Parametric FEM for Geometric Biomembranes

Andrea Bonito[a,1], Ricardo H. Nochetto[b,c,2], M. Sebastian Pauletti[b,*,3]

[a]*Department of Mathematics, Texas A&M University, College Station, TX, 77843, USA*
[b]*Department of Mathematics, University of Maryland ,College Park, MD, 20742, USA*
[c]*Institute for Physical Science and Technology, University of Maryland ,College Park, MD, 20742, USA*

Abstract

We consider geometric biomembranes governed by an L^2-gradient flow for bending energy subject to area and volume constraints (Helfrich model). We give a concise derivation of a novel vector formulation, based on shape differential calculus, and corresponding discretization via parametric FEM using quadratic isoparametric elements and a semi-implicit Euler method. We document the performance of the new parametric FEM with a number of simulations leading to dumbbell, red blood cell and toroidal equilibrium shapes while exhibiting large deformations.

Key words: Moving finite elements, Isoparametric elements, Geometric flow, Shape differential calculus, Gradient flow, Helfrich, Willmore, Bending energy, Biomembrane, Red blood cell

1. Introduction

Lipids consist of a hydrophilic head group and one or more hydrophobic hydrocarbon tails. When lipid molecules are immersed in aqueous environment at a proper concentration and temperature they spontaneously aggregate into a bilayer or membrane that forms an encapsulating bag called vesicle. This phenomenon is of interest in biology and biophysics because lipid membranes are ubiquitous in biological systems, and an understanding of vesicles provides an important element to understand real cells. Canhan and Helfrich [1, 2] were the first to introduce over 35 years ago, a model for the equilibrium shape of vesicles consisting of minimization of the *bending elasticity* or curvature energy. The structure of lipid membranes is that of a two dimensional, oriented, incompressible and viscous fluid. Phenomenological [1, 2] and rigorous continuum

mechanical [3, 4, 5] approaches agree that the membrane Γ is endowed with a bending or elastic energy. The simplest form of this energy is

$$\kappa \int_\Gamma h^2 + \kappa_G \int_\Gamma k,$$

where h and k are the mean and Gauss curvature respectively; and κ and κ_G are the constant bending coefficients. For closed surfaces without topological changes, the Gauss-Bonnet theorem [6, Section 8.5] yields the equivalence (up to scaling) between the above energy and the "Willmore" energy [7] defined by

$$W(\Gamma) := \frac{1}{2} \int_\Gamma h^2. \tag{1}$$

If the temperature and osmotic pressure of the vesicle do not change, the enclosed volume and surface area can be assumed to be conserved. The former is a consequence of the impermeability of the membrane. The latter is because the number of molecules remains fixed in each layer and the energetic cost of stretching or compressing the membrane is much larger than the cost of bending deformations. Refer to [8, 9, 10] for more details.

In this work we consider the Willmore energy model (1) with isoperimetric area and volume constraints. The combined effect of the bending elasticity with the surface and volume constraints generates a great variety of non-spherical shapes, in contrast to the characteristic spherical equilibrium shapes of simple liquids which are governed by isotropic surface tension. Describing the membrane by quantities all defined on the surface (energy, area and volume), equilibrium shapes are obtained as stationary states of a geometric evolution equation. For other aspects more related to the dynamics, the effect of the surrounding fluid should be taken into account. We study this effect in [11] and compare it with the geometric model.

Formally, the geometric model is a gradient flow for a suitable shape functional $J(\Gamma)$: find the evolution of the surface $\Gamma = \Gamma(t)$ such that its velocity $\boldsymbol{v}$ is given by

$$\langle \boldsymbol{v}, \boldsymbol{w} \rangle = -dJ(\Gamma; \boldsymbol{w}) \quad \forall \boldsymbol{w}, \tag{2}$$

where $dJ(\Gamma; \boldsymbol{w})$ is the shape derivative of $J(\Gamma)$ in the direction of $\boldsymbol{w}$ and $\langle \cdot, \cdot \rangle$ is a scalar product determining the type of flow [12].

The shape derivative of the Willmore energy (1) in three dimensions is given by

$$dW(\Gamma; \boldsymbol{w}) = \int_\Gamma \left(-\Delta_\Gamma h - \frac{1}{2}h^3 + 2kh \right) w, \tag{3}$$

where $w = \boldsymbol{\nu} \cdot \boldsymbol{w}$ is the normal component of $\boldsymbol{w}$. The L^2-gradient flow (i.e. $\langle \boldsymbol{v}, \boldsymbol{w} \rangle := \int_\Gamma \boldsymbol{v} \cdot \boldsymbol{w}$) obtained from (2) with $J = W$, namely using (3), is known as the Willmore flow and is a highly nonlinear 4-th order geometric partial differential equation (PDE) on $\Gamma(t)$. We refer to [13] for a general discussion of discrete gradient flows.

Parametric finite element methods (FEM) have already been proposed for the Willmore flow without constraints [14, 15] and with constraints [16]. A chief

difficulty is to make sense of Gauss curvature k within a variational framework. The scheme of Rusu [15] is the first of this class for (1) without constraints. That of Dziuk [14] copes with undesirable tangential motions observed in Rusu's scheme near equilibrium and presents a stability estimate for special initial conditions. In both cases, the formulation involves vector quantities (position and curvature). In contrast, Garcke et al [16] present a scalar scheme for (1) with constraints and evolve the interface in the direction of an averaged normal. The latter is somewhat related to the method of Bänsch et al for surface diffusion [17]. All these schemes are implemented with piecewise linear elements and exhibit difficulties to start; they are due to geometric inconsistency, a new concept that we discuss briefly in Section 4.4 and fully in [18]. Alternative techniques are also available in the literature, for instance the phase field approach [19, 20], threshold dynamics [21] and level set method [22]. A key advantage of our method is to have full access to the geometry at hand and be able to approximate curvature *semi-implicitly* (and so stably with modest timestep constraints of geometric nature). Moreover, the number of degrees of freedom associated with the parametric approach is that of a 2D problem, whereas for the phase field or level set methods a full 3D problem is to be solved, perhaps with the help of adaptive meshes or narrow band methods. These advantages are at the expense of difficulties in executing topological changes, especially in 3D.

In this paper we give a rather concise derivation of a novel vector formulation for (1) with constraints that hinges on shape differential calculus [23, 12]. In fact, we derive the following vector form of the shape derivative (Theorem 3.1)

$$
\begin{aligned}
dW(\Gamma; \boldsymbol{\phi}) = {} & \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} \cdot \nabla_\Gamma \boldsymbol{h} \\
& - \int_\Gamma (\nabla_\Gamma \boldsymbol{x} + \nabla_\Gamma \boldsymbol{x}^\mathsf{T}) \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \boldsymbol{h} + \frac{1}{2} \int_\Gamma \operatorname{div}_\Gamma \boldsymbol{h} \operatorname{div}_\Gamma \boldsymbol{\phi},
\end{aligned}
\tag{4}
$$

where $\boldsymbol{x}$ is the variable of integration or, with a slight abuse of notation, is also the identity over Γ. Since (4) is variational, it is the basis of a new parametric FEM with C^0-elements. We prefer quadratic isoparametric elements to linear elements, and discuss the reasons in Section 4.5. We evolve the computational domain at each time step via a semi-implicit Euler method; this is similar to [17, 16, 11, 24, 25, 13, 26, 14] and is discussed in Section 4.1.

The contributions of this paper are as follows:

- We derive the novel variational formulation (4) and corresponding parametric FEM. The derivation, being based on concepts from shape differential calculus, is rather concise.

- We implement the scheme using quadratic isoparametric elements. Previous parametric FEM use piecewise linears instead. We show the benefits of quadratics and explore further their use for curvature driven flows in [18].

- We do not need an explicit initial approximation to mean curvature to initialize the discrete Willmore flow (see (4)). This is rather tricky for piecewise linear FEM, due to geometric inconsistency, and a serious drawback of previous parametric approaches.

3

- We present a new and efficient algorithm to enforce the discrete isoperimetric volume and area constraints to machine precision. This hinges on a suitable Newton iteration.

- We introduce three computational tools, namely time and space adaptivity and mesh smoothing, which are important enhancements for large deformations and problems with disparate space-time scales. The use of mesh smoothing is indispensable for piecewise linears but not much so for quadratics.

- We include several numerical three dimensional simulations which lead to equilibrium shapes such as dumbbells, red cells, toroids, and exhibit large deformations. The evolutions as well as final shapes are rather stable.

The outline of the paper is as follows. In Section 2 we briefly introduced concepts from tensor algebra, tangential calculus and shape differential calculus. In Section 3 we present the continuous model for the biomembrane and derive a new variational formulation. We discretize this formulation in Section 4 after briefly introducing the finite element method for surfaces. In Section 5 we briefly discuss computational tools employed to enhance mesh quality and speed up computations. Finally, in Section 6 we present several numerical simulations.

2. Shape and Tangential Differential Calculus

Hereafter Γ denotes a compact, oriented, smooth d-hypersurface in $\mathbb{R}^{d+1}$ without boundary. The outer unit normal vector field is well defined and denoted by $\boldsymbol{\nu}$. Given a vector function $\boldsymbol{w}$ defined on Γ its normal component is denoted by $w := \boldsymbol{w} \cdot \boldsymbol{\nu}$.

Consider $\boldsymbol{v}$ a smooth vector field defined in $\mathcal{V} \subset \mathbb{R}^{d+1}$, a domain containing Γ. Then the surface $\Gamma = \Gamma(0)$ is deformed through an autonomous system of ODEs prescribed by $\boldsymbol{v}$. We denote the resulting sequence of perturbed surfaces by $\{\Gamma(t)\}_{t \geq 0}$. Under these considerations let J be a shape functional that assigns to each surface Γ a scalar value. Then the shape derivative of J in the direction of the vector field $\boldsymbol{v}$ is defined as the limit $dJ(\Gamma; \boldsymbol{v}) = \lim_{t \to 0} \frac{1}{t}(J(\Gamma(t)) - J(\Gamma))$. We are interested in shape functionals of the form

$$J(\Gamma) = \int_\Gamma \psi(x, \Gamma),$$

where the function ψ depends not only on the position but also on the geometry of Γ; for instance $\psi(x, \Gamma) = \frac{1}{2}h^2$ in (1).

For smooth surfaces, the signed distance function b is well defined in a neighborhood $\mathcal{V}$ of Γ. This provides a canonical extension to functions defined on Γ and is the basis of a simple differential calculus on Γ which uses the Euclidean differential calculus in the ambient neighborhood $\mathcal{V}$ of Γ [27, 12, 23].

Let f and $\boldsymbol{v}$ be C^1 scalar and vector valued functions defined on Γ and F and $\boldsymbol{V}$ canonical extensions to $\mathcal{V}$. Then we can define the tangential gradient of f as the orthogonal projection of ∇F to the tangent hyperplane of Γ

$$\nabla_\Gamma f := (\mathbf{I} - \boldsymbol{\nu} \otimes \boldsymbol{\nu}) \nabla F|_\Gamma.$$

4

This definition is independent of the extension . The tangential divergence of $\boldsymbol{v}$ is defined by $\operatorname{div}_\Gamma \boldsymbol{v} := \operatorname{tr} \nabla_\Gamma \boldsymbol{v}$ while the surface gradient of a vector and the surface divergence of a tensor are taken by columns. Finally for $C^2(\Gamma)$ functions or vectors, the surface Laplacian or Laplace Beltrami operator is defined by $\Delta_\Gamma \cdot := \operatorname{div}_\Gamma (\nabla_\Gamma \cdot)$.

