Commun. Comput. Phys.
doi: 10.4208/cicp.170611.130112a

Vol. **13**, No. 2, pp. 325-360
February 2013

Computation of Two-Phase Biomembranes with Phase Dependent Material Parameters Using Surface Finite Elements

Charles M. Elliott and Björn Stinner*

Mathematics Institute and Centre for Scientific Computing, University of Warwick, Coventry CV4 7AL, United Kingdom.

Received 17 June 2011; Accepted (in revised version) 13 January 2012

Available online 28 June 2012

Abstract. The shapes of vesicles formed by lipid bilayers with phase separation are governed by a bending energy with phase dependent material parameters together with a line energy associated with the phase interfaces. We present a numerical method to approximate solutions to the Euler-Lagrange equations featuring triangulated surfaces, isoparametric quadratic surface finite elements and the phase field approach for the phase separation. Furthermore, the method involves an iterative solution scheme that is based on a relaxation dynamics coupling a geometric evolution equation for the membrane surface with a surface Allen-Cahn equation. Remeshing and grid adaptivity are discussed, and in various simulations the influence of several physical parameters is investigated.

AMS subject classifications: 49Q10, 65M60, 74S05, 74G15, 74L15, 82B26, 92C10

Key words: Biomembrane, surface finite elements, relaxation dynamics.

1 Introduction

The basic components of cell boundaries and organelles are bilayers consisting of phospholipids that spontaneously form when introduced into an aqueous environment in appropriate concentration. Vesicles or bags formed by such biomembranes show a great variety of shapes and have been attracting interest from various fields. First, the geometry and composition are conjectured to contribute to and interact with cell processes. Second, the lipid bilayers possess intricate mechanical properties which partially are solid-like, namely they reveal a stiffness against stretching and bending, but are unable to sustain

*Corresponding author. *Email addresses:* c.m.elliott@warwick.ac.uk (C. M. Elliott), bjorn.stinner@warwick.ac.uk (B. Stinner)

shear stress and so also behave like a viscous fluid within each of the monolayers. Mathematicians finally are attracted by the geometric properties of the membrane but also by the patterns that phases within the membrane may form. Such phase separation phenomena are due to the different types of lipids of which the membrane consists. Here we wish to consider computational simulations and modeling in the context of a partial differential equation model which couples the geometric equation for the membrane surface to a diffusion equation on the surface which defines the surface phase separation. We begin by describing the model.

The classical biomembrane mechanics theory developed in [10, 27, 32] models the vesicle boundary as a hypersurface on which the following elastic energy functional is defined:

$$\mathcal{F}_{CEH}(\Gamma)=\int_\Gamma \frac{k_\kappa}{2}(\kappa-\kappa_s)^2+\int_\Gamma k_g g. \tag{1.1}$$

Here, the mean curvature (sum of the principal curvatures) of the membrane is denoted by κ and the Gaussian curvature by g and $k_\kappa>0$ (bending rigidity) and k_g (Gaussian bending rigidity) are material dependent elasticity parameters while the number κ_s is known as the spontaneous curvature. A lateral phase separation due to a decomposition of the different lipid molecules has been observed and recently been made visible [5,6,43]. Line tension is observed at the phase interfaces and in [34,35] an energy functional of the form

$$\mathcal{F}_{SI}(\Gamma,\gamma)=\mathcal{F}_B(\Gamma,\gamma)+\mathcal{F}_L(\Gamma,\gamma)=\sum_{i=1}^{2}\Big(\int_{\Gamma_i}\frac{k_\kappa^{(i)}}{2}(\kappa-\kappa_s^{(i)})^2+\int_{\Gamma_i}k_g^{(i)}g\Big)+\int_\gamma\bar{\sigma} \tag{1.2}$$

was proposed. The two-phase membrane, $\Gamma=\Gamma_1\cup\Gamma_2\cup\gamma$, consists of two smooth, not necessarily connected surfaces Γ_i with a common boundary γ which is the phase interface. The constant parameter $\bar{\sigma}$ denotes the energy density of the interfacial energy. An intricate issue is the smoothness across the phase interface. As in [35], Section II.B, we assume a C^1 surface which means that the external unit normal of the enclosed vesicle domain is continuous. This assumption is motivated by the fact that the lipid bilayer should be intact across the interface. However, higher order derivatives in general are discontinuous. For instance, we will see that the mean curvature is subject to a jump condition in equilibrium. But we want to mention that C^0 surfaces may be considered [33], motivated from the pictures in [6] which, on a macroscopic scale, reveal kinks at the phase interfaces.

In order to deal with the line energy we consider a phase field approach and introduce an order parameter c to distinguish the two phases. The states $c=-1$ and $c=1$ then correspond to the two phases and the phase interfaces are replaced by thin layers across which c changes is value rapidly but smoothly. To achieve this we replace the line energy $\mathcal{F}_L$ by a Ginzburg-Landau energy of the form

$$\mathcal{F}_{GL}(\Gamma,c)=\int_\Gamma\sigma\Big(\frac{\varepsilon}{2}|\nabla_\Gamma c|^2+\frac{1}{\varepsilon}\psi(c)\Big). \tag{1.3}$$

The function $\psi(c):=(1-c^2)^2/2$ is a double-well potential with minima in $c=1$ and $c=-1$ which favors the two phases. Denoting by $\boldsymbol{\nu}$ the external unit normal to Γ and

by $\boldsymbol{P}:=\boldsymbol{I}-\boldsymbol{\nu}\otimes\boldsymbol{\nu}$ the projection to the tangential space, the surface gradient of the order parameter is defined by $\nabla_\Gamma c:=\boldsymbol{P}\nabla c$. The gradient term $\varepsilon|\nabla_\Gamma c|^2$ in (1.3) acts as a penalty term which avoids rapid changes of c whilst the term $\psi(c)/\varepsilon$ favors the values ±1 of the order parameter. Here ε is a small length scale such that the thickness of the interfacial layers scales with ε. Finally, the coefficient σ is related to the line energy coefficient $\bar{\sigma}$ by

$$\bar{\sigma}=\frac{4}{3}\sigma. \tag{1.4}$$

The phase field approach has been successfully applied in various fields, see [11, 14]. In the context of two-phase membranes, the idea of replacing the line energy by the Ginzburg-Landau energy dates back to the early 90s [1, 36, 42].

The phase dependent physical parameters in (1.2) become functions of the order parameter c and we choose them to be of the form

$$k_\kappa(c)=\begin{cases} k_\kappa^{(1)}, & \text{if } 1\le c,\\ \dfrac{k_\kappa^{(1)}+k_\kappa^{(2)}}{2}+\dfrac{k_\kappa^{(1)}-k_\kappa^{(2)}}{4}c(3-c^2), & \text{if } -1<c<1,\\ k_\kappa^{(2)}, & \text{if } c\le -1, \end{cases} \tag{1.5}$$

and analogously for $\kappa_s(c)$ and $k_g(c)$. Other sufficiently smooth monotone interpolations between -1 and 1 may be used. The total membrane energy in this diffuse interface model then reads

$$\begin{aligned}\mathcal{F}_{DI}(\Gamma,c)&=\mathcal{F}_{MC}(\Gamma,c)+\mathcal{F}_{GC}(\Gamma,c)+\mathcal{F}_{GL}(\Gamma,c)\\ &=\int_\Gamma\frac{1}{2}k_\kappa(c)\big(\kappa-\kappa_s(c)\big)^2+\int_\Gamma k_g(c)g+\int_\Gamma\Big(\frac{\sigma\varepsilon}{2}|\nabla_\Gamma c|^2+\frac{\sigma}{\varepsilon}\psi(c)\Big).\end{aligned} \tag{1.6}$$

In this study we discuss a numerical method to compute equilibrium two-phase membrane shapes, subject to certain side conditions. Our approach is based on the surface phase field model. That is we look for local minimizers of the energy (1.6) and consider ε as an approximation parameter. In the limit as $\varepsilon\to0$ we recover the sharp interface model 1.2, [25]. We use relaxation dynamics consisting of the equations (1.7a) and (1.7b) below which correspond to an L_2 gradient flow dynamics of the energy (1.6) where the variation of the energy is carefully derived in [25]:

$$\begin{aligned}v_{\boldsymbol{\nu}}=&-\Delta_\Gamma\big(k_\kappa(c)(\kappa-\kappa_s(c))\big)-|\nabla_\Gamma\boldsymbol{\nu}|^2k_\kappa(c)(\kappa-\kappa_s(c))+\frac{1}{2}k_\kappa(c)\big(\kappa-\kappa_s(c)\big)^2\kappa\\ &-\nabla_\Gamma\cdot\big(k_g'(c)(\kappa\boldsymbol{I}+\nabla_\Gamma\boldsymbol{\nu})\nabla_\Gamma c\big)+\sigma\varepsilon\nabla_\Gamma c\otimes\nabla_\Gamma c:\nabla_\Gamma\boldsymbol{\nu}+\sigma\Big(\frac{\varepsilon}{2}|\nabla_\Gamma c|^2+\frac{1}{\varepsilon}\psi(c)\Big)\kappa\\ &-\sum_{i\in\mathcal{I}}\lambda_i\delta_\Gamma\mathcal{C}_i,\end{aligned} \tag{1.7a}$$

$$\begin{aligned}\varepsilon\omega\partial_t^\bullet c=&-\frac{1}{2}\big(\kappa-\kappa_s(c)\big)^2k_\kappa'(c)+k_\kappa(c)\big(\kappa-\kappa_s(c)\big)\kappa_s'(c)-gk_g'(c)\\ &+\varepsilon\sigma\Delta_\Gamma c-\frac{\sigma}{\varepsilon}\psi'(c)-\sum_{i\in\mathcal{I}}\delta_c\mathcal{C}_i,\end{aligned} \tag{1.7b}$$

where v_ν is the normal velocity, $\omega>0$ a kinetic coefficient and $\partial_t^\bullet c=\partial_t c+v_\nu \boldsymbol{\nu}\cdot\nabla c$ is the material derivative of c along the trajectories defined by the velocity field $v_\nu\boldsymbol{\nu}$, $\boldsymbol{\nu}$ denoting the unit normal on Γ. Furthermore, the λ_i are Lagrange multipliers for possible constraints which here are denoted by $\mathcal{C}_i(\Gamma,c)=0$, $i\in\mathcal{I}$ where $\mathcal{I}$ is an abstract index set. We have constraints on the total area $|\Gamma|$, on integrals of the form $\int_\Gamma h(c)$ and on the volume of the domain enclosed by Γ in mind. The notation $\delta_\Gamma\mathcal{C}_i$ stands for the variation with respect to the surface Γ, and similarly $\delta_c\mathcal{C}_i$ is the variation with respect to c. Note that with respect to the time scale for the evolution of the membrane surface we have taken a time scale defined by ω for the evolution of the phase interface on the membrane surface because of the appearance of ε on the left hand side of (1.7b). This follows from the usual asymptotics for the Allen-Cahn equation.

Our numerical approach is based on approximating the membrane with a triangulated surface and the surface fields by finite element functions. A suitable time discretization leads to an iterative method to calculate equilibrium configurations as steady states of the time dependent relaxation dynamics which couples a geometric evolution equation of Willmore-flow type for the membrane surface Γ to an advected surface Allen-Cahn equation for the phase field variable c.

In order to do this we require a variational formulation of the equations. The following fundamental geometric equation

$$\Delta_\Gamma \boldsymbol{x}=\kappa\boldsymbol{\nu} \tag{1.8}$$

which links the mean curvature κ of the surface Γ to the identity map $\boldsymbol{x}:\Gamma\to\Gamma$ is extremely useful. Observing that we may write $v_\nu\boldsymbol{\nu}=(\partial_t\boldsymbol{x}\cdot\boldsymbol{\nu})\boldsymbol{\nu}$ for the velocity of Γ we easily see that the surface evolution equation (1.7a) in fact is a first order in time and a fourth order in space equation. But thanks to the splitting into two second order equations (1.7a) and (1.8) we are only required to use H^1 conforming finite element spaces in their variational formulation.

We now make some comments about our numerical approach and the contributions of this paper:

- Our method combines the efficiency of triangulated surfaces for approximating the motion of hypersurfaces with the topological flexibility and algorithmic convenience of the phase field method to deal with the lateral phase separation.

- The surface is approximated using a union of curvilinear triangles using a quadratic isoparametric description over a base polyhedral surface with flat triangular faces. In order to evolve the discrete surface from the known old time level to the new time level, we parameterize the surface at the new time level over the surface at the old time level so that no global parameterization over a fixed domain is required. This means, unless we remesh, we use a fixed connectivity of vertices for the triangulation of the surface and we calculate the new positions of the triangle nodes given the current ones. Besides gaining higher order approximation we use isoparametric quadratic surface finite elements

(SFEs) rather than linear finite elements as in [24] in order to obtain a better approximation of the shape operator (or Weingarten map) $\nabla_\Gamma \nu$. We refer to [15,31] for a discussion of the approximation of geometric quantities with Lagrange SFEs.

• We use a semi-implicit discretization in time where only the terms to highest order and the Lagrange multipliers are taken implicitly. It results in a linear system whose variables comprise the Lagrange multipliers and at each node of the triangulation the values of the position vector of the new triangulation, the approximated mean curvature and the value of the phase field variable. The resulting system has a linear saddle point structure for the nodal positions and mean curvature nodal values coupled to a nonlinear system involving the Lagrange multipliers and a linear system for the phase field nodal values. We solve by factorizing the main matrix associated with the linear operator for the variables other than the Lagrange multipliers, solve a few linear systems depending on the number of Lagrange multipliers, and then perform a quasi-Newton iteration to compute the solution including the Lagrange multipliers.

• The surface phase field equation is solved using a finite element method on the triangulated surface generated by the approximation of the geometric surface equations. Thus the method is a variant of the evolving surface finite element method introduced in [22].

• We use a formulation in which the nodes of each element move with both a normal and tangential velocity. This is in contrast to the method in [24] (based on [20]) for the membrane evolution which evolves vertices purely in the normal direction. Here the time discrete normal velocity is obtained by a consistent discretization of the geometric surface equations but it also generates motion in the tangential direction which has some mesh smoothing properties. This formulation is based on the ideas in [3]. This method is less prone to grid distortion than the approach presented in [24]. We have found that mesh smoothing becomes a more desirable feature of a numerical scheme when the deformations are very strong which may happen, for example, when the initial surface is far from an equilibrium. Nevertheless for large deformations this tangential vertex motion may not be sufficient. For this reason we employ the global remeshing method presented in [21].

• In this paper we consider phase dependent material parameters whereas [24] is more restrictive with respect to the material parameters by requiring constant k_κ and k_g as well as $\kappa_s = 0$.

• For efficiency and accuracy we adaptively refine and coarsen the grid to ensure that the interfacial layers associated with the phase field variable are resolved by the mesh and that the strongly curved regions contain sufficient numbers of nodes.

• Besides presenting a computational tool we use it to:

  – Study the influence of the line tension parameter.

  – Investigate the impact of having differing Gaussian curvature bending rigidities.

– Investigate the stabilization of multiple rafts on vesicles with a sphere like topology.

– Compute non-spherical vesicles with multiple domains.

Finally we refer to other computational work.

• In this paper we compute quite general configurations. However for axisymmetric vesicle equilibrium shapes ode solvers are sufficient [35]. In this work we use a formulation which allows the use of H^1 conforming finite elements whereas in [28, 38] H^2 conforming finite elements with C^1 ansatz functions are used to define a discrete energy which is minimized using a black box conjugate gradient solver. In some cases this led to distorted meshes. In some simulations in [38] this was overcome by taking additional mesh regularization energies into account but convergence of the associated regularizing forces to zero in the long term could not be guaranteed.

• One might envisage a purely sharp interface numerical approach in which the surface is approximated by a triangulation and the phase boundaries on the surface are approximated by evolving piecewise linear or curvilinear curves, c.f. [4]. The evolution would then be defined by the sharp interface equations formulated in [25]. It is an open challenge to formulate computational sharp interface approaches which will handle topological changes. On the other hand phase field equations or diffuse interface models are well established in the flat situation for approximating or modeling energetic interfaces and has been used to describe the membrane itself, see [41] (for a method for vesicles close to a sphere), [16, 17, 44] and [37]. This approach can handle topological changes of the surface itself which may be a useful feature. In such models it is important to understand how the phase field formulation handles the dependence of the energy on Gauss curvature as the integral of the Gauss curvature is no longer an invariant of the evolution. Also topological changes of the membrane itself require severe rearrangements of the lipid molecules so that the validity of continuum phase field models may be questioned for such phenomena.

For the problem under consideration, provided there is no topological change of the surface, since the problem of computing the hypersurface involves a higher dimension than computing the (one- dimensional) phase interfaces we deem it advantageous to use the parametric approach together with, for instance, surface finite elements, in order to approximate the surface. This requires substantially fewer degrees of freedom than the phase field approach or level set methods, see the discussion in [14]. In our approach we use a surface phase field model which is useful with respect to the intra-membrane phase separation since it allows the study the appearance and stability of configurations with multiple domains (or rafts) of one phase in another matrix phase.

