude.

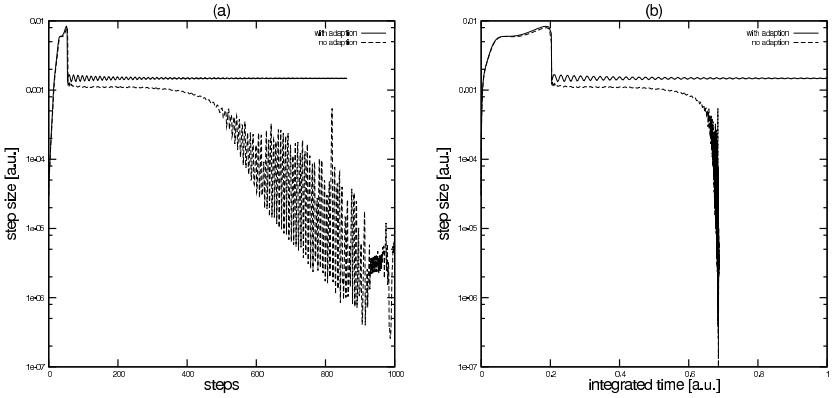

Figure 7.4: Step size during test simulations with and without mesh adaption techniques. (a) shows the step size as a function of the number of steps taken, (b) shows the step size as a function of the integrated time.

In contrast the second simulation improves the mesh quality by five steps of downhill minimization of the energy functional (discussed in Chapter 4.2) following each time step. During the first 55 steps both simulations display

similar step sizes, from then on the simulation improving the mesh quality performs better. Its step size displays oscillations with a period of 12 steps and an amplitude of approximately 10% of the step size. These oscillations stem from the coupling of the integration error and the step size with the element quality. Improving the element quality will allow larger step sizes, but if each step lowers the mesh quality, the quality improvement cannot keep up and the error grows again. Over time the surface relaxes, and its movement becomes slower, thus the change in the mesh quality per step decreases and so the amplitude of the oscillations in the step size[1]. The integration continues to larger times without showing the behavior of the first simulation.

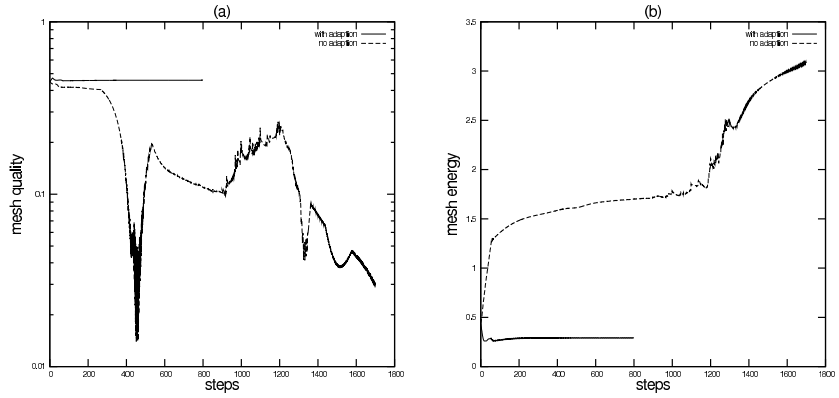

Figure 7.5: Mesh quality during test simulations with and without mesh adaption techniques. (a) shows the quality of the mesh measured by the smallest ratio of im–radius and circum–radius of all triangles, (b) shows the value of the energy functional used for the mesh adaption algorithm.

Figure 7.5 shows measures of the quality of a mesh, which is defined as the smallest ratio of im–radius and circum–radius of all elements with side lengths l_i,

$$q = \min_{\text{all elements}} \left(\frac{(l_2 + l_3 - l_1)(l_1 + l_3 - l_2)(l_1 + l_2 - l_3)}{2 l_1 l_2 l_3} \right). \tag{7.1}$$

In Fig. 7.5(a) we see the quality q of the mesh, as a function of the number of steps taken. For the simulation without mesh improvement we see a steep decay of the mesh quality around 400 steps into the simulation, which corresponds to the occurrence of thin triangles and decreasing step size. The following improvement of the quality is due to the inversion of the triangles (cf. Fig 7.6). The quality of the mesh in the simulation with mesh adaption is almost constant. Sub-figure (b) shows the value of the energy functional for both simulations, in the adaptive case the energy drops in an initial relaxation and then remains nearly constant. The value of the energy functional for the non adaptive case rises in the first 1000 steps. The high slope in the first steps is due to the large time steps taken. We see a second steep rise in the value of the energy functional after 1000 steps, which is caused by the unphysical buckling of the membrane, but there is no indication of the deterioration of the mesh quality at 400 steps,

[1] The step size does not increase because the global constraints keep the system frustrated.

ergo the energy functional itself is not a good measure of the mesh quality. The simulations are visualized in Fig. 7.6 and 7.7.

7.4 Micropipette Aspiration

A common method for experimental tests of the behavior of lipid membranes and in particular Red Blood Cells (RBC) is their aspiration into micropipettes [40, 7, 82, 83]. In these experiments pressure differences are used to aspirate a RBC into a pipette with an inner diameter of a few micron, while its deformation is monitored by microscopy.

Our goal here is not to provide new insights into the aspiration of RBCs. Instead, the virtual experiments serve as a testbed for simulations with external forces and geometric constraints. We are using a biconcave membrane as our initial condition. This shape is the result of simulating the relaxation of a membrane to its equilibrium shape. Therefore without the load applied by the pipette, the is bending rigidity almost balanced by the constraints.

The virtual pipette is aligned with the z-axis. The position of the center of the opening is $(0, 0, o_z)$. The the pressure difference Δp between the interior of the pipette and the surrounding medium is modeled as a constant stress s_p in the positive z–direction , which acts on the membrane surface $\vec{r}(v, w)$ facing the pipette, if the surface point is under the opening of the pipette. Its inner Radius is R_i and its outer radius R_a. We define the stress s_p as

$$s_p = \begin{cases} \Delta p & \vec{r} \cdot (1, 1, 0)^T < R_i \\ 0 & \text{otherwise} \end{cases} .$$

The rim of the pipette is modeled as a z–dependent stress $(0, 0, s_{rim})$. The functional form of

$$s_{rim}(\vec{r}) = \begin{cases} -100(\exp(9000(r_3 - o_z)^3) - 1) & \text{if } R_i < \vec{r} \cdot (1, 1, 0)^T < R_a \\ 0 & \text{else} \end{cases}$$

was chosen to provide a continuous stress that grows fast as the membrane enters the volume that is occupied by the rim of the pipette. We integrate the new stresses in the same manner as in section 2.3

$$f_{asp} = \vec{n} \int_\Omega \vec{n} \cdot (0, 0, s_{rim} + s_p)^T \psi dA \tag{7.2}$$

First we compare the aspiration of model membranes at different applied pressure differences (loads). We run two simulations with the same initial conditions and the same penalty on changes of the total mean curvature. The load applied by the pipette is varied by a factor of 10. The radius of the opening of the pipette is 58% of the cell radius. The outer rim of the pipette is 71% of the cells radius As expected from experimental experience, for smaller loads the length of the aspirated "tongue" decreases (cf. Figures 7.8 and 7.9) [82]. The invagination at the bottom of the cells is a result of the mean curvature constraint, it balances the curvature changes induced by the pipette at the top of the cell. Evans encountered these invaginations in his aspiration experiments with RBCs [40]. He reports a collapse of the cells at high loads. We cannot