The *second fundamental form* is $\nabla_\Gamma \boldsymbol{\nu}$. This symmetric tensor has the eigenvector $\boldsymbol{\nu}$ with zero eigenvalue. The remaining eigenvalues are the *principal curvatures* $\kappa_1, \ldots, \kappa_d$. We denote by $h := \kappa_1 + \cdots + \kappa_d$ the *total mean curvature*, by $k := \kappa_1 \cdots \kappa_d$ the *Gauss curvature*, and by $\boldsymbol{h} := h\boldsymbol{\nu}$ the *vector total mean curvature*. Using the previous differential operator definitions the following geometric identities follow [23, Ch. 8]:

$$\begin{aligned} \nabla_\Gamma \boldsymbol{x} = (\mathbf{I} - \boldsymbol{\nu} \otimes \boldsymbol{\nu}), \qquad &- \Delta_\Gamma \boldsymbol{x} = h\boldsymbol{\nu}, \\ \operatorname{div}_\Gamma \boldsymbol{\nu} = h, \qquad &\Delta_\Gamma \boldsymbol{\nu} = -|\nabla_\Gamma \boldsymbol{\nu}|^2 \boldsymbol{\nu} + \nabla_\Gamma h, \end{aligned} \tag{5}$$

where $\boldsymbol{x}$ is the identity function. Also observe that

$$\operatorname{div}_\Gamma \boldsymbol{v} = \operatorname{tr}(\nabla_\Gamma \boldsymbol{v}) = \mathbf{I} : \nabla_\Gamma \boldsymbol{v} = \nabla_\Gamma \boldsymbol{x} : \nabla_\Gamma \boldsymbol{v}.$$

We can extend some useful classical product rules (see [30, p. 30]) to tangential vector calculus: if f, $\boldsymbol{v}$ and $\mathbf{S}$ are $C^1(\Gamma)$ scalar, vector and tensor valued fields, then

$$\nabla_\Gamma (f\boldsymbol{v}) = f\nabla_\Gamma \boldsymbol{v} + \nabla_\Gamma f \otimes \boldsymbol{v}, \tag{6}$$

$$\operatorname{div}_\Gamma (f\boldsymbol{v}) = f \operatorname{div}_\Gamma \boldsymbol{v} + \boldsymbol{v} \cdot \nabla_\Gamma f, \tag{7}$$

$$\operatorname{div}_\Gamma (\mathbf{S}\boldsymbol{v}) = \mathbf{S} : \nabla_\Gamma \boldsymbol{v} + \boldsymbol{v} \cdot \operatorname{div}_\Gamma \mathbf{S}, \tag{8}$$

$$\operatorname{div}_\Gamma (f\mathbf{S}) = f \operatorname{div}_\Gamma \mathbf{S} + \mathbf{S}^\mathsf{T} \nabla_\Gamma f, \tag{9}$$

$$\nabla_\Gamma (\boldsymbol{v} \cdot \boldsymbol{w}) = \nabla_\Gamma \boldsymbol{v} \boldsymbol{w} + \nabla_\Gamma \boldsymbol{w} \boldsymbol{v}. \tag{10}$$

Tangential Stokes and Green formulas can be obtained (see [23, Ch. 8]). If f and $\boldsymbol{v}$ are $C^1(\Gamma)$ scalar and vector field, then

$$\int_\Gamma f \operatorname{div}_\Gamma \boldsymbol{v} + \int_\Gamma \nabla_\Gamma f \cdot \boldsymbol{v} = \int_\Gamma f\boldsymbol{v} \cdot \boldsymbol{h}. \tag{11}$$

3. Geometric Model for Biomembranes

The equilibrium shapes of biomembranes are stationary points of an L^2-gradient flow of an augmented bending energy J that accounts for the area and volume constraints. Let $A(\Gamma) := \int_\Gamma 1$ be the area of $\Gamma = \Gamma(t)$ and $V(\Gamma) := \frac{1}{d+1} \int_\Gamma \boldsymbol{x} \cdot \boldsymbol{\nu}$ be its enclosed volume, where $\boldsymbol{x}$ denotes the identity function on the domain of integration. Therefore, if λ, p are the area and volume Lagrange multipliers, the augmented functional J reads

$$J(\Gamma, \lambda, p) := \frac{1}{2} \int_\Gamma h^2 + \lambda \left(\int_\Gamma 1 - \int_{\Gamma^0} 1 \right) + \frac{p}{d+1} \left(\int_\Gamma \boldsymbol{x} \cdot \boldsymbol{\nu} - \int_{\Gamma^0} \boldsymbol{x} \cdot \boldsymbol{\nu} \right),$$

where Γ^0 is a given initial surface. For more details on imposing constraints using Lagrange multipliers see [31]. We now state the *Helfrich* problem, or equivalently the geometric flow for biomembrane modeling. Let G_T be the *trajectory space*

$$G_T := \{(\boldsymbol{x}, t) : \boldsymbol{x} \in \Gamma(t), t \in [0, T]\},$$

and $\mathbf{x} : G^T \to \mathbb{R}^{d+1}$ be the unique function defined by the property that $\mathbf{x}(\boldsymbol{x}, t) = \boldsymbol{x}$ for all $\boldsymbol{x} \in \Gamma(t)$, $t \in [0, T]$. We call $\mathbf{x}$ the *trajectory* of the evolution.

Problem 3.1 (Weak Helfrich flow). Given an initial surface Γ^0 and a final time $T > 0$, find the trajectory $\mathbf{x} : G_T \to \mathbb{R}^{d+1}$, the area multiplier $\lambda : [0, T] \to \mathbb{R}$ and the volume multiplier $p : [0, T] \to \mathbb{R}$ such that $\Gamma(0) = \Gamma^0$ and for all $t \in (0, T]$

$$\int_{\Gamma(t)} \dot{\mathbf{x}} \cdot \boldsymbol{\phi} = -dJ(\Gamma(t); \boldsymbol{\phi}) := -dW(\Gamma(t); \boldsymbol{\phi}) - \lambda \, dA(\Gamma(t); \boldsymbol{\phi}) - p \, dV(\Gamma(t); \boldsymbol{\phi}) \quad (12)$$

for all smooth $\boldsymbol{\phi} : \Gamma(t) \to \mathbb{R}^{d+1}$ and supplemented by the conservation relations

$$A(\Gamma(t)) = A(\Gamma(0)), \quad V(\Gamma(t)) = V(\Gamma(0)). \quad (13)$$

The shape derivatives of the area, volume and bending energy functionals are given by (see [12])

$$dA(\Gamma; \boldsymbol{\phi}) = \int_{\Gamma} h\phi \qquad dV(\Gamma; \boldsymbol{\phi}) = \int_{\Gamma} \phi, \quad (14)$$

$$dW(\Gamma, \boldsymbol{\phi}) = \int_{\Gamma} \nabla_{\Gamma}\phi \cdot \nabla_{\Gamma}h - \int_{\Gamma} h|\nabla_{\Gamma}\boldsymbol{\nu}|^2\phi + \frac{1}{2}\int_{\Gamma} h^3\phi. \quad (15)$$

Integrating (15) by parts, and using that $|\nabla_{\Gamma}\boldsymbol{\nu}|^2 = -h^2 + 2k$ in three dimensions [12], the Euler-Lagrange equation for J is given by

$$-\Delta_{\Gamma}h - \frac{1}{2}h^3 + 2kh + \lambda h + p = 0.$$

The familiar equations for W without constraints follows from $\lambda = p = 0$ [7].

Since neither the normal $\boldsymbol{\nu}$ nor the second fundamental form $\nabla_{\Gamma}\boldsymbol{\nu}$ are well defined on a globally C^0 piecewise polynomial surface, we now eliminate $\boldsymbol{\nu}$ as an explicit quantity in (15). This is achieved as in [14, 15], by allowing general variations of the manifold Γ, not necessarily in the normal direction, and using the total mean curvature $\boldsymbol{h} = h\boldsymbol{\nu}$ as a variable instead of h. However, our derivation being based on shape differential calculus is rather concise. First note that (14) give rise to the following vector form of the area and volume shape derivatives:

$$dA(\Gamma; \boldsymbol{\phi}) = \int_{\Gamma} \boldsymbol{h} \cdot \boldsymbol{\phi}, \qquad dV(\Gamma; \boldsymbol{\phi}) = \int_{\Gamma} \boldsymbol{\nu} \cdot \boldsymbol{\phi}. \quad (16)$$

The following Theorem establishes an equivalent formulation of the shape derivative (15) of Willmore energy, which does not explicitly involve the normal $\boldsymbol{\nu}$.

6

Theorem 3.1 (Vector form of dW). *For all smooth vector functions $\boldsymbol{\phi}$, we have*

$$dW(\Gamma; \boldsymbol{\phi}) = \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \boldsymbol{h} - \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} (\nabla_\Gamma \boldsymbol{x} + \nabla_\Gamma \boldsymbol{x}^\mathsf{T}) : \nabla_\Gamma \boldsymbol{h} + \frac{1}{2} \int_\Gamma \operatorname{div}_\Gamma \boldsymbol{h} \operatorname{div}_\Gamma \boldsymbol{\phi}.$$

Since $\nabla_\Gamma \boldsymbol{x} = \nabla_\Gamma \boldsymbol{x}^\mathsf{T} = \mathbf{I} - \boldsymbol{\nu} \otimes \boldsymbol{\nu}$ for as long as $\boldsymbol{x}$ stands for the identity over Γ, whether Γ is discrete or not, the symmetrization above is immaterial. However, writing the middle term implicitly would break the symmetry and could potentially be useful to reduce tangential motions. The proof of Theorem 3.1 requires the two following lemmas. The first one deals with the term $\int_\Gamma \nabla_\Gamma \phi \cdot \nabla_\Gamma h$ in (15).

Lemma 3.2. *For all smooth vector functions $\boldsymbol{\phi}$, we have the identity*

$$
\begin{aligned}
\int_\Gamma \nabla_\Gamma \phi \cdot \nabla_\Gamma h = {} & \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \boldsymbol{h} \\
& - \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} (\nabla_\Gamma \boldsymbol{x} + \nabla_\Gamma \boldsymbol{x}^\mathsf{T}) : \nabla_\Gamma \boldsymbol{h} - \int_\Gamma h \Delta_\Gamma \boldsymbol{\nu} \cdot \boldsymbol{\phi}.
\end{aligned}
\tag{17}
$$

PROOF. We first observe that $\nabla_\Gamma \boldsymbol{\nu}\, \boldsymbol{h} = \nabla_\Gamma \boldsymbol{\nu}^\mathsf{T} \boldsymbol{h} = 0$ because $\nabla_\Gamma \boldsymbol{\nu}$ is symmetric and $\boldsymbol{h}$ is normal. In view of (10), we obtain $\nabla_\Gamma h = \nabla_\Gamma (\boldsymbol{h} \cdot \boldsymbol{\nu}) = \nabla_\Gamma \boldsymbol{h} \boldsymbol{\nu}$. Then

$$
\begin{aligned}
\nabla_\Gamma \phi \cdot \nabla_\Gamma h = {} & \nabla_\Gamma (\boldsymbol{h} \cdot \boldsymbol{\nu}) \cdot \nabla_\Gamma (\boldsymbol{\phi} \cdot \boldsymbol{\nu}) \\
= {} & \nabla_\Gamma \boldsymbol{h} \boldsymbol{\nu} \cdot \nabla_\Gamma \boldsymbol{\phi} \boldsymbol{\nu} + \nabla_\Gamma \boldsymbol{h} \boldsymbol{\nu} \cdot \nabla_\Gamma \boldsymbol{\nu} \boldsymbol{\phi} \\
= {} & \nabla_\Gamma \boldsymbol{\phi} (\boldsymbol{\nu} \otimes \boldsymbol{\nu}) : \nabla_\Gamma \boldsymbol{h} + \nabla_\Gamma h \cdot \nabla_\Gamma \boldsymbol{\nu} \boldsymbol{\phi},
\end{aligned}
$$

where we have used again (10). We next employ (11) to integrate by parts the last term of the previous formula

$$\int_\Gamma \nabla_\Gamma h \cdot \nabla_\Gamma \boldsymbol{\nu} \boldsymbol{\phi} = - \int_\Gamma h \operatorname{div}_\Gamma (\nabla_\Gamma \boldsymbol{\nu} \boldsymbol{\phi}) = - \int_\Gamma h \nabla_\Gamma \boldsymbol{\nu} : \nabla_\Gamma \boldsymbol{\phi} - \int_\Gamma h \Delta_\Gamma \boldsymbol{\nu} \cdot \boldsymbol{\phi}.$$