We end the introduction with an outline of the paper. In the next section we present the variational formulation of the evolution problem and precisely state the constraints. Also the energy and Euler-Lagrange equations in the sharp interface limit as $\varepsilon \to 0$ are stated. We then introduce the SFEs and proceed with the spatial discretization in Section 3. Also mesh adaption and mesh regularity are discussed. In the final section we present

and discuss results of significant numerical experiments. This includes a convergence study in ε (including grid refinement), investigating the effect of varying the magnitude of the line tension and the Gaussian bending rigidities, the stabilization of multiple raft domains on sphere like vesicles and a study of non-sphere like vesicles. We indicate how the computational approach may be used to calculate phase diagrams.

For the implementation of the numerical schemes the finite-element software ALBERTA [39] has been employed, for solving the linear systems we have made use of the software UMFPACK [13], and for the visualization of vesicle shapes we used ParaView `http://paraview.org/`.

2 Modeling

2.1 Vesicles and constraints

We consider vesicles formed by lipid bilayers and define admissible phase field surfaces for the membrane energy (1.6) to be the smooth boundary $\Gamma=\partial\Omega$ of a bounded domain $\Omega\subset\mathbb{R}^3$ (the vesicle domain) together with a smooth field $c:\Gamma\to\mathbb{R}$, the order parameter or phase field variable.

The vesicle encloses a given volume denoted by V, such that the areas of the two phases are prescribed; we denote them by A_i, $i=1,2$ (we refer to [40], Section 2.4.4 for the physically relevant regime). In the context of the phase field model, the area constraints are taken into account with the help of the function

$$h(c)=\begin{cases}1, & \text{if } 1\leq c,\\ \frac{1}{2}c(3-c^2), & \text{if } -1<c<1,\\ -1, & \text{if } c\leq -1.\end{cases}$$

Again other monotone interpolations between -1 and 1 may be used. As $\varepsilon\to 0$ one expects that $\int_\Gamma h(c)\to|\Gamma_1|-|\Gamma_2|$ where Γ_i, $i=1,2$, are the domains of the two phases in the sharp interface limit, see Section 2.3. This motivates the conservation of $\int_\Gamma h(c)$ and $|\Gamma|=|\Gamma_1|+|\Gamma_2|$ instead of $|\Gamma_1|$ and $|\Gamma_2|$. We remark that this approach has been successfully applied previously in the context of Allen-Cahn systems on flat domains, c.f. [29]. The constraints read

$$\mathcal{C}_A(\Gamma,c)=0, \tag{2.1a}$$
$$\mathcal{C}_c(\Gamma,c)=0 \tag{2.1b}$$

in terms of the functionals

$$\mathcal{C}_A(\Gamma,c):=|\Gamma|-(A_1+A_2),\qquad \mathcal{C}_c(\Gamma,c):=\int_\Gamma h(c)-(A_1-A_2). \tag{2.2}$$

Defining the functional

$$\mathcal{C}_V(\Gamma,c):=|\Omega|-V=\frac{1}{3}\int_\Gamma \boldsymbol{x}\cdot\boldsymbol{\nu}-V \tag{2.3}$$

the volume constraint reads

$$\mathcal{C}_V(\Gamma,c)=0. \tag{2.4}$$

2.2 Gradient flow dynamics

Calculation of the variation of the energy (1.6) and the constraint functionals (2.2) and (2.3) is based on deforming the membrane and changing the order parameter along the trajectories associated with the deformation. Given a smooth normal vector field $\boldsymbol{w}=w\boldsymbol{\nu}$: $\Gamma\to\mathbb{R}^3$ and a smooth function $\eta:\Gamma\to\mathbb{R}$ we set

$$\begin{aligned}&\Gamma(\tau):=\big\{\boldsymbol{x}(\tau):=\boldsymbol{x}+\tau w(\boldsymbol{x})\boldsymbol{\nu}(\boldsymbol{x})|\boldsymbol{x}\in\Gamma\big\},\\&c(\tau):\Gamma(\tau)\to\mathbb{R},\quad c(\tau,\boldsymbol{x}(\tau)):=c(\boldsymbol{x})+\tau\eta(\boldsymbol{x}).\end{aligned}$$

The variation of the membrane energy is defined by

$$\big\langle\delta\mathcal{F}_{DI}(\Gamma,c),(w,\eta)\big\rangle=\frac{d}{d\tau}\mathcal{F}_{DI}\big(\Gamma(\cdot),c(\cdot)\big)\Big|_{\tau=0}$$

and similarly for the constraint functionals $\mathcal{C}_V$, $\mathcal{C}_A$ and $\mathcal{C}_c$.

We now define the relaxation dynamics by a weighted L^2 gradient flow. We consider the inner product

$$\mathcal{M}_\omega\big((v,\chi),(w,\eta);(\Gamma,c)\big):=\int_\Gamma\big(vw+\varepsilon\omega\chi\eta\big),$$

where $\omega>0$ is a kinetic coefficient. It yields a time scale which may speed up or slow down the phase separation in comparison with the membrane surface relaxation. The scaling in ϵ is such that the resulting Allen-Cahn equation for the order parameter c on the evolving surface Γ approximates a forced geodesic curvature in the sharp interface limit as $\varepsilon\to0$ in analogy to the phase field approximation of mean curvature flow in flat domains, [14,29].

Problem 2.1 (Gradient Flow Dynamics)**.** Suppose that an initial phase field surface (Γ^0,c^0), Γ^0 enclosing a domain Ω^0, is given such that $|\Omega^0|=V$, $|\Gamma^0|=A_1+A_2$ and $\int_{\Gamma^0}h(c^0)=A_1-A_2$. Find a family $\{(\Gamma(t),c(t))\}_{t\in[0,\infty)}$ with $(\Gamma(0),c(0))=(\Gamma^0,c^0)$ and functions $\lambda_i:[0,\infty)\to\mathbb{R}$, $i\in\mathcal{I}$, such that at each time $t\in[0,\infty)$

$$\begin{aligned}&\mathcal{M}_\omega\big(\big(v_{\boldsymbol{\nu}}(t),\partial_t^\bullet c(t)\big),\big(w,\eta\big);\big(\Gamma(t),c(t)\big)\big)\\&=-\big\langle\delta\mathcal{F}_{DI}(\Gamma(t),c(t)),(w,\eta)\big\rangle-\sum_{i\in\mathcal{I}}\lambda_i(t)\big\langle\delta\mathcal{C}_i(\Gamma(t),c(t)),(w,\eta)\big\rangle\end{aligned} \tag{2.5}$$

for all deformations (w,η) of $(\Gamma(t),c(t))$ and such that $0=\mathcal{C}_i(\Gamma(t),c(t))$, $i\in\mathcal{I}$, at each time $t\in[0,\infty)$.

The variations of the contributions to $\mathcal{F}_{DI}$ and of the constraints can be computed as presented in [25] and we confine ourselves to just stating the resulting formulae:

$$\begin{aligned}\big\langle \delta\mathcal{F}_{MC}(\Gamma,c),(w,\eta)\big\rangle &= \int_\Gamma \Big(\frac{1}{2}(\kappa-\kappa_s(c))^2 k_\kappa'(c) - k_\kappa(c)(\kappa-\kappa_s(c))\kappa_s'(c)\Big)\eta \\ &\quad + \int_\Gamma \Big(\Delta_\Gamma\big(k_\kappa(c)(\kappa-\kappa_s(c))\big) + |\nabla_\Gamma \boldsymbol{\nu}|^2 k_\kappa(c)(\kappa-\kappa_s(c))\Big)w \\ &\quad + \int_\Gamma \Big(-\frac{1}{2}k_\kappa(c)(\kappa-\kappa_s(c))^2\kappa\Big)w, \end{aligned} \tag{2.6a}$$

$$\big\langle \delta\mathcal{F}_{GC}(\Gamma,c),(w,\eta)\big\rangle = \int_\Gamma (g k_g'(c))\eta + \int_\Gamma \Big(\nabla_\Gamma\cdot\big(k_g'(c)(\kappa \boldsymbol{I} + \nabla_\Gamma\boldsymbol{\nu})\nabla_\Gamma c\big)\Big)w, \tag{2.6b}$$

$$\begin{aligned}\big\langle \delta\mathcal{F}_{GL}(\Gamma,c),(w,\eta)\big\rangle &= \int_\Gamma \Big(-\sigma\varepsilon\Delta_\Gamma c + \frac{\sigma}{\varepsilon}\psi'(c)\Big)\eta \\ &\quad + \int_\Gamma \Big(-\sigma\varepsilon\nabla_\Gamma c\otimes\nabla_\Gamma c : \nabla_\Gamma\boldsymbol{\nu} - \sigma\Big(\frac{\varepsilon}{2}|\nabla_\Gamma c|^2 + \frac{1}{\varepsilon}\psi(c)\Big)\kappa\Big)w, \end{aligned} \tag{2.6c}$$

$$\big\langle \delta\mathcal{C}_V(\Gamma,c),(w,\eta)\big\rangle = \int_\Gamma w, \tag{2.6d}$$

$$\big\langle \delta\mathcal{C}_A(\Gamma,c),(w,\eta)\big\rangle = \int_\Gamma -\kappa w, \tag{2.6e}$$

$$\big\langle \delta\mathcal{C}_c(\Gamma,c),(w,\eta)\big\rangle = \int_\Gamma h'(c)\eta - h(c)\kappa w. \tag{2.6f}$$

The strong equations (1.7a), (1.7b) emerge directly from (2.5).

2.3 Related sharp interface model

An asymptotic analysis of the stationary equations of (1.7a) and (1.7b) with constraints has been performed in [25]. For completeness we state the limiting problem as $\varepsilon\to 0$.

Admissible membranes surfaces $\Gamma=\Gamma_1\cup\gamma\cup\Gamma_2$ for the membrane energy $\mathcal{F}_{SI}$ are specified below (1.2). Recall that Γ_1 and Γ_2 are smooth, not necessarily connected hypersurfaces with smooth boundaries coinciding with γ which consists of a finite number of smooth curves and locally around γ the surface Γ can be parametrized by a C^1 map.

Limits of quantities on γ that may be discontinuous carry an upper index of the form $^{(1)}$ or $^{(2)}$ depending on whether γ is approached from Γ_1 or Γ_2 and by $[\cdot]_{(1)}^{(2)} = (\cdot)^{(2)} - (\cdot)^{(1)}$ we denote the jump across γ. Furthermore, by $\boldsymbol{\mu}$ we denote the outer co-normal of Γ_2 and by $\boldsymbol{\tau}$ a unit tangential vector field along γ such that $(\boldsymbol{\tau},\boldsymbol{\mu},\boldsymbol{\nu})$ is positively oriented. Fig. 1 gives an impression of the situation where γ consists of one curve only and Γ topologically is a sphere.

In a tube around γ, these vector fields $(\boldsymbol{\tau},\boldsymbol{\mu},\boldsymbol{\nu})$ may be extended to Γ and then are continuous across γ since Γ is a C^1 surface. The curvature quantities $h_\nu := -\boldsymbol{\tau}\cdot\nabla_\Gamma\boldsymbol{\nu}\boldsymbol{\tau}$ and $h_d := -\boldsymbol{\mu}\cdot\nabla_\Gamma\boldsymbol{\nu}\boldsymbol{\tau}$ are continuous but $h_p := -\boldsymbol{\mu}\cdot\nabla_\Gamma\boldsymbol{\nu}\boldsymbol{\mu}$ may be discontinuous across γ. Let us also remark that $\kappa = h_\nu + h_p$ and $g = h_\nu h_p - h_d^2$.

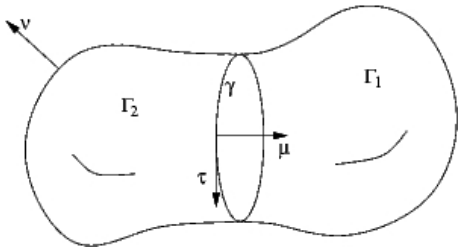

Figure 1: Sketch of a two-phase membrane with vectors $\boldsymbol{\tau}$, $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$ as occurring in the shape interface model in Section 2.3.

Problem 2.2 (Sharp Interface Equilibrium Equations)**.** Find a two-phase membrane $\Gamma = \Gamma_1 \cup \gamma \cup \Gamma_2$ and find Lagrange multipliers λ_V, $\lambda_A^{(1)}$ and $\lambda_A^{(2)}$ such that

$$0 = k_\kappa^{(i)} \Delta_{\Gamma_i} \kappa + k_\kappa^{(i)} |\nabla_{\Gamma_i} \boldsymbol{\nu}|^2 (\kappa - \kappa_s^{(i)}) - \frac{1}{2} k_\kappa^{(i)} (\kappa - \kappa_s^{(i)})^2 \kappa + \lambda_V - \lambda_A^{(i)} \kappa, \quad \text{on } \Gamma_i, \quad i = 1,2, \tag{2.7a}$$

$$0 = \left[k_\kappa (h_\nu + h_p - \kappa_s) + k_g h_\nu \right]_{(1)}^{(2)}, \qquad \text{on } \gamma, \tag{2.7b}$$

$$0 = -\left[k_\kappa \nabla_\Gamma (h_\nu + h_p) \right]_{(1)}^{(2)} \cdot \boldsymbol{\mu} + \left[k_g \right]_{(1)}^{(2)} \nabla_\gamma h_d \cdot \boldsymbol{\tau} - \bar{\sigma} h_\nu, \qquad \text{on } \gamma, \tag{2.7c}$$

$$\begin{aligned} 0 = &\left[\frac{k_\kappa}{2} (h_\nu + h_p - \kappa_s)^2 + k_g (h_\nu h_p - h_d^2) \right]_{(1)}^{(2)} \\ &- \left[(k_\kappa (h_\nu + h_p - \kappa_s) + k_g h_\nu) h_p \right]_{(1)}^{(2)} - \bar{\sigma} h_g + \left[\lambda_A \right]_{(1)}^{(2)}, \qquad \text{on } \gamma, \end{aligned} \tag{2.7d}$$

$$0 = |\Omega| - V, \tag{2.7e}$$

$$0 = |\Gamma_i| - A_i, \qquad i = 1,2. \tag{2.7f}$$

An asymptotic analysis of the stationary equations of (1.7a), (1.7b) with constraints (2.1a), (2.1b) and (2.4) has been performed in [25], too and in the limit as $\varepsilon \to 0$ the above Euler-Lagrange equations emerge.

2.4 Variational formulation

We aim for employing the method described in [3] in order to approximate the geometric evolution equation (1.7a). Note that we may write $v_\nu = \partial_t \boldsymbol{x} \cdot \boldsymbol{\nu}$ for the membrane normal velocity, recalling that $\boldsymbol{x}(t)$ denotes the identity map on the evolving surface $\Gamma(t)$. We then multiply (1.7a) with a test function $\chi : \Gamma \to \mathbb{R}$ and integrate over Γ. The first and fourth term on the right hand side may be integrated by parts and further evaluated, for instance

$$\begin{aligned} &\int_\Gamma -\Delta_\Gamma \big(k_\kappa(c)(\kappa - \kappa_s(c)) \big) \chi = \int_\Gamma \nabla_\Gamma \big(k_\kappa(c)(\kappa - \kappa_s(c)) \big) \cdot \nabla_\Gamma \chi \\ &= \int_\Gamma k_\kappa(c) \nabla_\Gamma \kappa \cdot \nabla_\Gamma \chi + k_\kappa'(c) \nabla_\Gamma c \cdot \nabla_\Gamma \chi (\kappa - \kappa_s(c)) - k_\kappa(c) \kappa_s'(c) \nabla_\Gamma c \cdot \nabla_\Gamma \chi. \end{aligned}$$

The first one of these terms and the third one of (1.7a) may be put to the left hand side. For fields $\boldsymbol{\xi}:\Gamma\to\mathbb{R}^3$ and $\phi,\chi:\Gamma\to\mathbb{R}$ let us now define the forms

$$\begin{aligned}
\mathcal{N}(\Gamma,\boldsymbol{\nu};\boldsymbol{\xi},\chi) &:= \int_\Gamma \boldsymbol{\xi}\cdot\boldsymbol{\nu}\chi,\\
\mathcal{C}(\Gamma,\kappa,c;\chi,\phi) &:= \int_\Gamma k_\kappa(c)\nabla_\Gamma\phi\cdot\nabla_\Gamma\chi+\frac{1}{2}k_\kappa(c)\big(\kappa-\kappa_s(c)\big)^2\phi\chi,\\
\mathcal{B}(\Gamma,\kappa,\nabla_\Gamma\boldsymbol{\nu},c;\chi) &:= \int_\Gamma k_\kappa'(c)(\kappa-\kappa_s(c))\nabla_\Gamma c\cdot\nabla_\Gamma\chi-k_\kappa(c)\kappa_s'(c)\nabla_\Gamma c\cdot\nabla_\Gamma\chi\\
&\quad+\int_\Gamma -|\nabla_\Gamma\boldsymbol{\nu}|^2k_\kappa(c)(\kappa-\kappa_s(c))\chi+\big(k_g'(c)(\kappa\boldsymbol{I}+\nabla_\Gamma\boldsymbol{\nu})\big)\nabla_\Gamma c\cdot\nabla_\Gamma\chi\\
&\quad+\int_\Gamma \sigma\varepsilon\nabla_\Gamma c\otimes\nabla_\Gamma c:\nabla_\Gamma\boldsymbol{\nu}\chi+\sigma\Big(\frac{\varepsilon}{2}|\nabla_\Gamma c|^2+\frac{1}{\varepsilon}\psi(c)\Big)\kappa\chi,\\
\mathcal{L}(\Gamma;\chi) &:= \int_\Gamma\chi, \qquad \mathcal{K}(\Gamma,\kappa;\chi) := \int_\Gamma -\kappa\chi, \qquad \mathcal{H}(\Gamma,\kappa,c;\chi) := \int_\Gamma -\kappa h(c)\chi.
\end{aligned}$$

Then a weak formulation of (1.7a) reads

$$\begin{aligned}
&\mathcal{N}(\Gamma,\boldsymbol{\nu};\partial_t\boldsymbol{x},\chi)-\mathcal{C}(\Gamma,\kappa,c;\chi,\kappa)\\
=&\mathcal{B}(\Gamma,\kappa,\nabla_\Gamma\boldsymbol{\nu},c;\chi)-\lambda_V\mathcal{L}(\Gamma;\chi)-\lambda_A\mathcal{K}(\Gamma,\kappa;\chi)-\lambda_c\mathcal{H}(\Gamma,\kappa,c;\chi)
\end{aligned}$$

for all test functions $\chi:\Gamma\to\mathbb{R}$.