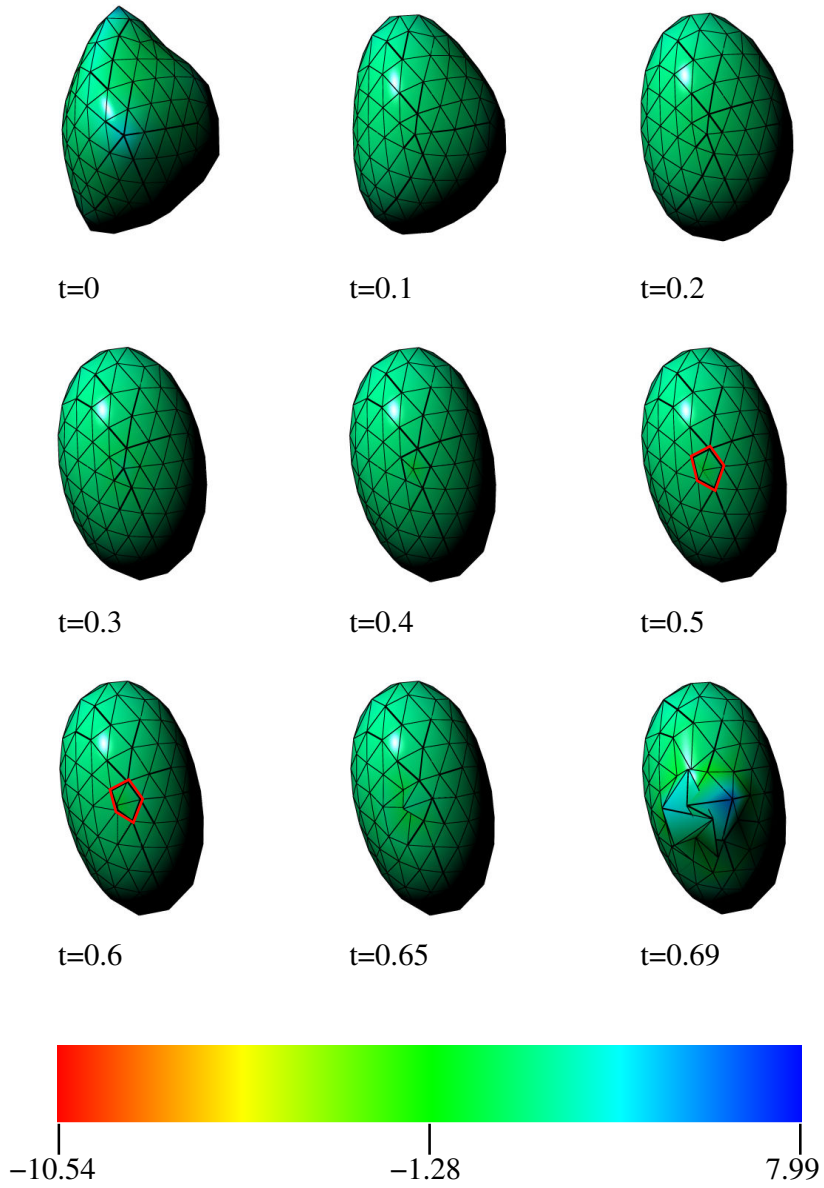

Figure 7.6: Visualization of a simulation without mesh quality improvement. The coloring indicates the mean curvature of the limit surface, the color lookup table is scaled to the range of mean curvature occurring in the simulation. Note the thin triangles at $t = 0.5$ (step 324). The red outline delineates five triangles around a vertex. At $t = 0.6$ (step 428) the two of the five triangles are nearly vanishingly thin. At $t = 0.65$ (step 503) we can see overlapping elements. The simulation then develops unrealistic buckling behavior at $t = 0.69$ (step 1683).

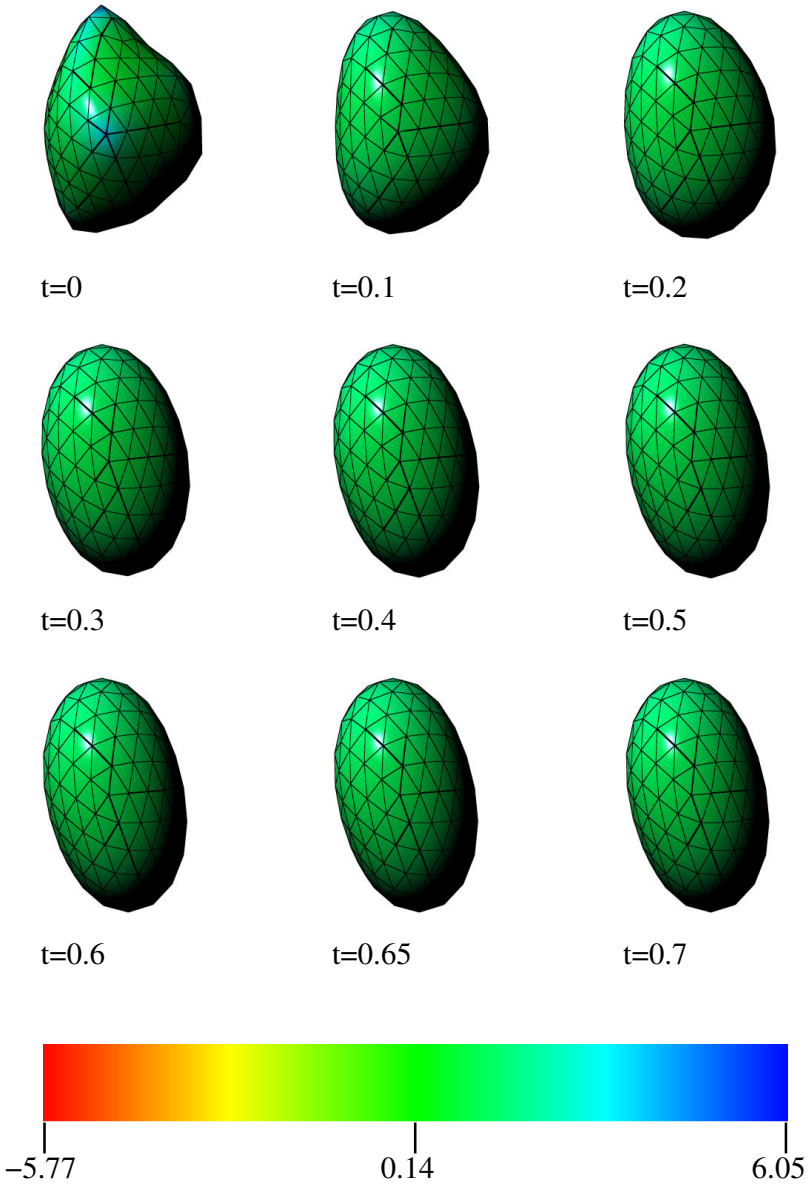

Figure 7.7: Visualization of a simulation with mesh quality improvement. The coloring indicates the mean curvature of the limit surface, the color lookup table is scaled to the range of mean curvature occurring in the simulation. We can see that the simulation proceeds to later times without the occurrence of distorted triangles.

reproduce this collapse. At high loads the simulated cell membranes intersect themselves and the simulation becomes unreliable. The self intersection in this situation is expected, for performance reasons we do not avoid self intersections by adding the appropriate force for cell–cell adhesion[2] and repulsion.

