A LEVEL SET APPROACH TO SHAPE DYNAMICS OF VESICLES

by

Tiankui Zhang

A Dissertation Submitted to the Faculty of the

DEPARTMENT OF PHYSICS

In Partial Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

In the Graduate College

THE UNIVERSITY OF ARIZONA

2020

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by: Tiankui Zhang, titled: A Level Set Approach to Shape Dynamics of Vesicles and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

______________________________ Charles William Wolgemuth (Department of Physics)	Date: ____________
______________________________ Srinivas Manne (Department of Physics)	Date: ____________
______________________________ Charles A Stafford (Department of Physics)	Date: ____________
______________________________ John A Milson (Department of Physics)	Date: ____________
______________________________ Calvin Zhang-Molina (Department of Mathematics)	Date: ____________

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

______________________________ Date: ____________
Charles William Wolgemuth
Dissertation Committee Chair
Department of Physics

Acknowledgements

My deepest gratitude goes to my mother 袁利群 (Yuan Li Qun) and my father 张军 (Zhang Jun). They brought me to this beautiful world and raised me up with love and care. They understand neither English nor physics, but taught me the most valuable lessons about morality and righteousness. I feel privileged to grow up as a curious and carefree child under their wings.

I am also extremely grateful to the Department of Physics at University of Arizona for offering me this opportunity to study and research in the United States. The past six years is and will be the most unique experience in my life. I feel most indebted to my advisor Professor Charles Wolgemuth, who inspires me with a cheerful personality, guides me with patience and selfless, supports me during difficult times. Without his teachings and support, I would not be able to stick through the past six years. I also need to thank many friendly administrative staffs in the department, Matthew Jones, Gardie Lueders, Carmen Montijo, Rohit Singh and many others whose service makes life much easier for the graduate students. I also need to thank many peer and senior students. I need to thank Ge XueZhen and Xiong Kun for taking me to grocery stores during the first year of my stay in Tucson. I need to thank Cheng Yihong for teaching me how to drive a car, which is necessary to survive in Tucson. I need to thank Mike for teaching me microscopy techniques. I need to thank both Alex for teaching me how to shoot a gun.

In the end, I have to thank those pioneering scientists and engineers who expanded our knowledge of this world through their creative and arduous work, and those who are building the modern civilization with their hands and perspiration. I feel very fortunate to be able to make a very small piece of contribution to our understanding of ourselves and this world.

I would like to end with a Chinese old saying from more than two thousand years ago，celebrating all travellers on this never ending journey into the vast unkonwns about nature

高山仰止 we may gaze up to the mountain's brow；
景行行止 we may travel along the greater road.
虽不能至 although we cannot hope to reach the goal，
心向往之 still we may push on with our untiring soul.

Table of contents

List of Figures

List of Tables

Abstract

Cell membranes are crucial to the life of cells and vesicles are important model systems for cell membranes. In this thesis, we aim to understand shape transformation of vesicles with theoretical tools from differential geometry and numerical techniques within the framework of level set method.

In Chapter 1, a breify introduction to membrane structure and membrane proteins is given. In Chapter 2, we introduce the mathemtical language for the description of surfaces and curves. In Chapter 3, variation for the Hamilton beyond Helfrich model is presented. In Chapter 4, we show how to discretize geometries of surface and curves with the level set method. A novel high order scheme for Hamilton-Jacobi equation with level set-defined boundary conditions is developed. In Chapter 5, we demonstrate accuray of our numerical scheme introduced in Chapter 4 with numerical experiments. In Chapter 6, we develop a semi-implicit level set method for the dynamics of single phase vesicles, biphasic vesicles and protine kinetics in three dimensional space. We explore how different parameters and physical conditions affect shape of vesicles.

Chapter 1 Introduction

1.1 Membrane Structure and Membrane Proteins

Cells are the fundamental units of all living organisms on earth, ranging from an E coli (*Escherichia coli*) of several microns to a blue whale up to 30 meters.[1.1] In view of the diversity of forms of life on our planet, the similarity of fundamental mechanisms across different species is extraordinary.

1.1.1 The Lipid Bilayer

Cell membranes are crucial to the life of cells. Cells are literally packed with membranes. The most important function of membranes is topological. Membranes, acting as a permeability barrier, compartmentalize cells into different organelles for different tasks. It is estimated that there are about 10^{14} cells in a human being and the total surface of membranes can cover an area of about $100\,\mathrm{km}^2$ [mouritsen2005life] [1.2]. All biological membranes consists of a very thin lipid bilayer (5nm) and protein molecules, held together mainly by noncovalent interactions. The most abundant membrane lipids are the **phospholipids**, which have a polar head group containing a phosphate group and two *hydrocarbon tails* with between 14 and 24 carbon atoms. In the physiological state, typical length of a lipid molecule is 2 nm and cross-sectional area 0.7 nm^2. Tails with at least one *unsaturated cis*-double bonds have a kink, which makes them shorter and more rigid than *saturated* tails. Major lipids in cell membranes include **phosphoglycerides**, **sphingolipids**, **glycolipids** and **cholesterol**. Among them, cholesterol takes the most peculiar shape, containing a rigid ring structure, attached to a single polar hydroxyl group and a short nonpolar hydrocarbon chain. It is interesting to observe that only negatively charged lipids are used by nature to construct membranes.

Lipid molecules are **amphiphilic**, i.e., they have a **hydrophilic** (water-loving) *polar* end and a **hydrophobic** (water-fearing) *nonploar* end. They are nature's own surfactants. The hydrophilic ends dissolve readily in water because they can form either favorable electrostatic interactions (if they are charged) or hydrogen bonds (if they are polar and uncharged) with water molecules . The uncharged and nonpolar ends cannot form energetically favorable interactions with water molecules and adjacent water molecules are forced into icelike cages surrounding hydrophobic ends. Those cage structures are more ordered, with lower entropy and assumes higher free energy. Exposed to water, amphiphilic molecules reorganize themselves to shield their hydrophobic ends from the water. Depending on their packing parameters[1.3] P, they can form either spherical *micelles* ($P < 1/3$), with tails inward, or *lamellar bilayers* ($P = 1$), with the hydrophobic tails sandwiched between the hydrophilic head groups. Line tension on the edge of a phospholipid bilayer can then seal it into a closed compartment known as an artificial *vesicle*.

The lipid bilayer is a two dimensional fluid, where lipid molecules can diffuse rapidly in the plane of the membrane. The fluidity of a lipid bilayer depends on both its composition and temperature. Lipids with shorter hydrocarbon chains or double bonds are more fluidic as they are more difficult

1.1. The material of this section comes primarily from Chapter 10 of [alberts2014molecular] (the bible of molecular cell biology) and [mouritsen2005life] (a monograph on lipids). Due to copyright issues, no graph is included and the reader is referred to the above mentioned texts for more detailed textual and graphical introduction on membrane structure and membrane proteins.

1.2. In comparison, the campus area of University of Arizona is about $1.5\,\mathrm{km}^2$ while Wuhan University covers an area of $3.5\,\mathrm{km}^2$.

1.3. The packing parameter P of a lipid molecules is defined by $P = v / a \cdot l$, where v is the volume of the lipid, a area of the head group, l length of the lipid. The packing parameter is a measure of the compatibility between the size of the head group and the size of the hydrophobic tail.

to pack compactly. A synthetic bilayer made from a single type of phospholipid changes from a **Ld** (liquid-disordered) phase to a **So** (solid-ordered) phase at a characteristic temperature, which is know as **thermotropic phase transition**. If the bilayer consists of more than one type of phospholipid, lipid molecules of the same kind tend to attract each other, separating out as a distinct domain with characteristic physical properties, know as **lateral phase separation**. Those segregated domains are termed **lipid rafts** which can provide specific protein binding and reaction sites. Since different phases of bilayer domains have different phase transition temperatures, the transition from a liquid disordered phase to a solid ordered phase now takes place over a range of temperatures. At a certain temperature, Ld phase and Lo phase can coexist, know as **phase coexistence** or **phase equilibrium**.

Cholesterol is an important modulator of the properties of lipid bilayers. Being an amphiphilic molecule, cholesterol can be easily incorporated into lipid bilayers with its -OH head group at the bilayer-water interface and the steroid skeleton inside the hydrophobic core. Mixed with phospholipids, cholesterol spans less than one monolayer leaflet of a typical bilayer and has a remarkable dual effect on membranes. One the one side, due to its peculiar shape and structure, cholesterol can not be compactly packed into the solid ordered phase of a lipid bilayer. On the other side, a loose packing in the liquid disordered phase is not energetically favorable. This frustration is released by the introduction of a new phase, the **Lo** (liquid-ordered) phase, first proposed by John Hjort Ipsen in 1987. On the one hand, lipid chains in the liquid-ordered phase have less conformational freedom, which makes the lipid bilayer thicker, more grid and less permeable to small water-soluble molecules. On the other hand, the liquid-ordered phase is still a liquid with considerable positional disorder and high lateral mobility of the membrane molecules, which is required for membrane function. Table 1.1 summarizes for different phases of a lipid bilayer the differences in positional freedom and internal freedom. In addition, the liquid-ordered phase can be stable below the phase transition temperature and cholesterol thus plays the role of an anti-freeze agent. This ability to stabilize the liquid ordered phase over a wide range of temperatures is one of the most remarkable properties of cholesterol.

	positional (translational) freedom	internal (conformational) freedom
liquid-disordered phase	yes	yes
liquid-ordered phase	yes	no
solid-ordered phase	no	no

Table 1.1. How the liquid-disordered phase, liquid-ordered phase, solid-ordered phase of a lipid bilayer differ in terms of positional freedom and internal freedom.

1.1.2 Membrane Proteins

Even though there are no genes coding for lipids, more than half of our genes are related to proteins that are either integral membrane proteins or bind peripherally to membranes. It is those membrane related proteins that perform most of the membrane's specific tasks and give each type of cell membrane its characteristic functional properties. Lipids grease the molecular machine controlled and run by proteins and DNA.

Membrane proteins can be associated with the lipid bilayer through various molecular interactions. Some of these forces are weak and nonspecific, others can be strong and long-ranged electrostatic forces. Formation of hydrogen bonds and even chemical bonds may also be involved. A huge and important class of membrane proteins spans the lipid bilayer with at least one membrane-spanning domain. The transmembrane part has hydrophobic amino acids interacting with the hydrophobic tails of the lipid molecule and sometimes hydrophilic or charged amino acids toward their interior, while outside the membrane water soluble part often carries the functional units of the proteins. The match of size of the hydrophobic transmembrane region from a membrane protein and the hydrophobic core of the lipid bilayer could be a way to sort different proteins to different parts of the membrane. The genes coding for the transmembrane domain of membrane proteins is one of the evolutionarily most conserved sequence, suggesting a universal lipid-protein interaction mechanism.

Membrane proteins are related to many functions of cells, e.g., transport, biochemical signaling, energy transduction, receptor-ligand interactions, nerve activity. In addition, membrane-bending proteins can deform bilayers. Some insert hydrophobic protein domains or have an amphiphilic α helix attached to one monolayer of the lipid bilayer, increasing the area of only one leaflet and creating a physical stress to bend the membrane. Some coat proteins form rigid scaffolds that deform the membrane or stabilize an already bent membrane. Some proteins cause membrane lipids of a specific packing parameter to cluster, locally inducing a particular spontaneous curvature.

Membrane proteins can rotate about an axis perpendicular to the plane of the bilayer (**rotational diffusion**) and move laterally within the membrane (**lateral diffusion**). Proteins can also influence the local structure and composition of the bilayers. On the other hand, physical curvature stress and lateral stress profile of the lipid bilayers can modulate structure and therefore function of the proteins. Research on protein-lipid interaction on different scales is still a rapidly developing field.

1.2 Vesicles : Model Systems for Cell Membranes

Unilamellar liposomes (also know as vesicles) are important model systems for cell membranes. Vesicles have been used to study membrane bending deformations [seifert1997configurations], lipid phase separation and phase coexistence [veatch2003separation], protein interaction with biomembranes [baumgart2011thermodynamics] and other physical processes on a cellular or subcellular level. These studies not only help us better understand biological membranes of living cells, but also provide insights into creating innovative drug delivery systems for treating cancer and other human diseases [noyhouzer2016ferrocene].

Vesicles have relatively simple lipid compositions and live in a more homogeneous environment than biomembranes in vivo [harayama2018understanding]. Still, complex shape transitions can happen by controlling temperature and osmotic pressure [yanagisawa2008shape], by mixing saturated and unsaturated lipids with cholesterol [baumgart2005membrane], by adding curvature inducing and sensing proteins to its environment [rossman2010influenza]. Remarkably, many observed vesicle shapes can be numerically computed by minimizing the Helfrich bending energy [helfrich1973elastic]. The Helfrich Hamiltonian can be generalized and augmented in a variety of ways to describe different membrane systems and dynamics. For instance, budding transitions of vesicles can be explained via adding the effect of area-difference elasticity [miao1994budding]. Stomatocyte-discocyte-echinocyte sequence of the human red blood cell can be numerically obtained when stretch and shear elasticity of the protein-based cytoskeleton is included [hw2002stomatocyte]. Shape transformations of vesicles with intramembrane domains can be captured by taking into account domain boundary energy [julicher1996shape]. Tank-treading of vesicles under shear flow is observed in simulation when vesicle deformation is coupled with hydrodynamic effects [biben2005phase]. More recently, endocytosis is studied where membrane shape transformation is affected by membrane protein kinetics [lowengrub2016numerical].

1.3 Structure of Thesis

Despite all of the efforts, it remains challenging to accurately simulate general lipid bilayers dynamics, which can involve lateral inhomogeneity in lipid composition, protein interaction and topological shape changes. One reason lies in the fact that bending forces for soft surfaces involve up to fourth order derivatives of the surface position, which can be difficult to compute convergently [guckenberger2016bending]. In addition, lipid bilayers can pinch off and fuse in important cellular events such as endocytosis, viral escaping and nerve signal transduction. Topological changes in those processes present difficulty to numerical handling due to presence of singularities. Moreover, membrane functions usually depend on formation of chemically and physically distinct local regions called *lipid rafts* [lingwood2010lipid], which maintain quite unique chemical compositions and physical properties. An universal physical principle and mathematical description for the interplay between membrane lateral inhomogeneity represented as lipid rafts and membrane geometry dynamics on a mesoscale is still lacking.

In this thesis, we aim to address some of the theoretical and numerical problems mentioned above. In Chapter 2, we give a brief introduction to the tensor language used to describe surfaces and curves. In Chapter 3, we then calculate bending forces of a membrane with locally varying bending moduli and line forces for a general phase boundary energy. Next, in Chapter 4, we give a concise introduction to the level set framwork. Numerical discretizations of surface and curve geometries are presented. Dynamical equations governing motion of surfaces and curves are presented. What's more, we develop a novel sixth-order accurate numerical scheme for Sussman's reinitialization equation and other Hamilton-Jacobi equations with a level set defined boundary condition. In Chapter 5, numerical experiments verifying our sixth order accurate schemes are presented. In particular, we show how to reinitialize the level set function, extend a scalar field away from the level surface and compute geodesics on the level surface with sixth order accuracy. It is important to note that we can thus compute bending forces of the Helfrich model with second order accuracy. In Chapter 6, partial differential equations governing shape dynamics of single and biphasic vesicles with and without protein interaction are presented. We explore effects of different parameters on shape of vesicles.

Even though there have been numerous studies in this area, our development is unique in several aspects. First, the vesicle membrane is assumed to be nonuniform and the resulting bending forces will have nontrivial tangential components. Second, a general Hamiltonian for phase boundary line energy is proposed and its variation with respect to position is calculated. Third, we adopt a highly accurate numerical scheme developed in [zhang2020sixth] to calculate bending forces with second order accuracy. Fourth, an adaptive semi-implicit scheme for the dynamical equation is presented. We make no assumptions on symmetry of the system and our simulation is carried out in fully three dimensional space.

Chapter 2

The Tensor Description of Embedded Surfaces and Curves

In this thesis, we treat vesicles as infinitely thin two dimensional surfaces embedded in three dimensional space, as is shown in Figure 2.1. This simplification can be justified by the fact that vesicle thickness (about 5 nm) is negligible compared compared with its size (several microns). In this chapter, we introduce necessary mathematical background on the tensor description of surfaces and curves, which are adopted from Pavel Grinfeld's textbook [grinfeld2013introduction] and are used throughout this thesis. We first introduce notations for the Euclidean space embedding surfaces and curves. Then we explain how to describe geometries of surfaces and curves in the tensor language.