In light of (6), we can write $\nabla_\Gamma \boldsymbol{h} = \nabla_\Gamma (h\boldsymbol{\nu}) = h \nabla_\Gamma \boldsymbol{\nu} + \nabla_\Gamma h \otimes \boldsymbol{\nu}$, whence

$$\int_\Gamma \nabla_\Gamma h \cdot \nabla_\Gamma \boldsymbol{\nu} \boldsymbol{\phi} = - \int_\Gamma \nabla_\Gamma \boldsymbol{\phi} : \nabla_\Gamma \boldsymbol{h} + \int_\Gamma (\nabla_\Gamma h \otimes \boldsymbol{\nu}) : \nabla_\Gamma \boldsymbol{\phi} - \int_\Gamma h \Delta_\Gamma \boldsymbol{\nu} \cdot \boldsymbol{\phi}.$$

Since we have shown that $\nabla_\Gamma h = \nabla_\Gamma \boldsymbol{h} \boldsymbol{\nu}$, using tensor product identities we get

$$(\nabla_\Gamma h \otimes \boldsymbol{\nu}) : \nabla_\Gamma \boldsymbol{\phi} = (\nabla_\Gamma \boldsymbol{h} \boldsymbol{\nu} \otimes \boldsymbol{\nu}) : \nabla_\Gamma \boldsymbol{\phi} = \nabla_\Gamma \boldsymbol{\phi} (\boldsymbol{\nu} \otimes \boldsymbol{\nu}) : \nabla_\Gamma \boldsymbol{h}.$$

Inserting this expression into the previous one, and using $2(\mathbf{I} - \boldsymbol{\nu} \otimes \boldsymbol{\nu}) = (\nabla_\Gamma \boldsymbol{x} + \nabla_\Gamma \boldsymbol{x}^\mathsf{T})$ as given by (5), leads to the asserted equality (17).

The second lemma provides an alternative expression for $\int_\Gamma h \Delta_\Gamma \boldsymbol{\nu} \cdot \boldsymbol{\phi}$ in (17).

Lemma 3.3. *For all smooth vector functions $\boldsymbol{\phi}$, we have the identity*

$$- \int_\Gamma h \Delta_\Gamma \boldsymbol{\nu} \cdot \boldsymbol{\phi} = \int_\Gamma h |\nabla_\Gamma \boldsymbol{\nu}|^2 \phi - \frac{1}{2} \int_\Gamma \nabla_\Gamma h^2 \cdot \boldsymbol{\phi}, \tag{18}$$

where $\phi = \boldsymbol{\nu} \cdot \boldsymbol{\phi}$ is the normal component of $\boldsymbol{\phi}$.

7

PROOF. From (5) it follws that $\boldsymbol{\phi} = \phi\boldsymbol{\nu} + \nabla_\Gamma\boldsymbol{x}\,\boldsymbol{\phi}$ and

$$h\Delta_\Gamma\boldsymbol{\nu}\cdot\boldsymbol{\phi} = -h|\nabla_\Gamma\boldsymbol{\nu}|^2\phi + h\nabla_\Gamma h\cdot\nabla_\Gamma\boldsymbol{x}\,\boldsymbol{\phi}.$$

As $\nabla_\Gamma\boldsymbol{x}$ is symmetric, and is the identity for tangent vectors, we see that

$$h\nabla_\Gamma h\cdot\nabla_\Gamma\boldsymbol{x}\,\boldsymbol{\phi} = \frac{1}{2}\nabla_\Gamma h^2\cdot\nabla_\Gamma\boldsymbol{x}\,\boldsymbol{\phi} = \frac{1}{2}\nabla_\Gamma\boldsymbol{x}\nabla_\Gamma h^2\cdot\boldsymbol{\phi} = \frac{1}{2}\nabla_\Gamma h^2\cdot\boldsymbol{\phi},$$

and the asserted expression (18) follows.

We are now in a position to prove Theorem 3.1.

PROOF OF THEOREM 3.1 Inserting (17) and (18) into (15) yields

$$dW(\Gamma;\boldsymbol{\phi}) = \int_\Gamma \nabla_\Gamma\boldsymbol{\phi} : \nabla_\Gamma\boldsymbol{h} - \int_\Gamma \nabla_\Gamma\boldsymbol{\phi}(\nabla_\Gamma\boldsymbol{x} + \nabla_\Gamma\boldsymbol{x}^\mathsf{T}) : \nabla_\Gamma\boldsymbol{h}$$
$$+ \frac{1}{2}\left(\int_\Gamma h^2\boldsymbol{h}\cdot\boldsymbol{\phi} - \int_\Gamma \nabla_\Gamma h^2\cdot\boldsymbol{\phi}\right).$$

In view of (11) we have $\int_\Gamma h^2\boldsymbol{h}\cdot\boldsymbol{\phi} = \int_\Gamma \operatorname{div}_\Gamma(h^2\boldsymbol{\phi})$. Moreover, invoking (7) we deduce

$$\operatorname{div}_\Gamma(h^2\boldsymbol{\phi}) = h^2\operatorname{div}_\Gamma\boldsymbol{\phi} + \nabla_\Gamma h^2\cdot\boldsymbol{\phi},$$
$$\operatorname{div}_\Gamma\boldsymbol{h} = \operatorname{div}_\Gamma(h\boldsymbol{\nu}) = \nabla_\Gamma h\cdot\boldsymbol{\nu} + h\operatorname{div}_\Gamma\boldsymbol{\nu} = h^2,$$

because $\operatorname{div}_\Gamma\boldsymbol{\nu} = h$. Consequently

$$\int_\Gamma h^2\boldsymbol{h}\cdot\boldsymbol{\phi} = \int_\Gamma \operatorname{div}_\Gamma\boldsymbol{h}\operatorname{div}_\Gamma\boldsymbol{\phi} + \int_\Gamma \nabla_\Gamma h^2\cdot\boldsymbol{\phi},$$

which implies the assertion.

4. Discrete Geometric Schemes for Biomembranes

We now present a discrete version of the Helfrich flow (12). We start with a semi-implicit Euler method for time stepping. We continue with the notion of isoparametric finite elements on surfaces. Together they yield the fully discrete scheme. We conclude with the algorithm for computing Lagrange multipliers.

4.1. Time Discretization

Consider the time partition $t_0 = 0 < \cdots < t_{M+1} = T$ of $[0, T]$, with time step $\tau_n := t_{n+1} - t_n$, for $n = 0, \ldots, M$. For advancing the surface from Γ^n to Γ^{n+1}, we resort to the same semi-implicit Euler method as in [32, 17, 16, 33, 24, 25, 13, 26, 14, 15]:

$$\mathbf{x}(\cdot, t_{n+1}) \approx \boldsymbol{X}^{n+1} := \boldsymbol{x} + \tau_n\boldsymbol{V}^{n+1}, \tag{19}$$

where $\boldsymbol{x}$ is the identity over Γ^n and $\boldsymbol{V}^{n+1}$ is an approximation of the velocity $\boldsymbol{v}(., t_{n+1})$. Note that both, $\boldsymbol{X}^{n+1}$ and $\boldsymbol{V}^{n+1}$ are approximation at time t_{n+1}

8

but are defined over Γ^n. The vector field $\boldsymbol{X}^{n+1}$ is an implicit description of the new surface Γ^{n+1} according to the following time discretization of (12)

$$\frac{1}{\tau_n} \int_{\Gamma^n} (\boldsymbol{X}^{n+1} - \boldsymbol{x}) \cdot \boldsymbol{\Phi} = -dJ^{n+1}(\Gamma^n; \boldsymbol{\Phi}) \quad \forall \boldsymbol{\Phi}, \tag{20}$$

and

$$\Gamma^{n+1} := \{ \boldsymbol{X}^{n+1}(\boldsymbol{x}) : \quad \forall \boldsymbol{x} \in \Gamma^n \}. \tag{21}$$

The time-discrete shape derivative $dJ^{n+1}(\Gamma^n)$ in (20) reads very much like expressions (16) for area and volume energies and Theorem 3.1 for Willmore energy, except that the geometry $(\Gamma, \boldsymbol{\nu}, \boldsymbol{x})$ is explicit at time t_n and the system is linear in the implicit unknowns $(\boldsymbol{X}^{n+1}, \boldsymbol{H}^{n+1})$. This is an effective *geometric linearization* of the highly nonlinear flow (12). Equation (21) means that the surface Γ^{n+1} is obtained as the image of $\boldsymbol{X}^{n+1}$, thereby preserving the connectivity of Γ^n. Consequently, topological changes are not admissible in this formulation. Equation (20) can be equivalently written in terms of $\boldsymbol{V}^{n+1} := \frac{1}{\tau_n}(\boldsymbol{X}^{n+1} - \boldsymbol{x})$ instead of $\boldsymbol{X}^{n+1}$. We prefer this choice because it helps make the exposition more coherent as this is the natural variable for membrane-fluid interaction [11]. So we will describe all our discrete schemes using $\boldsymbol{V}^{n+1}$ instead of $\boldsymbol{X}^{n+1}$.

In this vein, we now formulate a semidiscrete equation for curvature $\boldsymbol{H}^{n+1}$. The point of departure is the key geometric identity $-\Delta_\Gamma \boldsymbol{x} = \boldsymbol{h}$ written in (5). Combining this with (19) we get the following geometric PDE on Γ^n:

$$\boldsymbol{H}^{n+1} + \tau_n \Delta_{\Gamma^n} \boldsymbol{V}^{n+1} = -\Delta_{\Gamma^n} \boldsymbol{x}. \tag{22}$$

Equation (20) requires both $\boldsymbol{V}^{n+1}$ and $\boldsymbol{H}^{n+1}$. It is then natural to keep them separately, which entails an operator splitting of the underlying fourth order problem into two second order linear PDE on Γ^n.

4.2. Space Discretization

We recall the notion of finite elements on surfaces.

Definition 4.1 (Polyhedral surface). A pair $(\Gamma_h, \mathcal{T}_h)$ is a polyhedral surface if $\Gamma_h \subset \mathbb{R}^{d+1}$ and $\mathcal{T}_h$ is a finite family of closed, non degenerate, d-simplices in $\mathbb{R}^{d+1}$ such that:

- the intersection of two simplices in the family is either empty or a $(d-k)$-dimensional sub-simplex of both simplices with $k = 1, \ldots, d$, and
- $\Gamma_h = \bigcup_{K \in \mathcal{T}_h} K$.

Let $\hat{K} \subset \mathbb{R}^d$ be the master element, i.e. the convex hull spanned by the canonical basis $\{\boldsymbol{e}_i\}_{i=1}^d$. Given a d-simplex K in $\mathbb{R}^{d+1}$ let $\boldsymbol{F}_K : \hat{K} \to K$ be a injective affine map such that the vertices of $\hat{K}$ are mapped to the vertices of K.