Also the mean curvature equation (1.8) may be tested with a test function and integrated by parts. Defining

$$\mathcal{A}(\Gamma;\boldsymbol{\xi},\boldsymbol{\zeta}) := \int_\Gamma \nabla_\Gamma\boldsymbol{\zeta}:\nabla_\Gamma\boldsymbol{\xi}$$

for fields $\boldsymbol{\zeta},\boldsymbol{\xi}:\Gamma\to\mathbb{R}^3$, a weak formulation of (1.8) reads

$$\mathcal{A}(\Gamma;\boldsymbol{\xi},\boldsymbol{x})+\mathcal{N}(\Gamma,\boldsymbol{\nu};\boldsymbol{\xi},\kappa)=0$$

for all test functions $\boldsymbol{\xi}:\Gamma\to\mathbb{R}^3$.

For the weak formulation of the phase separation equation (1.7b) we introduce the forms

$$\begin{aligned}
\mathcal{M}(\Gamma;\phi,\chi) &:= \int_\Gamma\chi\phi,\\
\mathcal{J}(\Gamma,c;\phi,\chi) &:= \int_\Gamma \varepsilon\sigma\nabla_\Gamma\chi\cdot\nabla_\Gamma\phi+\frac{\sigma}{\varepsilon}2(c^2-1)^+\chi\phi,\\
\mathcal{S}(\Gamma,\kappa,c;\phi) &:= \int_\Gamma \frac{\sigma}{\varepsilon}2(1-c^2)^+c\phi-\frac{1}{2}(\kappa-\kappa_s(c))^2k_\kappa'(c)\phi\\
&\quad+\int_\Gamma k_\kappa(c)(\kappa-\kappa_s(c))\kappa_s'(c)\phi-gk_g'(c)\phi,\\
\mathcal{P}(\Gamma,c;\phi) &:= \int_\Gamma h'(c)\phi,
\end{aligned}$$

where $f^+ := \max\{f,0\}$ for any real quantity f. We remark that the above forms use the splitting $\psi'(c) = 2c(c^2-1)^+ - 2c(1-c^2)^+$, which we found convenient in the time discretization. We note that other splittings of the double well potential into convex and non convex parts have been introduced in [7,26] in the context of developing numerical methods that guarantee a decrease of energy independently of the time step. We finally obtain

$$\varepsilon\omega\mathcal{M}(\Gamma;\phi,\partial_t^\bullet c)+\mathcal{J}(\Gamma,c;\phi,c)=\mathcal{S}(\Gamma,\kappa,c;\phi)-\lambda_c\mathcal{P}(\Gamma,c;\phi)$$

for all test functions $\phi:\Gamma\to\mathbb{R}$.

Problem 2.3 (Variational Relaxation Flow)**.** Suppose that an initial phase field surface (Γ^0,c^0), Γ^0 enclosing a domain Ω^0, is given such that $|\Omega^0|=V$, $|\Gamma^0|=A_1+A_2$ and $\int_{\Gamma^0}h(c^0)=A_1-A_2$. Find a family $\{(\Gamma(t),\kappa(t),c(t))\}_{t\in[0,\infty)}$ with $\Gamma(0)=\Gamma^0$, $c(0)=c^0$ and functions $\lambda_i:[0,\infty)\to\mathbb{R}$, $i\in\mathcal{I}$, such that at each time $t\in[0,\infty)$

$$\begin{aligned}\mathcal{N}(\Gamma,\boldsymbol{\nu};\partial_t\boldsymbol{x},\chi)-\mathcal{C}(\Gamma,\kappa,c;\chi,\kappa)&=\mathcal{B}(\Gamma,\kappa,\nabla_\Gamma\boldsymbol{\nu},c;\chi)-\lambda_V\mathcal{L}(\Gamma;\chi)\\&\quad-\lambda_A\mathcal{K}(\Gamma,\kappa;\chi)-\lambda_c\mathcal{H}(\Gamma,\kappa,c;\chi),\end{aligned}\tag{2.8a}$$

$$\mathcal{A}(\Gamma;\boldsymbol{\xi},\boldsymbol{x})+\mathcal{N}(\Gamma,\boldsymbol{\nu};\boldsymbol{\xi},\kappa)=0,\tag{2.8b}$$

$$\varepsilon\omega\mathcal{M}(\Gamma;\phi,\partial_t^\bullet c)+\mathcal{J}(\Gamma,c;\phi,c)=\mathcal{S}(\Gamma,\kappa,c;\phi)-\lambda_c\mathcal{P}(\Gamma,c;\phi)\tag{2.8c}$$

for all test functions $\chi,\phi:\Gamma\to\mathbb{R}$, $\boldsymbol{\xi}:\Gamma\to\mathbb{R}^3$ and such that $0=\mathcal{C}_i(\Gamma(t),c(t))$, $i\in\{V,A,c\}$, at each time $t\in[0,\infty)$.

3 Numerical scheme

The discretization of the governing equations in Problem 2.3 is based on approximating the membrane Γ by triangulated surfaces Γ_h and using surface finite elements for both the surface and the Allen-Cahn equation on the surface [14,18,19,22,23]. The specific scheme that we use for the geometric evolution law is based on the approach of [3]. In order to avoid explicit coordinates on or parameterizations of the evolving surface, the surface at a subsequent time is computed via a parameterization over the actual surface. Since the present evolution equation is of fourth order the mean curvature is kept as an additional unknown quantity allowing for a splitting into two second order equations. Appropriate variational formulations involve only first order gradients so that H^1 conforming finite element spaces are sufficient.

3.1 Surface finite elements

Triangulated surfaces and isoparametric quadratic SFEs are carefully described in [8,15, 30] and we give a short introduction only. A *triangulated polyhedral surface* $\widetilde{\Gamma}_h$ is a polyhedron with planar triangular faces,

$$\widetilde{\Gamma}_h=\bigcup_{\widetilde{T}\in\widetilde{\mathcal{T}}_h}\widetilde{T},$$

where $\widetilde{\mathcal{T}}_h$ consists of a finite number of closed, non-degenerate triangles $\widetilde{T}$ such that the intersection of two different triangles is either empty or a common edge or a common vertex and such that each triangle has at least one edge in common with another triangle. Given a triangulated polyhedral surface $\widetilde{\Gamma}_h$, a *quadratic triangulated surface* Γ_h over $\widetilde{\Gamma}_h$ is of the form

$$\Gamma_h = \bigcup_{T\in\mathcal{T}_h} T,$$

where there exists a homeomorphism $\mathcal{F}:\widetilde{\Gamma}_h\to\Gamma_h$ such that

- for each $T\in\mathcal{T}_h$ there is a $\widetilde{T}\in\widetilde{\mathcal{T}}_h$ with $T=\mathcal{F}(\widetilde{T})$,
- $\mathcal{F}|_{\widetilde{T}}$ is a quadratic polynomial on each $\widetilde{T}\in\widetilde{\mathcal{T}}_h$,
- $\mathcal{F}$ leaves vertices unchanged.

It follows that each curved triangle $T\in\mathcal{T}_h$ can be parameterized by a quadratic polynomial $\Phi_T:\hat{T}\to T$ where $\hat{T}:=\{\lambda\in\mathbb{R}^3\,|\,\lambda_i\geq 0,\sum_i\lambda_i=1\}$ is a fixed reference triangle. Denoting the space of polynomials of degree two by $\mathbb{P}^2(\cdot)$ we have that $\Phi_T\in\mathbb{P}^2(\hat{T})$.

Given a quadratic triangulated surface Γ_h, the *isoparametric quadratic SFE space* is defined by

$$S_h(\Gamma_h):=\big\{\phi\in C^0(\Gamma_h)\big|\phi|_T\circ\Phi_T\in\mathbb{P}^2(\hat{T})\text{ on each }T\in\mathcal{T}_h\big\}.$$

The nodal variables are the evaluations at the images of the nodes of $\hat{T}$, i.e., at the images of the vertices and of the edge midpoints of $\hat{T}$. We denote the coordinates of these points by $\{\boldsymbol{x}_i\}_{i=1}^{N_h}$. Thus, N_h is the dimension of S_h. We denote the standard basis by $\{\phi_i\}_{i=1}^{N_h}$ characterized by $\phi_i(\boldsymbol{x}_j)=\delta_{ij}$ with δ_{ij} being the Kronecker symbol. Elements $\zeta_h\in S_h$ can uniquely be written in the form $\zeta_h(\boldsymbol{x})=\sum_i\zeta_i\phi_i(\boldsymbol{x})$ with coefficients $\zeta_i=\zeta_h(\boldsymbol{x}_i)$. We introduce the notation $\underline{\zeta}=(\zeta_i)_{i=1}^{N_h}$ for the coefficient vector. For discrete versions of three-dimensional fields $\boldsymbol{\xi}=\{\xi_k\}_{k=1}^3$ we introduce the finite element space $\boldsymbol{S}_h(\Gamma_h):=S_h^3(\Gamma_h)$. The standard basis of $\boldsymbol{S}_h$ is $\{\phi_i\boldsymbol{e}_k\}_{i,k=1}^{N_h,3}$, where $\boldsymbol{e}_k=\{\delta_{kj}\}_{j=1}^3$. We will employ the notation $\underline{\boldsymbol{\xi}}=\{\boldsymbol{\xi}_{i,k}\}_{i,k=1}^{N_h,3}$, where $\boldsymbol{\xi}_{i,k}=\boldsymbol{\xi}_h(\boldsymbol{x}_i)\cdot\boldsymbol{e}_k$. Generically, we denote the identity map on Γ_h by $\boldsymbol{x}_h$ and recall that the notion *isoparametric* refers to the fact that $\boldsymbol{x}_h\in\boldsymbol{S}_h(\Gamma_h)$.

We now consider pairs (Γ_h,c_h) where Γ_h is a quadratic triangulated surface that encloses a bounded domain Ω_h together with a scalar field $c_h\in S_h(\Gamma_h)$. For such objects we denote the external unit normal of the enclosed Ω_h by $\boldsymbol{\nu}_h$, which is well-defined at each point in the interior of each triangle $T\in\mathcal{T}_h$. The matrix $\boldsymbol{P}_h=\boldsymbol{I}-\boldsymbol{\nu}_h\otimes\boldsymbol{\nu}_h=\nabla_{\Gamma_h}\boldsymbol{x}_h$ stands for the projection onto the tangential space of Γ_h and, as $\boldsymbol{\nu}_h$, is well-defined at each point in the interior of each triangle $T\in\mathcal{T}_h$.

The weak formulation of (1.8) can be used to define a finite element function $\boldsymbol{\kappa}_h\in\boldsymbol{S}_h(\Gamma_h)$ that serves to approximate the mean curvature vector $\boldsymbol{\kappa}=\kappa\boldsymbol{\nu}$. It is defined via

$$\int_{\Gamma_h}\boldsymbol{\kappa}_h\cdot\boldsymbol{w}_h+\nabla_{\Gamma_h}\boldsymbol{x}_h:\nabla_{\Gamma_h}\boldsymbol{w}_h=0$$

which has to hold for all $w_h \in S_h(\Gamma_h)$. Furthermore, we need an approximation to the shape operator $\nabla_\Gamma \nu$ as appearing in the form $\mathcal{B}(\Gamma,\kappa,\nabla_\Gamma \nu,c;\chi)$. Partial integration on an admissible phase field surface yields

$$\int_\Gamma \nabla_\Gamma \nu : Z + \nu \cdot (\nabla_\Gamma \cdot Z) = \int_\Gamma \nabla_\Gamma \cdot (Z^T \nu) = \int_\Gamma (Z^T \nu) \cdot \kappa$$

for any smooth test function $Z : \Gamma \to \mathbb{R}^{3\times 3}$. This motivates defining the tensor valued field $Q_h \in S_h^{3\times 3}(\Gamma_h)$ via

$$\int_{\Gamma_h} Q_h : Z_h = -\int_{\Gamma_h} (\nabla_{\Gamma_h} \cdot Z_h) \cdot \nu_h + \int_{\Gamma_h} \nu_h \otimes \kappa_h : Z_h$$

for all tensor-valued test fields $Z_h \in S_h^{3\times 3}(\Gamma_h)$ in order to approximate the shape operator, [30]. If Γ_h approximates a sufficiently smooth surface Γ by interpolation then, as shown in [31], $\|\bar{Q}_h - \nabla_\Gamma \nu\|_{L^2(\Gamma)} = \mathcal{O}(h)$ for quadratic SFEs where $\bar{Q}_h$ is an appropriate lift of Q_h from Γ_h to Γ, see [31] for the details. Numerical experiments furthermore indicate that this convergence also holds true in $L^\infty(\Gamma)$. Another possibility for approximating the shape operator is to compute $\nabla_{\Gamma_h} \nu_h$ on every $T \in \mathcal{T}_h$, see [15] for the details.

3.2 Spatial discretization

For dynamic problems we consider families of triangulated surfaces $\{\Gamma_h(t)\}_{t\in I}$ where each $\Gamma_h(t)$ is a quadratic triangulated surface and the nodes $x_i(t)$ depend smoothly on the relaxation time t. The velocity $v_h(t,x) := \sum_i \partial_t x_i(t) \phi_i(t,x)$ is an element of $S_h(\Gamma_h(t))$ and is tacitly taken into account in the operator $\partial_t^\bullet$ whenever working on a triangulated surface. We remark that (see [22])

$$\partial_t^\bullet \phi_i = (\partial_t + v_h \cdot \nabla)\phi_i = 0, \quad \forall i = 1,\cdots,N_h. \tag{3.1}$$

Also the other t-dependent surface fields will become families of finite element functions, e.g., $\{\kappa_h(t)\}_t$ where the t dependence concerns the coefficient vector $\underline{\kappa}(t)$ but also the basis functions $\phi_i(t,\cdot)$ of $S_h(\Gamma_h(t))$. Observe that by the transport property of the basis functions (3.1) we have that

$$\partial_t^\bullet c_h = \sum_i \partial_t^\bullet (c_i \phi_i) = \sum_i \partial_t^\bullet c_i \phi_i + c_i \partial_t^\bullet \phi_i = \sum_i \partial_t c_i \phi_i.$$

Inserting this into (2.8c), a semi-discrete problem is derived from Problem 2.3 in a straightforward way:

Problem 3.1 (Semi-discrete Variational Relaxation Flow)**.** Suppose that an initial quadratic triangulated surface Γ_h^0 enclosing a domain Ω_h^0 and an initial order parameter $c_h^0 \in S_h(\Gamma_h^0)$ are given such that $|\Omega_h^0| = V$, $|\Gamma_h^0| = A_1 + A_2$ and $\int_{\Gamma_h^0} h(c^h) = A_1 - A_2$. Find a

family $\{(\Gamma_h(t),\kappa_h(t),c_h(t))\}_{t\in I}$, where $\Gamma_h(t)$ is a quadratic triangulated surface with identical map $\boldsymbol{x}_h(t)\in\boldsymbol{S}_h(\Gamma_h(t))$ for all $t\in I$, where $\kappa_h(t),c_h(t)\in S_h(\Gamma_h(t))$ for all $t\in I$ and where $\Gamma_h(0)=\Gamma_h^0$ and $c_h(0)=c_h^0$ and find functions $\lambda_{V,h},\lambda_{A,h},\lambda_{c,h}:[0,\infty)\to\mathbb{R}$ such that

$$\begin{aligned}&\mathcal{N}(\Gamma_h,\boldsymbol{\nu}_h;\partial_t\boldsymbol{x}_h,\chi_h)-\mathcal{C}(\Gamma_h,\kappa_h,c_h;\chi_h,\kappa_h)\\&\quad=\mathcal{B}(\Gamma_h,\kappa_h,\boldsymbol{Q}_h,c_h;\chi_h)-\lambda_{V,h}\mathcal{L}(\Gamma_h;\chi_h)-\lambda_{A,h}\mathcal{K}(\Gamma_h,\kappa_h;\chi_h)-\lambda_{c,h}\mathcal{H}(\Gamma_h,\kappa_h,c_h;\chi_h),\quad(3.2a)\\&\mathcal{A}(\Gamma_h;\boldsymbol{\xi}_h,\boldsymbol{x}_h)+\mathcal{N}(\Gamma_h,\boldsymbol{\nu}_h;\boldsymbol{\xi}_h,\kappa_h)=0,\quad(3.2b)\\&\varepsilon\omega\sum_{i=1}^{N_h}\mathcal{M}(\Gamma_h;\phi_h,\phi_i)\partial_t c_i+\mathcal{J}(\Gamma_h,c_h;\phi_h,c_h)=\mathcal{S}(\Gamma_h,\kappa_h,c_h;\phi_h)-\lambda_{c,h}\mathcal{P}(\Gamma_h,c_h;\phi_h)\quad(3.2c)\end{aligned}$$

for all test functions $\chi_h,\phi_h\in S_h(\Gamma_h(t))$ and $\boldsymbol{\xi}_h\in\boldsymbol{S}_h(\Gamma_h(t))$ and such that $0=\mathcal{C}_i(\Gamma_h(t),c_h(t))$, $i\in\{V,A,c\}$, at each time $t\in[0,\infty)$.