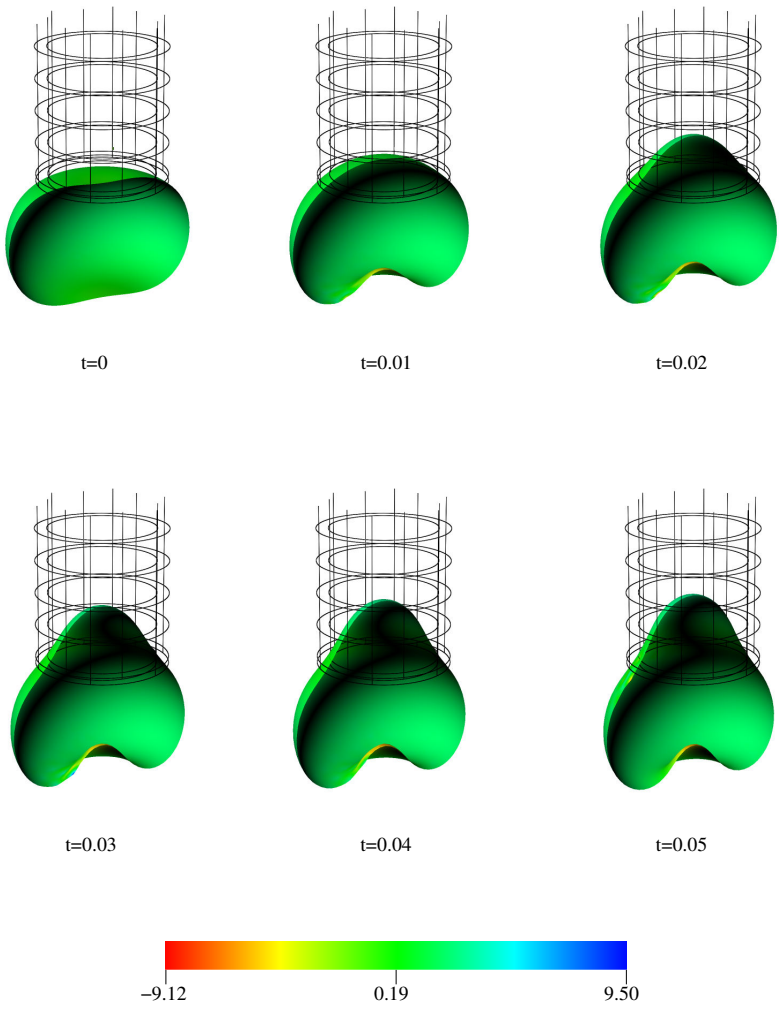

Figure 7.8: Simulation of the aspiration of a relaxed biconcave membrane into a micropipette.

Next we compare the effect of different penalty terms for changes of the integrated mean curvature. We use the same pressure difference as in the experiments shown in Figure 7.8, but we decrease the penalty by a factor of ten. No invagination forms at the bottom of the cell. The stresses arising form the bending rigidity of the membrane, exceed the forces from the bilayer coupling.

[2] At short distances two membranes experience repulsive forces, but thermal fluctuations can induce adhesive terms at longer distances [92].

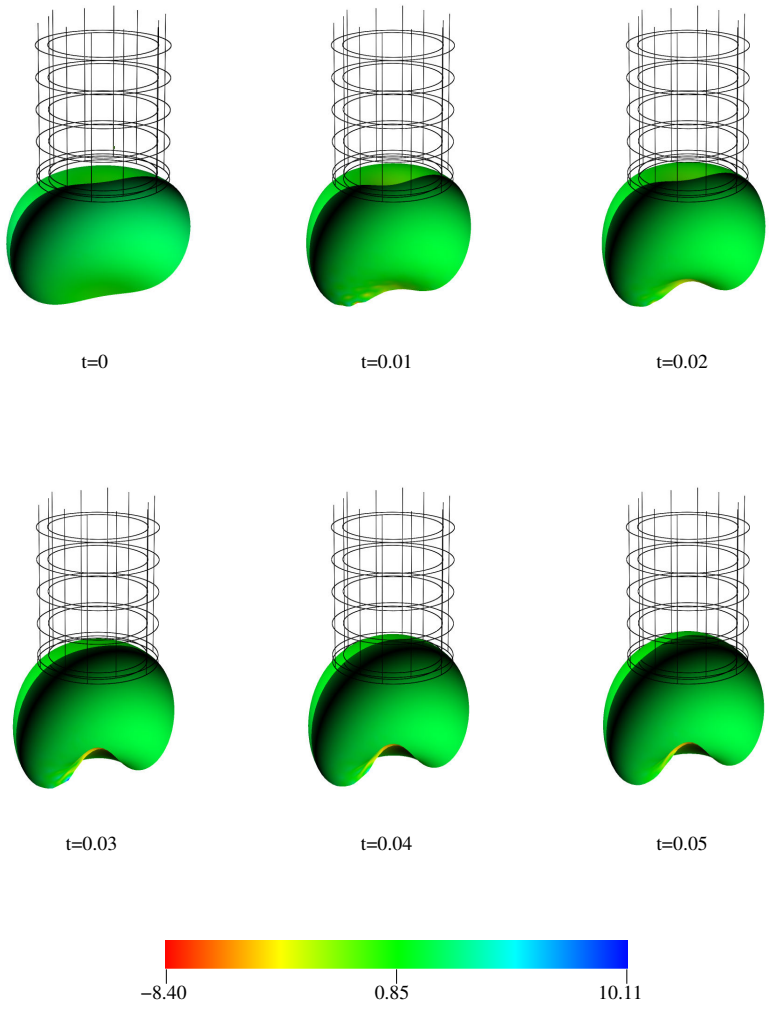

Figure 7.9: Simulation of the aspiration of a relaxed biconcave membrane into a micropipette.

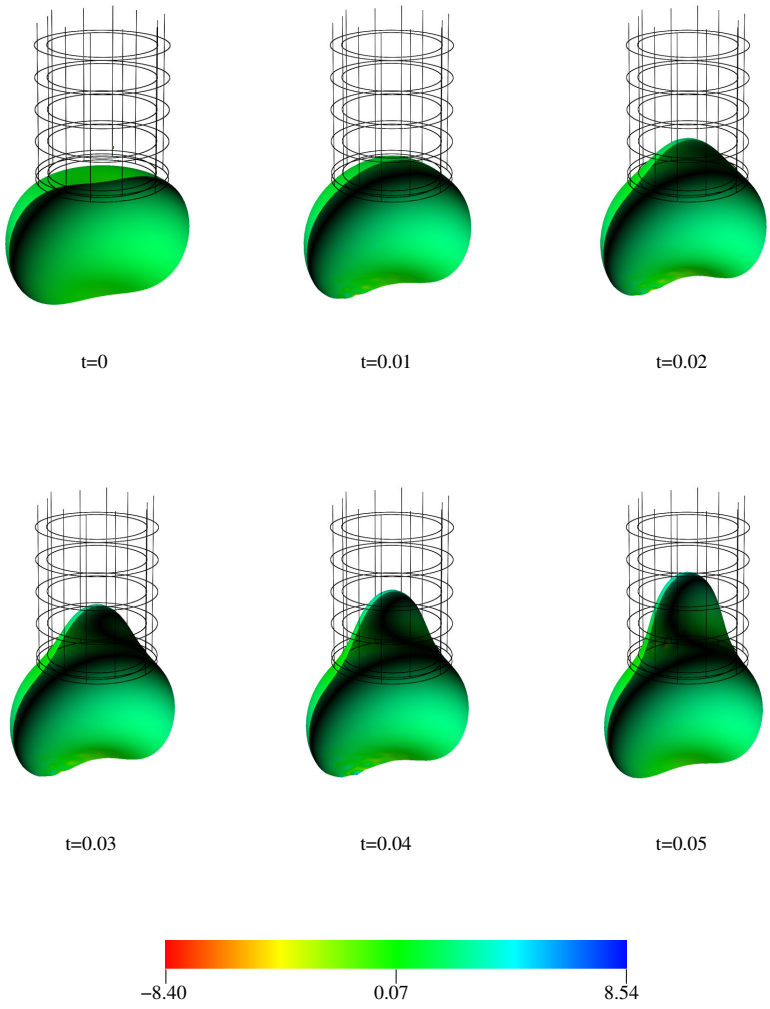

Figure 7.10: Simulation of the aspiration of a relaxed biconcave membrane into a micropipette.