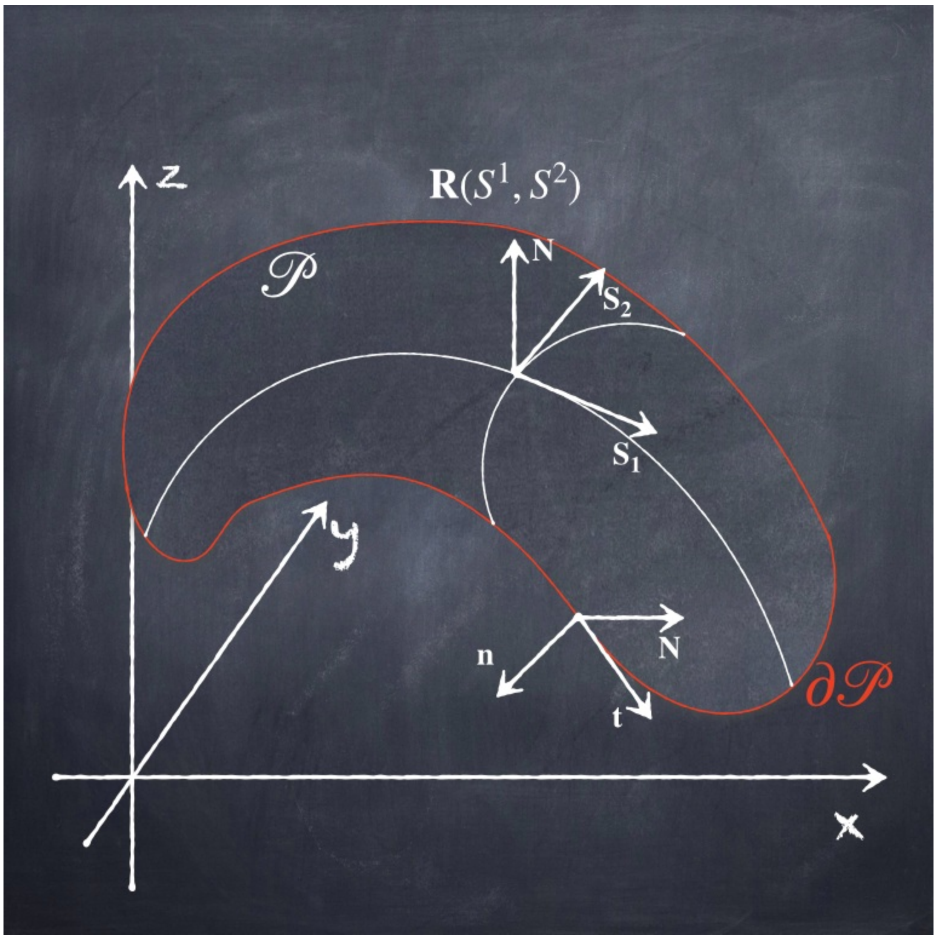

Figure 2.1. A two dimensional surface $\boldsymbol{R}(Z(S))$ embedded in a three dimensional space.

2.1 Euclidean Space

2.1.1 Tangent Space and Metrics of Euclidean Space

The physical space embedding surfaces and curves is a three dimensional Euclidean space with some arbitrary coordinates $Z^i, i \in \{1,2,3\}$, where the following convention about Latin indices are used

Convention 2.1. Latin letters i, j, k will be used to denote indices for the three dimensional Euclidean space.

The position vector $\boldsymbol{R}$ depends on coordinate Z (where contravariant indices i are ignored to simplify notation) in a continuous fashion and $\boldsymbol{R}(Z)$ are assumed to have as many orders of derivatives as we please with respect to Z. Then, at each point $\boldsymbol{R}$, **space coordinate vectors** $\boldsymbol{Z}_i \equiv \partial \boldsymbol{R} / \partial Z^i$ span a linear space. The components of the position vector generally varies from point to point $\boldsymbol{R} = R^i \boldsymbol{Z}_i$, where the following convention about repeated indices are used

Convention 2.2. Einstein summation convention is assumed in this thesis.

The **space metric tensor** is $Z_{ij} = \boldsymbol{Z}_i \cdot \boldsymbol{Z}_j$ and its inverse is defined by $Z^{ij} Z_{jk} = \delta^i_k$, where δ^i_k is the **Kronecker delta**. The determinant of Z_{ij} is denoted as Z. In the special case of a Euclidean space, the **Christoffel symbols** can be defined in an extrinsic way

$$ {}^{k}_{ij} = \boldsymbol{Z}^k \cdot \frac{\partial \boldsymbol{Z}_i}{\partial Z^j} = \quad \boldsymbol{Z}_i \cdot \frac{\partial \boldsymbol{Z}^k}{\partial Z^j}, \tag{2.1}$$

and thus derivatives of coordinate vectors can be written as

$$\frac{\partial \boldsymbol{Z}_i}{\partial Z^j} = {}^{k}_{ij} \boldsymbol{Z}_k, \tag{2.2}$$

$$\frac{\partial \boldsymbol{Z}^k}{\partial Z^j} = \quad {}^{k}_{ij} \boldsymbol{Z}_k. \tag{2.3}$$

2.1.1.1 Covariant Derivative and its Metrilinic Property

By definition, the **covariant derivatives** ∇_i of the contravariant components V^j and covariant components V_j of a vector $\boldsymbol{V} = V^j \boldsymbol{Z}_j = V_j \boldsymbol{Z}^j$ are

$$\nabla_i V_j = \frac{\partial V_j}{\partial Z^i} \quad {}^{k}_{ij} V_k \tag{2.4}$$

$$\nabla_i V^j = \frac{\partial V^j}{\partial Z^i} + {}^{j}_{ik} V^k \tag{2.5}$$

With this definition, it can be shown that the covariant derivatives of metrics are all zero. Indeed,

$$\nabla_i \boldsymbol{Z}_j = \frac{\partial \boldsymbol{Z}_j}{\partial Z^i} \quad {}^{k}_{ij} \boldsymbol{Z}_k = 0 \tag{2.6}$$

$$\nabla_i \boldsymbol{Z}^j = \frac{\partial \boldsymbol{Z}^j}{\partial Z^i} + {}^{j}_{ik} \boldsymbol{Z}^k = 0 \tag{2.7}$$

Since all covariant derivatives of coordinate vectors vanish, it is easy to conclude that any tensor formed from $\boldsymbol{Z}_i, \boldsymbol{Z}^j$ have vanishing covariant derivatives. In particular, the following metrics have vanishing covariant derivatives

$$Z_{ij} = \boldsymbol{Z}_i \cdot \boldsymbol{Z}_j \tag{2.8}$$

$$Z^{ij} = \boldsymbol{Z}^i \cdot \boldsymbol{Z}^j \tag{2.9}$$

$$\delta^i_j = \boldsymbol{Z}^i \cdot \boldsymbol{Z}_j \tag{2.10}$$

$$\delta^{ij}_{rs} = \begin{vmatrix} \delta^i_r & \delta^i_s \\ \delta^j_r & \delta^j_s \end{vmatrix} = \delta^i_r \delta^j_s \quad \delta^i_s \delta^j_r. \tag{2.11}$$

$$\delta^{ijk}_{rst} = \begin{vmatrix} \delta^i_r & \delta^i_s & \delta^i_t \\ \delta^j_r & \delta^j_s & \delta^j_t \\ \delta^k_r & \delta^k_s & \delta^k_t \end{vmatrix} = \delta^i_r \delta^{jk}_{st} \quad \delta^i_s \delta^{jk}_{rt} + \delta^i_t \delta^{jk}_{rs} \tag{2.12}$$

2.1.1.2 Determinants and Kronecker Delta

Consider a system of ${a^i}_j, i, j = 1 \cdots 3$. The **determinant** A of ${a^i}_j$ is defined as

$$|a^{\cdot}{}_{\cdot}| = e^{ijk}{a^1}_i{a^2}_j{a^3}_k = \frac{1}{3!}e^{ijk}e_{rst}{a^r}_i{a^s}_j{a^t}_k, \tag{2.13}$$

where e^{ijk} is the **permutation symbol** and we have

$$e^{ijk}{a^r}_i{a^s}_j{a^t}_k = |a^{\cdot}{}_{\cdot}|\, e^{rst} \tag{2.14}$$

$$e^{rst}e_{rst} = 3!. \tag{2.15}$$

With the relation between the permutation symbol and the Delta symbol

$$\delta^{ijk}_{rst} \equiv e^{ijk}e_{rst}, \tag{2.16}$$

we can rewrite Eqs. (2.13-2.15) as

$$|a^{\cdot}{}_{\cdot}| = \frac{1}{3!}\delta^{ijk}_{rst}{a^r}_i{a^s}_j{a^t}_k \tag{2.17}$$

$$\delta^{ijk}_{lmn}{a^r}_i{a^s}_j{a^t}_k = |a^{\cdot}{}_{\cdot}|\delta^{rst}_{lmn}, \tag{2.18}$$

$$\delta^{ijk}_{ijk} = 3!. \tag{2.19}$$

The above definition and relation can be generalized to $D \geq 3$ dimensions,

$$|a^{\cdot}{}_{\cdot}| = \frac{1}{D!}\delta^{i_1\cdots i_D}_{j_1\cdots j_D}{a^{j_1}}_{i_1}\cdots{a^{j_D}}_{i_D} \tag{2.20}$$

$$(D \quad k+1)\delta^{i_1\cdots i_k\ \ 1}_{j_1\cdots j_k\ \ 1} = \delta^{i_1\cdots i_k\ \ 1 i_k}_{j_1\cdots j_k\ \ 1 i_k}. \tag{2.21}$$

2.1.2 Relative Tensor and the Levi-Civita Tensor

Definition 2.3. *A tensor T^i_j is called a relative tensor of weight M if it changes according to the rule*

$$T^{i'}{}_{j'} = J^M T^i_j J^{i'}{}_i J^j{}_{j'}$$

where $J = |J^{\cdot}{}_{\cdot'}|$ is the determinant of the Jacobian $J^i{}_{i'} = \partial Z^i/\partial Z^{i'}$ when we change from an unprimed coordinate Z^i to a primed coordinate $Z^{i'}$.

Under a transformation from an unprimed coordinate Z^i to a primed coordinate $Z^{i'}$, the coordinate vector $\boldsymbol{Z}_i$, space metric tensor Z_{ij} and its determinant Z transforms as

$$\boldsymbol{Z}_{i'} = \frac{\partial \boldsymbol{R}}{\partial Z^i}\frac{\partial Z^i}{\partial Z^{i'}} = \boldsymbol{Z}_i J^i{}_{i'} \tag{2.22}$$

$$Z_{i'j'} = Z_{ij}J^i{}_{i'}J^j{}_{j'} \tag{2.23}$$

$$Z' = J^2 Z. \tag{2.24}$$

Thus the determinant of the metric tensor is a relative tensor of weight 2. If $J > 0$, the coordinate change preserves orientation and the volume element $\sqrt{Z}$ is then a relative tensor of weight 1:

$$\sqrt{Z'} = J\sqrt{Z}. \tag{2.25}$$

In Euclidean space, we can always take the unprimed coordinate to be the Cartesian coordinate, and $Z = 1$. Thus,

$$\sqrt{Z'} = J. \tag{2.26}$$

From Eq. (2.14), we have

$$e_{ijk}J^i{}_{i'}J^j{}_{j'}J^k{}_{k'} \;=\; Je_{i'j'k'} \tag{2.27}$$

$$e^{ijk}J^{i'}{}_iJ^{j'}{}_jJ^{k'}{}_k \;=\; J \quad ^1e^{i'j'k'}, \tag{2.28}$$

which gives

$$e_{i'j'k'} = J \ {}^1 e_{ijk} J^i_{\ i'} J^j_{\ j'} J^k \tag{2.29}$$

$$e^{i'j'k'} = J e^{ijk} J^{i'}_{\ \ i} J^{j'}_{\ \ j} J^{k'}_{\ \ k}. \tag{2.30}$$

Therefore, the permutation symbol e_{ijk} is a relative tensor of weight minus one and the permutation symbol e^{ijk} is a relative tensor of weight plus one. We can form **absolute tensors** out of relative tensors by multiplying volume element to add weight to or subtract weight from relative tensors. For example, the Levi-Civita symbols are absolute tensors defined by

$$\varepsilon^{ijk} = \frac{e^{ijk}}{\sqrt{Z}} \tag{2.31}$$

$$\varepsilon_{ijk} = \sqrt{Z} e_{ijk}. \tag{2.32}$$

In the next section, we shall see that the Levi-Civita symbols in two dimensional space are defined in a similar spirit.

2.2 Surfaces Embedded in Euclidean Space

2.2.1 Tangent Space, Metrics and Normal Space of Embedded Surfaces

Mathematically, a two dimensional surface embedded in the three dimensional Euclidean space can be represented as a mapping from $\mathbb{R}^2$ to $\mathbb{R}^3$: $Z(S)$: $(S^1, S^2) \to (Z^1, Z^2, Z^3)$ where $S^\alpha, \alpha \in \{1, 2\}$ and $Z^i, i \in \{1, 2, 3\}$ parametrize the surface and the ambient space respectively.

Convention 2.4. Greek letters α, β, γ will be used to denote indices for the embedded surfaces.

Note that Z^i can be arbitrary curvilinear coordinates. Since the position of the coordinate Z is $\boldsymbol{R}(Z)$, the surface may also be written as

$$\boldsymbol{R}(S) = \boldsymbol{R}(Z(S)). \tag{2.33}$$

All of the geometrical information for the surface is encoded in the mapping $\boldsymbol{R}(S)$. The **surface coordinate vectors** $\boldsymbol{S}_\alpha$ are defined as

$$\boldsymbol{S}_\alpha \equiv \frac{\partial \boldsymbol{R}}{\partial S^\alpha} = \frac{\partial Z^i}{\partial S^\alpha} \frac{\partial \boldsymbol{R}}{\partial Z^i} = Z^i_\alpha \boldsymbol{Z}_i \tag{2.34}$$

where $Z^i_\alpha \equiv \partial Z^i / \partial S^\alpha$ are called **shift tensors**, since it shifts coordinate vectors $\boldsymbol{Z}_i$ in ambient space to the tangent space of the surface spanned by $\{\boldsymbol{S}_\alpha\}$. Physically, the shift tensor Z^i_α represent ambient components of surface coordinate vectors. For a vector $\boldsymbol{V} = V^\alpha \boldsymbol{S}_\alpha$ living in the tangent space of the surface, its ambient components are $V^i = \boldsymbol{V} \cdot \boldsymbol{Z}^i = V^\alpha Z^i_\alpha$. Thus Z^i_α also shifts indices from tangent space of the surface to ambient Euclidean space. The **surface metric tensor** is

$$S_{\alpha\beta} \equiv \boldsymbol{S}_\alpha \cdot \boldsymbol{S}_\beta = Z_{ij} Z^i_\alpha Z^j_\beta \tag{2.35}$$

and its inverse is defined from

$$S^{\alpha\beta} S_{\beta\gamma} = \delta^\alpha_\gamma. \tag{2.36}$$

$S_{\alpha\beta}$ can lower indices of a tensor while $S^{\alpha\beta}$ can raise indices. For instance, multiplying Eq. (2.35) by $S^{\beta\gamma}$ gives

$$\boldsymbol{S}_\alpha \cdot \boldsymbol{S}^\gamma = Z^i_\alpha Z^\gamma_i = \delta^\gamma_\alpha. \tag{2.37}$$