When we had to compute the integrals we usually chose stable quadrature formulas that are exact for polynomials of degree eight. Some tests with formulas exact up to degree 16 have been carried out but gave no significant differences. Corresponding functionality is provided by the employed finite element software ALBERTA [39].

3.3 Time discretization

In order to discretize in time we consider times $\{t^m\}_{m\in\mathbb{N}}$ with $t^m\in[0,\infty)$, $t^m>t^{m-1}$ and $t^m\to\infty$ as $m\to\infty$ and set $\tau^m:=t^{m+1}-t^m$ for the time steps. Quantities at time t^m are denoted with an upper index m. At any time t^m the surface Γ_h^m is given by knowledge of $\check{\boldsymbol{x}}_h^m\in\boldsymbol{S}_h(\Gamma_h^m)$, the identical map on Γ_h^m. As mentioned before, the idea of discretizing the geometric evolution equation (2.8a) is to compute the surface at the subsequent time t^{m+1} via parameterizing it over the already computed surface at time level m which is achieved by computing $\boldsymbol{x}_h^{m+1}$ as an element of $\boldsymbol{S}_h(\Gamma_h^m)$. The link between $\check{\boldsymbol{x}}_h^m\in\boldsymbol{S}_h(\Gamma_h^m)$ and a computed $\boldsymbol{x}_h^m\in\boldsymbol{S}_h(\Gamma_h^{m-1})$ is given by $\check{\boldsymbol{x}}_{i,k}^{m+1}=\boldsymbol{x}_{i,k}^{m+1}$, $i=1,\cdots,N_h$, $k=1,2,3$. At the same time, we compute values for curvature and order parameter $\kappa_h^{m+1},c_h^{m+1}\in S_h(\Gamma_h^m)$ in the new nodes. To access the current values stored in $\kappa_h^m,c_h^m\in S_h(\Gamma_h^{m-1})$ we define $\check{\kappa}_h^m,\check{c}_h^m\in S_h(\Gamma_h^m)$ by $\check{\kappa}_i^m=\kappa_i^m$ and $\check{c}_i^m=c_i^m$, $i=1,\cdots,N_h$.

To compute the required approximation to the Weingarten map $\boldsymbol{Q}_h^m\in S_h^{3\times3}(\Gamma_h^m)$ we proceed as described in [24] which we, for convenience of the reader, repeat here. We first compute a field $\hat{\boldsymbol{\kappa}}_h^m\in\boldsymbol{S}_h(\Gamma_h^m)$ from

$$\int_{\Gamma_h^m}\hat{\boldsymbol{\kappa}}_h^m\cdot\boldsymbol{w}_h=\mathcal{A}(\Gamma_h^m;\check{\boldsymbol{x}}_h^m,\boldsymbol{w}_h)$$

for all $\boldsymbol{w}_h\in\boldsymbol{S}_h(\Gamma_h^m)$. Then we find $\boldsymbol{Q}_h^m\in S_h^{3\times3}(\Gamma_h^m)$ by solving

$$\int_{\Gamma_h^m}\boldsymbol{Q}_h^m:\boldsymbol{Z}_h=\int_{\Gamma_h^m}\left(-(\nabla_{\Gamma_h}\cdot\boldsymbol{Z}_h)\cdot\boldsymbol{\nu}_h^m+\boldsymbol{\nu}_h^m\otimes\hat{\boldsymbol{\kappa}}_h^m:\boldsymbol{Z}_h\right)$$

for all tensor-valued test fields $\boldsymbol{Z}_h\in S_h^{3\times3}(\Gamma_h)$.

Problem 3.2 (Fully Discrete Variational Relaxation Flow)**.** Suppose that an initial quadratic triangulated surface Γ_h^0 enclosing a domain Ω_h^0 and an initial order parameter $c_h^0 \in S_h(\Gamma_h^0)$ are given such that $|\Omega_h^0| = V$, $|\Gamma_h^0| = A_1 + A_2$ and $\int_{\Gamma_h^0} h(c_h^0) = A_1 - A_2$. Find a family $\{(\Gamma_h^m, \kappa_h^m, c_h^m)\}_{m\in\mathbb{N}}$, where $\kappa_h^m, c_h^m \in S_h(\Gamma_h^{m-1})$ and find series $\{\lambda_{V,h}^m, \lambda_{A,h}^m, \lambda_{c,h}^m\}_{m\in\mathbb{N}}$ such that

$$\begin{aligned}
&\mathcal{N}(\Gamma_h^m, \boldsymbol{\nu}_h^m; \boldsymbol{x}_h^{m+1}, \chi_h) - \tau^m \mathcal{C}(\Gamma_h^m, \check{\kappa}_h^m, \check{c}_h^m; \chi_h, \kappa_h^{m+1}) \\
&\quad = \mathcal{N}(\Gamma_h^m, \boldsymbol{\nu}_h^m; \check{\boldsymbol{x}}_h^m, \chi_h) + \tau^m \mathcal{B}(\Gamma_h^m, \check{\kappa}_h^m, \boldsymbol{Q}_h^m, \check{c}_h^m; \chi_h) - \lambda_{V,h}^{m+1} \tau^m \mathcal{L}(\Gamma_h^m; \chi_h) \\
&\qquad - \lambda_{A,h}^{m+1} \tau^m \mathcal{K}(\Gamma_h^m, \check{\kappa}_h^m; \chi_h) - \lambda_{c,h}^{m+1} \tau^m \mathcal{H}(\Gamma_h^m, \check{\kappa}_h^m, \check{c}_h^m; \chi_h), \qquad (3.3a) \\
&\mathcal{A}(\Gamma_h^m; \boldsymbol{\xi}_h, \boldsymbol{x}_h^{m+1}) + \mathcal{N}(\Gamma_h^m, \boldsymbol{\nu}_h^m; \boldsymbol{\xi}_h, \kappa_h^{m+1}) = 0, \qquad (3.3b) \\
&\varepsilon\omega \mathcal{M}(\Gamma_h^m; \phi_h, c_h^{m+1}) + \tau^m \mathcal{J}(\Gamma_h^m, \check{c}_h^m; \phi_h, c_h^{m+1}) \\
&\quad = \varepsilon\omega \mathcal{M}(\Gamma_h^m; \phi_h, \check{c}_h^m) + \tau^m \mathcal{S}(\Gamma_h^m, \check{\kappa}_h^m, \check{c}_h^m; \phi_h) - \lambda_{c,h}^{m+1} \tau^m \mathcal{P}(\Gamma_h^m, \check{c}_h^m; \phi_h) \qquad (3.3c)
\end{aligned}$$

for all test functions $\chi_h, \phi_h \in S_h(\Gamma_h^m)$, $\boldsymbol{\xi} \in \boldsymbol{S}_h(\Gamma_h^m)$ and such that $0 = \mathcal{C}_i(\Gamma_h^{m+1}, \check{c}_h^{m+1})$, $i \in \{V, A, c\}$ for each $m \in \mathbb{N} \cup \{0\}$.

We remark that the splitting of ψ yields an approximation of $\psi'(c)$ to be

$$2c^{n+1}((c^n)^2 - 1)^+ - 2c^n(1 - (c^n)^2)^+.$$

This linearization serves to make it easier to solve the resulting system of equations. The resulting discrete problem is a large linear system for the vector $\{c_h^{m+1}, \kappa_h^{m+1}, \boldsymbol{x}_h^{m+1}\}$ coupled to a small nonlinear system for the Lagrange multipliers. We state the solution algorithm in the Appendix. In short, we used the direct method UMFPACK [13] to factorize the system matrix and solve a few systems of linear equations (see Appendix A). After, a quasi-Newton iteration is carried out to compute the new Lagrange multipliers and the new shape including mean curvature and phase field variable (see Appendix B). This ideas dates back to [24] with the difference that the system matrix is as in [3] rather than as in [20] on which the method in [24] is based.

3.4 Damping of the tangential motion and stopping criterion

In contrast to the method used in [24] which is based on [20] the discretization in Section 3.3 leads to updates which also have tangential components ensuring good mesh properties, see [2] for details. However we have observed in our simulations that, in the long term, the tangential motion usually becomes dominating, i.e., the geometry of the shape merely changes but the nodes almost only move in tangential direction. Then the total relaxation time becomes unnecessarily large and, thus, we want to damp the tangential motion.

For this purpose, we extended the method from [3], Section 2.4, from linear to quadratic surface finite elements. On a quadratic triangulated surface Γ_h let $\boldsymbol{\tau}_{\mu,h}$, $\mu = 1,2$, denote

vector fields such that $\{\boldsymbol{\nu}_h,\boldsymbol{\tau}_{1,h},\boldsymbol{\tau}_{2,h}\}$ form an orthonormal basis of $\mathbb{R}^3$. In practice, we used the surface gradient of one of the barycentric coordinates to define $\boldsymbol{\tau}_{1,h}$ and then computed $\boldsymbol{\tau}_{2,h}$ from the vector product $\boldsymbol{\nu}_h\wedge\boldsymbol{\tau}_{1,h}$. In analogy to $\mathcal{N}$ we define the forms

$$\mathcal{T}_\mu(\Gamma_h,\boldsymbol{\tau}_{\mu,h};\boldsymbol{\xi}_h,\chi_h):=\int_{\Gamma_h}\boldsymbol{\xi}_h\cdot\boldsymbol{\tau}_{\mu,h}\chi_h,\qquad \mu=1,2,$$

for $\boldsymbol{\xi}_h\in\boldsymbol{S}_h(\Gamma_h)$, $\chi_h\in S_h(\Gamma_h)$.

Let $\alpha^m>0$ be a sequence of damping parameters which initially are small and towards the end of the simulation may increase (its adaptive choice will be discussed later on). We now augment the scheme in Problem 3.2 by two additional equations and change the curvature equation (3.3b). Apart from $\boldsymbol{x}_h^{m+1},\kappa_h^{m+1},c_h^{m+1}$ we now have to find $\beta_{\mu,h}^{m+1}\in S_h(\Gamma_h^m)$, $\mu=1,2$, such that

$$\begin{aligned}&\mathcal{A}(\Gamma_h^m;\boldsymbol{\xi}_h,\boldsymbol{x}_h^{m+1})+\mathcal{N}(\Gamma_h^m,\boldsymbol{\nu}_h^m;\boldsymbol{\xi}_h,\kappa_h^{m+1})\\&\qquad+\alpha^m\mathcal{T}_1(\Gamma_h^m,\boldsymbol{\tau}_{1,h}^m;\boldsymbol{\xi}_h,\beta_{1,h}^{m+1})+\alpha^m\mathcal{T}_2(\Gamma_h^m,\boldsymbol{\tau}_{2,h}^m;\boldsymbol{\xi}_h,\beta_{2,h}^{m+1})=0,\end{aligned}\tag{3.4a}$$

$$\mathcal{T}_\mu(\Gamma_h^m,\boldsymbol{\tau}_{\mu,h}^m;\boldsymbol{x}_h^{m+1},\eta_{\mu,h})-\tau^m\mathcal{M}(\Gamma_h^m;\eta_{\mu,h},\beta_{\mu,h}^{m+1})=\mathcal{T}_\mu(\Gamma_h^m,\boldsymbol{\tau}_{\mu,h}^m;\check{\boldsymbol{x}}_h^m,\eta_{\mu,h}),\quad\mu=1,2,\tag{3.4b}$$

for all test functions $\boldsymbol{\xi}\in\boldsymbol{S}_h(\Gamma_h^m)$, $\eta_{\mu,h}\in S_h(\Gamma_h^m)$, $\mu=1,2$, in addition to (3.3a), (3.3c) and the side constraints. The identities (3.4b) mean that the tangential motion in the directions $\boldsymbol{\tau}_{\mu,h}$ is stored in the $\beta_{\mu,h}$. The coupling in (3.4a) is such that the tangential motion that would emerge from its original equation (3.3b) is reduced by $\tau^m\alpha^m\boldsymbol{\tau}_{\mu,h}^m$.

In our simulations we do not fix the damping parameters but rather choose them in dependence on the velocities of vertices. Defining $\boldsymbol{v}_h^m\in\boldsymbol{S}_h(\Gamma_h^{m-1})$ and $\check{\boldsymbol{v}}_h^m\in\boldsymbol{S}_h(\Gamma_h^m)$ by

$$\boldsymbol{v}_h^m:=\frac{\boldsymbol{x}_h^m-\check{\boldsymbol{x}}_h^{m-1}}{\tau^{m-1}}\quad\text{ and }\quad\check{\boldsymbol{v}}_{i,k}^m:=\boldsymbol{v}_{i,k}^m,\quad i=1,\cdots,N_h,\quad k=1,2,3,$$

we compute the L^2 norms of the normal and the tangential portion

$$\bar{v}_{\boldsymbol{\nu}}^m:=\Big(\int_{\Gamma_h^m}\big(\boldsymbol{\nu}_h^m\cdot\check{\boldsymbol{v}}_h^m\big)^2\Big)^{\frac{1}{2}},\qquad\bar{v}_{\boldsymbol{\tau}}^m:=\Big(\int_{\Gamma_h^m}\big(\check{\boldsymbol{v}}_h^m-(\boldsymbol{\nu}_h^m\cdot\check{\boldsymbol{v}}_h^m)\boldsymbol{\nu}_h^m\big)^2\Big)^{\frac{1}{2}}.$$

The aim is to keep the tangential portion in a certain range with respect to the normal portion, i.e., $c_v\bar{v}_{\boldsymbol{\nu}}^m\le\bar{v}_{\boldsymbol{\tau}}^m\le C_v\bar{v}_{\boldsymbol{\nu}}^m$, where $0<c_v<C_v$ are two constants. As long as this holds true we just keep the damping parameter, $\alpha^{m+1}=\alpha^m$. If it turns out that $\bar{v}_{\boldsymbol{\tau}}^m$ is too big then we increase the damping parameter, $\alpha^{m+1}>\alpha^m$, so that the damping is enforced in the next time step. Similarly, we decrease it if the tangential portion is too small where we recall that some tangential motion is desired in order to keep a nice mesh. In practice, we usually set $c_v=0.1$ and $C_v=0.5$ and the increase or decrease of α^m was realized by multiplying with or dividing by a factor of 10. Additionally, we imposed lower and upper limits for α^m, usually of 10^{-6} and 10^4 and we usually started with $\alpha^0=10^{-4}$.

The criterion for stopping the computations and accepting a final surface as the equilibrium is based on the size of the computed normal velocity (and, hence, of the tangential velocity) and whether no Newton iteration steps are required any more to update the Lagrange multipliers. We usually stopped if the L^2 norm of the normal velocity normalized by the characteristic radius (4.2), i.e., $\|\vec{v}_\nu^m\|_{L^2(\Gamma_h^m)}/R_c$, reached a value of 10^{-3} and no Newton iteration steps had been carried out during the preceding 10 time steps where we typically required an accuracy of 10^{-8} for the constraints.

3.5 Mesh adaption

In order to reduce computation time it is desirable to adaptively refine and coarsen the grid. We use ideas as proposed in [24], Section 3.3.4, but for convenience describe them here. The employed finite element software ALBERTA [39] requires a marking function that provides an integer for each triangle indicating how many times it has to be refined (= bisected) or how many times it may be coarsened (if the integer is negative). We want to ensure that the interfacial layers are resolved by the mesh but also demand that the strongly curved regions contain sufficient numbers of nodes. For the latter regions we consider the quantity $|\nabla_\Gamma \boldsymbol{\nu}|^2 = \kappa^2 - 2g = \kappa_1^2 + \kappa_2^2$ where κ_1 and κ_2 are the principal curvatures. As an approximation on a quadratic polyhedral surface we consider the finite element function $s_\kappa \in S_h(\Gamma_h)$ with the nodal values $s_{\kappa,i} = |\boldsymbol{Q}_h(\boldsymbol{x}_i)|^2$.

Our *marking strategy* consists of three positive numbers $(N_{in}, N_{off}, N_\kappa)$ with the following meaning: The diameter of an element in the interfacial layer shall be smaller than ε/N_{in} and if the element belongs to one of the bulk phases then the diameter shall be smaller than ε/N_{off} and throughout the element diameter shall be smaller than N_κ/I_κ, where I_κ is the arithmetic mean of the values of $\sqrt{s_\kappa}$ in the nodes belonging to the element. If a triangle fails to meet any of these criteria then it is marked for one refinement. In turn, if its diameter is smaller than half the desired diameter then it is marked for one coarsening.