Chapter 8

Summary and Outlook

Mathematical models of deformations of lipid bilayers have been developed over the past three decades. Until recently no attempts have been made to solve these models for arbitrary geometries. Currently there are three approaches.

Qiang Du et al. [35] and Campelo and Hernández-Machado [18] developed a simulation of lipid bilayers based on a variant of the level set method, which has the advantage of naturally handling topological changes. This comes at the price of an implicit discretization of the membrane, which has a storage complexity worse than $O(h^2)$, h being the linear resolution. This can be traded of for a time complexity worse than $O(1)$ for the random access of the membrane data, thus the achievable resolution is limited. This becomes a problem if we want to simulate migrating cells. Fibroblasts move their leading edge by forming filopodia, which are long and thin protrusions driven by polarization of actin [4]. If we want to model this process using a level set method, we have to ensure that the distance between the filopodia is large compared to the size of a grid cell. Due to the large number of filopodia formed at the leading edge of migrating fibroblasts [4] much higher resolutions than the 64^3 voxels reported by [35] would be needed.

A second approach was attempted by Seth Green [46]. A finite element scheme was employed to solve an elastic thin shell model of red blood cells as developed by Evans and Skalak [41]. This approach combines effects from the membrane skeleton and the cell membrane into an effective model of red blood cells. Local area conservation and resistance to shear are included in the model as features of the spectrin network. Important terms coupling the two sheets of the lipid bilayer are absent. A simple linear elasticity model yielded stable results when simulating the aspiration of cells. The implementation of a more realistic model in terms of nonlinear thin shells, which is taking account of membrane curvature is unable to deal with the stiffness of the system and shows an unrealistic buckling behavior for simulations of micropipette aspiration experiments [46].

This thesis is the third approach towards solving a model of deformation of lipid bilayers. The goal was to develop a faithful computational model of the behavior of the lipid bilayer. The fluid character of the lipid bilayer is manifest in the model equations and used to maintain a high quality discretization of the model. The fourth order partial differential equations were solved by the same finite element scheme as used by Green. A global nonlinear constraint on the

integrated mean curvature has been discussed in the literature [72, 77, 92] as an important contribution to the dynamics. We enforced this constraint, and we saw in our the aspiration experiments, that the bilayer coupling has a significant influence on the dynamics of cell membranes. Our aspiration experiments do not show any unexpected behavior. The development of methods for handling constraints without introducing unphysical behavior into the simulations is the major result of this thesis.

We carried out additional numerical experiments using Lagrange multipliers to enforce constraints. The time step used by the Rosenbrock method collapsed to 0 at the first integration step. Recall the results on the relation of the number of degrees of freedom and enforcement of the global constraints (cf. section 7.2). There we saw that the exact solution at any time can not be expressed by a limited number of basis functions. Therefore we cannot expect that a solution of the algebraic equation that enforces the constraints on the discretized manifold exists. This explains the failure of algorithms, which enforce the constraints exactly. Therefore our use of terms penalizing changes in the quantities, which we want to conserve, is the method of choice. One might argue, that the cost for having to operate with a dense Jacobian is too high, but studies on vesicle shapes indicate that the constraint on the total mean curvature are of the form used here [72]. Therefore we are led to a dense Jacobian anyway, and the additional cost of enforcing a constant area and volume of the cell is small.

Many theoretical works on lipid vesicles and red blood cells discuss thermal fluctuations of vesicles [77, 92, 43, 89]. We disregard thermal fluctuations. There is no data on the importance of fluctuations for many cell types. Simson et al. [95] reported bending rigidities of $70k_bT$ and larger for Dictyostelium Discoideum, a model organism of assays of cell motility and chemotaxis[1]. For bending rigidities that are large compared to k_bT, thermal fluctuations are small and can therefore be neglected. Apart from the unclear relevance for many cell types, the simulation of thermal fluctuations is beyond what is feasible on current commodity hardware.

An open problem is the scaling of the simulation with an increased number of degrees of freedom. Comparing simulations at different resolutions we can see that the time step scales as n^{-1}. Furthermore, the global constraints force us to deal with a dense Jacobian, because any change of a variable will influence all others through the constraints. The time to assemble the dense Jacobian scales with n^2. The cost for solving the dense linear system (6.21) in each step scales even worse with n^3 [80]. This makes it obvious that for more complex geometries we need an algorithm that adapts the resolution only locally to keep the discretization error below a user defined threshold, thereby only adding the DOF we need. Traditional refinement algorithms adapt the mesh by subdividing triangles [59]. A drawback of this method is that we would change the topology of the mesh. This raises the problem that we cannot subdivide an edge of a triangle without changing the adjacent triangle as well. Otherwise we would introduce discontinuities in our function space (cf. Chapter 4). This propagates the refinement of one triangle into its neighbors. A solution to this problem was suggested in a series of papers [47, 57, 58]. Their idea is to rely on the refinement equation of subdivision surfaces and use this to refine the basis functions instead of the triangles themselves. We would rewrite nodal basis functions in terms

[1] Migration of cells along gradients of a chemo–attractant.

of translated and dilated copies of themselves. By construction, an old basis function is a linear combination of the functions introduced using the refinement equations. To construct a new basis of linear independent functions we would have to remove one of the basis functions on the finer mesh. Krysl et. al [57] suggest to remove the nodal basis function at the vertex that is already present in the coarse mesh. This refinement of basis functions associated with the vertices naturally enforces the continuity of the functions in the Sobolev space [57]. In addition to avoiding unnecessary DOF, this method is a natural setting for a multi-grid. The refinement equation defines an interpolation and restriction operator, and by removing the linear dependent functions, we constructed a hierarchical basis such that our degrees of freedom only affect their characteristic length scale.

An important step towards biological applications will be the integration of the algorithms developed here with Simmune [70, 69, 71], a tool that simulates biochemical processes in cells and their interactions. The mesh used in this work can be coupled to a finite volume discretization of the interior of the cell, and Simmune can be used to solve the reaction–diffusion equations on the cell membrane and in the cytosol. Besides the biochemistry of the cytosol, one will want to simulate the biophysical properties of the interior of the cell. It acts not only as a passive viscoelastic medium, but the cytoskeleton can also actively generate forces [4].