The **determinant of the metric tensor** is

$$S = \frac{1}{2!} e^{\alpha\beta} e^{\mu\nu} S_{\alpha\mu} S_{\beta\nu} \tag{2.38}$$

where $e^{\alpha\beta}$ is the **permutation symbol**. The **surface Levi-Civita tensor** is defined as

$$\varepsilon_{\alpha\beta} = \sqrt{S} e_{\alpha\beta} \tag{2.39}$$

$$\varepsilon^{\alpha\beta} = \frac{e^{\alpha\beta}}{\sqrt{S}}. \tag{2.40}$$

The **area element** is

$$|\boldsymbol{S}_1 \times \boldsymbol{S}_2| = \sqrt{S}, \tag{2.41}$$

which can be derived from

$$\begin{aligned}\boldsymbol{S}_1 \times \boldsymbol{S}_2 &= \varepsilon_{ijk} Z_1^j Z_2^k \boldsymbol{Z}^i = \frac{1}{2!}\varepsilon_{ijk} e^{\alpha\beta} Z_\alpha^j Z_\beta^k \boldsymbol{Z}^i \\ &= \frac{\sqrt{S}}{2!}\varepsilon_{ijk}(\varepsilon^{\alpha\beta}) Z_\alpha^j Z_\beta^k \boldsymbol{Z}^i = \frac{\sqrt{S}}{2!}\varepsilon^{ijk}\varepsilon_{\alpha\beta} Z_j^\alpha Z_k^\beta \boldsymbol{Z}_i \end{aligned}\tag{2.42}$$

and

$$\begin{aligned}|\boldsymbol{S}_1 \times \boldsymbol{S}_2|^2 &= \frac{\sqrt{S}}{2!}\varepsilon_{imn}\varepsilon^{\mu\nu} Z_\mu^m Z_\nu^n \frac{\sqrt{S}}{2!}\varepsilon^{ijk}\varepsilon_{\alpha\beta} Z_j^\alpha Z_k^\beta = \frac{S}{4}\delta_{imn}^{ijk}\delta_{\alpha\beta}^{\mu\nu} Z_\mu^m Z_\nu^n Z_j^\alpha Z_k^\beta \\ &= \frac{S}{4}\delta_{mn}^{jk}\delta_{\alpha\beta}^{\mu\nu} Z_\mu^m Z_\nu^n Z_j^\alpha Z_k^\beta = \frac{S}{4}(\delta_m^j \delta_n^k \quad \delta_n^j \delta_m^k)\delta_{\alpha\beta}^{\mu\nu} Z^m{}_\mu Z_\nu^n Z_j^\alpha Z_k^\beta \\ &= \frac{S}{4}(\delta_{\alpha\beta}^{\mu\nu} Z_\mu^m Z_\nu^n Z_m^\alpha Z_n^\beta \quad \delta_{\alpha\beta}^{\mu\nu} Z_\mu^m Z_\nu^n Z_n^\alpha Z_m^\beta) = \frac{S}{4}(\delta_{\alpha\beta}^{\mu\nu}\delta_\mu^\alpha\delta_\nu^\beta \quad \delta_{\alpha\beta}^{\mu\nu}\delta_\mu^\beta\delta_\nu^\alpha) \\ &= \frac{S}{4}(\delta_{\mu\nu}^{\mu\nu} \quad \delta_{\nu\mu}^{\mu\nu}) = \frac{S}{2}\delta_{\mu\nu}^{\mu\nu} = S. \end{aligned}\tag{2.43}$$

The unit normal $\boldsymbol{N} = N_i \boldsymbol{Z}^i$ is

$$\boldsymbol{N} = \frac{\boldsymbol{S_1} \times \boldsymbol{S_2}}{|\boldsymbol{S_1} \times \boldsymbol{S_2}|}, \tag{2.44}$$

which can also be written in a component form with shift tensors

$$N_i = \frac{1}{2!}\varepsilon_{ijk}\varepsilon^{\alpha\beta} Z_\alpha^j Z_\beta^k, \tag{2.45}$$

where ε_{ijk} is the **three dimensional Levi-Civita symbol**. The orthogonal relations $\boldsymbol{N}\cdot\boldsymbol{S}_\alpha = 0$ then translates to

$$N_i Z_\alpha^i = 0. \tag{2.46}$$

In figure 2.1, $\{\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{N}\}$ forms a right handed coordinate for the surface and we have the **completeness relation**

$$\boldsymbol{N} \otimes \boldsymbol{N} + \boldsymbol{S}^\alpha \otimes \boldsymbol{S}_\alpha = \mathbb{1}^{(3)}, \tag{2.47}$$

where $\otimes$ is the tensor product operator and $\mathbb{1}^{(n)}$ represents the identity matrix in n dimensional space. The corresponding component form is

$$N_j N^i + Z_j^\alpha Z_\alpha^i = \delta_j^i \tag{2.48}$$

$$N^i N^j + Z^{\alpha j} Z_\alpha^i = Z^{ij}. \tag{2.49}$$

2.2.2 Covariant Derivative and its Metrilinic Property

Due to the embedding in a Euclidean space, the **surface Christoffel symbol** $^{\alpha}_{\beta\gamma}$ can be defined in an extrinsic manner

$$^{\alpha}_{\beta\gamma} = \boldsymbol{S}^\alpha \cdot \frac{\partial \boldsymbol{S}_\beta}{\partial S^\gamma} = \quad \boldsymbol{S}_\beta \cdot \frac{\partial \boldsymbol{S}^\alpha}{\partial S^\gamma}, \tag{2.50}$$

which can be related to **ambient Christoffel symbols** $^{i}_{jk}$

$$^{\alpha}_{\beta\gamma} = Z_j^\alpha \frac{\partial Z_\beta^j}{\partial S^\gamma} + {}^{i}_{jk} Z_i^\alpha Z_\beta^j Z_\gamma^k. \tag{2.51}$$

The surface Christoffel symbols can also be defined in an intrinsic way

$$^{\mu}_{\beta\gamma} = \frac{1}{2} S^{\mu\alpha}\left(\frac{\partial S_{\alpha\beta}}{\partial S^\gamma} + \frac{\partial S_{\alpha\gamma}}{\partial S^\beta} \quad \frac{\partial S_{\beta\gamma}}{\partial S^\alpha}\right). \tag{2.52}$$

The **surface covariant derivatives** ∇_α of the surface components of a vector $\boldsymbol{V} = V^\alpha \boldsymbol{S}_\alpha = V_\alpha \boldsymbol{S}^\alpha$ living in the tangential space are thus

$$\nabla_\alpha V^\beta = \frac{\partial V^\beta}{\partial S^\alpha} + {}^{\beta}_{\alpha\gamma} V^\gamma \tag{2.53}$$

$$\nabla_\alpha V_\beta = \frac{\partial V_\beta}{\partial S^\alpha} \quad {}^{\gamma}_{\alpha\beta} V_\gamma. \tag{2.54}$$

The covariant derivative ∇_α is identity to the partial derivative ∂_α when applied to a scalar. Indeed,

$$\begin{aligned}\nabla_\gamma \boldsymbol{T} &= \frac{\partial(T^i \boldsymbol{Z}_i)}{\partial S^\gamma} = \frac{\partial T^i}{\partial S^\gamma}\boldsymbol{Z}_i + T^i \frac{\partial \boldsymbol{Z}_i}{\partial S^\gamma} = \frac{\partial T^i}{\partial S^\gamma}\boldsymbol{Z}_i + T^i \frac{\partial \boldsymbol{Z}_i}{\partial Z^j}\frac{\partial Z^j}{\partial S^\gamma} \\ &= \frac{\partial T^i}{\partial S^\gamma}\boldsymbol{Z}_i + T^i \;{}^{k}_{ij}\boldsymbol{Z}_k Z^j_\gamma = \left(\frac{\partial T^i}{\partial S^\gamma} + \;{}^{i}_{kj} Z^j_\gamma T^k\right)\boldsymbol{Z}_i \end{aligned} \tag{2.55}$$

$$\begin{aligned}\nabla_\gamma \boldsymbol{T} &= \frac{\partial(T_i \boldsymbol{Z}^i)}{\partial S^\gamma} = \frac{\partial T_i}{\partial S^\gamma}\boldsymbol{Z}^i + T_i \frac{\partial \boldsymbol{Z}^i}{\partial S^\gamma} = \frac{\partial T_i}{\partial S^\gamma}\boldsymbol{Z}^i + T_i \frac{\partial \boldsymbol{Z}^i}{\partial Z^j}\frac{\partial Z^j}{\partial S^\gamma} \\ &= \frac{\partial T_i}{\partial S^\gamma}\boldsymbol{Z}^i \quad T_i \;{}^{i}_{jk}\boldsymbol{Z}^k Z^j_\gamma = \left(\frac{\partial T_i}{\partial S^\gamma} \quad {}^{k}_{ji} Z^j_\gamma T_k\right)\boldsymbol{Z}^i. \end{aligned} \tag{2.56}$$

Thus, the surfaces covariant derivatives of the spatial components of a vector $\boldsymbol{T} = T^i \boldsymbol{Z}_i = T_i \boldsymbol{Z}^i$ are

$$\nabla_\gamma T^i = \frac{\partial T^i}{\partial S^\gamma} + \;{}^{i}_{kj} Z^j_\gamma T^k \tag{2.57}$$

$$\nabla_\gamma T_i = \frac{\partial T_i}{\partial S^\gamma} \quad {}^{k}_{ij} Z^j_\gamma T_k. \tag{2.58}$$

On the other hand,

$$\nabla_\gamma \boldsymbol{T} = \frac{\partial \boldsymbol{T}}{\partial Z^i}\frac{\partial Z^i}{\partial S^\gamma} = Z^i_\gamma \nabla_i \boldsymbol{T} = Z^i_\gamma \nabla_i T^j \boldsymbol{Z}_j. \tag{2.59}$$

Thus, we have the chain rule

$$\nabla_\gamma T^i = Z^j_\gamma \nabla_j T^i \tag{2.60}$$

$$\nabla_\gamma T_i = Z^j_\gamma \nabla_j T_i. \tag{2.61}$$

With those definitions, it can be shown that surface covariant derivatives of both surface metrics and space metrics vanishes

$$\nabla_\gamma S_{\alpha\beta}, \nabla_\gamma S^{\alpha\beta}, \nabla_\gamma S = 0 \tag{2.62}$$

$$\nabla_\gamma \varepsilon_{\alpha\beta}, \nabla_\gamma \varepsilon^{a\beta} = 0 \tag{2.63}$$

and with the chain rule

$$\nabla_\gamma \boldsymbol{Z}_i, \nabla_\gamma \boldsymbol{Z}^i = 0 \tag{2.64}$$

$$\nabla_\gamma Z_{ij}, \nabla_\gamma Z^{ij} = 0 \tag{2.65}$$

$$\nabla_\gamma \varepsilon_{ijk}, \nabla_\gamma \varepsilon^{ijk} = 0 \tag{2.66}$$

$$\nabla_\gamma \delta^i_j, \nabla_\gamma \delta^{ij}_{rs}, \nabla_\gamma \delta^{ijk}_{rst} = 0. \tag{2.67}$$

Remark 2.5. With the metrilinic property, we can adjust freely position of repeated indices within the covariant derivative operators.

2.2.3 Curvature Tensor

The information about curvature of a surface is encoded in the manner how the tangent space and normal space varies as we move from one point to another on the surface. Mathematically, it is encoded in $\nabla_\alpha \boldsymbol{S}_\beta$ and $\nabla_\alpha \boldsymbol{N}$. Indeed, the curvature tensor can be defined as

$$\begin{aligned} B_{\alpha\beta} &\equiv \boldsymbol{N}\cdot\nabla_\alpha \boldsymbol{S}_\beta = \boldsymbol{N}\cdot\nabla_\alpha(Z^i_\beta \boldsymbol{Z}_i) = \boldsymbol{N}\cdot(\nabla_\alpha Z^i_\beta \boldsymbol{Z}_i) = N_i \nabla_\alpha Z^i_\beta \\ &= \boldsymbol{S}_\beta \cdot \nabla_\alpha \boldsymbol{N} = Z^i_\beta \boldsymbol{Z}_i \cdot \nabla_\alpha (N_j \boldsymbol{Z}^j) = Z^i_\beta \nabla_\alpha N_i. \end{aligned} \tag{2.68}$$

Note that since $\nabla_\alpha \boldsymbol{N} = \partial_\alpha \boldsymbol{N}$, we have

$$B_{\alpha\beta} = \boldsymbol{S}_\beta \cdot \partial_\alpha \boldsymbol{N} = \boldsymbol{N}\cdot\partial_\alpha \boldsymbol{S}_\beta = \boldsymbol{N}\cdot\partial_\alpha\partial_\beta \boldsymbol{R} \tag{2.69}$$

and therefore $B_{\alpha\beta} = B_{\beta\alpha}$. Since $\boldsymbol{S}_\alpha$ and $\boldsymbol{N}$ spans a complete basis, the following decomposition follows

$$\frac{\partial \boldsymbol{S}_\alpha}{\partial S^\beta} = \frac{\partial \boldsymbol{S}_\alpha}{\partial S^\beta}\cdot(\boldsymbol{S}^\gamma \boldsymbol{S}_\gamma + \boldsymbol{N}\boldsymbol{N}) = \;{}^{\gamma}_{\alpha\beta}\boldsymbol{S}_\gamma + B_{\alpha\beta}\boldsymbol{N} \tag{2.70}$$

$$\frac{\partial \boldsymbol{S}^\alpha}{\partial S^\beta} = \frac{\partial \boldsymbol{S}^\alpha}{\partial S^\beta}\cdot(\boldsymbol{S}^\gamma \boldsymbol{S}_\gamma + \boldsymbol{N}\boldsymbol{N}) = \quad {}^{\alpha}_{\beta\gamma}\boldsymbol{S}^\gamma + B^\alpha_\beta \boldsymbol{N}. \tag{2.71}$$

We therefore have the **Weingarten's Formula** in vector notation

$$\begin{aligned}\nabla_\alpha \boldsymbol{N} &= \nabla_\alpha \boldsymbol{N}\cdot(\boldsymbol{S}^\beta \boldsymbol{S}_\beta + \boldsymbol{N}\boldsymbol{N}) = \quad B^\beta_\alpha \boldsymbol{S}_\beta && (2.72)\\ \nabla_\beta \boldsymbol{S}_\alpha &= \frac{\partial \boldsymbol{S}_\alpha}{\partial S^\beta} \quad {}^{\gamma}_{\alpha\beta}\boldsymbol{S}_\gamma = B_{\alpha\beta}\boldsymbol{N} && (2.73)\\ \nabla_\beta \boldsymbol{S}^\alpha &= \frac{\partial \boldsymbol{S}^\alpha}{\partial S^\beta} + {}^{\alpha}_{\beta\gamma}\boldsymbol{S}^\gamma = B^\alpha_\beta \boldsymbol{N}, && (2.74)\end{aligned}$$

whose component form gives

$$\begin{aligned}\nabla_\alpha N^i &= \quad Z^i_\beta B^\beta_\alpha && (2.75)\\ \nabla_\alpha Z^i_\beta &= N^i B_{\alpha\beta}. && (2.76)\end{aligned}$$

The only two scalar invariants of the curvature tensor is its trace $\mathrm{tr}(B)$ and determinant $\det(B)$, which are also known as the **mean curvature** K_M and **Gaussian curvature** K for the surface. Indeed,

$$\begin{aligned}K_M &\equiv \mathrm{tr}(B) = B^\alpha_\alpha = S^{\alpha\beta}B_{\alpha\beta} && (2.77)\\ K &\equiv \det(B) = \frac{1}{2!}\delta^{\alpha\beta}_{\mu\nu}B^\mu_\alpha B^\nu_\beta = \frac{1}{2}(B^\alpha_\alpha B^\beta_\beta \quad B^\beta_\alpha B^\alpha_\beta)\\ &= \frac{1}{2}(S^{\alpha\mu}S^{\beta\nu} \quad S^{\alpha\nu}S^{\beta\mu})B_{\alpha\mu}B_{\beta\nu} = \frac{1}{2}\varepsilon^{\alpha\beta}\varepsilon^{\mu\nu}B_{\alpha\mu}B_{\beta\nu}, && (2.78)\end{aligned}$$

where we used a very useful relation between the Levi-Civita tensor and metric tensor in two dimensional space

$$S^{\alpha\mu}S^{\beta\nu} \quad S^{\alpha\nu}S^{\beta\mu} = \varepsilon^{\alpha\beta}\varepsilon^{\mu\nu}. \qquad (2.79)$$

Convention 2.6. The mean curvature is defined such that the mean curvature of the unit sphere is 2.