Definition 4.2 (Piecewise polynomial surface). Given a polyhedral surface $(\Gamma_h, \mathcal{T}_h)$ and a positive integer m, let $\boldsymbol{F} : \Gamma_h \to \mathbb{R}^{d+1}$ be a C^0 map that leaves the vertices of Γ_h unchanged and such that $\boldsymbol{F}|_K$ is a polynomial of degree $\leq m$ for each $K \in \mathcal{T}_h$. Then the subordinate piecewise polynomial surface $\widetilde{\Gamma}_h$ of degree m is defined by $\widetilde{\Gamma}_h = \boldsymbol{F}(\Gamma_h)$.

A C^0-finite element space on $\widetilde{\Gamma}_h$ can then be defined as follows:

$$\mathbb{S}_h(\widetilde{\Gamma}_h) = \{\Phi \in C^0(\widetilde{\Gamma}_h) : \Phi \circ \widetilde{\boldsymbol{F}}_K \in \mathbb{P}^m(\hat{K}) \quad \forall \, K \in \mathcal{T}_h\},$$

where $\mathbb{P}^m(\hat{K})$ is the set of polynomial of degree $\leq m$ in d variables and $\widetilde{\boldsymbol{F}}_K = \boldsymbol{F} \circ \boldsymbol{F}_K$. We do not consider C^1-elements. We restrict both the surface and finite element space to have the same polynomial degree m for ; we discuss polynomial mismatch in [18].

Once the piecewise polynomial surface $\widetilde{\Gamma}_h^0$ approximating $\Gamma(0)$ has been defined, we can evolve it using the semi-implicit method of section 4.1. Since no confusion is possible, hereafter we simplify the notation, remove the tilde and simply denote by Γ_h^{n+1} the piecewise polynomial surface at time t_{n+1}. This surface is described by a vector field $\boldsymbol{X}^{n+1}$ defined over Γ^n, according to (21). Likewise, the finite element space becomes $\mathbb{S}_h^{n+1} = \mathbb{S}_h^{n+1}(\Gamma_h^n)$.

4.3. The Fully Discrete Scheme

We now couple the time discretization (20) and (22) with a space discretization using polynomial degree $m = 2$. We seek velocity and curvature $\boldsymbol{V}^{n+1}, \boldsymbol{H}^{n+1} \in \mathbb{S}_h^{n+1}$, and multipliers $p^{n+1}, \lambda^{n+1} \in \mathbb{R}$, satisfying

$$\int_{\Gamma_h^n} \boldsymbol{V}^{n+1} \cdot \boldsymbol{\Phi} = -dW_h^{n+1}(\Gamma_h^n; \boldsymbol{\Phi}) - \lambda^{n+1} dA_h^n(\Gamma_h^n; \boldsymbol{\Phi}) - p^{n+1} dV_h^n(\Gamma_h^n; \boldsymbol{\Phi}) \quad (23)$$

for all $\boldsymbol{\Phi} \in \mathbb{S}_h^{n+1}$ as well as

$$\int_{\Gamma_h^n} \boldsymbol{H}^{n+1} \cdot \boldsymbol{\Psi} - \tau_n \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \boldsymbol{V}^{n+1} : \nabla_{\Gamma_h^n} \boldsymbol{\Psi} = \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \boldsymbol{x} : \nabla_{\Gamma_h^n} \boldsymbol{\Psi}, \quad (24)$$

for all $\boldsymbol{\Psi} \in \mathbb{S}_h^{n+1}$. The parametric scheme (24) is due to Dziuk for the mean curvature flow [26]. It has been used more recently for other flows [32, 17, 16, 11, 18, 33, 24, 25, 13, 14, 34, 35, 15]. On the other hand, $dW_h^{n+1}(\Gamma_h^n; \boldsymbol{\Phi})$ approximates $dW(\Gamma; \boldsymbol{\Phi})$ in Theorem 3.1 and reads

$$dW_h^{n+1}(\Gamma_h^n; \boldsymbol{\Phi}) = \int_{\Gamma_h^n} \nabla_{\Gamma_h^n} \boldsymbol{H}^{n+1} : \nabla_{\Gamma_h^n} \boldsymbol{\Phi}$$

$$- \int_{\Gamma_h^n} (\nabla_{\Gamma_h^n} \boldsymbol{x} + \nabla_{\Gamma_h^n} \boldsymbol{x}^{\mathsf{T}}) \nabla_{\Gamma_h^n} \boldsymbol{\Phi} : \nabla_{\Gamma_h^n} \boldsymbol{H}^{n+1} \quad (25)$$

$$+ \frac{1}{2} \int_{\Gamma_h^n} \operatorname{div}_{\Gamma_h^n} \boldsymbol{H}^{n+1} \operatorname{div}_{\Gamma_h^n} \boldsymbol{\Phi},$$

whereas $dA_h^n(\Gamma_h^n; \boldsymbol{\Phi})$ and $dV_h^n(\Gamma_h^n; \boldsymbol{\Phi})$ approximate (16)

$$dA_h^n(\Gamma_h^n; \boldsymbol{\Phi}) = \int_{\Gamma_h^n} \boldsymbol{H}^n \cdot \boldsymbol{\Phi}, \qquad dV_h^n(\Gamma_h^n; \boldsymbol{\Phi}) = \int_{\Gamma_h^n} \boldsymbol{\Phi} \cdot \boldsymbol{\nu}. \quad (26)$$

Finally, we enforce the area and volume constraints

$$A(\Gamma_h^{n+1}) = A(\Gamma_h^n), \qquad V(\Gamma_h^{n+1}) = V(\Gamma_h^n).$$

We explain the actual computation of multipliers λ^{n+1}, p^{n+1} in Subsection 4.6.

4.4. Geometric Consistency

The chief geometric identity $\boldsymbol{h} = -\Delta_\Gamma \boldsymbol{x}$ needs to be relaxed at the discrete level. According to [18] we say that a discrete triple $(\Gamma, \boldsymbol{X}, \boldsymbol{H})$ is geometrically consistent if it satisfies the weak relation:

$$\boldsymbol{X}, \boldsymbol{H} \in \mathbb{S}_h : \quad \int_\Gamma \boldsymbol{H} \cdot \boldsymbol{\Phi} = \int_\Gamma \nabla_\Gamma \boldsymbol{X} : \nabla_\Gamma \boldsymbol{\Phi} \quad \forall \boldsymbol{\Phi} \in \mathbb{S}_h \qquad (27)$$

However, if $\boldsymbol{X} = \boldsymbol{x}$ is the identity over Γ, then (27) may yield oscillations in both direction and magnitude for $\boldsymbol{H}$ and polynomial degree $m = 1$. Lack of convergence has already been reported in the literature [14, 16], whereas convergence of $\boldsymbol{H}$ with order $m - 1$ has been proved [34, 36]. We note that (19) decouples $\boldsymbol{X}^{n+1}$ and $\boldsymbol{x}$, the identity over Γ_h^n, and thereby makes (24) geometrically consistent.

The situation is even more dramatic when mesh modification is involved, because we have no access to the exact geometry, and leads to a new paradigm: *where to place the nodes to keep geometric consistency and accuracy.* To illustrate this issue, consider the example of Figure 1 with $m = 1$ (linear finite elements) in $2D$. Both the exact unit circle Γ and its piecewise linear approximation Γ_h are given (left). Bisection of every edge of Γ_h yields a piecewise linear vector pair $(\boldsymbol{X}, \boldsymbol{H})$ satisfying (27) with $\boldsymbol{H}$ much bigger in magnitude to 1 at the old vertices and with wrong orientation at the midpoints. This strange behavior is however consistent with the curvature of a smooth curve going through the nodes (right), but quite distinct from the unit circle (left). Therefore, the issue at stake is that placing the new nodes over Γ_h, namely overrelying on the approximate curve, has undesirable geometric effects.
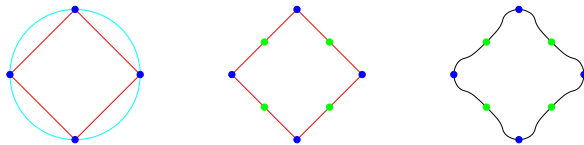
Figure 1: Placing the new nodes created by bisection on the curve is equivalent to replacing the circle being approximated (left) by a smooth curve passing through all the nodes (right). This yields geometric artifacts.

Similar geometric artifacts occur when doing mesh coarsening and smoothing, which together with mesh refinement are important ingredients to maintain a balanced computational effort and mesh quality for any polynomial degree $m \geq 1$. A remedy is briefly explored in Section 5 and in full detail in [18].

4.5. Linears vs Quadratics

The simulations in Section 6 are obtained using quadratics ($m = 2$). We now list their advantages against linears.

- *Initialization.* To start our scheme (24)-(23) we need a discrete representation of Γ_h^0 but not of $\boldsymbol{H}^0$. This appears to be an advantage of our method with respect to those in [16, 33, 14, 15] for Willmore energy. However, we also need $\boldsymbol{H}^0$ to deal with $dA_h^0(\Gamma_h^0; \Phi)$ on (26). We have noticed that a geometrically inconsistent initial triple $(\Gamma^0, \boldsymbol{X}^0, \boldsymbol{H}^0)$ may prevent ours, as well as the other methods, to start for $m = 1$. Dziuk proposes in [14] an intricate computation for the initial triple to be geometrically consistent. The issue disappears with quadratics $m = 2$. They can also be used to create a geometrically consistent triple for $m = 1$. In fact, we let $\boldsymbol{X}^0$ be the identity over Γ_h^0, compute $\boldsymbol{H}^0$ explicitly from (27), and finally interpolate at the linear nodes; the resulting triple is geometrically consistent and the method is practical. We have used this idea for initialization in Figures 2 and 3.

- *Time-steps.* The only time-step restriction of our approach, as well as those in [16, 33, 14, 15], is to avoid node crossing while advancing the surface from Γ_h^n to Γ_h^{n+1}. However, taking small time-steps may represent an almost explicit, and so geometrically inconsistent, computation of curvature $\boldsymbol{H}^{n+1}$ from (24). This undesirable effect has been observed for $m = 1$ and is responsible for initialization difficulties and numerical oscillations associated with mesh modification. The issue does not occur with $m = 2$.

- *Tangential motions.* Undesirable tangential motions, particularly close to equilibrium, have been reported [14, 15, 16]. Our method does exhibit such motions for $m = 1$ as documented in Figures 2 and 3. They may lead to severe mesh distortions and compromise simulations. The use of mesh smoothing resolves the problem as can be seem by comparing left and middle columns of Figure 2. More importantly, $m = 2$ bypasses the problem altogether without mesh smoothing (right column of Figure 2).

- *Mesh quality.* For the ellipsoid evolutions of Figure 2, quadratics maintain the mesh quality at a value roughly of 0.4 (right column) whereas linears with smoothing keep it at about 0.5 (mid column); see Figure 3. These values are excellent. This situation is rather typical in our simulations in Section 6, which run with $m = 2$ without mesh smoothing except for the large deformation of the twisted banana (see Figures 12 and 13). On the other hand, linears without smoothing blow up in finite time (Figure 2 left column).

4.6. Constraints

We present a method to compute the discrete Lagrange multipliers of (23). This is an instance of the more general method described in [37] to enforce discrete isoperimetric constraints to machine precision.