Appendix A

Basis Functions

Here we list the finite element basis functions that we use on regular triangles. The generation of basis functions on irregular triangles is explained in Chapter 5.

$$
\begin{aligned}
b_0 &= -1/3w + 1/2w^2 + 1/2wv - 1/2w^2v + 1/6w^3v - 1/6wv^3 - 1/6v \\
&\quad + 1/12 - 1/3w^3 + 1/6v^3 + 1/12w^4 - 1/12v^4 \\
b_1 &= -1/6w + 1/2wv + 1/2v^2 - 1/2wv^2 - 1/6w^3v + 1/6wv^3 \\
&\quad - 1/3v + 1/12 + 1/6w^3 - 1/3v^3 - 1/12w^4 + 1/12v^4 \\
b_2 &= -1/6w - 1/2wv + 1/2w^2v - 1/6w^3v + 1/3wv^3 + 1/6v \\
&\quad + 1/12 + 1/6w^3 - 1/3v^3 - 1/12w^4 + 1/6v^4 \\
b_3 &= 1/2 - w^2 - wv - v^2 + w^2v + wv^2 - 1/6w^3v - 1/6wv^3 \\
&\quad + 2/3w^3 + 2/3v^3 - 1/12w^4 - 1/12v^4 \\
b_4 &= 1/6w - 1/2wv + 1/2wv^2 + 1/3w^3v - 1/6wv^3 \\
&\quad - 1/6v + 1/12 - 1/3w^3 + 1/6v^3 + 1/6w^4 - 1/12v^4 \\
b_5 &= 1/6v^3 - 1/6wv^3 - 1/12v^4 \\
b_6 &= 1/6w + 1/2wv + 1/2v^2 - w^2v - 1/2wv^2 + 1/3w^3v \\
&\quad - 1/6wv^3 + 1/3v + 1/12 - 1/3w^3 - 1/3v^3 + 1/6w^4 - 1/12v^4 \\
b_7 &= 1/3w + 1/2w^2 + 1/2wv - 1/2w^2v - wv^2 - 1/6w^3v \\
&\quad + 1/3wv^3 + 1/6v + 1/12 - 1/3w^3 - 1/3v^3 - 1/12w^4 + 1/6v^4 \\
b_8 &= 1/6w^3 - 1/12w^4 - 1/6w^3v \\
b_9 &= 1/6wv^3 + 1/12v^4 \\
b_{10} &= 1/6w^3 - 1/12w^4 - 1/6w^3v + 1/2w^2v \\
&\quad + 1/2wv^2 - 1/6wv^3 + 1/6v^3 - 1/12v^4 \\
b_{11} &= 1/12w^4 + 1/6w^3v.
\end{aligned}
$$

Appendix B

Quadrature

These are the Gaussian quadrature points and weights given in canonical coordinates (u,v). Which were derived from [36] using $\alpha+\beta+\gamma=1$ i.e choosing the right permutations of the points. All quadrature points up to $n=13$ are given. All simulations in this thesis were done using 6 quadrature points. With less quadrature points the error of the numerical integration was undesireably large, or the simulation would even become unstable. Higher numbers of quadrature points are not used as the cost for the integration is proportional to the number of quadrature points.

n	w_i	u_i	v_i
1	1.000000000000000	0.333333333333333	0.333333333333333
3	0.333333333333333	0.666666666666667	0.166666666666667
	0.333333333333333	0.166666666666667	0.666666666666667
	0.333333333333333	0.166666666666667	0.166666666666667
4	-0.562500000000000	0.333333333333333	0.333333333333333
	0.520833333333333	0.600000000000000	0.200000000000000
	0.520833333333333	0.200000000000000	0.600000000000000
	0.520833333333333	0.200000000000000	0.200000000000000
6	0.223381589678011	0.108103018168070	0.445948490915965
	0.223381589678011	0.445948490915965	0.108103018168070
	0.223381589678011	0.445948490915965	0.445948490915965
	0.109951743655322	0.816847572980459	0.091576213509771
	0.109951743655322	0.091576213509771	0.816847572980459
	0.109951743655322	0.091576213509771	0.091576213509771

n	w_i	u_i	v_i
7	0.225000000000000	0.333333333333333	0.333333333333333
	0.132394152788506	0.059715871789770	0.470142064105115
	0.132394152788506	0.470142064105115	0.059715871789770
	0.132394152788506	0.470142064105115	0.470142064105115
	0.125939180544827	0.797426985353087	0.101286507323456
	0.125939180544827	0.101286507323456	0.797426985353087
	0.125939180544827	0.101286507323456	0.101286507323456
12	0.116786275726379	0.501426509658179	0.249286745170910
	0.116786275726379	0.249286745170910	0.501426509658179
	0.116786275726379	0.249286745170910	0.249286745170910
	0.050844906370207	0.872821971016996	0.063089014491502
	0.050844906370207	0.063089014491502	0.872821971016996
	0.050844906370207	0.063089014491502	0.063089014491502
	0.082851075618374	0.053145049844817	0.310352451033784
	0.082851075618374	0.053145049844817	0.636502499121399
	0.082851075618374	0.310352451033784	0.053145049844817
	0.082851075618374	0.310352451033784	0.636502499121399
	0.082851075618374	0.636502499121399	0.053145049844817
	0.082851075618374	0.636502499121399	0.310352451033784
13	-0.149570044467682	0.333333333333333	0.333333333333333
	0.175615257433208	0.479308067841920	0.260345966079040
	0.175615257433208	0.260345966079040	0.479308067841920
	0.175615257433208	0.260345966079040	0.260345966079040
	0.053347235608838	0.869739794195568	0.065130102902216
	0.053347235608838	0.065130102902216	0.869739794195568
	0.053347235608838	0.065130102902216	0.065130102902216
	0.077113760890257	0.048690315425316	0.312865496004874
	0.077113760890257	0.048690315425316	0.638444188569810
	0.077113760890257	0.312865496004874	0.048690315425316
	0.077113760890257	0.312865496004874	0.638444188569810
	0.077113760890257	0.638444188569810	0.312865496004874
	0.077113760890257	0.638444188569810	0.048690315425316

Appendix C

Coefficients of the Rosenbrock Methods

The coefficients of all Rosenbrock methods used in this thesis are given here. All coefficients not listed are set to 0.

$\gamma = 0.3$	
$\alpha_{21} = 0.5$	$\gamma_{21} = -0.1121794871794876$
$\alpha_{31} = 0.28$	$\gamma_{31} = 2.54$
$\alpha_{32} = 0.72$	$\gamma_{32} = -3.84$
$\alpha_{41} = 0.28$	$\gamma_{41} = 29/75$
$\alpha_{42} = 0.72$	$\gamma_{42} = -0.72$
$\alpha_{43} = 0$	$\gamma_{43} = 1/30$
$b_1 = 2/3$	$\hat{b}_1 = 0.4799002800355166$
$b_2 = 0$	$\hat{b}_2 = 0.5176203811215082$
$b_3 = 1/30$	$\hat{b}_3 = 0.002479338842975209$
$b_4 = 0.3$	$\hat{b}_4 = 0$

Table C.1: Coeffcients for the ROWDAIND2 scheme [68].