2.2.4 The Gaussian Curvature and Riemann Curvature Tensor in 2D

The Gaussian curvature K is completely intrinsic and can be defined independent of how the surface is embedded. Actually, the Gaussian curvature is the only invariant of the **Riemann curvature tensor** in two dimensional space. The Riemann curvature tensor $R_{\gamma\delta\alpha\beta}$ is defined by

$$[\nabla_\alpha, \nabla_\beta]T_\gamma = R_{\gamma\delta\alpha\beta}T^\delta, \qquad (2.80)$$

which gives the explicit form

$$R_{\gamma\delta\beta\alpha} \equiv \frac{\partial \quad {}_{\gamma,\alpha\delta}}{\partial S^\beta} \quad \frac{\partial \quad {}_{\gamma,\beta\delta}}{\partial S^\alpha} + {}^{\omega}_{\beta\delta} \quad {}_{\omega,\gamma\alpha} \quad {}^{\omega}_{\alpha\delta} \quad {}_{\omega,\gamma\beta}. \qquad (2.81)$$

The following symmetric properties hold for the Riemann curvature tensor $R_{\alpha\beta\mu\nu}$.

i. Anti-symmetry of (α,β) and (μ,ν)

$$R_{\alpha\beta\mu\nu} = \quad R_{\beta\alpha\mu\nu}, R_{\alpha\beta\mu\nu} = \quad R_{\alpha\beta\nu\mu} \qquad (2.82)$$

ii. **The first Bianchi identity**

$$R_{\alpha\beta\mu\nu} + R_{\alpha\nu\beta\mu} + R_{\alpha\mu\nu\beta} = 0 \qquad (2.83)$$

iii. Symmetry of (α,β) and (μ,ν)

$$R_{\alpha\beta\mu\nu} = R_{\mu\nu\alpha\beta}. \qquad (2.84)$$

Let us count the number of degree of freedom of the Riemann curvature tensor $R_{\alpha\beta\mu\nu}$ in D dimensional space. Due to Eq. (2.82), there are $M = D(D \quad 1)/2$ ways of choosing nontrivial pairs of (α,β) and (μ,ν). Due to Eq. (2.84), there are $M + M(M \quad 1)/2 = M(M+1)/2$ ways of choosing pairs of (a,b) and (c,d). In addition, Eq. (2.83) imposes $D(D \quad 1)(D \quad 2)(D \quad 3)/4!$ constraints. Thus, the number of independent degree of freedom is

$$\frac{1}{2}M(M+1) \quad \frac{D(D \quad 1)(D \quad 2)(D \quad 3)}{4!} = \frac{D^2(D^2 \quad 1)}{12}.$$

If $D=2$, there is only one degree of freedom. Thus, in 2D, tensors satisfying symmetric properties of the Riemann curvature tensor must differ only by the multiplication of a scalar invariant. One can verify that Eqs. (2.82-2.84) hold for both $\varepsilon^{\alpha\beta}\varepsilon^{\mu\nu}$ and $S^{\alpha\mu}S^{\beta\nu} \quad S^{\alpha\nu}S^{\beta\mu}$ and the proportional constant between them is 1 by taking $\alpha=\mu=1, \beta=\nu=2$, which gives Eq. (2.79). The Gaussian curvature K can then be defined as the multiplication constant between the Riemann curvature tensor and $\varepsilon_{\alpha\beta}\varepsilon_{\mu\nu}$ (or $S_{\alpha\mu}S_{\beta\nu} \quad S_{\alpha\nu}S_{\beta\mu}$):

$$R_{\alpha\beta\mu\nu}=K\varepsilon_{\alpha\beta}\varepsilon_{\mu\nu}=K(S_{\alpha\mu}S_{\beta\nu} \quad S_{\alpha\nu}S_{\beta\mu}), \tag{2.85}$$

and

$$K=\frac{R_{1212}}{S}=\frac{1}{4}\varepsilon^{\alpha\beta}\varepsilon^{\gamma\delta}R_{\gamma\delta\alpha\beta}=\frac{1}{2}S^{\gamma\alpha}S^{\delta\beta}R_{\gamma\delta\alpha\beta}. \tag{2.86}$$

To show the consistence between the extrinsic determinant definition Eq. (2.78) and the intrinsic definition Eq. (2.86), we need a *Remarkable theorem*, Gauss's Theorema Egregium.

2.2.5 Gauss-Codazzi equation and Gauss's Theorema Egregium

Applying the definition Eq. (2.80) for the Riemann curvature tensor to the tangent vector $\boldsymbol{S}_\gamma$ gives the **Gauss-Codazzi equation**

$$[\nabla_\alpha, \nabla_\beta]\boldsymbol{S}_\gamma=R_{\gamma\delta\alpha\beta}\boldsymbol{S}^\delta, \tag{2.87}$$

whose explicit form is

$$(\nabla_\alpha B_{\beta\gamma} \quad \nabla_\beta B_{\alpha\gamma})\boldsymbol{N}+(B_{\alpha\gamma}B_{\beta\delta} \quad B_{\beta\gamma}B_{\alpha\delta})\boldsymbol{S}^\delta=R_{\gamma\delta\alpha\beta}\boldsymbol{S}^\delta. \tag{2.88}$$

The normal component of Eq. (2.88) leads to the **Codazzi equation**

$$\nabla_\alpha B_{\beta\gamma}=\nabla_\beta B_{\alpha\gamma}, \tag{2.89}$$

which shows that $\nabla_\alpha B_{\beta\gamma}$ is fully symmetric with respect to its indices. The tangential component of Eq. (2.88) gives us the **Gauss's Theorema Egregium** (translated from Latin as the Remarkable theorem)

$$B_{\alpha\gamma}B_{\beta\delta} \quad B_{\beta\gamma}B_{\alpha\delta}=R_{\gamma\delta\alpha\beta}. \tag{2.90}$$

With Eq. (2.90), it is obvious that Eq. (2.78) and Eq. (2.86) gives the same definition for the Gaussian curvature. Equivalent forms of Eq. (2.90) are also used:

$$B_{\alpha\gamma}B_{\beta\delta} \quad B_{\beta\gamma}B_{\alpha\delta}=K\varepsilon_{\gamma\delta}\varepsilon_{\alpha\beta} \tag{2.91}$$

$$B^\gamma_\alpha B^\delta_\beta \quad B^\gamma_\beta B^\delta_\alpha=K\delta^{\gamma\delta}_{\alpha\beta}, \tag{2.92}$$

whose contraction of γ, α and δ, β gives

$$K_M B^\delta_\beta \quad B^\alpha_\beta B^\delta_\alpha=K\delta^\delta_\beta \tag{2.93}$$

$$(K_M)^2 \quad B^\alpha_\beta B^\beta_\alpha=2K \tag{2.94}$$

2.2.6 Differential Operators in Space and Surface

The full gradient operator ∇ is

$$\nabla=\boldsymbol{Z}^i\nabla_i=\boldsymbol{Z}_i\nabla^i, \tag{2.95}$$

which can be decomposed into a normal component and a tangential component:

$$\nabla=(\boldsymbol{N}\boldsymbol{N}+\boldsymbol{S}^\alpha\boldsymbol{S}_\alpha)\cdot\nabla=\boldsymbol{N}\frac{\partial}{\partial N}+\boldsymbol{S}^\alpha\nabla_\alpha=\boldsymbol{N}\frac{\partial}{\partial N}+\nabla_\parallel, \tag{2.96}$$

where $\partial/\partial N$ is the directional derivative in $\boldsymbol{N}$ direction

$$\frac{\partial}{\partial N}\equiv\boldsymbol{N}\cdot\nabla=N^i\nabla_i, \tag{2.97}$$

and we used the chain rule

$$\boldsymbol{S}_\alpha \cdot \nabla = (\boldsymbol{S}_\alpha \cdot \boldsymbol{Z}^i)\nabla_i = Z^i_\alpha \nabla_i = \nabla_\alpha. \tag{2.98}$$

The surface gradient operator $\nabla_{\|}$ is defined as

$$\nabla_{\|} \equiv \boldsymbol{S}^\alpha \nabla_\alpha = \nabla \quad \boldsymbol{N}\frac{\partial}{\partial N}. \tag{2.99}$$

The component form of Eq. (2.96) is

$$\nabla_j = (N_j N^i + Z^\alpha_j Z^i_\alpha)\nabla_i = N_j N^i \nabla_i + Z^\alpha_j \nabla_\alpha. \tag{2.100}$$

The surface gradient of a scalar field F restricted to the surface is

$$\nabla_{\|} F = \boldsymbol{S}^\alpha \nabla_\alpha F = \nabla F \quad \boldsymbol{N}\frac{\partial F}{\partial N} = \boldsymbol{Z}^i(\nabla_i F \quad N_i N_j \nabla^j F) \tag{2.101}$$

The surface divergence of a vector field $\boldsymbol{F} = F^\alpha \boldsymbol{S}_\alpha + F\boldsymbol{N} = F^i \boldsymbol{Z}_i$ is

$$\begin{aligned} \nabla_{\|} \cdot \boldsymbol{F} &= \boldsymbol{S}^\alpha \cdot \nabla_\alpha \boldsymbol{F} = \boldsymbol{S}^\alpha \cdot \nabla_\alpha (F^\beta \boldsymbol{S}_\beta + F\boldsymbol{N}) = \nabla_\alpha F^\alpha \quad K_M F \\ &= \nabla \cdot \boldsymbol{F} \quad \boldsymbol{N} \cdot \frac{\partial \boldsymbol{F}}{\partial N} = \nabla_i F^i \quad N_i N^j \nabla_j F^i. \end{aligned} \tag{2.102}$$

Applying Eq. (2.102) to $\boldsymbol{F} = \boldsymbol{N}$, we have

$$\nabla \cdot \boldsymbol{N} = \nabla_{\|} \cdot \boldsymbol{N} = \quad K_M, \tag{2.103}$$

where we used $\boldsymbol{N} \cdot \partial \boldsymbol{N} / \partial N = 0$. Eq. (2.103) is also a frequently used definition for mean curvature. The **Laplace-Beltrami** operator (Laplacian on surface) of a scalar field F is

$$\begin{aligned} \Delta_{\|} F &= \nabla_{\|} \cdot \nabla_{\|} F = \boldsymbol{S}^\alpha \cdot \nabla_\alpha (\boldsymbol{S}_\beta \nabla^\beta F) = \boldsymbol{S}^\alpha \cdot \boldsymbol{S}_\beta (\nabla_\alpha \nabla^\beta F) + (\boldsymbol{S}^\alpha \cdot \nabla_\alpha \boldsymbol{S}_\beta)\nabla^\beta F = \nabla_\alpha \nabla^\alpha F \\ &= \left(\nabla \quad \boldsymbol{N}\frac{\partial}{\partial N}\right) \cdot \left(\nabla F \quad \boldsymbol{N}\frac{\partial F}{\partial N}\right) \\ &= \nabla \cdot \left(\nabla F \quad \boldsymbol{N}\frac{\partial F}{\partial N}\right) \quad \left(\boldsymbol{N}\frac{\partial}{\partial N}\right) \cdot \left(\nabla F \quad \boldsymbol{N}\frac{\partial F}{\partial N}\right) \\ &= \Delta F \quad \nabla \cdot \boldsymbol{N}\frac{\partial F}{\partial N} \quad \boldsymbol{N} \cdot \nabla \frac{\partial F}{\partial N} \quad \boldsymbol{N} \cdot \frac{\partial (\nabla F)}{\partial N} + \boldsymbol{N} \cdot \frac{\partial}{\partial N}\left(\boldsymbol{N}\frac{\partial F}{\partial N}\right) \\ &= \Delta F + K_M \frac{\partial F}{\partial N} \quad \frac{\partial^2 F}{\partial N^2} \quad \boldsymbol{N} \cdot \frac{\partial (\nabla F)}{\partial N} + \boldsymbol{N} \cdot \left(\boldsymbol{N}\frac{\partial^2 F}{\partial N^2} + \frac{\partial \boldsymbol{N}}{\partial N}\frac{\partial F}{\partial N}\right) \\ &= \Delta F + K_M \frac{\partial F}{\partial N} \quad \frac{\partial^2 F}{\partial N^2} \quad \boldsymbol{N} \cdot \frac{\partial (\nabla F)}{\partial N} + \frac{\partial^2 F}{\partial N^2} \\ &= \Delta F + K_M \frac{\partial F}{\partial N} \quad \boldsymbol{N} \cdot \frac{\partial (\nabla F)}{\partial N} \\ &= \Delta F + K_M \frac{\partial F}{\partial N} \quad N^i N^j \nabla_j \nabla_i F \\ &= \nabla^i \nabla_i F + K_M N^j \nabla_j F \quad N^i N^j \nabla_j \nabla_i F. \end{aligned} \tag{2.104}$$

2.3 Curves Embedded in Surface

2.3.1 Metrics and Covariant Derivatives

In Figure 2.1, the boundary of an open patch of surface $\mathcal{P}$ is denoted by $\partial\mathcal{P}$, which is an embedded curve on the surface. Mathematically, this curve can be represented as a mapping from $\mathbb{R}$ to $\mathbb{R}^3$ through composition of $S(U)$: $U^\Phi \to S^\alpha$ and $Z(S)$: $S^\alpha \to Z^i$, where U^Φ, $\Phi = 1$ is the contravariant curve coordinate.

Convention 2.7. Capital Greek letters Φ, Ψ will be used to denote curve indices.

We may also write the curve as $\boldsymbol{R}(U)=\boldsymbol{R}(Z(S(U)))$. Similar to embedded surfaces, the **curve coordinate vector** is

$$\boldsymbol{U}_\Phi = \frac{\partial \boldsymbol{R}}{\partial U^\Phi} = \frac{\partial \boldsymbol{R}}{\partial Z^i}\frac{\partial Z^i}{\partial U^\Phi} = \boldsymbol{Z}_i Z^i_\Phi \tag{2.105}$$

$$= \frac{\partial \boldsymbol{R}}{\partial S^\alpha}\frac{\partial S^\alpha}{\partial U^\Phi} = \boldsymbol{S}_\alpha S^\alpha_\Phi = \boldsymbol{Z}_i Z^i_\alpha S^\alpha_\Phi, \tag{2.106}$$

where the **shift tensor** $S^\alpha_\Phi \equiv \partial S^\alpha / \partial U^\Phi$ and $Z^i_\Phi \equiv \partial Z^i / \partial U^\Phi$. The covariant **metric tensor** is

$$U_{\Phi\Psi} = \boldsymbol{U}_\Phi \cdot \boldsymbol{U}_\Psi = S_{\alpha\beta} S^\alpha_\Phi S^\beta_\Psi, \tag{2.107}$$

and it inverse is defined by

$$U^{\Phi\Psi} U_{\Psi\Omega} = \delta^\Phi_\Omega, \tag{2.108}$$

which is

$$S^\Phi_\alpha S^\alpha_\Psi = \delta^\Phi_\Psi. \tag{2.109}$$

The determinant of the metric tensor is

$$U = e^\Phi e^\Psi U_{\Phi\Psi}, \tag{2.110}$$

where e^Φ is the one dimensional permutation symbol and is therefore always equal to one. The **curve Christoffel symbol** is defined intrinsically as

$${}^{\Theta}_{\Phi\Psi} = \frac{1}{2} U^{\Theta\Omega}\left(\frac{\partial U_{\Omega\Phi}}{\partial U^\Psi} + \frac{\partial U_{\Omega\Psi}}{\partial U^\Phi} \quad \frac{\partial U_{\Phi\Psi}}{\partial U^\Omega}\right), \tag{2.111}$$

which is equivalent to an extrinsic definition

$${}^{\Theta}_{\Phi\Psi} = \frac{\partial \boldsymbol{U}_\Phi}{\partial U^\Psi} \cdot \boldsymbol{U}^\Theta = \quad \frac{\partial \boldsymbol{U}^\Theta}{\partial U^\Psi} \cdot \boldsymbol{U}^\Phi. \tag{2.112}$$

With this definition, we find

$$\nabla_\Omega U_{\Phi\Psi}, \nabla_\Omega U^{\Phi\Psi}, \nabla_\Omega U = 0. \tag{2.113}$$

The curve Christoffel symbol is related to the surface Christoffel symbol by the equation

$${}^{\Omega}_{\Phi\Psi} = \quad {}^{\alpha}_{\beta\gamma} S^\Omega_\alpha S^\beta_\Phi S^\gamma_\Psi + \frac{\partial S^\alpha_\Phi}{\partial U^\Psi} S^\Omega_\alpha. \tag{2.114}$$

The **curve covariant derivatives** of the components of curve vector $\boldsymbol{T} = T^\Phi \boldsymbol{U}_\Phi = T_\Phi \boldsymbol{U}^\Phi$ are

$$\nabla_\Theta T^\Phi = \frac{\partial T^\Phi}{\partial U^\Phi} + \quad {}^{\Phi}_{\Theta\Omega} T^\Omega \tag{2.115}$$

$$\nabla_\Theta T_\Phi = \frac{\partial T_\Phi}{\partial U^\Theta} \quad {}^{\Omega}_{\Theta\Phi} T_\Omega. \tag{2.116}$$