Since both $dA_h^n(\Gamma_h^n; \boldsymbol{\Phi})$ and $dV_h^n(\Gamma_h^n; \boldsymbol{\Phi})$ in (26) are explicit, then (23) and (24) can be rewritten as

$$\mathcal{E}(\boldsymbol{V}^{n+1}, \boldsymbol{H}^{n+1}) := \boldsymbol{F}_w + \lambda^{n+1}\boldsymbol{F}_a + p^{n+1}\boldsymbol{F}_v, \qquad (28)$$
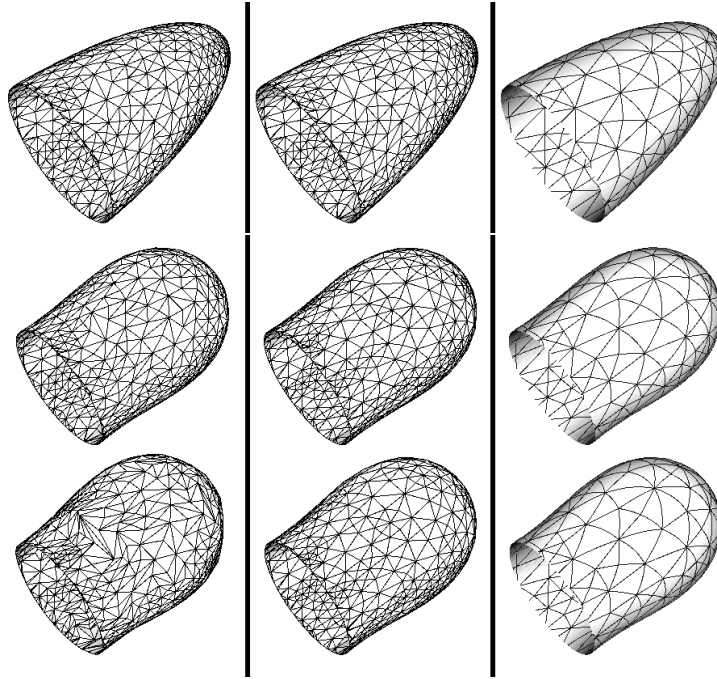
Figure 2: An initial ellipsoid of aspect ratio 4x1x1 is subject to a Helfrich flow. The mesh evolution under scheme (23) is shown using piecewise linears (left), piecewise linears with smoothing (center), and piecewise quadratics (right). The snapshots are taken at times $t = 0, 0.4, 1.4$. All three meshes have the same number of degrees of freedoms and the quadratic mesh has 4 times fewer elements. Using piecewise linears produces a blow up of the computed bending energy in finite time due to mesh distortion. This does not occur if mild mesh smoothing is used or if quadratic elements (without smoothing) are used. See Figure 3 for a more quantitative behavior.
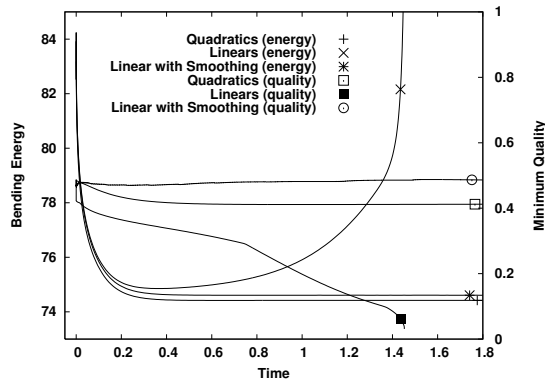
Figure 3: Energy (left scale) and mesh quality (right scale) as a function of time for the simulations described in Figure 2. For linear elements blow up occurs at $t \approx 1.4$ and correlates with mesh distortion. Quadratics conserve the mesh quality without smoothing.

13

where $\boldsymbol{F}_w$, $\boldsymbol{F}_a$, and $\boldsymbol{F}_v$ are the forcing functions in system (23)-(24) corresponding to energies W, A and V; note that $\mathcal{E}$ is linear in its two arguments. The idea of the method is to solve (28) by solving a similar system for three different right hand sides, and adjust the multipliers to verify the area and volume constraints

$$A(\Gamma_h^{n+1}) = A(\Gamma_h^n), \qquad V(\Gamma_h^{n+1}) = V(\Gamma_h^n). \tag{29}$$

More precisely, let $(\boldsymbol{V}_s{}^{n+1}, \boldsymbol{H}_s{}^{n+1})$ be the solution of

$$\mathcal{E}(\boldsymbol{V}_s{}^{n+1}, \boldsymbol{H}_s{}^{n+1}) = \boldsymbol{F}_s,$$

for $s = w, a, v$. Invoking the linearity of system (28) we have

$$\boldsymbol{V}^{n+1} = \boldsymbol{V}_w{}^{n+1} + \lambda^{n+1}\boldsymbol{V}_a{}^{n+1} + p^{n+1}\boldsymbol{V}_v{}^{n+1}$$

where λ^{n+1} and p^{n+1} are determined so that (29) is satisfied. Given $\boldsymbol{V}_w{}^{n+1}$, $\boldsymbol{V}_a{}^{n+1}$ and $\boldsymbol{V}_v{}^{n+1}$, the pair (λ^{n+1}, p^{n+1}) is a root of $f : \mathbb{R}^2 \to \mathbb{R}^2$ defined by

$$f(\lambda, p) := \begin{bmatrix} A(\Gamma_h(\lambda, p)) - A(\Gamma_h^n) \\ V(\Gamma_h(\lambda, p)) - V(\Gamma_h^n) \end{bmatrix}, \tag{30}$$

where $\Gamma_h(\lambda, p) := \Gamma_h^n + \tau\left(\boldsymbol{V}_w^{n+1} + \lambda\boldsymbol{V}_a^{n+1} + p\boldsymbol{V}_v^{n+1}\right)$. A quasi-Newton method can now be used to find the roots λ^{n+1} and p^{n+1} of f. The derivatives of f can be obtained via shape differential calculus.

Lemma 4.1 (Jacobian of f). *For τ sufficiently small, the differential of f in (30) is given by*

$$Df(\lambda, p) = \tau \begin{bmatrix} \int_{\Gamma_h} \mathrm{div}_{\Gamma_h} \boldsymbol{V}_a & \int_{\Gamma_h} \mathrm{div}_{\Gamma_h} \boldsymbol{V}_v \\ \int_{\Gamma_h} \boldsymbol{\nu} \cdot \boldsymbol{V}_a & \int_{\Gamma_h} \boldsymbol{\nu} \cdot \boldsymbol{V}_v \end{bmatrix},$$

where $\Gamma_h = \Gamma_h(\lambda, p)$.

PROOF. We define an extension $\tilde{\boldsymbol{V}}_s$ of $\boldsymbol{V}_s$ for $s = w, a, v$ in a tubular neighborhood $\mathcal{V}$ of Γ_h^n as follows:

$$\tilde{\boldsymbol{V}}_s(x) = \boldsymbol{V}_s(y_s), \quad \text{where} \quad y_s \in \Gamma_h^n \quad \text{satisfies} \quad y_s + \tau\boldsymbol{V}_s(y_s) = x \in \mathcal{V}.$$

Note that this extension is independent of λ and p and is well defined provided that τ is sufficiently small. Since

$$A(\Gamma_h(\lambda, p)) = \int_{\Gamma_h(\lambda, p)} 1, \quad V(\Gamma_h(\lambda, p)) = \int_{\Omega_h(\lambda, p)} 1,$$

invoking the shape derivatives (14), we deduce

$$\partial_\lambda A(\Gamma_h(\lambda, p)) = \tau \int_{\Gamma_h(\lambda, p)} \boldsymbol{H} \cdot \tilde{\boldsymbol{V}}_a = \tau \int_{\Gamma_h(\lambda, p)} \nabla_\Gamma \boldsymbol{X} : \nabla_\Gamma \boldsymbol{V}_a = \tau \int_{\Gamma_h(\lambda, p)} \mathrm{div}_\Gamma \tilde{\boldsymbol{V}}_a$$

and

$$\partial_\lambda V(\Gamma_h(\lambda, p)) = \tau \int_{\Gamma_h(\lambda, p)} \tilde{\boldsymbol{V}}_v \cdot \nu.$$

This concludes the proof.

14

Remark 4.1 (Initial guess). To derive a good initial guess, we recall that

$$A(\Gamma) = \frac{1}{d} \int_\Gamma \mathrm{div}_\Gamma \, \boldsymbol{x}, \qquad V(\Gamma) = \frac{1}{d+1} \int_\Gamma \boldsymbol{x} \cdot \boldsymbol{\nu},$$

and manipulate these formulas to approximate $A(\Gamma_h(\lambda, p))$ and $V(\Gamma_h(\lambda, p))$. To do so, we replace the domain of integration $\Gamma_h(\lambda, p)$ by Γ_h^n and the identity on $\Gamma_h(\lambda, p)$ by $\boldsymbol{x} + \tau \boldsymbol{V}^{n+1}$ to infer that

$$A(\Gamma_h(\lambda, p)) \approx \frac{\tau}{d} \left(\int_{\Gamma_h^n} \mathrm{div}_{\Gamma_h^n} \boldsymbol{V}_w + \lambda \int_{\Gamma_h^n} \mathrm{div}_{\Gamma_h^n} \boldsymbol{V}_a + p \int_{\Gamma_h^n} \mathrm{div}_{\Gamma_h^n} \boldsymbol{V}_v \right) + A(\Gamma_h^n)$$

$$V(\Gamma_h(\lambda, p)) \approx \frac{\tau}{d+1} \left(\int_{\Gamma_h^n} \boldsymbol{\nu} \cdot \boldsymbol{V}_w + \lambda \int_{\Gamma_h^n} \boldsymbol{\nu} \cdot \boldsymbol{V}_a + p \int_{\Gamma_h^n} \boldsymbol{\nu} \cdot \boldsymbol{V}_v \right) + V(\Gamma_h^n).$$

In view of (29) we impose that the terms in parenthesis vanish. If $\alpha_s = \int_{\Gamma_h^n} \mathrm{div}_{\Gamma_h^n} \boldsymbol{V}_s$ and $\beta_s = \int_{\Gamma_h^n} \boldsymbol{\nu} \cdot \boldsymbol{V}_s$ with $s = w, a, v$, then we can solve the following system for λ and p to get a good initial guess:

$$\begin{bmatrix} \alpha_a & \alpha_v \\ \beta_a & \beta_v \end{bmatrix} \begin{bmatrix} \lambda \\ p \end{bmatrix} = \begin{bmatrix} -\alpha_w \\ -\beta_w \end{bmatrix}.$$

Remark 4.2 (Implementation). If a direct solver is used to solve the system, then it is only necessary to do the factorization once and use it to solve all the systems. This implies that the computational cost of using this method to impose the constraints is basically the same as not having the constraints. On the other hand, if an iterative solver is used then the system has to be solved 3 times, but each system is independent from the others so a parallel implementation is very straightforward.

Remark 4.3 (Comparison with penalization). The previous method exhibits a couple of advantages over penalization: penalization is sensitive to the penalization parameter and the solution oscillates when reaching equilibrium. These drawbacks do not happen with the previous method.

5. Computational Tools

A drawback of the parametric FEM is the deterioration of mesh quality caused by tangential motion of nodes. Another issue is the disparate space-time scales typical of fourth order problems. In this section, we briefly describe numerical tools to circumvent these issues and speed up computations. Full details are given in [37]. As an illustration Figure 4 shows a view of the final mesh obtained for the flow of Figure 5 with and without space adaptivity.