$\gamma = 0.435866521508459$	
$\alpha_{21} = 0.87173304301691801$	$\gamma_{21} = -\alpha_{21}$
$\alpha_{31} = 0.84457060015369423$	$\gamma_{31} = -0.90338057013044082$
$\alpha_{32} = -0.11299064236484185$	$\gamma_{32} = 0.054180672388095326$
$\alpha_{41} = 0$	$\gamma_{41} = 0.24212380706095346$
$\alpha_{42} = 0$	$\gamma_{42} = -1.2232505839045147$
$\alpha_{43} = 1$	$\gamma_{43} = 0.54526025533510214$
$b_1 = 0.24212380706095346$	$\hat{b}_1 = 0.37810903145819369$
$b_2 = -1.2232505839045147$	$\hat{b}_2 = -0.096042292212423178$
$b_3 = 1.5452602553351020$	$\hat{b}_3 = 0.5$
$b_4 = 0.435866521508459$	$\hat{b}_4 = 0.2179332607542295$

Table C.2: Coeffcients for the ROS34PW2 scheme [85].

„ $\gamma = 1.0685790213016289$	
$\alpha_{21} = 2.5155456020628817$	$\gamma_{21} = -\alpha_{21}$
$\alpha_{31} = 0.50777280103144085$	$\gamma_{31} = -0.87991339317106512$
$\alpha_{32} = 0.75$	$\gamma_{32} = -0.96014187766190695$
$\alpha_{41} = 0.13959081404277204$	$\gamma_{41} = -0.41731389379448741$
$\alpha_{42} = -0.33111001065419338$	$\gamma_{42} = 0.41091047035857703$
$\alpha_{43} = 0.82040559712714178$	$\gamma_{43} = -1.3558873204765276$
„ $b_1 = 0.22047681286931747$	$\hat{b}_1 = 0.31300297285209688$
$b_2 = 0.0027828278331185935$	$\hat{b}_2 = -0.28946895245112692$
$b_3 = 0.0071844787635140066$	$\hat{b}_3 = 0.97646597959903003$
$b_4 = 0.76955588053404989$	$\hat{b}_4 = 0$

Table C.3: Coeffcients for the ROS34PW3 scheme [85].

$\gamma = 0.43586652150845900$	
$\alpha_{21} = 0.5$	$\gamma_{21} = -0.5$
$\alpha_{31} = 0.55729261836499822$	$\gamma_{31} = -0.64492162993321323$
$\alpha_{32} = 0.19270738163500176$	$\gamma_{32} = 0.063491801247597734$
$\alpha_{41} = -0.30084516445435860$	$\gamma_{41} = 0.0093606009252719842$
$\alpha_{42} = 1.8995581939026787$	$\gamma_{42} = -0.25462058718013519$
$\alpha_{43} = -0.59871302944832006$	$\gamma_{43} = -0.32645441930944352$
$b_1 = 0.052900072579103834$	$\hat{b}_1 = 0.14974465479289098$
$b_2 = 1.3492662311920438$	$\hat{b}_2 = 0.70051069041421810$
$b_3 = -0.91013275270050265$	$\hat{b}_3 = 0$
$b_4 = 0.50796644892935516$	$\hat{b}_4 = 0.14974465479289098$

Table C.4: Coeffcients for the ROSI2P1 scheme [84].

Bibliography

[1] 754-1985 IEEE Standard for Binary Floating-Point Arithmetic. ISBN 1-5593-7653-8.

[2] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277, 1995.

[3] R. C. Aiken, editor. *Stiff computaion.* Oxford University Press, 1985.

[4] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson. *Molecular Biology of the Cell.* Garland Publishing, 4th edition, 2002.

[5] R. An and Z. C. Tu. The application of exterior differential forms in variational problems on manifolds. arXiv:math-ph/0307007, Jul 2003.

[6] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide.* Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[7] G. Artmann, K. Sung, T. Horn, D. Whittemore, G. Norwich, and S. Chien. Micropipette aspiration of human erythrocytes induces echinocytes via membrane phospholipid translocation. *Biophys J*, 72(3):1434–41, Mar 1997.

[8] S. Balay, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[9] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.

[10] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[11] R. Bank and R. Smith. Mesh smoothing using a posteriori error estimates. *SIAM Journal on Numerical Analysis*, 34:979–997, 1997.

[12] T. Baumgart, S. T. Hess, and W. W. Webb. Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension. *Nature*, 425(6960):821–4, Oct 2003.

[13] M. Bertram. Biorthogonal loop-subdivison wavelets. *Computing*, 72:29–39, 2004.

[14] N. Blanchard and C. Hivroz. The immunological synapse: the more you look the less you know... *Biol Cell*, 94(6):345–54, Oct 2002.

[15] D. Boal. *Mechanics of the Cell.* Cambridge University Press, 2001.

[16] A. Bossavit. *Computational Electromagnetism.* Academic Press, 1998.

[17] D. Braess. *Finite elements.* Cambridge University Press, second edition, 2001.

[18] F. Campelo and A. Hernández-Machado. Dynamic model and stationary shape of fluid vesicles. arXiv:cond-mat/0603437v1, Mar 2006.

[19] P. Canham. The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell. *J. theoretical Biology*, 26:61–81, 1970.

[20] R. Capovilla and J. Guven. Stresses in lipid membranes. *J. Phys. A*, 35:6233–6247, 2002.

[21] R. Capovilla, J. Guven, and J. A. Santiago. Deformations of the geometry of lipid vesicles. *J. Phys. A*, 36:6281–6295, 2003.

[22] R. Capovilla and G. J. Stress and geometry in lipid vesicles. *J. Phys.: Condens. Matter*, 16:S2178–S2191, 2004.

[23] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *computer-aided design*, 10(6):351–355, 1978.

[24] S. Chen, B. Merriman, S. Osher, and P. Smereka. A simple level set method for solving stefan problems. *Journal of Computational Physics*, 135:8–29, 1997.

[25] P. G. Ciarlet and J. L. Lions, editors. *Handbook of Numerical Analysis*, volume 2. Elsevier Science Publishers B. V., 1991.

[26] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: A new paradigm for thin-shell finite-element analysis. http://www.multires.caltech.edu/pubs/pubs.htm, 2000.

[27] V. Cristini, J. Blawzdziewicz, and M. Loewenberg. An adaptive mesh algorithm for evolving surfaces: Simulations of drop breakup and coalescence. *Journal of Computational Physics*, 168:445–463, 2001.

[28] C. F. Curtis and J. Hirschfelder. Integration of stiff equations. *Proceedings of the National Academy of Science*, 38:235–243, 1952.

[29] W. DeLano. The pymol molecular graphics system. http://www.pymol.org, 2002. DeLano Scientific, San Carlos, CA, USA.

[30] M. Desbrun, E. Kanso, and Y. Tong. Discrete differential forms for computational modeling. In *International Conference on Computer Graphics and Interactive Techniques archive ACM SIGGRAPH 2005 Courses*, 2005. Avialable at http://geometry.caltech.edu/pubs/DKT05.pdf.

[31] M. Deserno. Notes on differential geometry with special emphasis on surfaces in $\mathbb{R}^3$. Published online as lecture notes at the Max-Planck-Institut für Polymerforschung, Mainz, Germany., May 2004. http://www.mpip-mainz.mpg.de/~deserno/scripts/diff_geom/diff_geom.pdf.

[32] G. A. Di Mazo. RODAS5(4) méthodes de rosenbrock d'ordre 5(4) adaptées aux problèmes diffèrentiels algébriques. Master's thesis, Université de Geneve, 1993.

[33] J. Dongarra. Preface: Basic Linear Algebra Subprograms Technical (Blast) Forum Standard I. 16(1):1–111, Spring 2002.

[34] J. Dongarra. Preface: Basic Linear Algebra Subprograms Technical (Blast) Forum Standard II. 16(2):115–199, Summer 2002.