Note that since

$$\nabla_\Theta \boldsymbol{T} = \frac{\partial \boldsymbol{T}}{\partial U^\Theta} = \frac{\partial S^\alpha}{\partial U^\Theta}\frac{\partial \boldsymbol{T}}{\partial S^\alpha} = S^\alpha_\Theta \nabla_\alpha \boldsymbol{T}, \tag{2.117}$$

chain rule still holds:

$$\nabla_\Theta = S^\alpha_\Theta \nabla_\alpha. \tag{2.118}$$

To compute the curve covariant derivatives of the components of a surface vector $\boldsymbol{T} = T^\alpha \boldsymbol{S}_\alpha = T_\alpha \boldsymbol{S}^\alpha$, we first compute

$$\begin{aligned}\nabla_\Theta \boldsymbol{T} &= \frac{\partial \boldsymbol{T}}{\partial U^\Theta} = \frac{\partial (T^\alpha \boldsymbol{S}_\alpha)}{\partial U^\Theta} = \frac{\partial T^\alpha}{\partial U^\Theta}\boldsymbol{S}_\alpha + T^\alpha \frac{\partial \boldsymbol{S}_\alpha}{\partial U^\Theta} = \frac{\partial T^\alpha}{\partial U^\Theta}\boldsymbol{S}_\alpha + T^\alpha S^\beta_\Theta \frac{\partial \boldsymbol{S}_\alpha}{\partial S^\beta} \\ &= \frac{\partial T^\alpha}{\partial U^\Theta}\boldsymbol{S}_\alpha + T^\alpha S^\beta_\Theta (\ {}^{\gamma}_{\alpha\beta}\boldsymbol{S}_\gamma + B_{\alpha\beta}\boldsymbol{N}) = \left(\frac{\partial T^\gamma}{\partial U^\Theta} + T^\alpha S^\beta_\Theta \ {}^{\gamma}_{\alpha\beta}\right)\boldsymbol{S}_\gamma + T^\alpha S^\beta_\Theta B_{\alpha\beta}\boldsymbol{N}\end{aligned} \tag{2.119}$$

$$\begin{aligned}\nabla_\Theta \boldsymbol{T} &= \frac{\partial \boldsymbol{T}}{\partial U^\Theta} = \frac{\partial (T_\alpha \boldsymbol{S}^\alpha)}{\partial U^\Theta} = \frac{\partial T_\alpha}{\partial U^\Theta}\boldsymbol{S}^\alpha + T_\alpha \frac{\partial \boldsymbol{S}^\alpha}{\partial U^\Theta} = \frac{\partial T_\alpha}{\partial U^\Theta}\boldsymbol{S}^\alpha + T_\alpha S^\beta_\Theta \frac{\partial \boldsymbol{S}^\alpha}{\partial S^\beta}\boldsymbol{N} \\ &= \frac{\partial T_\alpha}{\partial U^\Theta}\boldsymbol{S}^\alpha + T_\alpha S^\beta_\Theta (B^\alpha_\beta \boldsymbol{N} \quad {}^{\alpha}_{\beta\gamma}\boldsymbol{S}^\gamma) = \left(\frac{\partial T_\gamma}{\partial U^\Theta} \quad T_\alpha S^\beta_\Theta \ {}^{\alpha}_{\beta\gamma}\right)\boldsymbol{S}^\gamma + T_\alpha S^\beta_\Theta B^\alpha_\beta \boldsymbol{N}.\end{aligned} \tag{2.120}$$

On the other hand, we should be able to write

$$\begin{aligned}\nabla_\Theta(T^\alpha \boldsymbol{S}_\alpha) &= \nabla_\Theta T^\alpha \boldsymbol{S}_\alpha + T^\alpha S^\beta_\Theta \nabla_\beta \boldsymbol{S}_\alpha = \nabla_\Theta T^\alpha \boldsymbol{S}_\alpha + T^\alpha S^\beta_\Theta B_{\alpha\beta}\boldsymbol{N} && (2.121)\\ \nabla_\Theta(T_\alpha \boldsymbol{S}^\alpha) &= \nabla_\Theta T_\alpha \boldsymbol{S}^\alpha + T_\alpha S^\beta_\Theta \nabla_\beta \boldsymbol{S}^\alpha = \nabla_\Theta T_\alpha \boldsymbol{S}^\alpha + T_\alpha S^\beta_\Theta B^\alpha_\beta \boldsymbol{N}. && (2.122)\end{aligned}$$

Thus by comparison,

$$\begin{aligned}\nabla_\Theta T^\gamma &\equiv \frac{\partial T^\gamma}{\partial U^\Theta} + T^\alpha S^\beta_\Theta \ {}^{\gamma}_{\alpha\beta} && (2.123)\\ \nabla_\Theta T_\gamma &\equiv \frac{\partial T_\gamma}{\partial U^\Theta} \quad T_\alpha S^\beta_\Theta \ {}^{\alpha}_{\beta\gamma}. && (2.124)\end{aligned}$$

Similarly, to compute the curve covariant derivatives of the components of a space vector $\boldsymbol{T} = T^i \boldsymbol{Z}_i = T_i \boldsymbol{Z}^i$, we first compute

$$\begin{aligned}\nabla_\Theta \boldsymbol{T} &= \frac{\partial \boldsymbol{T}}{\partial U^\Theta} = \frac{\partial(T^i\boldsymbol{Z}_i)}{\partial U^\Theta} = \frac{\partial T^i}{\partial U^\Theta}\boldsymbol{Z}_i + T^i \frac{\partial \boldsymbol{Z}_i}{\partial U^\Theta} = \frac{\partial T^i}{\partial U^\Theta}\boldsymbol{Z}_i + T^i \frac{\partial Z^j}{\partial S^\alpha}\frac{\partial S^\alpha}{\partial U^\Theta}\frac{\partial \boldsymbol{Z}_i}{\partial Z^j}\\ &= \frac{\partial T^i}{\partial U^\Theta}\boldsymbol{Z}_i + T^i Z^j_\alpha S^\alpha_\Theta \ {}^{k}_{ij}\boldsymbol{Z}_k = \left(\frac{\partial T^k}{\partial U^\Theta} + T^i Z^j_\alpha S^\alpha_\Theta \ {}^{k}_{ij}\right)\boldsymbol{Z}_k && (2.125)\\ \nabla_\Theta \boldsymbol{T} &= \frac{\partial \boldsymbol{T}}{\partial U^\Theta} = \frac{\partial(T_i\boldsymbol{Z}^i)}{\partial U^\Theta} = \frac{\partial T_k}{\partial U^\Theta}\boldsymbol{Z}^k + T_i \frac{\partial \boldsymbol{Z}^i}{\partial U^\Theta} = \frac{\partial T_k}{\partial U^\Theta}\boldsymbol{Z}^k + T_i \frac{\partial Z^j}{\partial S^\alpha}\frac{\partial S^\alpha}{\partial U^\Theta}\frac{\partial \boldsymbol{Z}^i}{\partial Z^j}\\ &= \frac{\partial T_k}{\partial U^\Theta}\boldsymbol{Z}^k \quad T_i Z^j_\alpha S^\alpha_\Theta \ {}^{i}_{jk}\boldsymbol{Z}^k = \left(\frac{\partial T_k}{\partial U^\Theta} \quad T_i Z^j_\alpha S^\alpha_\Theta \ {}^{i}_{jk}\right)\boldsymbol{Z}^k. && (2.126)\end{aligned}$$

Thus,

$$\begin{aligned}\nabla_\Theta T^k &= \frac{\partial T^k}{\partial U^\Theta} + T^i Z^j_\alpha S^\alpha_\Theta \ {}^{k}_{ij} && (2.127)\\ \nabla_\Theta T_k &= \frac{\partial T_k}{\partial U^\Theta} \quad T_i Z^j_\alpha S^\alpha_\Theta \ {}^{i}_{jk}. && (2.128)\end{aligned}$$

The metrilinic properties follows easily from chain rule.

$$\begin{aligned}\nabla_\Theta \boldsymbol{Z}_i, \nabla_\Theta \boldsymbol{Z}^i &= 0 && (2.129)\\ \nabla_\Theta Z_{ij}, \nabla_\Theta Z^{ij}, \nabla_\Theta Z &= 0 && (2.130)\\ \nabla_\Theta \delta^i_j, \nabla_\Theta \delta^{ij}_{rs}, \nabla_\Theta \delta^{ijk}_{rst} &= 0. && (2.131)\end{aligned}$$

The covariant derivatives of surface metrics vanishes except for surface coordinate vectors.

$$\begin{aligned}\nabla_\Theta \boldsymbol{S}_\alpha &= S^\beta_\Theta \nabla_\beta \boldsymbol{S}_\alpha = S^\beta_\Theta B_{\beta\alpha}\boldsymbol{N} && (2.132)\\ \nabla_\Theta \boldsymbol{S}^\alpha &= S^\beta_\Theta \nabla_\beta \boldsymbol{S}^\alpha = S^\beta_\Theta B^\alpha_\beta \boldsymbol{N} && (2.133)\\ \nabla_\Theta S_{\alpha\beta}, \nabla_\Theta S^{\alpha\beta}, \nabla_\Theta S &= 0 && (2.134)\\ \nabla_\Theta \varepsilon_{\alpha\beta}, \nabla_\Theta \varepsilon^{\alpha\beta} &= 0. && (2.135)\end{aligned}$$

2.3.2 Levi-Civita Symbols and Invariant Derivatives

The **curve Levi-Civita symbols** are defined similarly

$$\varepsilon_\Phi = \sqrt{U} e_\Phi, \varepsilon^\Phi = \frac{e^\Phi}{\sqrt{U}}. \qquad (2.136)$$

Since there is only one element in $U_{\Phi\Psi}$, $U_{11} = U, U^{11} = 1/U$ and it is trivial to write

$$\begin{aligned}e^\Phi U_{\Phi\Psi} &= U e_\Psi && (2.137)\\ e_\Phi U^{\Phi\Psi} &= U \ {}^{1} e^\Psi && (2.138)\\ \varepsilon_\Phi \varepsilon_\Psi &= U_{\Phi\Psi} && (2.139)\\ \varepsilon^\Phi \varepsilon^\Psi &= U^{\Phi\Psi} && (2.140)\\ \varepsilon^\Phi \varepsilon_\Psi &= \delta^\Phi_\Psi. && (2.141)\end{aligned}$$

Just like the surface gradient operator $\nabla_{\|} \equiv \boldsymbol{S}^{\alpha}\nabla_{\alpha}$, we can form an invariant derivative ∇_s along the curve by contraction of curve Levi-Civita tensor and the curve covariant gradient:

$$\nabla_s \equiv \varepsilon^{\Phi}\nabla_{\Phi} \tag{2.142}$$

and

$$\nabla_s^2 \equiv (\varepsilon^{\Phi}\nabla_{\Phi})(\varepsilon_{\Psi}\nabla^{\Psi}) = \nabla^{\Phi}\nabla_{\Phi}. \tag{2.143}$$

Conversely the covariant curve derivative is related to the invariant derivative by

$$\nabla_{\Phi} \;=\; \varepsilon_{\Phi}\nabla_s. \tag{2.144}$$

2.3.3 Darboux Frame

Now, the invariant unit tangent of the curve can be defined as

$$\begin{aligned} \boldsymbol{t} &= \varepsilon^{\Phi}\boldsymbol{U}_{\Phi} = \varepsilon^{\Phi}\nabla_{\Phi}\boldsymbol{R} = \nabla_s\boldsymbol{R} \\ &= \varepsilon^{\Phi}S_{\Phi}^{\alpha}\boldsymbol{S}_{\alpha} = \nabla_s S^{\alpha}\boldsymbol{S}_{\alpha} \\ &= \varepsilon^{\Phi}Z_{\Phi}^{i}\boldsymbol{Z}_i = \nabla_s Z^i\boldsymbol{Z}_i. \end{aligned} \tag{2.145}$$

The contravariant components of $\boldsymbol{t} = t^{\alpha}\boldsymbol{S}_{\alpha} = t_{\alpha}\boldsymbol{S}^{\alpha}$ are

$$t^{\alpha} \;=\; \boldsymbol{t}\cdot\boldsymbol{S}^{\alpha} = \nabla_s S^{\alpha} = \varepsilon^{\Phi}S_{\Phi}^{\alpha} \tag{2.146}$$

$$t_{\alpha} \;=\; \boldsymbol{t}\cdot\boldsymbol{S}_{\alpha} = \nabla_s S_{\alpha}. \tag{2.147}$$

The space components of $\boldsymbol{t} = t^i\boldsymbol{Z}_i$ are

$$t^i = \boldsymbol{t}\cdot\boldsymbol{Z}^i = \nabla_s Z^i = \varepsilon^{\Phi}Z_{\Phi}^{i}. \tag{2.148}$$

The following relations are trivial results of the definition for $\boldsymbol{t}$:

$$\boldsymbol{U}_{\Phi} \;=\; \varepsilon_{\Phi}\boldsymbol{t} \tag{2.149}$$

$$S_{\Phi}^{\alpha} \;=\; \varepsilon_{\Phi}t^{\alpha} \tag{2.150}$$

$$Z_{\Phi}^{i} \;=\; \varepsilon_{\Phi}t^{i}. \tag{2.151}$$

The unit normal $\boldsymbol{n}$ within the surface is defined as the **two dimensional cross product** of $\boldsymbol{t}$.

Definition 2.8. *In general, in D dimension, the cross product involves $(D\quad 1)$ vectors $U_{(1)}^{i}, \cdots,$ $U_{(D\quad 1)}^{i}$ and the component V_i of the resulting cross product is*

$$V_i = \varepsilon_{ij_1\cdots j_{D\quad 1}}U_{(1)}^{j_1}\cdots U_{(D\quad 1)}^{j_{D\quad 1}}. \tag{2.152}$$

With the definition Eq. (2.152) for the general cross product, the two dimensional cross product of $\boldsymbol{t}$ is

$$\boldsymbol{n} = n^{\alpha}\boldsymbol{S}_{\alpha} = \varepsilon^{\alpha\beta}t_{\beta}\boldsymbol{S}_{\alpha}. \tag{2.153}$$

The components of $\boldsymbol{n}$ in surface and space are

$$n_{\alpha} \;=\; \boldsymbol{S}_{\alpha}\cdot\boldsymbol{n} = \varepsilon_{\alpha\beta}t^{\beta} = \varepsilon_{\alpha\beta}\nabla_s S^{\beta} = \varepsilon_{\alpha\beta}\varepsilon^{\Phi}S_{\Phi}^{\beta} \tag{2.154}$$

$$n^{\alpha} \;=\; \boldsymbol{S}^{\alpha}\cdot\boldsymbol{n} = \varepsilon^{\alpha\beta}t_{\beta} = \varepsilon^{\alpha\beta}\nabla_s S_{\beta} \tag{2.155}$$

$$n^{i} \;=\; \boldsymbol{Z}^{i}\cdot\boldsymbol{n} = n^{\alpha}Z_{\alpha}^{i} = \varepsilon^{\alpha\beta}t_{\beta}Z_{\alpha}^{i}. \tag{2.156}$$

Along $\partial\mathcal{P}$, $\{\boldsymbol{t}, \boldsymbol{N}, \boldsymbol{n}\}$ forms an orthonormal coordinate systems known as the **Darboux frame**. The orthonormality of $\boldsymbol{t}$ and $\boldsymbol{n}$ can be seen from

$$\boldsymbol{t}\cdot\boldsymbol{t} = t^{\alpha}t_{\alpha} = \varepsilon^{\Phi}\boldsymbol{U}_{\Phi}\cdot\varepsilon^{\Psi}\boldsymbol{U}_{\Psi} = \varepsilon^{\Phi}\varepsilon^{\Psi}U_{\Phi\Psi} = U^{\Phi\Psi}U_{\Phi\Psi} = 1 \tag{2.157}$$

$$\boldsymbol{n}\cdot\boldsymbol{n} = n^{\alpha}n_{\alpha} = \varepsilon^{\alpha\beta}t_{\beta}\varepsilon_{\alpha\gamma}t^{\gamma} = \delta_{\alpha\gamma}^{\alpha\beta}t_{\beta}t^{\gamma} = \delta_{\gamma}^{\beta}t_{\beta}t^{\gamma} = t_{\beta}t^{\beta} = 1 \tag{2.158}$$

$$\boldsymbol{t}\cdot\boldsymbol{n} = t^{\alpha}n_{\alpha} = t_{\alpha}n^{\alpha} = t_{\alpha}\varepsilon^{\alpha\beta}t_{\beta} = 0. \tag{2.159}$$

The completeness relation takes a special form in the Darboux frame

$$\boldsymbol{t}\otimes\boldsymbol{t}+\boldsymbol{N}\otimes\boldsymbol{N}+\boldsymbol{n}\otimes\boldsymbol{n}=\mathbb{1}^{(3)}. \tag{2.160}$$

whose component form is

$$\begin{aligned} t^i t_j + n^i n_j + N^i N_j &= \delta^i_j & (2.161)\\ t^i t^j + n^i n^j + N^i N^j &= Z^{ij}. & (2.162)\end{aligned}$$

Within the surface, we have the completeness relation

$$\boldsymbol{S}^\alpha\otimes\boldsymbol{S}_\alpha=\boldsymbol{t}\otimes\boldsymbol{t}+\boldsymbol{n}\otimes\boldsymbol{n}=\mathbb{1}^{(2)}, \tag{2.163}$$

whose component form is

$$\begin{aligned} t^\alpha t_\beta + n^\alpha n_\beta &= \delta^\alpha_\beta & (2.164)\\ t^\alpha t^\beta + n^\alpha n^\beta &= S^{\alpha\beta}. & (2.165)\end{aligned}$$

The decomposition of the coordinate vector $\boldsymbol{S}_\alpha$ and $\boldsymbol{Z}_i$ in the Darboux frame is

$$\begin{aligned} \boldsymbol{S}_\alpha &= t_\alpha\boldsymbol{t}+n_\alpha\boldsymbol{n} & (2.166)\\ \boldsymbol{Z}_i &= t_i\boldsymbol{t}+n_i\boldsymbol{n}+N_i\boldsymbol{N}. & (2.167)\end{aligned}$$

The invariant operator ∇_s is the directional derivative along $\boldsymbol{t}$:

$$\nabla_s=\varepsilon^\Phi\nabla_\Phi=\varepsilon^\Phi\nabla_\Phi S^\alpha\nabla_\alpha=t^\alpha\nabla_\alpha=\boldsymbol{t}\cdot\nabla \tag{2.168}$$