We perform an explicit mesh adaption strategy and execute the marking algorithm usually at the beginning of about every fifth time step, followed by the mesh adaption. These routines involve interpolation and restriction operations to obtain the values in the new nodes as described in [39]. In general, this leads to an increase of the total energy and, in particular, the surface data are no longer consistent in the following sense. For a triangulated surface in (or close to) equilibrium equation (3.2b) is fulfilled and relates $\boldsymbol{x}_h$ and κ_h but mesh adaption typically destroys this relation. We observed that this usually increases the nodal velocity and for this reason we perform a couple of time steps before considering another mesh adaptation. We further observed in the late stage of the simulation when the system has almost relaxed that mesh changes are no longer required. Altogether, this mesh adaption procedure does not ensure that the above mentioned criteria are fulfilled during the whole relaxation but they are satisfied in the long run and, hence, for the final shapes that we are essentially interested in.

3.6 Remeshing

The method that we use for the geometric evolution equation in general leads to a good mesh behavior by moving the vertices in tangential direction in an appropriate way, see [2]. Nevertheless, when the deformations became strong even this incorporated tangential vertex motion turned out to be insufficient. In particular, when necks formed then triangles became somewhat elongated such that the ratio of the longest edge to the shortest one became large. For this reason we employ the global remeshing method presented in [21]. It is based on computing a conformal map to a reference manifold on which a nice mesh is given. Then this nice mesh is mapped to the actual triangulated surface using the inverse of the computed map. Since the inverse is conformal, too, the new surface mesh is of the same quality as the mesh on the reference manifold.

We have employed this method only when computing some of the surfaces presented in Section 4.4, namely those where the initial shape has been obtained by deforming the two-sphere S^2. In the following description, the reference manifold therefore is S^2, too.

As a quality measure $q(\Gamma_h)$ of a quadratic triangulated surface Γ_h we have used the minimal value of the sinus of the interior angles of the elements,

$$q(\Gamma_h) := \min\{q_T | T \in \mathcal{T}_h\}, \qquad q_T := \min\{\sin(\alpha) | \alpha \text{ inner angle of } T\}. \tag{3.5}$$

With inner angles of a curved triangle $T \in \mathcal{T}_h$ we mean all angles of the four flat triangles formed by neighboring nodes: Recalling that any $T \in \mathcal{T}_h$ has six nodes, three of them corresponding to the vertices and three located on the edges (i.e., the images of the nodes of $\hat{T}$ under the map Φ_T, see Section 3.1), we consider the three planar triangles formed by a vertex and the nodes on the adjacent edges and the planar triangle formed by the three nodes on the edges.

One will expect that the mesh quality essentially depends on the triangulated polyhedral surface $\widetilde{\Gamma}_h$ underlying a quadratic triangulated surface Γ_h (as introduced in Section 3.1). For this reason, it should be sufficient to work with linear SFEs, which has the further advantage to be cheaper and this indeed turned out to be practical. We introduce the space

$$\widetilde{S}_h(\widetilde{\Gamma}_h) := \{\phi \in C^0(\widetilde{\Gamma}_h) | \phi \text{ is linear on each } \widetilde{T} \in \widetilde{\mathcal{T}}_h\}$$

of dimension $\widetilde{N}_h$ and denote its vector valued variant by $\widetilde{\boldsymbol{S}}_h(\widetilde{\Gamma}_h)$. We remark that $\widetilde{\Gamma}_h^m$ is obtained from a computed Γ_h^m by restricting the corresponding identity map $\boldsymbol{x}_h^m$ to $\widetilde{\boldsymbol{S}}_h(\widetilde{\Gamma}_h)$ as this yields the identity map on $\widetilde{\Gamma}_h^m$ (intuitively, we drop the edge midpoints and linearly interpolate between the remaining vertices). The vertices of $\widetilde{\Gamma}_h$ are denoted by $\{\widetilde{\boldsymbol{x}}_i\}_{i=1}^{\widetilde{N}_h}$ and let us introduce the notation $\underline{\widetilde{\boldsymbol{u}}} = (\widetilde{\boldsymbol{u}}_h(\widetilde{\boldsymbol{x}}_i))_{i=1}^{\widetilde{N}_h}$ for a function $\widetilde{\boldsymbol{u}}_h \in \widetilde{\boldsymbol{S}}_h(\widetilde{\Gamma}_h)$.

We assume that the mesh on S^2 that we want to map to $\widetilde{\Gamma}_h$ is given in terms of a finite element function $\widetilde{\boldsymbol{y}}_h \in \widetilde{\boldsymbol{S}}_h(\widetilde{\Gamma}_h)$. In practice, initial meshes Γ_h^0 usually were obtained by deforming a mesh on S^2. This mesh on S^2 was created by starting with a cube with vertices on S^2, inserting a diagonal edge on each face so that a triangulated surface with

flat triangles-a special case of a quadratic triangulated surface-is obtained and then projecting the nodes on the edges to S^2. We then subdivided by bisection, starting with the longest edge and projecting new nodes to S^2 (the algorithm is described in the manual of the employed software [39] but see also [31] for further illustration). This procedure resulted in quadratic triangulated surfaces with nodes on S^2 of a quality $q>0.498$ for up to 18 refinements. The vertex positions of this "nice" quadratic triangulated mesh on S^2 before deforming it to Γ_h^0 were stored in the vector $\underline{\widetilde{\boldsymbol{y}}}$ associated with the finite element function $\widetilde{\boldsymbol{y}}_h \in \widetilde{\boldsymbol{S}}_h(\widetilde{\Gamma}_h)$. We remark that the grid adaption introduced in Section 3.5 also involves refinement and coarsening of the mesh on S^2 defined by $\widetilde{\boldsymbol{y}}_h$ where, again, new vertices are projected to S^2.

The method described in [21] consists of computing a minimizer of the Dirichlet integral among the functions mapping to S^2. The latter constraint is taken into account with Lagrange multipliers ρ_i, $i=1,\cdots,\widetilde{N}_h$, leading to the functional

$$F(\underline{\widetilde{\boldsymbol{u}}},\rho) := \frac{1}{2}\int_{\widetilde{\Gamma}_h} \left|\nabla_{\widetilde{\Gamma}_h}\widetilde{\boldsymbol{u}}_h\right|^2 + \sum_{i=1}^{N_h}\rho_i\left(|\widetilde{\boldsymbol{u}}_h(\widetilde{\boldsymbol{x}}_i)|^2-1\right), \tag{3.6}$$

where $\rho=(\rho_i)_{i=1}^{\widetilde{N}_h}$. The problem then is to solve $0=F'(\underline{\widetilde{\boldsymbol{u}}},\rho)$ for which we employ the Newton method where we use $(\underline{\widetilde{\boldsymbol{y}}},0)$ as an initial guess. To compute the new search direction $(F''(\cdot))^{-1}F'(\cdot)$ a linear system of saddle point structure needs to be solved and there we apply the generalized minimal residual method, GMRES. We remark that conformal maps to S^2 are only unique up to the conformal group of S^2 which consists of stereographic projections of the Möbius transformations of the complex plane (these include dilations, inversions, rotations and translations) [21]. To ensure uniqueness one could "divide by this group", i.e., taking the additional constraints into account by augmenting (3.6) with further terms and Lagrange multipliers as described in [21] which improves the stability of the numerical method. But the above method of computing an approximation to any solution has turned out to be sufficient for our purpose.

As mentioned, the goal is to map the vertices $\{\widetilde{\boldsymbol{y}}_h(\widetilde{\boldsymbol{x}}_i)\}_i$ including their connections to $\widetilde{\Gamma}_h$ by $\widetilde{\boldsymbol{u}}_h^{-1}$ in order to obtain a nice mesh there. The function $\widetilde{\boldsymbol{u}}_h$ defines another mesh $\widetilde{\boldsymbol{u}}_h(\widetilde{\Gamma}_h)$ with vertices on S^2 and the first step is to project the vertices $\{\widetilde{\boldsymbol{y}}_h(\widetilde{\boldsymbol{x}}_i)\}_i$ to $\widetilde{\boldsymbol{u}}_h(\widetilde{\Gamma}_h)$. By the piecewise linearity of $\widetilde{\boldsymbol{u}}_h$ it is then straightforward to apply $\widetilde{\boldsymbol{u}}_h^{-1}$ to the projected vertices. The upshot is a new triangulated surface $\widetilde{\Gamma}_h^{new}$ with a mesh of the same topology but almost the same quality as the mesh of $\widetilde{\boldsymbol{y}}_h(\widetilde{\Gamma}_h)$-almost because of the involved projections and numerical errors arising from minimizing (3.6) with an iterative method.

In order to obtain a new quadratic triangulated surface Γ_h^{new} we simply pick the edge midpoints of $\widetilde{\Gamma}_h^{new}$ as new vertices, i.e., we linearly interpolate. This is certainly no particularly sophisticated method yet recall that we are not that much interested in approximating the relaxation dynamics but rather in the final relaxed shapes.

In a similar simple fashion we deal with the surface fields c_h^m, κ_h^m that are required for the subsequent time step. One first projects the fields to the space $\widetilde{S}_h(\widetilde{\Gamma}_h^m)$ yielding

functions $\widetilde{c}_h^m$, $\widetilde{\kappa}_h^m$. The values in the vertices of the new mesh $\widetilde{\Gamma}_h^{m,new}$ then are computed by evaluating $\widetilde{c}_h^m$ and $\widetilde{\kappa}_h^m$ there. And finally, new functions $c_h^{m,new}$,$\kappa_h^{m,new} \in S_h(\Gamma_h^{m,new})$ are defined by finding the values in the edge midpoints by interpolation. In general, $\mathcal{C}_i(\Gamma_h^{m,new},c_h^{m,new})=0$, $i\in\{V,A,c\}$, is not fulfilled any more. But the errors have been small enough in practice such that the Newton iteration in the subsequent time step has been able to restore the constraints.

Similar to mesh adaption, in practice, mesh smoothing is no longer required when the system has almost relaxed in the late stage of the simulation. In particular, characteristic values such as the energy and the Lagrange multipliers of the final shape are not affected (modulo numerical errors) by the preceding mesh operations during the relaxation. This has been checked by comparing results computed with and without the mesh operations for problems where the mesh did not degenerate too much. In Fig. 2 we display some meshes revealing the effect of our remeshing method.

4 Simulation results

As mentioned in the introduction we are mainly interested in computing equilibrium shapes of two-phase vesicles and we compute these by a relaxation dynamics which reduces the total membrane energy. The phase field methodology employed to model the phase separation on the membrane easily allows for topological changes during the relaxation. We present an example where this happens before we turn our attention to the equilibrium shapes.

This initial shape is a cigar of (dimensionless) length 4 and diameter 1 with spherical caps, a shape of revolution around the symmetry axis $\{\boldsymbol{x}=(x_1,x_2,x_3)^T\in\mathbb{R}^3 | x_2=x_3=0.5\}$. Fig. 3 displays it including the initial phase separation where the order parameter is given by

$$c^0(\boldsymbol{x})=\begin{cases} 1, & \text{if } 2.25\le x_1, \\ x_1-1.25, & \text{if } 0.25\le x_1\le 2.25, \\ -1, & \text{if } x_1\le 0.25. \end{cases} \tag{4.1}$$

The initial grid was obtained by gluing together four coarsely triangulated surfaces of unit cubes, refining the triangles and projecting onto the surface until it had 4610 nodes. No mesh adaption has been carried out during the relaxation. The enclosed volume and area are $V\approx 2.88$ and $A_1+A_2\approx 12.57$, respectively and the area difference is set to $A_1-A_2=4.71$. Furthermore, we set $\sigma=2$, $k_\kappa^{(1)}=k_\kappa^{(2)}=1$, $\kappa_s^{(1)}=-2$, $\kappa_s^{(2)}=2$ and $k_g^{(1)}=k_g^{(2)}=0$, $\varepsilon=0.3$ and $\omega=0.01$. The time step was fixed at $\tau=5\times 10^{-5}$.

Almost immediately the red phase also occurs at the other tip (Fig. 3, upper middle), which is energetically favorable due to the spontaneous curvatures and already involves a topological change since the two red domains are separate. The shape of a two-phase dumbbell emerges (Fig. 3, upper right), however it is not stable. Due to some numerical noise, the axial symmetry breaks and the red domains merge together while two sep-

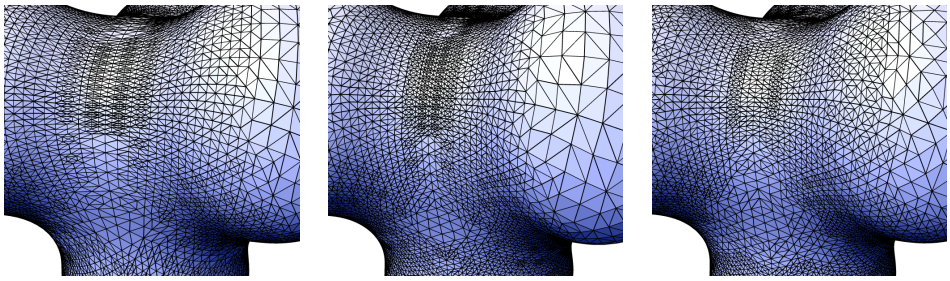

Figure 2: Example for the remeshing method described in Section 3.6. For each curved triangle of a quadratic triangulated surface we display the four flat sub-triangles as specified below (3.5). On the left, a mesh with a quality of $q \approx 0.29$ before remeshing. In our simulations the mesh never degenerated that much, this is for illustration purpose only. In the middle the mesh after applying the global remeshing method, $q \approx 0.47$. Often, some mesh adaption (here a rather severe one) is necessary after remeshing. On the right we display the mesh after this adaption.

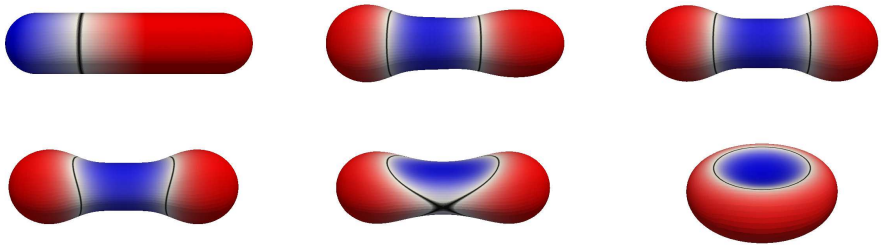

Figure 3: Relaxation of a cigar with lateral phase separation (upper left) to a two-phase dumbbell (upper right) which then becomes unstable (lower left) and relaxes to a two-phase discocyte (lower right). Here and in the following figures showing vesicle shapes we visualize the quadratic triangulated surfaces by resorting to the four flat sub-triangles of each curved triangle as specified below (3.5). The vertices of this sub-triangulation are the nodes of the quadratic triangulated surface in which we know the value of c_h. The color-code is based on linearly interpolating these nodal values on each flat sub-triangle. While red and blue stand for the pure phases the dark region indicates the phase interface position by means of the zero level set of c_h. The grid is quite coarse but these images only serve to illustrate that topological changes with respect to the phase separation can occur during relaxation and are accounted for by the proposed method.

arated blue domain form so that we finally end up with a two-phase discocyte (Fig. 3, lower row).

Lets us now turn our attention to equilibrium configurations. Shapes of two-phase vesicles are characterized by several dimensionless numbers that are computed from the physical parameters and to which we will refer in the subsequent sections.

1. *Reduced volume*: Given a membrane area A_1+A_2, the *characteristic radius* of a sphere of the same area,
$$R_c := \sqrt{(A_1+A_2)/4\pi}, \tag{4.2}$$
can serve as a length scale of the problem. The reduced volume is then defined as the volume divided by the volume of a ball with radius R_c,
$$v_r := \frac{V}{4\pi R_c^3/3}.$$

2. *Relative domain size*: The portion of the "blue" phase area,

$$x_r := \frac{A_2}{A_1 + A_2}.$$

3. *Reduced line tension*: The *invagination length* $\xi_i := k_\kappa^{(1)}/\bar{\sigma}$ serves to characterize the competition between bending and line tension. The reduced line tension is then the ratio between the characteristic radius and the invagination length,

$$\bar{\sigma}_r := \frac{R_c}{\xi_i} = \frac{R_c}{k_\kappa^{(1)}}\bar{\sigma}.$$

4. *Bending moduli ratio*: Ratio of the mean curvature bending rigidities,

$$b_r := \frac{k_\kappa^{(2)}}{k_\kappa^{(1)}}.$$

5. *Normalized saddle-splay moduli difference:* Thanks to the continuity of h_ν and h_d, Eq. (2.7b) can be written as

$$0 = \left[k_\kappa(h_\nu + h_p - \kappa_s)\right]_{(1)}^{(2)} + \left[k_g\right]_{(1)}^{(2)} h_\nu,$$

and a short calculation shows that (2.7d) may be written as

$$0 = \left[\frac{k_\kappa}{2}(h_\nu - h_p - \kappa_s)^2\right]_{(1)}^{(2)} - \left[k_g\right]_{(1)}^{(2)} h_d^2 - \bar{\sigma} h_g + \left[\lambda_A\right]_{(1)}^{(2)}.$$

In view of (2.7c) we see that the Gaussian curvature bending rigidities influence the phase interface position only via their difference $k_g^{(1)} - k_g^{(2)}$. An appropriate dimensionless quantity is

$$\Delta_g := \frac{k_g^{(1)} - k_g^{(2)}}{k_\kappa^{(1)}}.$$

4.1 Convergence in ε

We have observed convergence of the method as the mesh has been refined and the time discretization error has been negligible in comparison with the spatial discretization error, which is analogous to the results in [24], Section 4.3. We therefore confine ourselves on reporting on the convergence as $\varepsilon \to 0$ only.