5.1. Time Adaptivity

Time-step control for nonlinear time dependent fourth order problems is indispensable for two reasons:
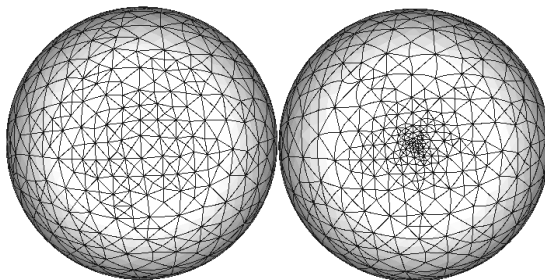
Figure 4: The initial ellipsoid of Figure 5 below was subject to a discrete Helfrich's flow. Starting with the same initial mesh two simulations were run, one with space adaptivity (left) and the other one without (right). The pictures show the spherical caps of the final equilibrium meshes (cf. 4th frame of Figure 5). Space adaptivity allows for the of a smaller number of degrees of freedoms and a consequent decrease on the computational time without any appreciable difference in the geometric quantities of interest.

- *Disparate time-scales.* It is typical that fast scales coexist with slow scales. We illustrate this with Example 6.3 for a twisted banana. At the beginning the time scale is fast relative to the interval of evolution, whereas it turns out to be relatively slow towards the end. Time-step control helps avoid negligible evolution.
- *Node crossing.* This is the only geometric constraint for the time-step but it is crucial for the integrity of the mesh and success of the entire simulation.

We employ the following algorithm, fully described in [37]. If ρ_K is the diameter of the largest inscribed ball in $K \in \mathcal{T}$, and z is a generic node of $\mathcal{T}$, then we let $d(z)$ be the nodal function that takes the minimum of ρ_K over all $K \in \mathcal{T}$ that share z. The quantity $\vartheta \frac{d(z)}{|V(z)|}$ gives the largest time-step to move node z without entangling the mesh, provided $\vartheta \leq 1/2$. Practice indicates that $\vartheta = 1/3$ is a good choice for linear meshes whereas $\vartheta = 1/6$ is the safe choice for quadratic meshes controlled by the hybrid method of [37].

5.2. Space Adaptivity and Geometric Consistency

The idea is to equidistribute the pointwise error, as proposed in [17] and studied in [36]; see also [38, 28, 29]. This maintains a balance between accuracy and number of degrees of freedom, and makes simulations affordable. This is achieved via atomic refinement and coarsening operations based on the domain pointwise error (see [37]). We use the maximum strategy, and refer to [39, 40] for details.

To avoid geometric artifacts due to geometric inconsistency, as explained in Section 4.4, we first interpolate H in the new mesh and next solve (27) for X. This procedure is stable, efficient, and keeps the existing geometric accuracy for any polynomial degree $m \geq 1$. We refer to [18] for a full discussion including proofs of these assertions.

5.3. Mesh Improvement

If a parametric FEM is used to discretize a geometric evolution, then it creates a discrete flow of the mesh. The mesh quality tends to deteriorate as the mesh gets transported. This effect is most pronounced for large domain deformations, and may lead to blow up in finite time. Figures 2 and 3 depict such a situation for linear FEM. It is thus significant to include mesh quality control as part of the design of a robust method. We implement a mesh optimization routine which operates on stars and selectively reallocates the centered node so as to:

- improve the star quality;
- preserve the shape of the approximate surface;
- maintain the local mesh size;
- produce negligible changes of the finite element functions defined on the mesh.

Our method, described in [37], consists of a hybrid affine-quadratic approach for isoparametric surfaces, which extends mesh smoothing techniques from linears to quadratics. More precisely, after mesh smoothing has been performed on the underlying simplicial mesh (the skeleton), we relocate the position of the quadratic midnodes but restrict it to lie not too far from the corresponding simplicial midnode. Mesh modification yields geometric inconsistencies, especially for linear elements, which can be cured with the algorithm in [18].

6. Numerical Results

In this section we present numerical simulations for the geometric biomembrane model of Problem 3.1 using the fully discrete scheme (24)-(23) for quadratics polynomials $m = 2$. We do not employ the computational tools of Section 5 except for the twisted banana, which exhibits very large domain deformations. These tools are indispensable for linears $m = 1$. The algorithm was coded in C using the adaptive finite element library ALBERTA [41, 40]. The description of each simulation is included in the caption of the corresponding figure. In all simulations the volume and surface area are preserved to machine precision.

Axisymmetric equilibrium shapes have been obtained in [42, 43] by reducing the problem to a system of ODEs. Our simulations preserve the axisymmetry during the evolution without imposing it. We refer to [16, 33, 14] for 3 dimensional simulations with piecewise linears with and without axial symmetry.

6.1. Dumbbell Bars

For axisymmetric initial ellipsoids it follows that 2 of its 3 axes are equal. Also as the bending energy is invariant under dilations what actually matters for the flow is the aspect ratio of the ellipsoid axes. When the third axis of the initial configuration is greater than about twice the length of the other axes, the evolution gives rise to dumbbell shapes. We present three simulations for

17

the aspects ratios of 8x1x1 (Figure 5), 4x1x1 (Figure 6) and 2x1x1 (Figure 7). The last one is not literally in the family but makes the transition phase to the "pill" shaped family through the sphere to the next family of Section 6.2.

We also include a non axisymmetric initial ellipsoid (Figure 8). The simulation reveals that at equilibrium it ends being axisymmetric.

6.2. Red Blood Cells

This family of shapes, also called discocytes, occurs when the third axis on the initial axisymmetric ellipsoid is less than about half the length of the other axes. We present two simulations for aspects ratios 3x3x1 (Figure 9) and 5x5x1 (Figure 10). Pinching is observed when the third axis is at least 5 times smaller than the others. For higher aspects ratios the upper and lower sides actually cross. This self-intersection is allowed by the parametric FEM, which is local in nature, but the evolution is meaningless for biomembrane modeling. The aspect ratio 5x5x1 yields an equilibrium shape which barely touches the other side (see Figure 10).

6.3. Twisted Banana

The twisted banana is interesting for bending flows because it is not axisymmetric and also has two different bendings in it (the banana bending plus the twist). The time-step adaptivity was crucial for this simulation as we detected two very different time scales along the evolution: the initial time-step is 10^{-6} whereas the total running time is $t = 5$. Moreover, the mesh smoothing algorithm is turn on to cope with very large deformations, First we introduce the shape (Figure 11) and next show the full simulation (Figure 12) and the two time scales (Figures 13 and 14).

6.4. Toroids

Figure 16 shows a nonsymmetric donut shape configuration. The stationary state is not the Clifford torus, as conjectured by Willmore, because the evolution is subject to area and volume constraints.

6.5. Conclusions

From the simulations presented in this chapter the following conclusions can be drawn:

- **Spherical caps**. For a geometric flow when the initial shape has distinctive ends (as in Simulations 6.1 and 6.3), the formation of spherical-shaped ends connected by a cylindrical neck seems to be the most effective way to minimize the energy. In fact, the formation of spherical-shaped ends decreases the energy several orders of magnitude more than straightening of a bended shape (see Simulation 6.3).

- **Red cells**. For a geometric flow when the initial shape is disc like (as Simulation 6.2) the evolution to decrease the energy is characterized by the thickening of the outer circular edge with the formation of a depression in the center.

18

8x1x1 Initial Ellipsoid

Time: 0.000000

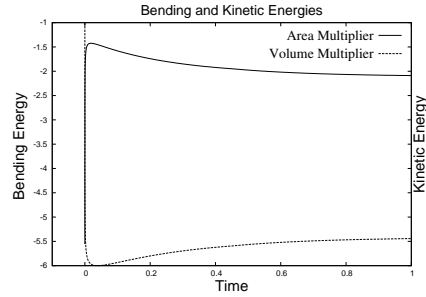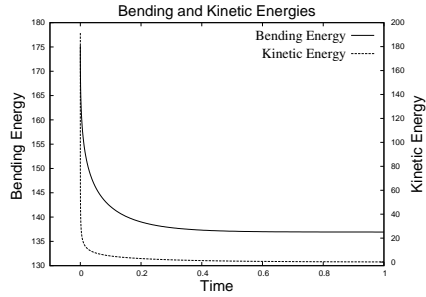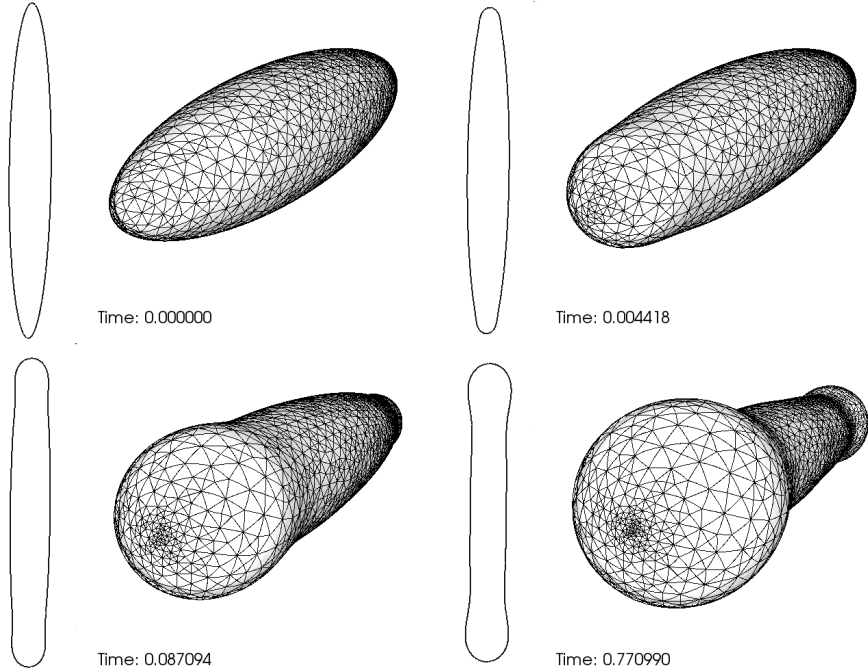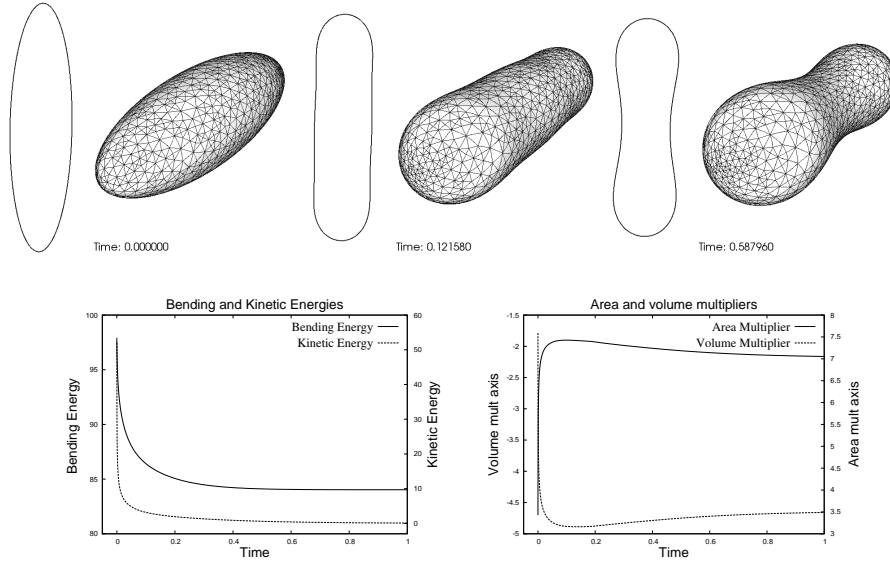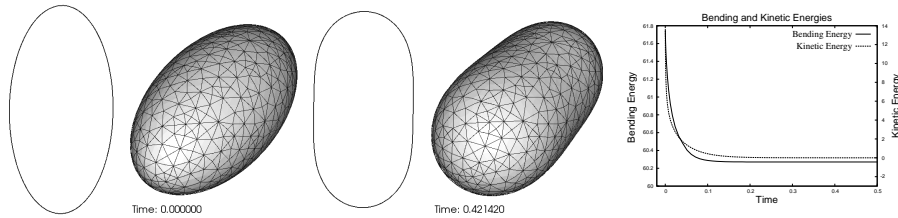Time: 0.004418

Time: 0.087094

Time: 0.770990

Figure 5: Evolution of an initial axisymmetric ellipsoid of aspect ratio 8x1x1. For each frame the picture on the right is a 3D view of the surface mesh whereas that on the left is a 2D cut through a symmetry plane. The evolution is characterized by the formation of spherical caps connected by a strongly cylindrical and long neck (dumbbell shape). The two bottom graphs show the evolution of the corresponding bending and kinetic energies (left) and the area and volume Lagrange multipliers (right). Notice that the bending energy approaches the equilibrium as the kinetic energy approaches 0. The bending energy for this aspect ratio is reduced approximately 22%.