The directional derivative along $\boldsymbol{n}$ is denoted by $\nabla_\perp$:

$$\nabla_\perp\equiv\boldsymbol{n}\cdot\nabla=n^\alpha\nabla_\alpha. \tag{2.169}$$

The surface derivative ∇_α can be decomposed into derivatives along the tangential and normal direction

$$\nabla_\alpha = (n^\beta n_\alpha+t^\beta t_\alpha)\nabla_\beta=n_\alpha n^\beta\nabla_\beta+t_\alpha t^\beta\nabla_\beta=n_\alpha\nabla_\perp+t_\alpha\nabla_s. \tag{2.170}$$

The decomposition of the space derivative ∇ in the Darboux frame is

$$\nabla=(\boldsymbol{t}\boldsymbol{t}+\boldsymbol{N}\boldsymbol{N}+\boldsymbol{n}\boldsymbol{n})\cdot\nabla=\boldsymbol{t}\nabla_s+\boldsymbol{N}\frac{\partial}{\partial N}+\boldsymbol{n}\nabla_\perp. \tag{2.171}$$

2.3.4 Curve Curvature

Rate of changes of the Darboux frame as we move along the curve gives curvature of the curve. Unlike surfaces, curves are intrinsically Euclidean and have no intrinsic curvature. All curvatures of curves are extrinsic and depend explicitly on the embedding manner. The rate of change of the Darboux frame is

$$\begin{aligned} \nabla_s\boldsymbol{t} &= \nabla_s(t^\alpha\boldsymbol{S}_\alpha)=\nabla_s t^\alpha\boldsymbol{S}_\alpha+t^\alpha\nabla_s\boldsymbol{S}_\alpha=\nabla_s t^\alpha(t_\alpha\boldsymbol{t}+n_\alpha\boldsymbol{n})+t^\alpha\nabla_s S^\beta\nabla_\beta\boldsymbol{S}_\alpha \\ &= n_\alpha\nabla_s t^\alpha\boldsymbol{n}+t^\alpha t^\beta B_{\alpha\beta}\boldsymbol{N}\equiv k_g\boldsymbol{n}+k_n\boldsymbol{N} & (2.172)\\ \nabla_s\boldsymbol{n} &= \nabla_s(n^\alpha\boldsymbol{S}_\alpha)=\nabla_s n^\alpha\boldsymbol{S}_\alpha+n^\alpha\nabla_s\boldsymbol{S}_\alpha=\nabla_s n^\alpha(t_\alpha\boldsymbol{t}+n_\alpha\boldsymbol{n})+n^\alpha\nabla_s S^\beta\nabla_\beta\boldsymbol{S}_\alpha \\ &= t_\alpha\nabla_s n^\alpha\boldsymbol{t}+n^\alpha t^\beta B_{\alpha\beta}\boldsymbol{N}= \quad k_g\boldsymbol{t}+\tau_g\boldsymbol{N} & (2.173)\\ \nabla_s\boldsymbol{N} &= t^\alpha\nabla_\alpha\boldsymbol{N}= \quad t^\alpha B_{\alpha\beta}\boldsymbol{S}^\beta= \quad t^\alpha B_{\alpha\beta}(t^\beta\boldsymbol{t}+n^\beta\boldsymbol{n}) \\ &= \quad t^\alpha t^\beta B_{\alpha\beta}\boldsymbol{t} \quad t^\alpha n^\beta B_{\alpha\beta}\boldsymbol{n}= \quad k_n\boldsymbol{t} \quad \tau_g\boldsymbol{n} & (2.174)\\ \nabla_\perp\boldsymbol{N} &= n^\alpha\nabla_\alpha\boldsymbol{N}= \quad n^\alpha B^\beta_\alpha\boldsymbol{S}_\beta= \quad n^\alpha(t_\beta\boldsymbol{t}+n_\beta\boldsymbol{n})B^\beta_\alpha \\ &= \quad n^\alpha t^\beta B_{\alpha\beta}\boldsymbol{t} \quad n^\alpha n^\beta B_{\alpha\beta}\boldsymbol{n}= \quad \tau_g\boldsymbol{t} \quad B_\perp\boldsymbol{n} & (2.175)\end{aligned}$$

where we defined the **geodesic curvature**

$$k_g=n_\alpha\nabla_s t^\alpha=\varepsilon_{\alpha\beta}t^\beta\nabla_s t^\alpha=\varepsilon_{\alpha\beta}\nabla_s S^\beta\nabla^2_s S^\alpha, \tag{2.176}$$

the **normal curvature**

$$k_n=t^\alpha t^\beta B_{\alpha\beta}=\nabla_s S^\alpha\nabla_s S^\beta B_{\alpha\beta}, \tag{2.177}$$

and the **geodesic torsion**

$$\tau_g = n^\alpha t^\beta B_{\alpha\beta} = n_\alpha t^\beta B^\alpha_\beta = \varepsilon_{\alpha\gamma} t^\gamma t^\beta B^\alpha_\beta = \varepsilon_{\alpha\gamma} \nabla_s S^\gamma \nabla_s S^\beta B^\alpha_\beta, \tag{2.178}$$

and

$$B_\perp \equiv n^\alpha n^\beta B_{\alpha\beta}. \tag{2.179}$$

Convention 2.9. smaller circles of radius $r < 1$ on a unit sphere will have geodesic curvature $k_g = \quad \sqrt{1 \quad r^2}/r$.

Obviously, $\begin{pmatrix} k_n & \tau_g \\ \tau_g & B_\perp \end{pmatrix}$ represents the curvature tensor $B_{\alpha\beta}$ in the Darboux frame. Since curvature scalars are invariant, along the curve, we have

$$\begin{aligned} K_M &= k_n + B_\perp &(2.180)\\ K &= k_n B_\perp \quad \tau_g^2. &(2.181)\end{aligned}$$

From the above definition, invariant derivatives of t^α and n^α are

$$\begin{aligned} \nabla_s t^\alpha &= \nabla_s(\boldsymbol{t} \cdot \boldsymbol{S}^\alpha) = \nabla_s \boldsymbol{t} \cdot \boldsymbol{S}^\alpha + \boldsymbol{t} \cdot \nabla_s S^\beta \nabla_\beta \boldsymbol{S}^\alpha = k_g n^\alpha &(2.182)\\ \nabla_s n^\alpha &= \nabla_s(\boldsymbol{n} \cdot \boldsymbol{S}^\alpha) = \nabla_s \boldsymbol{n} \cdot \boldsymbol{S}^\alpha + \boldsymbol{n} \cdot \nabla_s S^\beta \nabla_\beta \boldsymbol{S}^\alpha = \quad k_g t^a. &(2.183)\end{aligned}$$

Since $\nabla_s \boldsymbol{Z}^i, \nabla_s \boldsymbol{Z}_i = 0$, we can also write

$$\begin{aligned} \nabla_s t^i &= k_g n^i + k_n N^i &(2.184)\\ \nabla_s n^i &= \quad k_g t^i + \tau_g N^i &(2.185)\\ \nabla_s N^i &= \quad k_n t^i \quad \tau_g n^i. &(2.186)\end{aligned}$$

Chapter 3

Variation of the Hamiltonian for Soft Fludic Elastic Surfaces

In the 1970s, Helfrich proposed an elegant geometric model for lipid bilayers and successfully explained the biconcave shape of red blood cells [helfrich1973elastic], which says the elastic energy density of lipid bilayers is proportional to the mean curvature squared of the lipid bilayer. At the local scale, lipid fluidity, lipid stretching, lipid tilting and bilayer coupling can all contribute to the energy of the bilayer. It is therefore quite surprising that all the local chemistry and physics of lipids becomes just a small set of emergent parameters of a geometric Hamiltonian. On the large scale, the forms of the Hamiltonian is limited by requirement of symmetry and should only depend on geometric surface scalars. A natural expansion is proposed in [deserno2015fluid]

$$H = \int \mathrm{d}A\{C^{(0)} + C^{(1)}K_M + C^{(2,1)}K_M^2 + C^{(2,2)}K + C^{(3,1)}K_M^3 + C^{(3,2)}K_MK + C^{(4,1)}K_M^4 + C^{(4,2)}K_M^2K + C^{(4,3)}K^2 + C^{(4,4)}(\nabla_\alpha K_M)(\nabla^\alpha K_M) + \mathcal{O}(\text{length}\ \ ^{5})\}. \tag{3.1}$$

Usually only terms up to $\mathcal{O}(\text{length}\ \ ^{2})$ are adopted, although there are researches suggesting stability effects of higher order terms [siegel2010fourth]. For inhomogenous bilayers, the phenomenological coefficients can depend on local concentration of proteins and composition of lipids.

For lipid bilayers with open edges or/and phase coexistence, line energy can not be neglected. Based on a similar analysis of symmetry, a natural expansion of a general line energy can be written as

$$H = \int \mathrm{d}l\,\{c^{(0)} + c^{(1)}k_g + c^{(2)}k_n + c^{(3)}\tau_g + c^{(4)}B_\perp + c^{(5)}k_g^2 + c^{(6)}k_n^2 + \cdots\}. \tag{3.2}$$

This general line integral can be used to describe a variety of interactions. For instance, energy contribution of open edges of nanodiscoids stabilized by edge-reactant salts can be written as $H = \int_{\partial\mathcal{P}} \mathrm{d}l\,\{\tau + c_1(k_g \quad k_0)^2 + c_2k_n^2\}$ [morris2019solvated]. For membranes with phase coexistence, differences in Gaussian bending moduli will lead to terms like $H = \int \mathrm{d}l\,c^{(1)}k_g$ [baumgart2005membrane]. Based on the problem of interest, special forms of Eq. (3.1) and Eq. (3.2) can be chosen.

3.1 Variation of Surface Geometry

The variation of the Helfrich bending energy was first carried out by Ou-Yang [zhong1989bending]. Then this process is made more elegant by Capovilla and Guven via adopting covariant notations [capovilla2002stresses]. In most computations, vesicles are assumed to be homogeneous and all phenomenological coefficients constant. Thus the resulting bending forces from a general curvature Hamiltonian Eq. (3.1) is still lacking. In the following computation, we shall assume that all the phenomenological coefficients in Eq. (3.1) vary locally on the surface.

3.1.1 Variation of Metrics and Curvatures of Embedded Surface

Consider a variation of the position of the membrane given by $\boldsymbol{R}(S)\rightarrow\boldsymbol{R}(S)+\delta\boldsymbol{R}(S)$. Let us denote $\delta\boldsymbol{R}(S)$ by $\boldsymbol{W}$ and decompose it into tangential and normal directions: $\delta\boldsymbol{R}=\boldsymbol{W}(S)=W\boldsymbol{N}+W^\alpha\boldsymbol{S}_\alpha$. The variation of the tangent vectors and the normal $\boldsymbol{N}$ is

$$\begin{aligned}
\delta\boldsymbol{S}_\alpha &= \delta(\nabla_\alpha\boldsymbol{R})=\nabla_\alpha\delta\boldsymbol{R}=\nabla_\alpha\boldsymbol{W}=\nabla_\alpha(W\boldsymbol{N}+W^\beta\boldsymbol{S}_\beta)\\
&= \nabla_\alpha W\boldsymbol{N}\quad WB_\alpha^\beta\boldsymbol{S}_\beta+\nabla_\alpha W^\beta\boldsymbol{S}_\beta+W^\beta B_{\alpha\beta}\boldsymbol{N}\\
&= (\nabla_\alpha W+W^\beta B_{\alpha\beta})\boldsymbol{N}+(\nabla_\alpha W^\beta\quad WB_\alpha^\beta)\boldsymbol{S}_\beta && (3.3)\\
\delta\boldsymbol{N} &= \delta\boldsymbol{N}\cdot(\boldsymbol{S}_\alpha\boldsymbol{S}^\alpha+\boldsymbol{N}\boldsymbol{N})=\quad\boldsymbol{N}\cdot\delta\boldsymbol{S}_\alpha\boldsymbol{S}^\alpha=\quad(\nabla_\alpha W+W^\beta B_{\alpha\beta})\boldsymbol{S}^\alpha && (3.4)
\end{aligned}$$

The variation of metrics are

$$\begin{aligned}
\delta S_{\alpha\beta} &= \delta\boldsymbol{S}_\alpha\cdot\boldsymbol{S}_\beta+\boldsymbol{S}_\alpha\cdot\delta\boldsymbol{S}_\beta=(\nabla_\alpha W_\beta\quad WB_{\alpha\beta})+(\nabla_\beta W_\alpha\quad WB_{\alpha\beta})\\
&= \nabla_\alpha W_\beta+\nabla_\beta W_\alpha\quad 2WB_{\alpha\beta} && (3.5)\\
\delta S^{\alpha\beta} &= \delta_\gamma^\beta\delta S^{\alpha\gamma}=S^{\beta\sigma}S_{\sigma\gamma}\delta S^{\alpha\gamma}=S^{\beta\sigma}(\quad S^{\alpha\gamma}\delta S_{\sigma\gamma})\\
&= \quad S^{\beta\sigma}S^{\alpha\gamma}(\nabla_\sigma W_\gamma+\nabla_\gamma W_\sigma\quad 2WB_{\sigma\gamma})\\
&= \quad(\nabla^\alpha W^\beta+\nabla^\beta W^\alpha\quad 2WB^{\alpha\beta}) && (3.6)\\
\delta S &= \frac{1}{2!}e^{\alpha\beta}e^{\mu\nu}(\delta S_{\alpha\mu}S_{\beta\nu}+S_{\alpha\mu}\delta S_{\beta\nu})=e^{\alpha\beta}e^{\mu\nu}S_{\alpha\mu}\delta S_{\beta\nu}\\
&= S\varepsilon^{\alpha\beta}\varepsilon^{\mu\nu}S_{\alpha\mu}(\nabla_\beta W_\nu+\nabla_\nu W_\beta\quad 2WB_{\beta\nu})\\
&= S(S^{\alpha\mu}S^{\beta\nu}\quad S^{\alpha\nu}S^{\beta\mu})S_{\alpha\mu}(\nabla_\beta W_\nu+\nabla_\nu W_\beta\quad 2WB_{\beta\nu})\\
&= S(2S^{\beta\nu}\quad S^{\beta\nu})(\nabla_\beta W_\nu+\nabla_\nu W_\beta\quad 2WB_{\beta\nu})\\
&= 2S(\nabla_\beta W^\beta\quad WK_M)=2(\nabla_\parallel\cdot\boldsymbol{W})S, && (3.7)
\end{aligned}$$

where we used

$$\delta(\delta_\beta^\alpha)=\delta(S^{\alpha\gamma}S_{\gamma\beta})=0. \tag{3.8}$$

The variation of the surface Levi-Civita tensor and area differential $\mathrm{d}A$ is

$$\begin{aligned}
\delta\varepsilon_{\alpha\beta} &= \delta(\sqrt{S}e_{\alpha\beta})=\frac{\delta S}{2S}\sqrt{S}e_{\alpha\beta}=(\nabla_\gamma W^\gamma\quad WK_M)\varepsilon_{\alpha\beta} && (3.9)\\
\delta\varepsilon^{\alpha\beta} &= \delta\left(\frac{e^{\alpha\beta}}{\sqrt{S}}\right)=\quad\frac{\delta S}{S}\frac{e^{\alpha\beta}}{\sqrt{S}}=\quad(\nabla_\gamma W^\gamma\quad WK_M)\varepsilon^{\alpha\beta} && (3.10)\\
\delta(\mathrm{d}A) &= \delta(\sqrt{S}\mathrm{d}S^1\mathrm{d}S^2)=\frac{\delta S}{2S}\sqrt{S}\mathrm{d}S^1\mathrm{d}S^2=(\nabla_\alpha W^\alpha\quad WK_M)\mathrm{d}A=(\nabla_\parallel\cdot\boldsymbol{W})\mathrm{d}A. && (3.11)
\end{aligned}$$

To find variation of the curvature tensor, we need to find variation of of either $\nabla_\alpha\boldsymbol{N}$ or $\nabla_\alpha\boldsymbol{S}_\beta$:

$$\delta(\nabla_\gamma\boldsymbol{N})=\nabla_\gamma\delta\boldsymbol{N}=\quad\nabla_\gamma(\nabla_\alpha W+W^\beta B_{\alpha\beta})\boldsymbol{S}^\alpha\quad(\nabla_\alpha W+W^\beta B_{\alpha\beta})B_\gamma^\alpha\boldsymbol{N}. \tag{3.12}$$