We studied a rotationally symmetric problem and relaxed cigars of length 4 and diameter 1 with spherical caps and symmetry axis $\{\boldsymbol{x} = (x_1, x_2, x_3)^T \in \mathbb{R}^3 | x_2 = x_3 = 0.5\}$ to nonsymmetric dumbbells, see Fig. 4. The initial grids were obtained as described in the

example at the beginning of this section. We recall the values $V\approx 2.88$, $A_1+A_2\approx 12.57$ and $A_1-A_2=4.71$. Initial data for the order parameter are given by (4.1) again.

We chose the material parameters $\sigma=1$, $k_\kappa^{(1)}=1.5$, $k_\kappa^{(2)}=1$, $\kappa_s^{(1)}=\kappa_s^{(2)}=0$ and $k_g^{(1)}=k_g^{(2)}=0$ and set the kinetic coefficient $\omega=0.1$. With respect to mesh adaption, the strategy $(N_{in},N_{off},N_\kappa)=(1.6,0.3,4)$ has been used where the value $N_\kappa=4$ is chosen such that it does not impact on the grid, i.e., the mesh size is determined by ε only. The time step was fixed at $\tau=(\varepsilon/30)^2$.

Table 1: Convergence in ε, test with cigars relaxing to non-symmetric dumbbells as described in Section 4.1, values of the discrete energy (4.3) and the Lagrange multiplier $\lambda_{V,h}$ at time $t=0.45$ as well as orders of convergence computed according to (4.4).

ε	N_h	$\mathcal{F}_h$	eoc$(\mathcal{F}_h)$	$\lambda_{V,h}$	eoc$(\lambda_{V,h})$
$0.3/2^{0.0}$	01730	58.8500	-	27.20821	-
$0.3/2^{0.5}$	02738	58.8159	1.2273	27.24698	3.3533
$0.3/2^{1.0}$	03634	58.7935	1.2376	27.25911	2.1723
$0.3/2^{1.5}$	06274	58.7790	1.0641	27.26482	3.6868
$0.3/2^{2.0}$	09114	58.7689	1.3357	27.26641	3.5202
$0.3/2^{2.5}$	13458	58.7626	-	27.26688	-

The values shown in Table 1 reveal convergence as $\varepsilon\to 0$. For instance, we display the discrete energy

$$\mathcal{F}_h=\int_{\Gamma_h}\frac{1}{2}k_\kappa(c_h)\big(\kappa_h-\kappa_s(c_h)\big)^2+k_g(c_h)(\kappa_h^2-|\boldsymbol{Q}_h|^2)+\frac{\sigma\varepsilon}{2}\big|\nabla_{\Gamma_h}c_h\big|^2+\frac{\sigma}{\varepsilon}\psi(c_h) \tag{4.3}$$

and the Lagrange multiplier for the volume constraint. The experimental order of convergence for a quantity f that depends on ε is defined by

$$\text{eoc}(f)(\varepsilon)=\frac{\log\big(|f(\sqrt{2}\varepsilon)-f(\varepsilon)|/|f(\varepsilon)-f(\varepsilon/\sqrt{2})|\big)}{\log(\sqrt{2})}. \tag{4.4}$$

In Fig. 4 we see that also the shape profiles, displayed around the necks, converge. Moreover, the transition points marking the zero level sets of c_h are displayed and converge also as $\varepsilon\to 0$.

The mean curvature is expected to jump in the limit because condition (2.7b) reads

$$0=k_\kappa^{(2)}\kappa^{(2)}-k_\kappa^{(1)}\kappa^{(1)} \quad\Leftrightarrow\quad \frac{k_\kappa^{(1)}}{k_\kappa^{(2)}}=\frac{\kappa^{(1)}}{\kappa^{(2)}} \tag{4.5}$$

and we have that $k_\kappa^{(1)}\neq k_\kappa^{(2)}$. We computed the values of the mean curvature κ and of the bending moment

$$q:=k_\kappa(c)\kappa \tag{4.6}$$

along the profiles. In Fig. 5 the values are displayed for several values of ε around the phase interface position. We see that, as $\varepsilon\to 0$, a jump of the mean curvature indeed

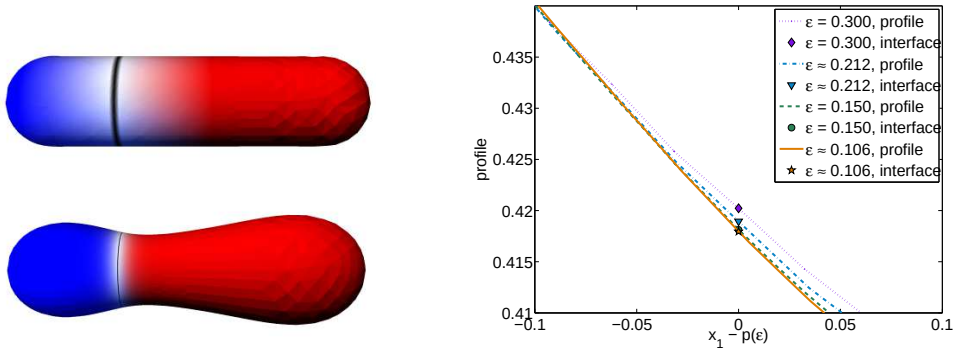

Figure 4: Convergence in ε, test with cigars relaxing to non-symmetric dumbbells as described in Section 4.1. We display the initial and the relaxed shape. The colors red and blue indicate the pure phases and the position of the phase interface is black. The profiles of the relaxed dumbbell shapes around the necks with the phase transition region for several values of ε are shown, too. More precisely, we intersected the shapes with the half plane $\{x\in\mathbb{R}^3 | x_2=0.5, x_3\geq 0\}$ and display the distance to the symmetry axis. In addition, we indicate the position of the phase interface. For convenience we shifted the profiles along the symmetry axis such that the phase interfaces are at the position 0 where we stress that the continuum energy (1.6) is invariant under such operations.

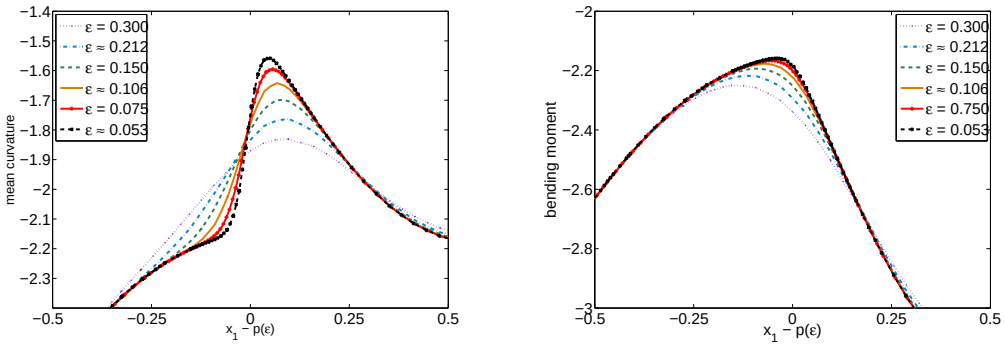

Figure 5: Convergence in ε, test with cigars relaxing to non-symmetric dumbbells as described in Section 4.1. We display the mean curvature and the bending moment q as defined in (4.6) along the profiles displayed in Fig. 4 for several values of ε. The shapes have been shifted along the symmetry axis such that the phase interfaces are at the position.

emerges. In turn, the curves showing the bending moment q appear to converge to a continuous curve, possibly with a kink.

4.2 Increasing the line tension

We studied the influence of the surface tension parameter σ in an axisymmetric setting as in the Figs. 4B and 4C of [5]. Starting with a cigar again we have the values $A_1+A_2=12.5664$, $V\approx 3.1835$, $A_1-A_2\approx 1.5080$ and we set $k_\kappa^{(1)}=k_\kappa^{(2)}=1$, $k_g^{(1)}=k_g^{(2)}=0$, $\kappa_s^{(1)}=\kappa_s^{(2)}=0$. The dimensionless parameters are $v_r\approx 0.76$, $x_r\approx 0.44$, $b_r=1$, $\Delta_g=0$. Furthermore, we set $\omega=0.1$, $\varepsilon=0.1$.

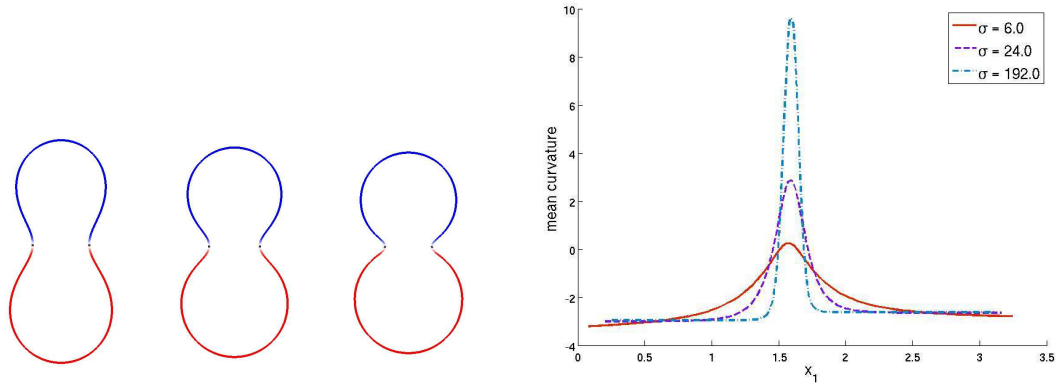

Figure 6: Influence of the line tension with simulation parameters as described in Section 4.2. The profiles of axisymmetric vesicles are shown for the values $\sigma=6,24,192$ which corresponds to $\bar{\sigma}_r=8,32,256$. The colors red and blue indicate the pure phases whilst the phase interface, approximated by the zero-level-set of c_h, is colored in black. We also display the values of the mean curvature along the profiles from the top to the bottom.

We performed simulations for values of σ between 6 and 192 (or $\bar{\sigma}_r$ between 8 and 256). Fig. 6 shows the profiles of some relaxed shapes, i.e., the intersections of the shape with the plane $\{\boldsymbol{x}|x_2=0.5\}$ containing the symmetry axis. As σ increases the neck becomes more and more pronounced. This is supported by the graph below the profiles in Fig. 6 showing the values of the mean curvature along one side of the profiles where we see a peak emerging as σ increases. We also observe that the mean curvature in the pure phases becomes nearly constant, i.e., the shapes of the pure phases become spherical.

4.3 Impact of different Gaussian curvature bending rigidities

A difference of the Gaussian curvature bending rigidities effectively shifts the interface out of the neck of a two-phase dumbbell as in Fig. 4. Consider an axisymmetric setting as in [5], Fig. 5C and 5D. We started with the same cigar as in the previous sections as initial shape and appropriate initial values for the phase field variable. Parameters were $A_1+A_2=12.5664$, $V\approx 3.1835$, $A_1-A_2\approx 1.5080$, $k_\kappa^{(1)}=k_\kappa^{(2)}=1$, $\kappa_s^{(1)}=\kappa_s^{(2)}=0$ and $\sigma=24.45$ so that $v_r\approx 0.76$, $x_r\approx 0.44$, $\bar{\sigma}_r=32.6$ and $b_r=1$ in consistency with the data in [5].

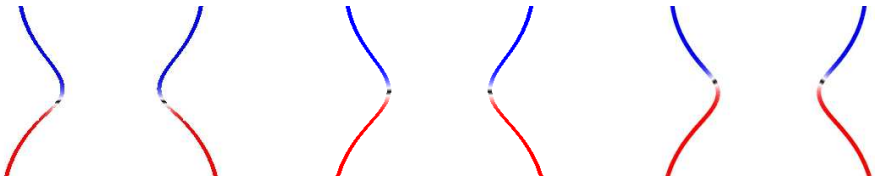

Figure 7: Influence of the Gaussian curvature bending rigidities on the position of the phase interface, simulation parameters as described in Section 4.3. We display the profiles of axisymmetric vesicles around the neck shown for the values $\Delta_g=-4,0,4$ from left to right. The colors red and blue indicate the pure phases and the position of the phase interface is black.

Furthermore, we set $\omega=0.1$, $\varepsilon=0.1$. Changing $\Delta_g:=k_g^{(1)}-k_g^{(2)}$ from -4 to 4, our method indeed reproduced the predicted behavior of shifting the phase interface. Fig. 7 shows some intersections of the shapes with the plane $\{\boldsymbol{x}|x_2=0.5\}$ that contains the symmetry axis around the necks.

4.4 Stabilization of multiple domains

Many equilibrium vesicle shapes formed by two-phase membranes exhibit symmetries and some of them are even axisymmetric. The Euler-Lagrange equations in this setting can be written as a system of ordinary differential equations, for instance see [12], that are easier and cheaper to solve than the nonlinear partial differential equations in the general setting. Our method may be used to investigate the phase diagram for arbitrary equilibrium shapes and in this section we present an example where the spontaneous curvature causes the appearance of non-axisymmetric shapes.

In a first set of simulations we chose $k_\kappa^{(1)}=k_\kappa^{(2)}=1$, $k_g^{(1)}=k_g^{(2)}=0$, $\kappa_s^{(2)}=0$, $\sigma=1$ and vary $\kappa_s^{(1)}$. We consider different initial shapes that are distinguished by the number of red domains. Enclosed volume and area are fixed to $V\approx 2.6226$ and $A_1+A_2=12.5664$ and we set $A_1-A_2\approx 2.4504$. To be more precise with the initial shapes, we start with cigar shaped vesicles where on one or on both tips red phases appear and we start with deformed spheres with three or four tips. The latter ones initially enclosed a larger volume than the desired value because otherwise the quality of the initial grid would have been too bad. We then started the method in combination with a deflation process that leads to the desired volume, i.e., the value V in (2.3) depends linearly on t until the desired value is attained. The dimensionless parameters are $v_r\approx 0.63$, $x_r\approx 0.4025$, $\bar{\sigma}_r\approx 1.33$, $b_r=1$, $\Delta_g=0$. Furthermore, we set $\omega=0.01$, $\varepsilon=0.1$.

In Fig. 8 we show some relaxed shapes with one to four tips in the upper row. In the graph below the energy is displayed in dependence of the spontaneous curvature $\kappa_s^{(1)}$ of the red phase. We see that for $\kappa_s^{(1)}$ between 0 and about -1.7 the shape with one simply connected red domain energetically is most favorable. If $\kappa_s^{(1)}$ decreases until about -2.3, the shape with two red tips attached to a blue bulged cylinder has less energy. Below that value, three tips are favored and at about -4.1 the regime with four tips starts to have least energy.

The curves are based on trying to compute the equilibrium energy at points $\kappa_s^{(1)}=0,-0.2,-0.4,\cdots,-7$. We were unable to do so for all configurations because of instabilities. For instance, after increasing $\kappa_s^{(1)}$ to -2 the configuration with four red tips became unstable and the vesicle relaxed to a shape with three red tips and after decreasing $\kappa_s^{(1)}$ to -7.6 a vesicle with six red tips emerged.

Different bending rigidities together with the spontaneous curvature can also stabilize multiple domains of one phase embedded in another phase on vesicles close to spheres. We set $k_\kappa^{(1)}=1$, $k_\kappa^{(2)}=8$, $k_g^{(1)}=k_g^{(2)}=0$, $\kappa_s^{(1)}=-8$, $\kappa_s^{(2)}=0$ and $\sigma=1$. The initial shape was

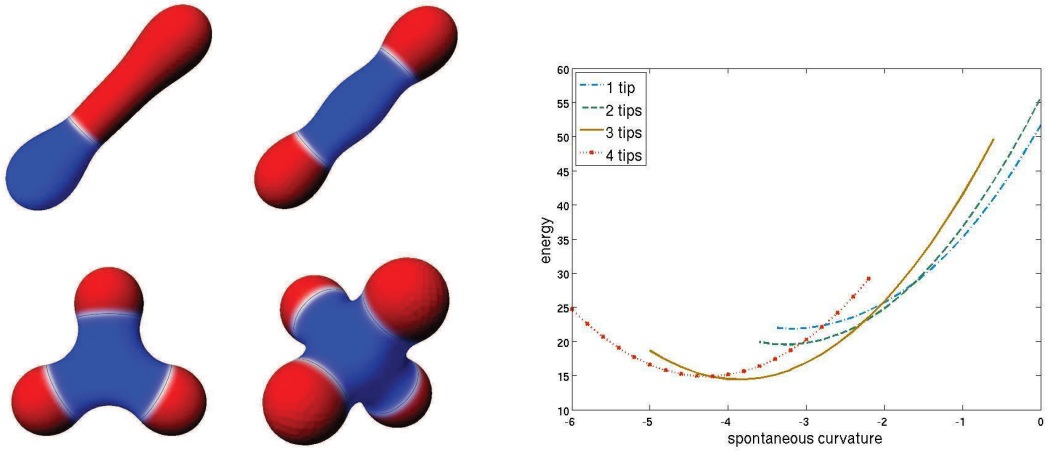

Figure 8: Equilibrium shapes and their energy in dependence of the number of the spontaneous curvature $\kappa_s^{(1)}$ for the red phase and the number of red domains. Simulation parameters are as described in Section 4.4. In particular, $\kappa_s^{(2)}=0$ in all cases. On top, four equilibrium shapes for spontaneous curvatures $\kappa_s^{(1)}=-1$ (top left), $\kappa_s^{(1)}=-2$ (top right), $\kappa_s^{(1)}=-3$ (bottom left), $\kappa_s^{(1)}=-4$ (bottom right).