4x1x1 Initial Ellipsoid

Figure 6: Evolution of an initial axisymmetric ellipsoid of aspect ratio 4x1x1. For each frame the picture on the right is a 3D view of the surface mesh and that on the left is a 2D cut through a symmetry plane. This evolution displays the formation of spherical caps connected by a relatively short neck, but still gives rise to a dumbbell shape (compare with Figure 5). The two bottom graphs show the evolution of the corresponding bending and kinetic energies (left) and the area and volume Lagrange multipliers (right). The bending energy for this aspect ratio is reduced approximately 14%.

2x1x1 Initial Ellipsoid

Figure 7: Evolution of an initial axisymmetric ellipsoid of aspect ratio 2x1x1. For each frame the picture on the left is a 3D view of the surface mesh and that on the right is a 2D cut through a symmetry plane. This evolution does not exhibit the formation of separate spherical caps but rather a pill shaped configuration. (compare with Figures 5 and 6). The two bottom graphs show the evolution of the corresponding bending and kinetic energies (left) and the area and volume Lagrange multipliers (right). The bending energy for this aspect ratio is reduced approximately 2%.

2x3x5 Initial Ellipsoid

Time: 0.000000   Time: 0.309340   Time: 4.576800
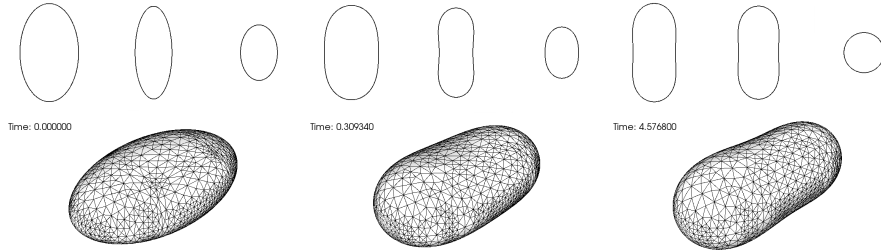
Figure 8: Evolution of a non-axisymmetric ellipsoid of aspect ratio 2x3x5. For each frame the picture on the bottom is a 3D view of the surface mesh and the pictures on the top from left to right are three cuts through the coordinate planes. The evolution seems to produce an axisymmetric equilibrium.
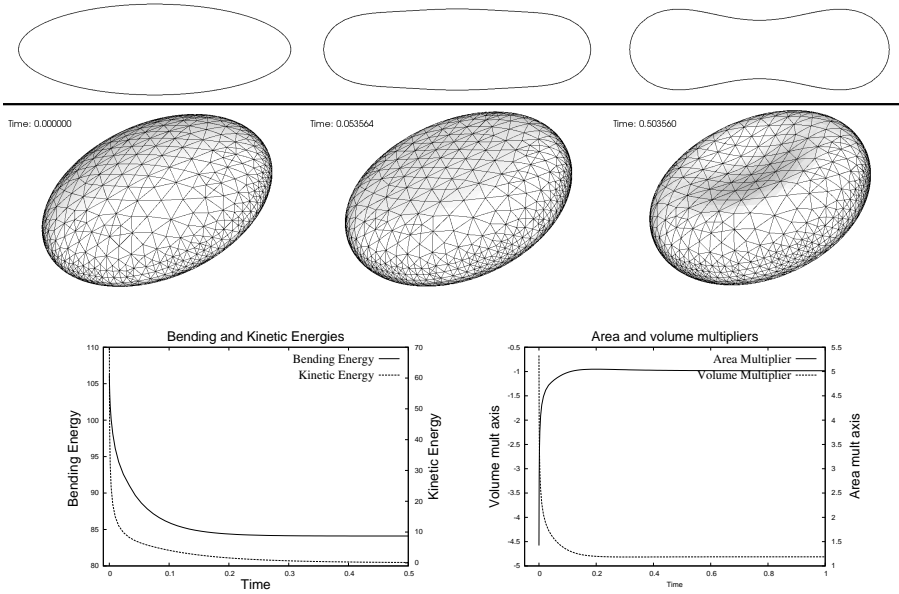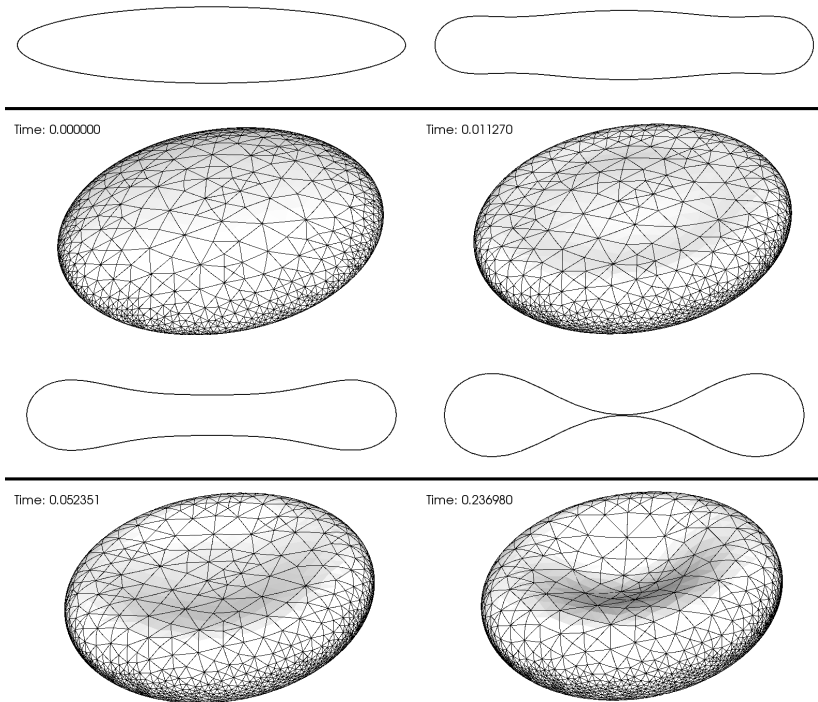
3x3x1 Initial Ellipsoid

Time: 0.000000   Time: 0.053564   Time: 0.503560

Figure 9: Evolution of an initial axisymmetric ellipsoid of aspect ratio 3x1x1. For each frame the picture on the bottom is a 3D view of the surface mesh and that on the top is a 2D cut through a symmetry plane. This evolution is characterized by the formation of a depression in the center together with a rounding and thickening of the outer circular edge (red blood cell). The two bottom graphs show the evolution of the corresponding bending and kinetic energies (left) and the area and volume Lagrange multipliers (right). The bending energy for this aspect ratio is reduced approximately 24%.

5x5x1 Initial Ellipsoid

Figure 10: Evolution of an initial axisymmetric ellipsoid of aspect ratio 5x5x1. For each frame the picture on the bottom is a 3D view of the surface mesh and that on the top is a 2D cut through symmetry plane. The equilibrium is characterized by the formation of an extreme depression of the center to the point of almost pinching (red blood cell). During the evolution the thickening of the outer circular edge occurs faster than the motion on the center, producing a depressed circular ring in between the outer edge and the center (second frame). This in turn is responsible for the appearance of a center bump instead of a depression. Later the evolution continues to squeeze this bump to a depression at the expense of more thickening and rounding of the outer circular edge.
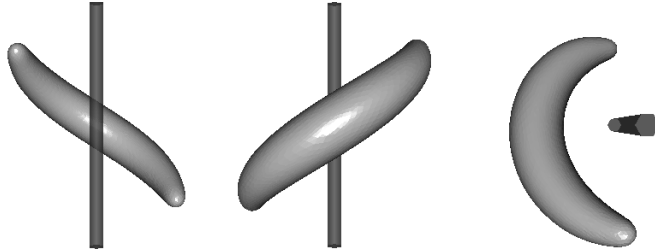
Figure 11: Initial configuration of the twisted banana. We see from left to right the front, back and top views of the initial twisted banana shape. To help visualize the shape it is rendered next to a thin right cylinder with its axis in the z direction.
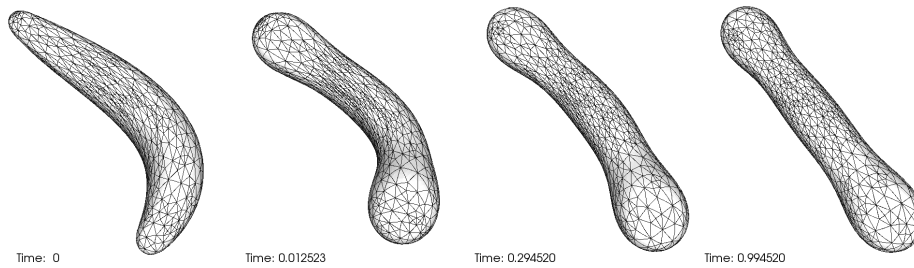