Then the variation of curvature tensor is

$$\begin{aligned}
\delta B_{\alpha\beta} &= \quad\delta(\boldsymbol{S}_\beta\cdot\nabla_\alpha\boldsymbol{N})=\quad\delta\boldsymbol{S}_\beta\cdot\nabla_\alpha\boldsymbol{N}\quad\boldsymbol{S}_\beta\cdot\nabla_\alpha\delta\boldsymbol{N}\\
&= B_\alpha^\gamma\delta\boldsymbol{S}_\beta\cdot\boldsymbol{S}_\gamma\quad\boldsymbol{S}_\beta\cdot\nabla_\alpha\delta\boldsymbol{N}\\
&= B_\alpha^\gamma(\nabla_\beta W_\gamma\quad WB_{\beta\gamma})+\nabla_\alpha(\nabla_\mu W+W^\nu B_{\mu\nu})\boldsymbol{S}^\mu\cdot\boldsymbol{S}_\beta\\
&= B_\alpha^\gamma\nabla_\beta W_\gamma\quad WB_{\beta\gamma}B_\alpha^\gamma+\nabla_\alpha\nabla_\beta W+\nabla_\alpha(W^\nu B_{\beta\nu})\\
&= \nabla_\alpha B_{\beta\nu}W^\nu+B_{\alpha\nu}\nabla_\beta W^\nu+B_{\beta\nu}\nabla_\alpha W^\nu\quad B_{\beta\gamma}B_\alpha^\gamma W+\nabla_\alpha\nabla_\beta W. && (3.13)
\end{aligned}$$

The variation of the mean curvature is

$$\begin{aligned}
\delta K_M &= \delta(S^{\alpha\beta}B_{\alpha\beta})=\delta S^{\alpha\beta}B_{\alpha\beta}+S^{\alpha\beta}\delta B_{\alpha\beta}\\
&= (2WB^{\alpha\beta}\quad\nabla^\alpha W^\beta\quad\nabla^\beta W^\alpha)B_{\alpha\beta}+S^{\alpha\beta}\{B_\alpha^\gamma\nabla_\beta W_\gamma\quad WB_{\beta\gamma}B_\alpha^\gamma+\nabla_\alpha\nabla_\beta W+\nabla_\alpha(W^\nu B_{\beta\nu})\}\\
&= (2WB_\beta^\alpha B_\alpha^\beta\quad 2B_{\alpha\beta}\nabla^\alpha W^\beta)+B_\alpha^\gamma\nabla^\alpha W_\gamma\quad WB_\gamma^\alpha B_\alpha^\gamma+\nabla_\alpha\nabla^\alpha W+B_\nu^\alpha\nabla_\alpha W^\nu+W^\nu\nabla_\alpha B_\nu^\alpha\\
&= WB_\beta^\alpha B_\alpha^\beta+\nabla_\alpha\nabla^\alpha W+W^\nu\nabla_\alpha B_\nu^\alpha\\
&= WB_\beta^\alpha B_\alpha^\beta+\nabla_\alpha\nabla^\alpha W+W^\nu\nabla^\alpha B_{\alpha\nu}\\
&= W\,(K_M^2\quad 2K)+\nabla_\alpha\nabla^\alpha W+W^\nu\nabla_\nu K_M. && (3.14)
\end{aligned}$$

The variation of the Gaussian curvature is

$$\begin{aligned}
\delta K &= \delta\Big(\frac{1}{2!}\varepsilon^{\alpha\beta}\varepsilon^{\mu\nu}B_{\alpha\mu}B_{\beta\nu}\Big) = \frac{1}{2}e^{\alpha\beta}e^{\mu\nu}\delta\Big(\frac{1}{S}B_{\alpha\mu}B_{\beta\nu}\Big)\\
&= \frac{1}{2}e^{\alpha\beta}e^{\mu\nu}\Big(\quad\frac{\delta S}{S^2}B_{\alpha\mu}B_{\beta\nu}\Big) + \frac{1}{2}e^{\alpha\beta}e^{\mu\nu}\frac{1}{S}(\delta B_{\alpha\mu}B_{\beta\nu} + B_{\alpha\mu}\delta B_{\beta\nu})\\
&= \frac{1}{2}e^{\alpha\beta}e^{\mu\nu}\Big(\frac{1}{S}B_{\alpha\mu}B_{\beta\nu}\Big)\Big(\quad\frac{\delta S}{S}\Big) + e^{\alpha\beta}e^{\mu\nu}\frac{1}{S}B_{\alpha\mu}\delta B_{\beta\nu}\\
&= K\Big(\quad\frac{\delta S}{S}\Big) + (S^{\alpha\mu}S^{\beta\nu}\quad S^{\alpha\nu}S^{\beta\mu})B_{\alpha\mu}\delta B_{\beta\nu}\\
&= 2K(WK_M\quad\nabla_\beta W^\beta) + (K_M S^{\beta\nu}\quad B^{\nu\beta})\delta B_{\beta\nu}\\
&= 2K(WK_M\quad\nabla_\beta W^\beta) +\\
&\quad (K_M S^{\beta\nu}\quad B^{\nu\beta})(\nabla_\beta B_{\nu\gamma}W^\gamma + B_{\beta\gamma}\nabla_\nu W^\gamma + B_{\nu\gamma}\nabla_\beta W^\gamma\quad B_{\nu\gamma}B^\gamma_\beta W + \nabla_\beta\nabla_\nu W)\\
&= 2K(WK_M\quad\nabla_\beta W^\beta)\\
&\quad + K_M S^{\beta\nu}(\nabla_\beta B_{\nu\gamma}W^\gamma + B_{\beta\gamma}\nabla_\nu W^\gamma + B_{\nu\gamma}\nabla_\beta W^\gamma\quad B_{\nu\gamma}B^\gamma_\beta W + \nabla_\beta\nabla_\nu W)\\
&\quad\quad B^{\nu\beta}(\nabla_\beta B_{\nu\gamma}W^\gamma + B_{\beta\gamma}\nabla_\nu W^\gamma + B_{\nu\gamma}\nabla_\beta W^\gamma\quad B_{\nu\gamma}B^\gamma_\beta W + \nabla_\beta\nabla_\nu W)\\
&= 2K(WK_M\quad\nabla_\beta W^\beta)\\
&\quad + K_M(\nabla_\beta B^\beta_\gamma W^\gamma + B^\nu_\gamma\nabla_\nu W^\gamma + B^\beta_\gamma\nabla_\beta W^\gamma\quad B^\beta_\gamma B^\gamma_\beta W + \nabla_\beta\nabla^\beta W)\\
&\quad\quad B^{\nu\beta}(\nabla_\beta B_{\nu\gamma}W^\gamma + B_{\beta\gamma}\nabla_\nu W^\gamma + B_{\nu\gamma}\nabla_\beta W^\gamma\quad B_{\nu\gamma}B^\gamma_\beta W + \nabla_\beta\nabla_\nu W)\\
&= 2K(WK_M\quad\nabla_\beta W^\beta)\\
&\quad + K_M(\nabla_\beta B^\beta_\gamma W^\gamma + 2B^\beta_\gamma\nabla_\beta W^\gamma\quad B^\beta_\gamma B^\gamma_\beta W + \nabla_\beta\nabla^\beta W)\\
&\quad\quad B^{\nu\beta}(\nabla_\beta B_{\nu\gamma}W^\gamma + 2B_{\nu\gamma}\nabla_\beta W^\gamma\quad B_{\nu\gamma}B^\gamma_\beta W + \nabla_\beta\nabla_\nu W)\\
&= (2KK_M + B^{\nu\beta}B_{\nu\gamma}B^\gamma_\beta\quad K_M B^\beta_\gamma B^\gamma_\beta)W + K_M\nabla_\beta\nabla^\beta W\quad B^{\nu\beta}\nabla_\nu\nabla_\beta W\\
&\quad\quad 2K\nabla_\beta W^\beta + (K_M\nabla_\beta B^\beta_\gamma\quad B^{\nu\beta}\nabla_\beta B_{\nu\gamma})W^\gamma + 2(K_M B^\beta_\gamma\quad B^{\nu\beta}B_{\nu\gamma})\nabla_\beta W^\gamma\\
&= KK_M W + K_M\nabla_\beta\nabla^\beta W\quad B^{\nu\beta}\nabla_\nu\nabla_\beta W\\
&\quad\quad 2K\nabla_\beta W^\beta + (K_M\nabla_\beta B^\beta_\gamma\quad B^{\nu\beta}\nabla_\beta B_{\nu\gamma})W^\gamma + 2K\delta^\beta_\gamma\nabla_\beta W^\gamma\\
&= KK_M W + K_M\nabla_\beta\nabla^\beta W\quad B^{\nu\beta}\nabla_\nu\nabla_\beta W + W^\gamma\nabla_\gamma K
\end{aligned}\tag{3.15}$$

where we used

$$2KK_M + B^{\nu\beta}B_{\nu\gamma}B^\gamma_\beta\quad K_M B^\beta_\gamma B^\gamma_\beta = 2KK_M + (K_M B^\beta_\gamma\quad K\delta^\beta_\gamma)B^\gamma_\beta\quad K_M B^\beta_\gamma B^\gamma_\beta = KK_M \tag{3.16}$$

$$K_M B^\beta_\gamma\quad B^{\nu\beta}B_{\nu\gamma} = K\delta^\beta_\gamma \tag{3.17}$$

and

$$K_M\nabla_\beta B^\beta_\gamma\quad B^{\nu\beta}\nabla_\beta B_{\nu\gamma} = \frac{1}{2}\nabla_\gamma(K_M)^2\quad\frac{1}{2}\nabla_\gamma(B^{\nu\beta}B_{\nu\beta}) = \frac{1}{2}\nabla_\gamma(2K) = \nabla_\gamma K. \tag{3.18}$$

In the next part, we apply the above results to calculate various forms of surface integrals.

3.1.2 Variation of Surface Hamiltonian

Example 3.1. (variation of a surface functional) Consider an energy functional defined as $\int_{\mathcal{P}} f\,\mathrm{d}A$. Then,

$$\begin{aligned}
\delta\int_{\mathcal{P}} f\,\mathrm{d}A &= \int_{\mathcal{P}}\Big(\delta f + f\frac{\delta(\mathrm{d}A)}{\mathrm{d}A}\Big)\mathrm{d}A\\
&= \int_{\mathcal{P}}\{\delta f + f(\nabla_\alpha W^\alpha\quad WK_M)\}\mathrm{d}A\\
&= \int_{\mathcal{P}}(\delta f\quad\nabla_\alpha f W^\alpha\quad fK_M W)\mathrm{d}A + \int_{\mathcal{P}}\nabla_\alpha(fW^\alpha)\mathrm{d}A\\
&= \int_{\mathcal{P}}(\delta f\quad\nabla_\alpha f W^\alpha\quad fK_M W)\mathrm{d}A + \int_{\partial\mathcal{P}} n_\alpha f W^\alpha\mathrm{d}l\\
&= \int_{\mathcal{P}}\delta\boldsymbol{R}\cdot\Big(\frac{\delta f}{\delta\boldsymbol{R}}\quad\nabla_\parallel f\quad fK_M\boldsymbol{N}\Big)\mathrm{d}A + \int_{\partial\mathcal{P}}\delta\boldsymbol{R}\cdot(f\boldsymbol{n})\mathrm{d}l
\end{aligned}\tag{3.19}$$

where in the last step we used Stokes' law to convert a surface integral into a line integral. Thus forces from the energy density f is

$$\frac{\delta\int_{\mathcal{P}} f\mathrm{d}A}{\delta\boldsymbol{R}} = \int_{\mathcal{P}}\left(\quad\frac{\delta f}{\delta\boldsymbol{R}} + \nabla_{\|} f + fK_M\boldsymbol{N}\right)\mathrm{d}A + \int_{\partial\mathcal{P}}(\quad f\boldsymbol{n})\mathrm{d}l \tag{3.20}$$

where the integrand in the surface integral represents force per unit area and that in the line integral represents boundary force.

Example 3.2. (variation of surface area) The surface area of patch $\mathcal{P}$ is

$$A = \int_{\mathcal{P}}\mathrm{d}A, \tag{3.21}$$

whose functional derivative is

$$\frac{\delta A}{\delta\boldsymbol{R}} = \int_{\mathcal{P}}(K_M\boldsymbol{N})\mathrm{d}A + \int_{\partial\mathcal{P}}(\quad\boldsymbol{n})\mathrm{d}A \tag{3.22}$$

which can be interpreted as surface tension.

Example 3.3. (variation of enclosed volume) The enclosed volume V of a closed surface $\mathcal{P}$ is

$$V = \frac{1}{3}\oiint_{\mathcal{P}}\boldsymbol{R}\cdot\boldsymbol{N}\mathrm{d}A \tag{3.23}$$

and

$$\begin{aligned}
\delta V &= \frac{1}{3}\int_{\mathcal{P}}\left(\boldsymbol{W}\cdot\boldsymbol{N} + \boldsymbol{R}\cdot\delta\boldsymbol{N} + \boldsymbol{R}\cdot\boldsymbol{N}\frac{\delta A}{\mathrm{d}A}\right)\mathrm{d}A \\
&= \frac{1}{3}\int_{\mathcal{P}}\{W \quad \boldsymbol{R}\cdot\boldsymbol{S}^{\alpha}(\nabla_{\alpha}W + W^{\beta}B_{\alpha\beta}) + \boldsymbol{R}\cdot\boldsymbol{N}(\nabla_{\alpha}W^{\alpha} \quad WK_M)\}\mathrm{d}A \\
&= \frac{1}{3}\int_{\mathcal{P}}\{W \quad \boldsymbol{R}\cdot\boldsymbol{S}^{\alpha}\nabla_{\alpha}W \quad \boldsymbol{R}\cdot\boldsymbol{S}^{\alpha}W^{\beta}B_{\alpha\beta} + \boldsymbol{R}\cdot\boldsymbol{N}\nabla_{\alpha}W^{\alpha} \quad \boldsymbol{R}\cdot\boldsymbol{N}WK_M\}\mathrm{d}A \\
&= \frac{1}{3}\int_{\mathcal{P}}\{W + W\nabla_{\alpha}(\boldsymbol{R}\cdot\boldsymbol{S}^{\alpha}) \quad \boldsymbol{R}\cdot\boldsymbol{S}^{\alpha}W^{\beta}B_{\alpha\beta} \quad W^{\alpha}\nabla_{\alpha}(\boldsymbol{R}\cdot\boldsymbol{N}) \quad \boldsymbol{R}\cdot\boldsymbol{N}WK_M\}\mathrm{d}A \\
&= \frac{1}{3}\int_{\mathcal{P}}\{W + W(\boldsymbol{S}_{\alpha}\cdot\boldsymbol{S}^{\alpha} + \boldsymbol{R}\cdot K_M\boldsymbol{N}) \quad \boldsymbol{R}\cdot\boldsymbol{S}^{\alpha}W^{\beta}B_{\alpha\beta} \quad W^{\alpha}(\quad\boldsymbol{R}\cdot B^{\beta}_{\alpha}\boldsymbol{S}_{\beta}) \quad \boldsymbol{R}\cdot\boldsymbol{N}WK_M\}\mathrm{d}A \\
&= \frac{1}{3}\int_{\mathcal{P}}\{W + 2W\}\mathrm{d}A \\
&= \int_{\mathcal{P}}W\mathrm{d}A.
\end{aligned} \tag{3.24}$$

Thus,

$$\frac{\delta V}{\delta\boldsymbol{R}} = \oiint_{\mathcal{P}}(\quad\boldsymbol{N})\mathrm{d}A. \tag{3.25}$$

Example 3.4. (variation of energy due to material properties) Let $f = f(S)$ be a function of the surface coordinate. It can be used to represent energy due to scalar fields such as a phase order parameter describing lipid species or a local Lagrange multiplier (a local tension field) to enforce incompressibility condition. Note that f describes material properties like a color field not material quantities. Thus local stretching or compression will not affect its value. Therefore, $\delta f = 0$ and Eq.(3.20) reduces to

$$\frac{\delta\int_{\mathcal{P}} f\mathrm{d}A}{\delta\boldsymbol{R}} = \int_{\mathcal{P}}(\nabla_{\|} f + fK_M\boldsymbol{N})\mathrm{d}A + \int_{\partial\mathcal{P}}(\quad f\boldsymbol{n})\mathrm{d}l \tag{3.26}$$

Note that $\delta f = 0$ means $f(S)$ as a function of surface coordinates will not change under a virtual position variation $\delta\boldsymbol{R}$ but $f(\boldsymbol{R})$ will change. Indeed,

$$\delta f(\boldsymbol{R}) \;=\; f(\boldsymbol{R} \quad \delta\boldsymbol{R}) \quad f(\boldsymbol{R}) = \quad \delta\boldsymbol{R}\cdot\nabla f. \tag{3.27}$$

Therefore, if the surface moves with velocity $\boldsymbol{V} = V\boldsymbol{N} + V^{\alpha}\boldsymbol{S}_{\alpha}$, then the PDE (partial differential equation) satisfied by $f(\boldsymbol{R}, t)$ reads

$$\frac{\partial f(\boldsymbol{R}, t)}{\partial t} = \quad \frac{\delta\boldsymbol{R}}{\delta t}\cdot\nabla f = \quad \boldsymbol{V}\cdot\nabla f \tag{3.28}$$

which is simply the advection equation for field f.