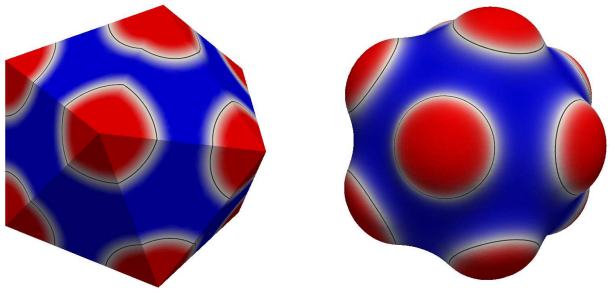

Figure 9: Initial configuration and equilibrium shape for a vesicle with multiple rafts. Simulation parameters are stated in Section 4.4.

a slightly deformed icosahedron where the 12 tips are occupied by the first phase, see Fig. 9 on the left. Further data are $V\approx 3.8134$, $A_1+A_2=12.5664$ and $A_1-A_2\approx -2.3894$ so that $v_r\approx 0.91$, $x_r\approx 0.595$, $\bar{\sigma}_r\approx 1.33$, $b_r=8$, $\Delta_g=0$. We also set $\omega=0.05$, $\varepsilon=0.1$. The relaxed shape is displayed in Fig. 9 on the right. This two-phase membrane shape turned out to be stable with respect to changes in ε, the mesh size, small perturbations of the sizes of the domains of the first phase (the red one) and small perturbations of the initial shape. In fact, in Fig. 9 on the left we display a gently perturbed icosahedron which can be spotted by comparing the distances between the red domains. We remark that too large values for ε lead to coalescence of the red domains. If different material parameters in the two phases are not taken into account then the red domains will merge as well as has already been discovered in [44].

4.5 Nonspherical vesicles

Our method also allows for investigating non-spherical vesicles with phase separation. As an example, we start with a torus with main radii 1 and 0.15 centered in $\boldsymbol{x}=0$ and extended in directions x_1 and x_2. We want to investigate the effect of multiple phase changes along the torus and, thus, initialize the order parameter such that the two phases alternate. The parameters are

$$V=0.444083,\quad A_1+A_2=5.921605,\quad A_1-A_2=-1.184321,$$
$$k_\kappa^{(1)}=1,\quad k_\kappa^{(2)}=1.25,\quad \kappa_s^{(1)}=\kappa_s^{(2)}=0,\quad k_g^{(1)}=-0.83,\quad k_g^{(2)}=-1.0375$$

and, if not otherwise stated, $\sigma=0.2$. The dimensionless parameters are $x_r\approx 0.6$ and $b_r=1.25$ (v_r and $\bar{\sigma}_r$ require the characteristic radius R_c which has been defined for spherical vesicles only). The material parameters are comparable to those in [9]. Furthermore, we set $\omega=0.01$, $\varepsilon=0.05$.

We have mainly been interested in the stability of multiple phase interfaces along the torus and found that for the same simulation parameters different initial configurations lead to different locally stable equilibria. In fact, four to eight red domains along the torus appear to be stable which has been checked by slightly distorting the position of the domains, enclosed volume and ε. Interestingly, the shape with four domains has less symmetry than the initial shape where the red domains were distributed at equal distance. Fig. 10 displays some relaxed shapes. We remark that when starting with nine red domains we observed coarsening and ended up with six red domains.

We found that increasing the number of red domains leads to a growth of the total energy which is almost linear. From the graph at the bottom of Fig. 10 we infer that the increase is essentially associated with the line energy which is almost proportional to the number of phase interfaces across the torus since its radius does not change very much. A small increase of the bending energy can be observed, too, which we link to the fact that the shape is less able to deform if the number of red domains increases.

In another set of simulations each of the two phases are connected and we vary σ, see Fig. 11 on the left for the initial configuration and on the right for a relaxed shape. The phase interfaces lead to indentions which, as expected, become more pronounced when the line tension is increased. We remark that a configuration as in Fig. 12 on the left does not seem to lead to a stable shape. The torus shrinks reducing the lengths of the phase interfaces and such that the blue phase (with the higher bending rigidity) becomes flatter (see Fig. 12 on the right). Both are favorable for the energy and apparently the increase of energy due to the increased bending of the red phase is compensated. In fact, we even observed an acceleration of the shrinking process. Moreover, the energy of the shape in Fig. 12 on the right is 147.9562 and, thus, lower than the energy of the relaxed shape obtained from the initial configuration as in Fig. 11 on the left with the same parameters, which is 152.9947. However, analyzing this behavior further requires investigating the remeshing method described in Section 3.6 for non-spherical shapes and is left for future research.

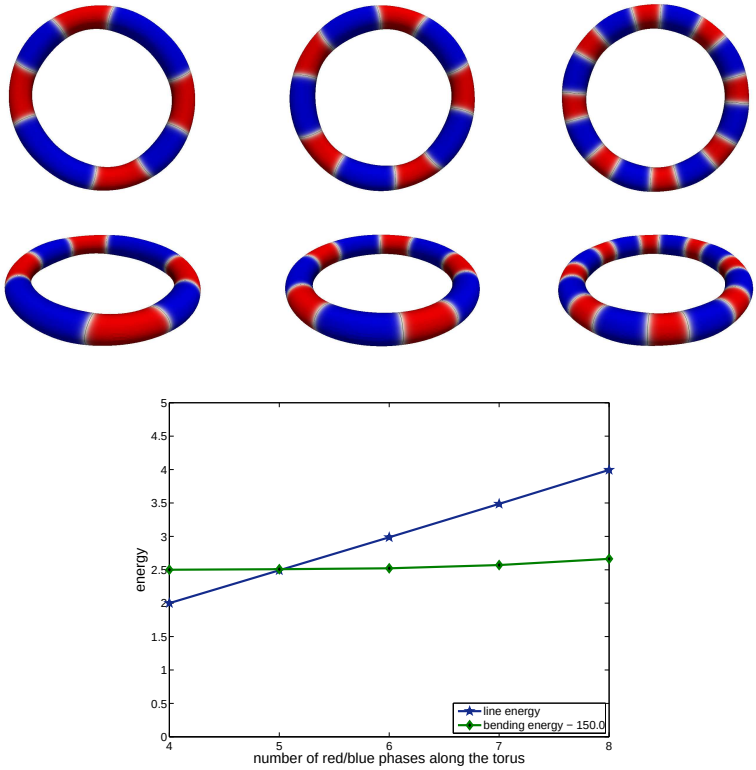

Figure 10: On top, relaxed shapes of tori differing in the number of phase transitions. Below we display the energy over the number of red domains split up into the line energy and the bending energy (where we subtracted 150 from the latter one for an easier comparison). The simulation parameters are given in Section 4.5.

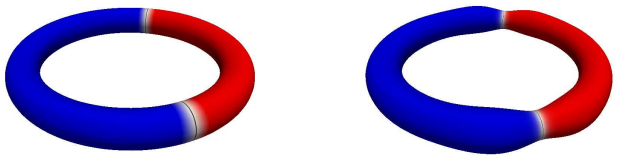

Figure 11: Torus with phase separation, initial shape and relaxed shape for $\sigma = 6.4$. The other simulation parameters are as described in Section 4.5.

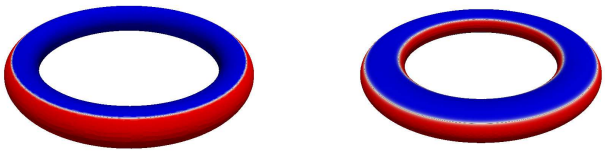

Figure 12: Torus with phase separation, initial shape on the left. At time $t = 0.02$, the shape on the right is obtained. Simulation parameters as described in Section 4.5.

Acknowledgments

This work was supported by the UK Engineering and Physical Sciences Research Council, grant EP/G010404.

Appendix

A Discrete system

In order to solve the fully discrete system in each time step we proceed in a similar way to [24]. Let us define the matrices

$$\begin{aligned}
&A^m=(A^m_{i,k,j,l})_{i,k,j,l=1}^{N_h,3,N_h,3}, && A^m_{i,k,j,l}:=\mathcal{A}(\Gamma^m_h;\phi_i\boldsymbol{e}_k,\phi_j\boldsymbol{e}_l),\\
&N^m=(N^m_{i,k,j})_{i,k,j=1}^{N_h,3,N_h}, && N^m_{i,k,j}:=\mathcal{N}(\Gamma^m_h,\boldsymbol{\nu}^m_h;\phi_i\boldsymbol{e}_k,\phi_j),\\
&T^m_\mu=(T^m_{\mu;i,k,j})_{i,k,j=1}^{N_h,3,N_h}, && T^m_{\mu;i,k,j}:=\mathcal{T}_\mu(\Gamma^m_h,\boldsymbol{\tau}^m_{\mu,h};\phi_i\boldsymbol{e}_k,\phi_j), \qquad \mu=1,2,\\
&M^m=(M^m_{i,j})_{i,j=1}^{N_h}, && M^m_{i,j}:=\mathcal{M}(\Gamma^m_h;\phi_i,\phi_j),\\
&C^m=(C^m_{i,j})_{i,j=1}^{N_h}, && C^m_{i,j}:=\mathcal{C}(\Gamma^m_h,\check{\kappa}^m_h,\check{c}^m_h;\phi_i,\phi_j),\\
&I^m=(I^m_{i,j})_{i,j=1}^{N_h}, && I^m_{i,j}:=\varepsilon\omega\mathcal{M}(\Gamma^m_h;\phi_i,\phi_j)+\tau^m\mathcal{J}(\Gamma^m_h,\check{c}^m_h;\phi_i,\phi_j).
\end{aligned}$$

The objects A^m, N^m, T^m_1 and T^m_2 rather are tensors but they can be understood as matrices in connection with coefficient vectors $\underline{\boldsymbol{\xi}}=\{\boldsymbol{\xi}_{i,k}\}_{i,k=1}^{N_h,3}$ and $\underline{\chi}=\{\chi_i\}_{i=1}^{N_h}$ of finite element functions $\boldsymbol{\xi}_h\in\boldsymbol{S}(\Gamma^m_h)$ and $\chi_h\in S_h(\Gamma^m_h)$. In fact, matrix-vector products like $A^m\underline{\boldsymbol{\xi}}$, $(N^m)^T\underline{\boldsymbol{\xi}}$ and $T^m_\mu\underline{\chi}$ are naturally defined by

$$\big(A^m\underline{\boldsymbol{\xi}}\big)_{i,k}=\sum_{j,l=1}^{N_h,3}A^m_{i,k,j,l}\boldsymbol{\xi}_{j,l}, \qquad \big((N^m)^T\underline{\boldsymbol{\xi}}\big)_j=\sum_{i,k=1}^{N_h,3}N^m_{i,k,j}\boldsymbol{\xi}_{i,k}, \qquad \big(T^m_\mu\underline{\chi}\big)_{i,k}=\sum_{j=1}^{N_h}T^m_{\mu;i,k,j}\chi_j.$$

In the same spirit we define the vectors

$$\begin{aligned}
&\underline{b}^m=\{b^m_i\}_{i=1}^{N_h}, && b^m_i:=\mathcal{N}(\Gamma^m_h,\boldsymbol{\nu}^m_h;\check{\boldsymbol{x}}^m_h,\phi_i)+\tau^m\mathcal{B}(\Gamma^m_h,\check{\kappa}^m_h,\boldsymbol{Q}^m_h,\check{c}^m_h;\phi_i),\\
&\underline{l}^m=\{l^m_i\}_{i=1}^{N_h}, && l^m_i:=\tau^m\mathcal{L}(\Gamma^m_h;\phi_i),\\
&\underline{k}^m=\{k^m_i\}_{i=1}^{N_h}, && k^m_i:=\tau^m\mathcal{K}(\Gamma^m_h,\check{\kappa}^m_h;\phi_i),\\
&\underline{h}^m=\{h^m_i\}_{i=1}^{N_h}, && h^m_i:=\tau^m\mathcal{H}(\Gamma^m_h,\check{\kappa}^m_h,\check{c}^m_h;\phi_i),\\
&\underline{t}^m_\mu=\{t^m_{\mu;i}\}_{i=1}^{N_h}, && t^m_{\mu;i}:=\tau^m\mathcal{T}_\mu(\Gamma^m_h,\boldsymbol{\tau}^m_{\mu,h};\check{\boldsymbol{x}}^m_h,\phi_i), \qquad \mu=1,2,\\
&\underline{s}^m=\{s^m_i\}_{i=1}^{N_h}, && s^m_i:=\varepsilon\omega\mathcal{M}(\Gamma^m_h;\phi_i,\check{c}^m_h)+\tau^m\mathcal{S}(\Gamma^m_h,\check{\kappa}^m_h,\check{c}^m_h;\phi_i),\\
&\underline{p}^m=\{p^m_i\}_{i=1}^{N_h}, && p^m_i:=\tau^m\mathcal{P}(\Gamma^m_h,\check{c}^m_h;\phi_i).
\end{aligned}$$

In each time step we have to solve the system consisting of (3.4a), (3.3a), (3.4b), (3.3c) plus the constraints. With the above definitions, the system is equivalent to

$$\begin{pmatrix} A^m & N^m & \alpha^m T_1^m & \alpha^m T_2^m & 0 \\ (N^m)^T & \tau^m C^m & 0 & 0 & 0 \\ \alpha^m (T_1^m)^T & 0 & -\alpha^m \tau^m M^m & 0 & 0 \\ \alpha^m (T_2^m)^T & 0 & 0 & -\alpha^m \tau^m M^m & 0 \\ 0 & 0 & 0 & 0 & I^m \end{pmatrix} \begin{pmatrix} \underline{\boldsymbol{x}}^{m+1} \\ \underline{\kappa}^{m+1} \\ \underline{\beta}_1^{m+1} \\ \underline{\beta}_2^{m+1} \\ \underline{c}^{m+1} \end{pmatrix}$$
$$= \begin{pmatrix} 0 \\ \underline{b}^m \\ \underline{t}_1^m \\ \underline{t}_1^m \\ \underline{s}^m \end{pmatrix} - \lambda_{V,h}^{m+1} \begin{pmatrix} 0 \\ \underline{l}^m \\ 0 \\ 0 \\ 0 \end{pmatrix} - \lambda_{A,h}^{m+1} \begin{pmatrix} 0 \\ \underline{k}^m \\ 0 \\ 0 \\ 0 \end{pmatrix} - \lambda_{c,h}^{m+1} \begin{pmatrix} 0 \\ \underline{h}^m \\ 0 \\ 0 \\ \underline{p}^m \end{pmatrix}$$

plus the constraints. Let us define

$$J^m := \begin{pmatrix} A^m & N^m & \alpha^m T_1^m & \alpha^m T_2^m & 0 \\ (N^m)^T & \tau^m C^m & 0 & 0 & 0 \\ \alpha^m (T_1^m)^T & 0 & -\alpha^m \tau^m M^m & 0 & 0 \\ \alpha^m (T_2^m)^T & 0 & 0 & -\alpha^m \tau^m M^m & 0 \\ 0 & 0 & 0 & 0 & I^m \end{pmatrix}, \tag{A.1a}$$

$$y^{m+1} := \begin{pmatrix} \underline{\boldsymbol{x}}^{m+1} \\ \underline{\kappa}^{m+1} \\ \underline{\beta}_1^{m+1} \\ \underline{\beta}_2^{m+1} \\ \underline{c}^{m+1} \end{pmatrix}, \qquad r^m := (J^m)^{-1} \begin{pmatrix} 0 \\ \underline{b}^m \\ \underline{t}_1^m \\ \underline{t}_1^m \\ \underline{s}^m \end{pmatrix}, \tag{A.1b}$$

$$d_V^m := \begin{pmatrix} \underline{\boldsymbol{u}}_V^m \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} := (J^m)^{-1} \begin{pmatrix} 0 \\ \underline{l}^m \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.1c}$$

$$d_A^m := \begin{pmatrix} \underline{\boldsymbol{u}}_A^m \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{pmatrix} := (J^m)^{-1} \begin{pmatrix} 0 \\ \underline{k}^m \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{A.1d}$$

$$d_c^m := \begin{pmatrix} \underline{\boldsymbol{u}}_c^m \\ \cdot \\ \cdot \\ \cdot \\ \underline{\zeta}_c^m \end{pmatrix} := (J^m)^{-1} \begin{pmatrix} 0 \\ \underline{h}^m \\ 0 \\ 0 \\ \underline{p}^m \end{pmatrix}, \tag{A.1e}$$

where we remark that the last N_h entries of d_V^m and d_A^m vanish and $\underline{\zeta}_c^m = (I^m)^{-1} \underline{p}^m$ because of the block structure of J^m. In order to solve the above four linear systems we used a direct method [13]. Computing the matrix factorizations of the two blocks of J^m turned

out to be the most costly part of the whole solution algorithm, typically requiring 80-90% of the computation time.