[35] Q. Du, C. Liu, and X. Wang. Simulating the deformations of vesicle membranes under elastic bending energy in three dimensions. *Journal of Computational Physics*, 212:757–777, 2005.

[36] D. A. Dunavant. High degree efficient symmetrical gaussian quadrature rules for the triangle. *International Journal for Numerical Methods in Engineering*, 21:1129–1148, 1985.

[37] N. Dyn, D. Leviatan, and D. Levin, editors. *Multivariate Approximation and Applications*. Cambridge University Press, 2001.

[38] S. Elcott, Y. Tong, P. Kanso, E. Schrï£¡er, and M. Desbrun. Stable, circulation-preserving, simplicial fluids. *CM Transactions on Graphics (TOG)*, 26:Article No. 4, 2007.

[39] E. Evans. Bending resistance and chemically induced moments in membrane bilayers. *Biophys J*, 14(12):923–31, Dec 1974.

[40] E. A. Evans. Bending elastic modulus of red blood cell membrane derived from buckling instability in micropipet aspiration tests. *Biophysical Journal*, 43(1):27–30, 1983.

[41] E. A. Evans and R. Skalak. *Mechanics and Thermodynamics of Biomembranes*. CRC Press, 1980.

[42] G. Farin. *Curves and Surfaces for CAGD – A Practical Guide*. Morgan Kaufmann Publishers, fith edition, 2002.

[43] J.-B. Fournier, D. Lacoste, and E. Rapha? Fluctuation spectrum of fluid membranes coupled to an elastic meshwork: jump of the effective surface tension at the mesh size. *Phys Rev Lett*, 92(1):018102, Jan 2004.

[44] C. W. Gear and L. R. Petzold. Ode methods for the solution of differential/algebraic systems. *SIAM Numerical Analysis*, 21:716–728, 1984.

[45] E. Giladi. Finite elements lecture notes. The lecture notes were available at http://www.rpi.edu/ gilade/Math6860.htm, as of March 7, 2007 they can be found at http://www.cs.rpi.edu/ flaherje/pdf/.

[46] S. Green. *Multilevel, Subdivision-Based, Thin Shell Finite Elements: Development and an Application to Red Blood Cell Modeling.* PhD thesis, University of Washington, 2003.

[47] E. Grinspun, P. Krysl, and P. S. Oder. Charms: A simple framework for adaptive simulation. 2002.

[48] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations*, volume I. Springer, 2 nd edition, 1993.

[49] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations*, volume II. Springer, 2 nd edition, 1996.

[50] J. Happel. *Low Reynolds number hydrodynamics, with special applications to particulate media.* Prentice-Hall international series in the physical and chemical engineering sciences. Prentice-Hall, 1965.

[51] W. Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Z Naturforsch [C]*, 28(11):693–703, 1973.

[52] H. Heller, M. Schaefer, and S. K. Molecular dynamics simulation of a bilayer of 200 lipids in the gel and in the liquid-crystal phases. *J. Phys. Chem.*, 97:8343–8360, 1993.

[53] A. N. Hirani. *Discrete Exterior Calculus.* PhD thesis, California Institue of Technology, 2003.

[54] B. Houston, M. B. Nielsen, C. Batty, and K. Nilsson, O. Museth. Hierarchical RLE level set: A compact and versatile deformable surface representation. *ACM Transactions on Graphics*, 25(1):151 – 175, 2006.

[55] C. A. Janeway, P. Travers, M. Walport, and M. J. Shlomchik. *Immunobiology.* Garland Publishing, fifth edition, 2001.

[56] P. A. Janmey and P. K. J. Kinnunen. Biophysical properties of lipids and dynamic membranes. *Trends Cell Biol*, 16(10):538–546, Oct 2006.

[57] P. Krysl, E. Grinspun, and P. Schroder. Natural hierarchical refinement for finite element methods. 2002.

[58] P. Krysl, A. Trivedi, and B. Zhu. Object-oriented hierarchical mesh refinement with charms. *Int. J. Numer. Meth. Engng*, 00:0, 2002.

[59] J. Lang. *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems.* Lecture Notes in Computational Science and Engineering. Springer, 2001.

[60] J. Lang, W. Cao, W. Huang, and R. D. Russell. A two-dimensional moving finite element method with local refinement based on a posteriori error estimates. Avialable at: citeseer.ifi.unizh.ch/567395.html.

[61] J. Lang and J. Verwer. ROS3P – an accurate third-order rosenbrock solver designed for parabolic problems. *BIT Numerical Mathematics*, 41:731–??, 2001.

[62] G. Le Vey. Differential algebraic equations - a new look at the index. 1994.

[63] S.-J. E. Lee, Y. Hori, and A. K. Chakraborty. Low t cell receptor expression and thermal fluctuations contribute to formation of dynamic multifoval synapses in thymocytes. *PNAS*, 100(8):4383–4388, 2003.

[64] S.-J. E. Lee, Y. Hori, J. T. Groves, M. L. Dustin, and A. K. Chakraborty. Correlation of a dynamic model for immunological synapse formation with effector functions: two pathways to synapse formation. *Trends Immunol*, 23(10):492–9, Oct 2002.

[65] S.-J. E. Lee, Y. Hori, J. T. Groves, M. L. Dustin, and A. K. Chakraborty. The synapse assembly model. *TRENDS in Immunology*, 23(10):500–502, 2002.

[66] C. T. Loop. Smooth subdivision based on triangles. Master's thesis, University of Utah, 1987.

[67] F. Losasso, F. Gibou, and R. Fedkiw. Simulating water and smoke with an octree data structure. *SIGGRAPH 2004, ACM TOG*, 23:457–462, 2004.

[68] C. Lubich and M. Roche. Rosenbrock methods for differential-algebraic systems with solution-dependent singular matrix multiplying the derivative. *Computing*, 43:325–342, 1990.

[69] M. Meier-Schellersheim. *The immune system as a complex system, description and simulation of the interactions of its components.* PhD thesis, II. Institue for Theoretical Physics, University of Hamburg, 2001.

[70] M. Meier-Schellersheim and G. Mack. Simmune a tool for simulating and analyzing immune system behavior. arXiv:cs.MA/9903017, Mar. 1999.

[71] M. Meier-Schellersheim, X. Xu, B. Angermann, E. J. Kunkel, T. Jin, and R. N. Germain. Key role of local regulation in chemosensing revealed by a new molecular interaction-based modeling method. *PLoS Comput Biol*, 2(7):e82, Jul 2006.

[72] Miao, Seifert, Wortis, and Dï£¡ereiner. Budding transitions of fluid-bilayer vesicles: The effect of area-difference elasticity. *PHYSICAL REVIEW. E. STATISTICAL PHYSICS, PLASMAS, FLUIDS, AND RELATED INTERDISCIPLINARY TOPICS*, 49(6):5389–5407, Jun 1994.

[73] M. Nakahara. *Geometry, Topology and Physics.* Adam Hilger, 1990.

[74] B. Nichols. Cell biology: without a raft. *Nature*, 436:638–639, 2005.

[75] S. Osher and R. Fedkiw. Level set methods: An overview and some recent results. *J. Comput. Phys.*, 169:463–502, 2001.

[76] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[77] L. Peliti. Amphiphilic membranes. arXiv:cond-mat/9501076, Jan 1995.

[78] D. Peng, B. Merriman, S. Osher, H.-K. Zhao, and M. Kang. A pde-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.