Figure 12: Evolution of the twisted banana shape of Figure 11. The frames show a 3D view of the surface mesh. The simulation was run in the time interval $[0, 5]$. The final equilibrium was reached at $t \approx 0.6$. Two different time scales associated to distinct geometric motions were observed. In Figures 13 and 14 we describe the two time scales leading to the second and forth frames respectively.
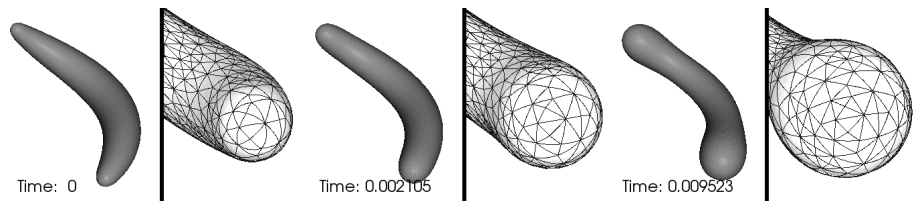
Figure 13: Time zooming to the interval $[0, 0.02]$ for the evolution of the twisted banana described in Figure 12. For each frame the picture on the left is a 3D view of the whole surface and that on the right is a 3D mesh view of the lower end suitably zoomed. This is the fast time scale of the simulation and is characterized by the formation of spherical caps and disappearance of the twist.
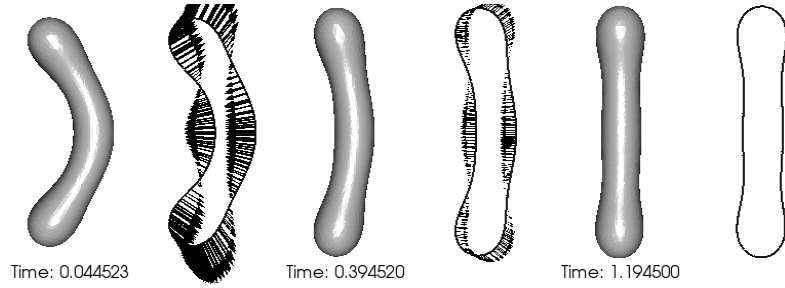
Figure 14: Time zooming to the interval $[0.04, 1.3]$ for the evolution of the twisted banana described in Figure 12. For each frame the picture on the left is a 3D view of the surface and that on the right is a 2D cut through a symmetry plane together with the velocity field depicted with arrows. Observe that the arrows disappeared in the last frame, when equilibrium is reached. This is the slow time scale of the evolution, and is characterized by the straightening of the boomerang-like shape of the fast time scale evolution (Figure 13).
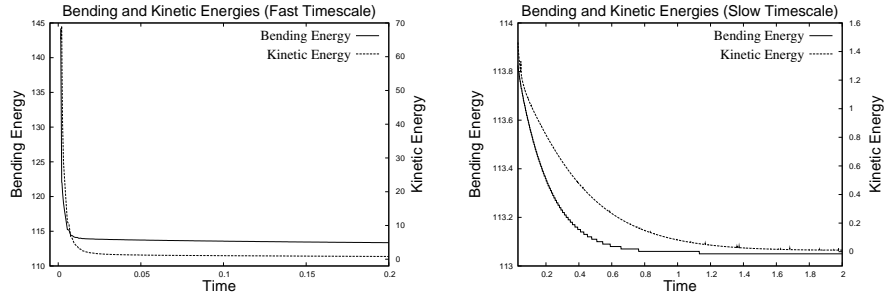

Figure 15: Kinetic and bending energy plot for the twisted banana simulation. Each graph is correspond to the time interval of the fast and slow scale of the simulation. For the full time scale of the simulation the energy drops appear quite abrupt. The spherical caps are completely formed in the interval $[0, 0.01]$ and the evolution falsely seems to reach a steady state. The bending the energy decreases about 21% in this time interval $t = 0.01$. After this the bending energy decreases only a tiny 0.3% in a relatively long time interval $t \approx 1.0$.
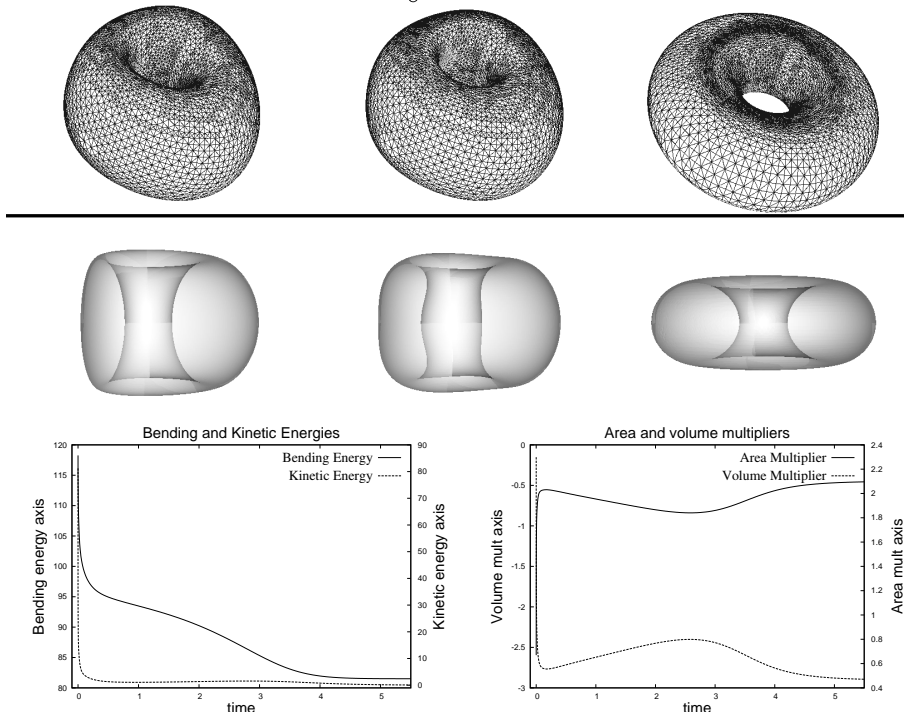
Initial Nonsymmetric Donut

Figure 16: Evolution of an initial nonsymmetric donut-shaped surface. Each frame shows a 3D transparent view of the surface (bottom) and the corresponding mesh (top). The evolution follows an asymmetric path and exhibits an energy decrease and multipliers behavior quite different from previous examples with genus 1 topology.

- **Exponential decay of kinetic energy**. The geometric flow shows a clear exponential decay of the kinetic energy when approaching the equilibrium shape. The rate of decay seems to depend in a non trivial way on the equilibrium shape.

Acknowledgments

References

[1] P. Canham, The minimum energy of bending as a possible explanation of the biconcave shape of the human red blood cell, Journal of Theoretical

25

Biology 26 (1) (1970) 61–81.

[2] W. Helfrich, Elastic properties of lipid bilayers - theory and possible experiments, Zeitschrift Fur Naturforschung C-A Journal Of Biosciences 28 (1973) 693.

[3] J. Jenkins, The equations of mechanical equilibrium of a model membrane, SIAM J. Appl. Math. 32 (4) (1977) 755–764.

[4] D. Steigmann, Fluid films with curvature elasticity, Arch. Ration. Mech. Anal. 150 (2) (1999) 127–152.

[5] D. Hu, P. Zhang, W. E, Continuum theory of a moving membrane, Phys. Rev. E (3) 75 (4) (2007) 041605, 11.

[6] S. Montiel, A. Ros, Curves and surfaces, Vol. 69 of Graduate Studies in Mathematics, American Mathematical Society, Providence, RI, 2005, translated and updated from the 1998 Spanish edition by the authors.

[7] T. J. Willmore, Riemannian geometry, Oxford Science Publications, The Clarendon Press Oxford University Press, New York, 1993.

[8] E. Evans, R. Skalak, Mechanics and thermodynamics of biomembranes, CRC Press.

[9] U. Seifert, Configurations of fluid membranes and vesicles, Advances in Physics 46 (1) (1997) 13–137.

[10] D. Steigmann, E. Baesu, R. Rudd, J. Belak, M. McElfresh, On the variational theory of cell-membrane equilibria, Interfaces Free Bound. 5 (4) (2003) 357–366.

[11] A. Bonito, R. Nochetto, M. Pauletti, Adaptive FEM for coupled fluid-membrane interaction problems. application to biomembranes, in preparation.

[12] J. Sokołowski, J.-P. Zolésio, Introduction to shape optimization, Vol. 16 of Springer Series in Computational Mathematics, Springer-Verlag, Berlin, 1992, shape sensitivity analysis.

[13] G. Doğan, P. Morin, R. Nochetto, M. Verani, Discrete gradient flows for shape optimization and applications, Comput. Methods Appl. Mech. Engrg. 196 (37-40) (2007) 3898–3914.

[14] G. Dziuk, Computational parametric willmore flow, Numer. Math. 111 (1) (2008) 55–80.

[15] R. E. Rusu, An algorithm for the elastic flow of surfaces, Interfaces Free Bound. 7 (3) (2005) 229–239.

[16] J. Barrett, H. Garcke, R. Nürnberg, Parametric approximation of willmore flow and related geometric evolution equations, preprint.

[17] E. Bänsch, P. Morin, R. Nochetto, A finite element method for surface diffusion: the parametric case, J. Comput. Phys. 203 (1) (2005) 321–343.

[18] A. Bonito, R. Nochetto, M. Pauletti, Geometrically consistent refinement, in preparation.

[19] Q. Du, C. Liu, X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, J. Comput. Phys. 198 (2) (2004) 450–468.

[20] Q. Du, C. Liu, X. Wang, Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions, J. Comput. Phys. 212 (2) (2006) 757–777.

[21] S. Esedoglu, S. Ruuth, R. Tsai, Threshold dynamics for high order geometric motions, Interfaces Free Bound. 10 (3) (2008) 263–282.

[22] M. Droske, M. Rumpf, A level set formulation for Willmore flow, Interfaces Free Bound. 6 (3) (2004) 361–378.

[23] M. C. Delfour, J.-P. Zolésio, Shapes and geometries, Vol. 4 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2001, analysis, differential calculus, and optimization.

[24] K. Deckelnick, G. Dziuk, Error analysis of a finite element method for the Willmore flow of graphs, Interfaces Free Bound. 8 (1) (2006) 21–46.

[25] K. Deckelnick, G. Dziuk, C. Elliott, Computation of geometric partial differential equations and mean curvature flow, Acta Numer. 14 (2005) 139–232.

[26] G. Dziuk, An algorithm for evolutionary surfaces, Numer. Math. 58 (6) (1991) 603–611.

[27] D. Gilbarg, N. Trudinger, Elliptic partial differential equations of second order, 2nd Edition, Vol. 224 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1983.

[28] K. Mekchay, Convergence of adaptive finite element methods, Ph.D. thesis, University of Maryland (2005).

[29] K. Mekchay, P. Morin, R. Nochetto, Afem for the laplace-beltrami operator: design and conditional contraction property, to appear.

[30] M. Gurtin, An introduction to continuum mechanics, Vol. 158 of Mathematics in Science and Engineering, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], New York, 1981.

[31] M. Giaquinta, S. Hildebrandt, Calculus of variations. I, Vol. 310 of Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Springer-Verlag, Berlin, 1996, the Lagrangian formalism.

[32] E. Bänsch, Finite element discretization of the Navier-Stokes equations with a free capillary surface, Numer. Math. 88 (2) (2001) 203–235.

[33] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, R. Rusu, A finite element method for surface restoration with smooth boundary conditions, Comput. Aided Geom. Design 21 (5) (2004) 427–445.

[34] C. Heine, Isoparametric finite element approximations of curvature on hypersurfaces, preprint from his webpage.

[35] O. Lakkis, R. Nochetto, A posteriori error analysis for the mean curvature flow of graphs, SIAM J. Numer. Anal. 42 (5) (2005) 1875–1898 (electronic).

[36] A. Demlow, Higher-order finite element methods and pointwise error estimates for elliptic problems on surfaces, SIAM J. Numer. Anal.To appear.

[37] A. Bonito, R. Nochetto, M. Pauletti, Computational tools for free boundary geometric flows, in preparation.

[38] A. Demlow, G. Dziuk, An adaptive finite element method for the Laplace-Beltrami operator on implicitly defined surfaces, SIAM J. Numer. Anal. 45 (1) (2007) 421–442 (electronic).

[39] I. Kossaczký, A recursive approach to local mesh refinement in two and three dimensions, J. Comput. Appl. Math. 55 (3) (1994) 275–288.

[40] A. Schmidt, K. Siebert, Design of adaptive finite element software, Vol. 42 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2005, the finite element toolbox ALBERTA, With 1 CD-ROM (Unix/Linux).

[41] D. Köster, O. Kriessl, K. Siebert, Design of finite element tools for coupled surface and volume meshes, Numer. Math. Theor. Meth. Appl. 1 (3) (2008)) 245–274.

[42] J. Jenkins, Static equilibrium configurations of a model red blood cell, J. Math. Biology 4 (2) (1977) 150–169.

[43] U. Seifert, K. Berndl, R. Lipowsky, Shape transformations of vesicles: Phase diagram for spontaneous- curvature and bilayer-coupling models, Phys. Rev. A 44 (2) (1991) 1182–1202.