We then obtain

$$y^{m+1} = r^m - \lambda_{V,h}^{m+1} d_V^m - \lambda_{A,h}^{m+1} d_A^m - \lambda_{c,h}^{m+1} d_c^m, \tag{A.2}$$

and the Lagrange multipliers have to be such that

$$\mathcal{C}_i(\Gamma_h^{m+1}, \check{c}_h^{m+1}) = 0, \qquad i \in \{V, A, c\}. \tag{A.3}$$

B Solution algorithm

Eq. (A.2) suggests to consider y^{m+1} and, hence, also Γ_h^{m+1} as a function of the unknown λ_i^{m+1} which have to be a root of (A.3), i.e., we have to solve

$$0 = f(\lambda^{m+1}) := \begin{pmatrix} \bar{\mathcal{C}}_V(\lambda^{m+1}) \\ \bar{\mathcal{C}}_A(\lambda^{m+1}) \\ \bar{\mathcal{C}}_c(\lambda^{m+1}) \end{pmatrix},$$

where $\lambda^{m+1} = (\lambda_{V,h}^{m+1}, \lambda_{A,h}^{m+1}, \lambda_{c,h}^{m+1})$ and

$$\bar{\mathcal{C}}_i(\lambda^{m+1}) := \mathcal{C}_i(\Gamma_h^{m+1}(\lambda^{m+1}), \check{c}_h^{m+1}(\lambda^{m+1})), \qquad i \in \{V, A, c\}.$$

As proposed in [24] we perform a quasi-Newton iteration for this purpose. For the Newton method we would need the derivative of f. We see from (A.2) and (A.1c) that a change in $\lambda_{V,h}^{m+1}$ leads to a change of $\underline{\boldsymbol{x}}^{m+1}(\lambda^{m+1})$ in the direction $-\underline{\boldsymbol{u}}_V^m$. Writing $-\boldsymbol{u}_{V,h}^m \in \boldsymbol{S}_h(\Gamma_h^m)$ for the finite element function corresponding to $-\underline{\boldsymbol{u}}_V^m$ and defining $\check{\boldsymbol{u}}_{V,h}^m \in \boldsymbol{S}_h(\Gamma_h^{m+1}(\lambda^{m+1}))$ in the usual way by setting $\check{\boldsymbol{u}}_{V,i,k}^m := \boldsymbol{u}_{V,i,k}^m$, $i = 1, \cdots, N_h$, $k = 1,2,3$, we see that the change in $\lambda_{V,h}^{m+1}$ can also be understood as a deformation of $\Gamma_h^{m+1}(\lambda^{m+1})$ in the direction $-\check{\boldsymbol{u}}_{V,h}^m$. Hence, $\partial_{\lambda_{V,h}^{m+1}} \bar{\mathcal{C}}_V(\lambda^{m+1})$ is the change of the volume enclosed by $\Gamma_h^{m+1}(\lambda^{m+1})$ when deforming in direction $-\check{\boldsymbol{u}}_{V,h}^m$. Identity (2.6d) suggests that (see [24] for the details)

$$\partial_{\lambda_{V,h}^{m+1}} \bar{\mathcal{C}}_V(\lambda^{m+1}) = \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \boldsymbol{\nu}_h^{m+1}(\lambda^{m+1}) \cdot \left(-\check{\boldsymbol{u}}_{V,h}^m\right).$$

Similarly, changes in $\lambda_{A,h}^{m+1}$ and $\lambda_{c,h}^{m+1}$ correspond to deformations in directions $-\check{\boldsymbol{u}}_{A,h}^m$ and $-\check{\boldsymbol{u}}_{c,h}^m$, respectively, so that

$$\partial_{\lambda_{A,h}^{m+1}} \bar{\mathcal{C}}_V(\lambda^{m+1}) = \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \boldsymbol{\nu}_h^{m+1}(\lambda^{m+1}) \cdot \left(-\check{\boldsymbol{u}}_{A,h}^m\right),$$

$$\partial_{\lambda_{c,h}^{m+1}} \bar{\mathcal{C}}_V(\lambda^{m+1}) = \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \boldsymbol{\nu}_h^{m+1}(\lambda^{m+1}) \cdot \left(-\check{\boldsymbol{u}}_{c,h}^m\right).$$

For computing $\partial_{\lambda_{i,h}^{m+1}}\bar{\mathcal{C}}_A(\lambda^{m+1})$ we want to proceed analogously, using identity (2.6e). But instead of computing the variation of the area of a quadratic triangulated surface exactly as in [24] we use the mean curvature $\kappa_h^{m+1}(\lambda^{m+1})$ which is contained in $y^{m+1}(\lambda^{m+1})$ by means of $\underline{\kappa}^{m+1}(\lambda^{m+1})$:

$$\begin{aligned}
\partial_{\lambda_{V,h}^{m+1}}\bar{\mathcal{C}}_A(\lambda^{m+1}) &\approx \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \big(-\check{\kappa}_h^{m+1}(\lambda^{m+1})\boldsymbol{\nu}_h^{m+1}(\lambda^{m+1})\big)\cdot\big(-\check{\boldsymbol{u}}_{V,h}^m\big),\\
\partial_{\lambda_{A,h}^{m+1}}\bar{\mathcal{C}}_A(\lambda^{m+1}) &\approx \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \big(-\check{\kappa}_h^{m+1}(\lambda^{m+1})\boldsymbol{\nu}_h^{m+1}(\lambda^{m+1})\big)\cdot\big(-\check{\boldsymbol{u}}_{A,h}^m\big),\\
\partial_{\lambda_{c,h}^{m+1}}\bar{\mathcal{C}}_A(\lambda^{m+1}) &\approx \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \big(-\check{\kappa}_h^{m+1}(\lambda^{m+1})\boldsymbol{\nu}_h^{m+1}(\lambda^{m+1})\big)\cdot\big(-\check{\boldsymbol{u}}_{c,h}^m\big).
\end{aligned}$$

In order to approximate the partial derivatives of $\bar{\mathcal{C}}_c(\lambda^{m+1})$, identity (2.6f) suggests to use $\kappa_h^{m+1}(\lambda^{m+1})$ again but also $c_h^{m+1}(\lambda^{m+1})$. Recall that η in (2.6f) is the variation of c along the trajectories defined by the deformation field. From (A.1c) and (A.1d) we see if $\lambda_{V,h}^{m+1}$ or $\lambda_{A,h}^{m+1}$ change then these variations of the order parameter vanish but, in view of (A.1e) and (A.2), changing $\lambda_{c,h}^{m+1}$ leads to a variation of $\check{c}_h^{m+1}(\lambda^{m+1})$ in direction $-\check{\zeta}_{c,h}^m \in S_h(\Gamma_h^{m+1}(\lambda^{m+1}))$ which is the finite element function corresponding to the coefficient vector $-\underline{\zeta}_c^m$. Hence,

$$\begin{aligned}
\partial_{\lambda_{V,h}^{m+1}}\bar{\mathcal{C}}_c(\lambda^{m+1}) &\approx \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \big(h(\check{c}_h^{m+1})\check{\kappa}_h^{m+1}\boldsymbol{\nu}_h^{m+1}\big)(\lambda^{m+1})\cdot\check{\boldsymbol{u}}_{V,h}^m,\\
\partial_{\lambda_{A,h}^{m+1}}\bar{\mathcal{C}}_c(\lambda^{m+1}) &\approx \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \big(h(\check{c}_h^{m+1})\check{\kappa}_h^{m+1}\boldsymbol{\nu}_h^{m+1}\big)(\lambda^{m+1})\cdot\check{\boldsymbol{u}}_{A,h}^m,\\
\partial_{\lambda_{c,h}^{m+1}}\bar{\mathcal{C}}_c(\lambda^{m+1}) &\approx \int_{\Gamma_h^{m+1}(\lambda^{m+1})} \big(h(\check{c}_h^{m+1})\check{\kappa}_h^{m+1}\boldsymbol{\nu}_h^{m+1}\big)(\lambda^{m+1})\cdot\check{\boldsymbol{u}}_{c,h}^m\\
&\qquad - \int_{\Gamma_h^{m+1}(\lambda^{m+1})} h'(\check{c}_h^{m+1}(\lambda^{m+1}))\check{\zeta}_c^m.
\end{aligned}$$

We denote the approximation of $Df(\lambda^{m+1})$ by $H(\lambda^{m+1})$. The quasi-Newton iteration to compute the values λ^{m+1} reads

$$\lambda^{m+1,k+1} = \lambda^{m+1,k} - \big(H(\lambda^{m+1,k})\big)^{-1} f(\lambda^{m+1,k}). \tag{B.1}$$

The values $\lambda^{m+1,0}=\lambda^m$, $\lambda^{0,0}=0$ were always taken as initial choice. We stopped the iteration when the values $\bar{\mathcal{C}}_V(\lambda^{m+1,k+1})/V$, $\bar{\mathcal{C}}_A(\lambda^{m+1,k+1})/(A_1+A_2)$ and $\bar{\mathcal{C}}_c(\lambda^{m+1,k+1})/(A_1-A_2)$ were reduced below a given tolerance that usually was set to about 10^{-8}. Damping has never been required to ensure convergence of the quasi-Newton method.

References

[1] D. Andelman, T. Kawakatsu and K. Kawasaki, Equilibrium shape of two-component unilamellar membranes and vesicles, Europhys. Lett., 19 (1992), 57–62.

[2] J. W. Barrett, H. Garcke and R. Nürnberg, On the parametric finite element approximation of evolving hypersurfaces in $\mathbb{R}^3$, J. Comput. Phys., 227 (2008), 4281–4307.
[3] J. W. Barrett, H. Garcke and R. Nürnberg, Parametric approximation of Willmore flow and related geometric evolution equations, SIAM J. Sci. Comput., 31 (2008), 225–253.
[4] J. W. Barrett, H. Garcke and R. Nürnberg, Parametric approximation of surface clusters driven by isotropic and anisotropic surface energies, Interf. Free Bound., 12 (2010), 187–234.
[5] T. Baumgart, S. Das, W. Webb and J. Jenkins, Membrane elasticity in giant vesicles with fluid phase coexistence, Biophys. J., 89 (2005), 1067–1084.
[6] T. Baumgart, S. Hess and W. Webb, Imaging coexisting fluid domains in biomembrane models coupling curvature and line tension, Nature, 425 (2003), 821–824.
[7] J. F. Blowey and C. M. Elliott, The Cahn-Hilliard gradient theory for phase separation with non–smooth free energy part ii: Numerical analysis, Euro. J. Appl. Math., 3 (1993), 147–179.
[8] S. Brenner and L. Scott, The Mathematical Theory of Finite Element Methods, No. 15 in Texts in Applied Mathematics, Springer, Third ed., 2007.
[9] F. Campelo, J.-M. Allain and M. B. Amar, Periodic lipidic membrane tubes, Europhys. Lett., 77 (2007), 38006.
[10] P. Canham, The minimum energy of bending as a possible explanation of the biconcave shape of the red blood cell, J. Theor. Biol., 26 (1970), 61–81.
[11] L.-Q. Chen, Phase-field models for microstructure evolution, Ann. Rev. Mat. Res., 32 (2002), 113–140.
[12] S. Das, J. Jenkins and T. Baumgart, Neck geometry and shape transitions in vesicles with co-existing fluid phases: role of gaussian curvature stiffness vs. spontaneous curvature, Europhys. Lett., 86 (2009), 48003–1–6.
[13] T. A. Davies, Algorithm 832: Umfpack, an unsymmetric-pattern multifrontal method, ACM Trans. Math. Software, 30 (2004), 196–199.
[14] K. Deckelnick, G. Dziuk and C. M. Elliott, Computation of geometric partial differential equations and mean curvature flow, Acta Numer., 14 (2005), 139–232.
[15] A. Demlow, Higher-order finite element methods and pointwise error estimates for elliptic problems on surfaces, SIAM J. Numer. Anal., 47 (2009), 805–827.
[16] Q. Du, C. Liu and X. Wang, A phase field approach in the numerical study of the elastic bending energy for vesicle membranes, J. Comput. Phys., 198 (2004), 450–468.
[17] Q. Du, C. Liu and X. Wang, Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions, J. Comput. Phys., 212 (2006), 757–777.
[18] G. Dziuk, Finite elements for the Beltrami operator on arbitrary surfaces, in Partial Differential Equations and Calculus of Variations, S. Hildebrandt and R. Leis, eds., Vol. 1357 of Lecture Notes in Mathematics, Springer, 1988, 142–155.
[19] G. Dziuk, An algorithm for evolutionary surfaces, Numerische Mathematik, 58 (1991), 603–611.
[20] G. Dziuk, Computational parametric Willmore flow, Numerische Mathematik, 111 (2008), 55–80.
[21] G. Dziuk and U. Clarenz, Numerical methods for conformally parametrized surfaces, CPDw04-Interphase 2003: Numerical Methods for Free Boundary Problems, workshop at the Isaac Newton Institute, `http://www.newton.cam.ac.uk/webseminars/pg+ws/2003/cpd/cpdw04/0415/dziuk/`, (2003).
[22] G. Dziuk and C. M. Elliott, Finite elements on evolving surfaces, IMA J. Numer. Anal., 25 (2007), 385–407.
[23] G. Dziuk and C. M. Elliott, Surface finite elements for parabolic equations, J. Comput. Math.,

25 (2007), 385–407.
[24] C. Elliott and B. Stinner, Modeling and computation of two phase geometric biomembranes using surface finite elements, J. Comput. Phys., 229 (2010), 6585–6612.
[25] C. Elliott and B. Stinner, A surface phase field model for two-phase biological membranes, SIAM J. Appl. Math., 70 (2010), 2904–2928.
[26] C. M. Elliott and A. M. Stuart, The global dynamics of discrete semilinear parabolic equations, SIAM J. Numer. Anal., 30 (1993), 1622–1663.
[27] E. Evans, Bending resistance and chemically induced moments in membrane bilayers, Biophys. J., 14 (1974), 923–931.
[28] F. Feng and W. Klug, Finite element modeling of liquid bilayer membranes, J. Comput. Phys., 220 (2006), 394–408.
[29] H. Garcke, B. Nestler, B. Stinner and F. Wendler, Allen-Cahn systems with volume constraints, Math. Mod. Meth. Appl. Sci., 18 (2008), 1347–1381.
[30] C.-J. Heine, Computations of Form and Stability of Rotating Drops with Finite Elements, PhD thesis, Faculty for Mathematics, Informatics and Natural Sciences, University of Aachen, 2003.
[31] C.-J. Heine, Isoparametric finite element approximation of curvature on hypersurfaces, Fakultät für Mathematik und Physik, University of Freiburg, Preprint, 26 (2004).
[32] W. Helfrich, Elastic properties of lipid bilayers: theory and possible experiments, Z. Naturforschung, C28 (1973), 693–703.
[33] M. Helmers, Snapping elastic curves as a one-dimensional analogue of two-component lipid bilayers, to appear in Math. Mod. Meth. Appl. Sci., DOI: 10.1142/S0218202511005234, (2011).
[34] F. Jülicher and R. Lipowsky, Domain-induced budding of vesicles, Phys. Rev. Lett., 70 (1993), 2964–2967.
[35] F. Jülicher and R. Lipowsky, Shape transformations of vesicles with intramembrane domains, Phys. Rev. E, 53 (1996), 2670–2683.
[36] T. Kawakatsu, D. Andelman, K. Kawasaki and T. Taniguchi, Phase transitions and shapes of two component membranes and vesicles I: strong segregation limit, J. Phys. II (France), 3 (1993), 971–997.
[37] J. S. Lowengrub, A. Rätz and A. Voigt, Phase-field modeling of the dynamics of multicomponent vesicles: spinodal decomposition, coarsening, budding and fission, Phys. Rev. E, 79 (2009), 031926–1–13.
[38] L. Ma and W. Klug, Viscous regularization and *r*-adaptive remeshing for finite element analysis of lipid membrane mechanics, J. Comput. Phys., 227 (2008), 5816–5835.
[39] A. Schmidt and K. G. Siebert, Design of adaptive finite element software: the finite element toolbox ALBERTA, Vol. 42 of Lecture Notes in Computational Science and Engineering, Springer, 2005.
[40] U. Seifert, Configurations of fluid membranes and vesicles, Adv. Phys., 46 (1997), 1–137.
[41] T. Taniguchi, Shape deformations and phase separation dynamics of two-component vesicles, Phys. Rev. Lett., 76 (1996), 4444–4447.
[42] T. Taniguchi, K. Kawasaki, D. Andelman and T. Kawakatsu, Phase transitions and shapes of two component membranes and vesicles II: weak segregation limit, J. Phys. II (France), 4 (1994), 1333–1362.
[43] S. L. Veatch and S. L. Keller, Separation of liquid phases in giant vesicles of ternary mixtures of phospholipids and cholesterol, Biophys. J., 85 (2003), 3074–3083.
[44] X. Wang and Q. Du, Modelling and simulations of multi-component lipid membranes and open membranes via diffuse interface approaches, J. Math. Biol., 56 (2008), 347–371.