[79] A. G. Petrov and I. Bivas. Elastic and flexoelectric aspects of out of plane fluctuations in biological and model membranes. *Progress in Surface Sci.*, 16, 1984.

[80] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C The Art of Scientific Computing.* Cambridge University Press, 1992.

[81] S. Y. Qi, J. T. Groves, and A. K. Chakraborty. Synaptic pattern formation during cellular recognition. *PNAS*, 98(12), 2001.

[82] R. P. RAND. MECHANICAL PROPERTIES OF THE RED CELL MEMBRANE. II. VISCOELASTIC BREAKDOWN OF THE MEMBRANE. *Biophys J*, 17:303–316, Jul 1964.

[83] R. P. RAND and A. C. BURTON. MECHANICAL PROPERTIES OF THE RED CELL MEMBRANE. I. MEMBRANE STIFFNESS AND INTRACELLULAR PRESSURE. *Biophys J*, 45:115–135, Mar 1964.

[84] J. Rang and L. Angermann. New rosenbrock methods of order 3 for pdaes of index 2. Technical Report Mathematik-Bericht Nr. 2005/3, Institut fr Mathematik, Technische Universitï£¡ Clausthal, 2005.

[85] J. Rang and A. L. New rosenbrock w-methods of order 3 for partial differential algebraic wquations of index 1. *BIT Nummerical MAthematics*, 45:761–787, 2005.

[86] P. Rentrop, M. Roche, and G. Steinebach. The application of rosenbrock-wanner type methods with stepsize control in differential-algebraic equations. *Numerische Mathematik*, 55:545–563, 1989.

[87] R. Richter, J. Frauendiener, and V. M. General relativity and quantum cosmology, abstract. arXiv:gr-qc/0608041, Aug 2006.

[88] M. Roche. Rosenbrock methods for differential algebraic equations. *Numerische Mathematik*, 52:45–63, 1988.

[89] S. A. Safran. *Statistical Thermodynamics of Surfaces, Interfaces, amd Membranes*, volume 90 of *Frontiers in physics*. Addison-Wesley Publishing Company, 1994.

[90] A. Sandu, J. G. Verwer, J. G. Blom, E. J. Spee, and F. A. Porta. Benchmarking stiff ode solvers for atmospheric chemistry problems ii: Rosenbrock solvers. *Athmospheric Environment*, 31(20):3459–3472, 1997.

[91] P. Schröder and U. Reif. Curvature integrability of subdivision surfaces. 2001.

[92] U. Seifert. Configuration of fluid membranes and vesicles. *Advances in Physics*, 46(1):13–137, 1997.

[93] J. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Science*. Cambridge University Press, 1999.

[94] M. Sheetz and S. Singer. Biological membranes as bilayer couples. A molecular mechanism of drug-erythrocyte interactions. *Proc Natl Acad Sci U S A*, 71(11):4457–61, Nov 1974.

[95] R. Simson, E. Wallraff, J. Faix, J. Niewï£¡ner, G. Gerisch, and E. Sackmann. Membrane bending modulus and adhesion energy of wild-type and mutant cells of Dictyostelium lacking talin or cortexillins. *Biophys J*, 74(1):514–522, Jan 1998.

[96] J. Stam. Evaluation of loop subdivision surfaces. Technical report, Alias|wavefront, Inc.

[97] J. Stam. Exact evaluation of catmull-clark subdivision surfaces at arbitrary parameter values. 2001.

[98] E. M. Stokely and S. Y. Wu. Surface parametrisation and curvature measurement of arbitrary 3-d objects: Five practical methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8):833–840, Aug. 1992.

[99] Stoker. *Differential Geometry*, volume XX of *Pure and Applied Mathematics*. Wiley-Interscience, 1969.

[100] G. Strang and G. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Inc., 1973.

[101] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2000.

[102] M. Sussmann, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.

[103] S. Svetina, M. Brumen, and Žekš. Lipid bilayer elasticity and the bilayer couple interpretation of red cell shape transformations and lysis. *studia biophysica*, 110(1-3):177–184, 1985.

[104] S. Svetina and B. Zeks. Bilayer couple as a possible mechanism of biological shape formation. *Biomed Biochim Acta*, 44(6):979–86, 1985.

[105] Z. Tu and Z. Ou-Yang. Lipid membranes with free edges. *Phys Rev E Stat Nonlin Soft Matter Phys*, 68(6 Pt 1):061915, Dec 2003.

[106] Z. C. Tu and Z. C. Ou-Yang. Variational problems in elastic theory of biomembranes, smectic-a liquid crystals, and carbon related structures. arXiv:math-ph/0506055, Jun 2005.

[107] G. G. Walter. *Wavelets and Other Orthogonal Systems With Applications.* CRC Press, 1994.

[108] R. Waugh, J. Song, S. Svetina, and B. Zeks. Local and nonlocal curvature elasticity in bilayer membranes by tether formation from lecithin vesicles. *Biophys J*, 61(4):974–82, Apr 1992.

[109] V. Weaver. personal communication.

[110] R. E. White. *An Introduction to the Finite Element Method with Applications to Nonlinear Problems.* John Wiley & Sons, 1985.

[111] T. Yeung, P. C. Georges, L. A. Flanagan, B. Marg, M. Ortiz, M. Funaki, N. Zahir, W. Ming, V. Weaver, and P. A. Janmey. Effects of substrate stiffness on cell morphology, cytoskeletal structure, and adhesion. *Cell Motil Cytoskeleton*, 60(1):24–34, Jan 2005.

[112] Zhong-can and Helfrich. Bending energy of vesicle membranes: General expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders. *PHYSICAL REVIEW. A*, 39(10):5280–5288, May 1989.

[113] J. Zimmerberg and S. McLaughlin. Membrane curvature: how BAR domains bend bilayers. *Curr Biol*, 14(6):R250–2, Mar 2004.

[114] D. Zorin and P. Schröder, editors. *Subdivision for Modeling and Animation.* ACM, 2000.

Acknowledgments

First I would like to thank my advisors Gerhard Mack and Martin Meier-Schellersheim for permitting my work on this fascinating topic so far from High Energy Physics. I am most grateful towards Ronald N. Germain for the kind reception in the Lympocyte Biology Section of the Laboratory of Immunology. I spend a lot of time discussing with Frederick Klauschen, I thank him for sharing his insights and his efforts in making our office more habitable. Alison Wise worked with me on the visualizations of signal transduction pathways; despite our depressing basement she would always contribute a song lightning up the mood. I learned a lot about biology, I thank everyone in the LBS for showing me this intriguing field. Discussing with Irena Štefanová was always a great enjoyment. I thank Jackson Egen and Ina Ifrim for teaching me experimental Methods. Lilly Koo always managed to cheer me up. Thank you to my family and unnamed friends for supporting and encouraging me.

The time I spend researching this thesis was above all the time I could spend with my wife Anke Knauf, who had the courage to join me in an uncertain venture. I love you.

During my stay at the II. Institute for Theoretical Physics in the Department of Physics of the University of Hamburg this work was supported by the Physics Graduate School "Future Developments in Particle Physics", which is supported by the German Research Foundation (DFG).
My research in the Laboratory of Immunology and in the Program in Systems Immunology and Infectious Disease Modeling (PSIIM) at the National Institute of Allergy and Infectious Diseases was supported by the Intramural Research Program of the US National Institutes of Health.
This research was supported, in part, by the Intramural Research Program of the US National Institutes of Health (NIH), National Institute of Allergy and Infectious Diseases.