Example 3.5. (incompressibility condition and local Lagrange Multiplier) Suppose physical forces on an elastic surface leads to a velocity field $\boldsymbol{V} = V\boldsymbol{N} + V^{\alpha}\boldsymbol{S}_{\alpha}$ on the surface. Now we add a local Lagrange multiplier $\oint f dA$ to enforce incompressibility. We assume that force density is proportional to velocity and this proportionality constant is ζ. Then the additional velocity field $\boldsymbol{V}_f$ from the condition of incompressibility derives from $\zeta\boldsymbol{V}_f = \boldsymbol{S}_{\alpha}\nabla^{\alpha} f + K_M f\boldsymbol{N}$. Incompressibility implies vanishing surface divergence for total velocity:

$$\nabla_{\|}\cdot(\boldsymbol{V} + \boldsymbol{V}_f) = 0, \tag{3.29}$$

which gives an equation to calculate the local Lagrange multiplier f

$$\nabla_{\|}\cdot(\zeta\boldsymbol{V}_f) = \nabla_{\alpha}\nabla^{\alpha} f \quad (K_M)^2 f = \quad \nabla_{\|}\cdot(\zeta\boldsymbol{V}). \tag{3.30}$$

Example 3.6. (variation of a concentration field due to passive advection) Consider a concentration field c living on the surface. If the surface moves with velocity $\boldsymbol{V} = V\boldsymbol{N} + V^{\alpha}\boldsymbol{S}_{\alpha}$, during a differential time interval Δt, the differential changes of surface position is given by $\delta\boldsymbol{R} = \boldsymbol{W} = \Delta t\boldsymbol{V}$. Under this variation, the concentration field $c \to c + \delta c$ and an infinitesimal area $\mathrm{d}A \to \mathrm{d}A + \delta(\mathrm{d}A)$. Since there is only stretching and compression, the total amount of c within a small patch of membrane with area $\mathrm{d}A$ remains invariant. We therefore have $c\mathrm{d}A = (c + \delta c)(\mathrm{d}A + \delta(\mathrm{d}A))$, which gives

$$\delta c = \quad c\frac{\delta(\mathrm{d}A)}{\mathrm{d}A} = \quad (\nabla_{\|}\cdot\boldsymbol{W})c, \tag{3.31}$$

where we ignored the second order term $\delta c\delta(\mathrm{d}A)$.

Example 3.7. (variation of a concentration dependent energy due to passive advection of the concentration field) Now consider a Hamiltonian $\int_{\mathcal{P}} f(c)\mathrm{d}A$ depending only on the concentration c of some membrane molecules, so $\delta f = (\delta f/\delta c)\delta c$. Then,

$$\begin{aligned}
&\delta\int_{\mathcal{P}} f(c)\mathrm{d}A\\
= &\int_{\mathcal{P}}\left(\frac{\delta f}{\delta c}\delta c + f\frac{\delta(\mathrm{d}A)}{\mathrm{d}A}\right)\mathrm{d}A\\
= &\int_{\mathcal{P}}\left(\quad \frac{\delta f}{\delta c}c\frac{\delta(\mathrm{d}A)}{\mathrm{d}A} + f\frac{\delta(\mathrm{d}A)}{\mathrm{d}A}\right)\mathrm{d}A\\
= &\int_{\mathcal{P}}\left(f \quad \frac{\delta f}{\delta c}c\right)(\nabla_{\alpha}W^{\alpha} \quad WK_M)\mathrm{d}A\\
= &\int_{\mathcal{P}}(\quad W^{\alpha}\nabla_{\alpha} \quad WK_M)\left(f \quad \frac{\delta f}{\delta c}c\right)\mathrm{d}A + \int_{\mathcal{P}}\nabla_{\alpha}\left\{\left(f \quad \frac{\delta f}{\delta c}c\right)W^{\alpha}\right\}\mathrm{d}A\\
= &\quad \int_{\mathcal{P}}\left\{W^{\alpha}\nabla_{\alpha}\left(f \quad \frac{\delta f}{\delta c}c\right) + \left(f \quad \frac{\delta f}{\delta c}c\right)K_M W\right\}\mathrm{d}A + \int_{\partial\mathcal{P}} n^{\alpha}\left(f \quad \frac{\delta f}{\delta c}c\right)W^{\alpha}\mathrm{d}A\\
= &\quad \int_{\mathcal{P}}\delta\boldsymbol{R}\cdot\left\{\nabla_{\|}\left(f \quad \frac{\delta f}{\delta c}c\right) + \left(f \quad \frac{\delta f}{\delta c}c\right)K_M\boldsymbol{N}\right\}\mathrm{d}A + \delta\boldsymbol{R}\cdot\int_{\partial\mathcal{P}}\left(f \quad \frac{\delta f}{\delta c}c\right)\boldsymbol{n}\mathrm{d}A.
\end{aligned} \tag{3.32}$$

Thus

$$\frac{\delta \int_{\mathcal{P}} f(c)\mathrm{d}A}{\delta \boldsymbol{R}} = \int_{\mathcal{P}} \left\{ \nabla_{\|}\left(f \quad c\frac{\partial f}{\partial c}\right) + K_M \left(f \quad c\frac{\partial f}{\partial c}\right)\boldsymbol{N} \right\} \mathrm{d}A + \int_{\partial \mathcal{P}} \left\{ \quad \left(f \quad c\frac{\partial f}{\partial c}\right)\boldsymbol{n} \right\} \mathrm{d}l. \tag{3.33}$$

which is basically Eq. (3.26) with the substitution

$$f \to f \quad c\frac{\partial f}{\partial c}.$$

Since f in Eq. (3.26) can be interpreted as a local surface tension, we may say that dependence of the surface energy density f on concentration c of membrane molecules leads to an extra tension term of $\quad c(\partial f / \partial c)$. Let us denote by $\sigma_c[f(c)]$ this equivalent tension for a concentration dependent surface energy $f(c)$

$$\sigma_c[f(c)] = f(c) \quad c\frac{\partial f(c)}{\partial c}. \tag{3.34}$$

Example 3.8. (incompressibility and local area elasticity) Solving for a local Lagrange multiplier can be computationally expensive and numerically inaccurate. In another approach from [aland2014diffuse], a scalar field c is introduced to measure local compression and stretching. Physically, c may be interpreted as density of lipid molecules of the vesicle. The hypothesis of being incompressible means density of lipids should remain constant during surface deformation. Dynamics of c is dictated by the conservation law

$$\frac{\partial c}{\partial t} + \nabla_{\|} \cdot (c\boldsymbol{v}) = 0, \tag{3.35}$$

where $\boldsymbol{v}$ is the velocity field of the vesicle. Initial value of c is set to be 1 everywhere. When $c > 1 (c < 1)$, the vesicle is locally compressed(stretched). Then an energy penalty $\int f_{\mathrm{in}}(c) dA$ is added to the Hamiltonian of the system where $f_{\mathrm{in}}(c)$ has a local minimum at $c = 1$. Conventional choices for $f_{\mathrm{in}}(c)$ are

$$\begin{aligned} f_{1,\mathrm{in}}(c) &= \frac{\mu}{2}(c \quad 1)^2 & (3.36)\\ f_{2,\mathrm{in}}(c) &= \frac{\mu}{2}\left(\frac{1}{c} \quad 1\right)^2 & (3.37)\\ f_{3,\mathrm{in}}(c) &= \frac{\mu}{2}\left(c + \frac{1}{c} \quad 2\right)^2, & (3.38) \end{aligned}$$

and

$$\begin{aligned} \sigma_c[f_{1,\mathrm{in}}(c)] &= \frac{1}{2}(c \quad 1)^2 \quad c(c \quad 1) = \frac{1}{2}(c \quad 1)(\quad c \quad 1) = \frac{1}{2}(1 \quad c^2) & (3.39)\\ \sigma_c[f_{2,\mathrm{in}}(c)] &= \frac{1}{2}\left(\frac{1}{c} \quad 1\right)^2 \quad c\left(\frac{1}{c} \quad 1\right)\left(\quad \frac{1}{c^2}\right) = \frac{1}{2}\left(\frac{1}{c} \quad 1\right)\left(\frac{3}{c} \quad 1\right) & (3.40)\\ \sigma_c[f_{3,\mathrm{in}}(c)] &= \frac{1}{2}\left(\frac{1}{c} + c \quad 2\right)^2 \quad c\left(\frac{1}{c} + c \quad 2\right)\left(1 \quad \frac{1}{c^2}\right) = \frac{1}{2c^2}(3 + c)(1 \quad c)^3. & (3.41) \end{aligned}$$

We require $f_{\mathrm{in}}(c)$ 1) approach ∞ when c approaches 0 or ∞ 2) when $c \sim 1 + \epsilon$, be approximately quadratic in the small deviation ϵ. Obviously, none of the conventional choices meet our requirements. We therefore take

$$f_{\mathrm{in}}(c) = f_{1,\mathrm{in}}(c) + f_{2,\mathrm{in}}(c) = \frac{\mu}{2}\left[\left(\frac{1}{c} \quad 1\right)^2 + (1 \quad c)^2\right]. \tag{3.42}$$

The equivalent tensor related to $f_{\mathrm{in}}(c)$ is

$$\sigma_c[f_{\mathrm{in}}(c)] = \frac{\mu}{2}\left[\left(\frac{1}{c} \quad 1\right)\left(\frac{3}{c} \quad 1\right) + (1 \quad c^2)\right]. \tag{3.43}$$

Example 3.9. (variation of a concentration field due to active motion) Consider a biomembrane where protein and lipid molecules reside. The energy functional typically depends on the concentration of those molecules. If we vary the position of those molecules, we can find the associated force on those them and derive dynamical equations for them. Suppose the variation of the position of the molecules is due to the action of an instantaneous tangential velocity $\boldsymbol{V} = V^\alpha \boldsymbol{S}_\alpha$ for a small time period Δt. Then the **variation of the position of the molecules** is $\delta_p \boldsymbol{R} = \boldsymbol{W} = \Delta t \boldsymbol{V} = \Delta t V^\alpha \boldsymbol{S}_\alpha$, where the subscript p is used to stress that this variation is related to postilion changes of protein molecules (in contrast to the one without this subscript related to changes in surface position). The variation of the total amount of molecules within a small patch of surface is due to flux through the boundary of this patch

$$\delta_p \int_{\mathcal{P}} c_p \, dA \;=\; \Delta t \int_{\partial\mathcal{P}} (c_p \boldsymbol{V}) \cdot \boldsymbol{n} dl = \int_{\partial\mathcal{P}} n_\alpha c_p \, \Delta t V^\alpha dl = \int_{\mathcal{P}} \nabla_\alpha (c_p \, \Delta t V^\alpha) dA. \tag{3.44}$$

Thus, the variation of the concentration is given by

$$\delta_p c_p = \quad \nabla_\alpha (c_p \Delta t V^\alpha) = \quad \nabla_\alpha (c_p W^\alpha). \tag{3.45}$$

Example 3.10. (variation of a concentration dependent Hamiltonian due to active motion of material quantities) Now the variation of an energy density f due to $\delta_p c_p$ is thus $\delta_p f = \frac{\partial f}{\partial c_p} \delta c_p$ and

$$\begin{aligned}
\delta_p \int_{\mathcal{P}} f(c_p) \mathrm{d}A &= \int_{\mathcal{P}} \delta_p f \, \mathrm{d}A = \int_{\mathcal{P}} \frac{\partial f}{\mathrm{d}c_p} \delta c_p \mathrm{d}A \\
&= \quad \int_{\mathcal{P}} \frac{\partial f}{\partial c_p} \nabla_\alpha (c_p \, W^\alpha) \mathrm{d}A \\
&= \int_{\mathcal{P}} \left(c_p W^\alpha \nabla_\alpha \left(\frac{\partial f}{\partial c_p} \right) \right) \mathrm{d}A \quad \int_{\mathcal{P}} \nabla_\alpha \left(\frac{\partial f}{\partial c_p} c_p \, W^\alpha \right) \mathrm{d}A \\
&= \int_{\mathcal{P}} \left(c_p \, W^\alpha \nabla_\alpha \left(\frac{\partial f}{\partial c_p} \right) \right) \mathrm{d}A \quad \int_{\partial\mathcal{P}} \frac{\partial f}{\partial c_p} c_p \, n_\alpha W^\alpha \mathrm{d}l,
\end{aligned} \tag{3.46}$$

which gives

$$\frac{\delta_p \int_{\mathcal{P}} f(c_p) \mathrm{d}A}{\delta \boldsymbol{R}_p} = \int_{\mathcal{P}} \left(\quad c_p \nabla_{\parallel} \left(\frac{\partial f}{\partial c_p} \right) \right) \mathrm{d}A + \int_{\partial\mathcal{P}} \left(\frac{\partial f}{\partial c_p} c_p \boldsymbol{n} \right) \mathrm{d}l. \tag{3.47}$$

We thus can interpret the integrand $\left(\quad c_p \nabla_{\parallel} \frac{\partial f}{\partial c_p} \right)$ of the surface integral in Eq. (3.47) as molecular forces from lipids to protein molecules.

Example 3.11. (variation of bending energy without spontaneous curvature) Consider a variable bending moduli $\kappa(S)$ and assume that $\delta\kappa = 0$.

$$\begin{aligned}
& \delta \int_{\mathcal{P}} \frac{\kappa}{2} K_M^2 \mathrm{d}A \\
= & \int_{\mathcal{P}} \left\{ \kappa K_M \delta K_M + \frac{\kappa}{2} K_M^2 \frac{\delta(\mathrm{d}A)}{\mathrm{d}A} \right\} \mathrm{d}A \\
= & \int_{\mathcal{P}} \left\{ \kappa K_M (W B^\alpha_\beta B^\beta_\alpha + \nabla^\alpha \nabla_\alpha W + W^\nu \nabla_\nu K_M) + \frac{\kappa}{2} K_M^2 (\nabla_\alpha W^\alpha \quad W K_M) \right\} \mathrm{d}A \\
= & \int_{\mathcal{P}} \left\{ \kappa W \left(K_M B^\alpha_\beta B^\beta_\alpha \quad \frac{1}{2} K_M^3 \right) + \kappa K_M \nabla^\alpha \nabla_\alpha W + \kappa \left(W^\alpha K_M \nabla_\alpha K_M + \frac{1}{2} K_M^2 \nabla_\alpha W^\alpha \right) \right\} \mathrm{d}A \\
= & \int_{\mathcal{P}} \left\{ \kappa W \left(\frac{1}{2} K_M^3 \quad 2 K_M K \right) + W \nabla_\alpha \nabla^\alpha (\kappa K_M) + \nabla^\alpha [\kappa K_M \nabla_\alpha W \quad W \nabla_\alpha (\kappa K_M)] \right\} \mathrm{d}A \\
& + \int_{\mathcal{P}} \left\{ \quad \frac{1}{2} K_M^2 W^\alpha \nabla_\alpha \kappa + \nabla_\alpha \left(\frac{\kappa}{2} K_M^2 W^\alpha \right) \right\} \mathrm{d}A
\end{aligned}$$

$$\begin{aligned}
&= \int_{\mathcal{P}}\left\{W\left[\kappa\left(\frac{1}{2}K_M^3 \quad 2K_MK\right)+\Delta_{\|}(\kappa K_M)\right] \quad \frac{1}{2}W^\alpha K_M^2\nabla_\alpha\kappa\right\}\mathrm{d}A\\
&\quad+\int_{\mathcal{P}}\left\{\nabla_\alpha\left[\kappa K_M\nabla^\alpha W \quad W\nabla^\alpha(\kappa K_M)+\frac{\kappa}{2}K_M^2W^\alpha\right]\right\}\mathrm{d}A,
\end{aligned}$$

where we used

$$\begin{aligned}
K_MB^\alpha_\beta B^\beta_\alpha \quad \frac{1}{2}K_M^3 &= K_M(K_M^2 \quad 2K) \quad \frac{1}{2}K_M^3=\frac{1}{2}K_M^3 \quad 2K_MK\\
\kappa K_M\nabla^\alpha\nabla_\alpha W &= \quad \nabla^\alpha(\kappa K_M)\nabla_\alpha W+\nabla^\alpha(\kappa K_M\nabla_\alpha W)\\
&= W\nabla_\alpha\nabla^\alpha(\kappa K_M)+\nabla^\alpha[\kappa K_M\nabla_\alpha W \quad W\nabla_\alpha\kappa K_M]\\
\kappa\left(W^\alpha K_M\nabla_\alpha K_M+\frac{1}{2}K_M^2\nabla_\alpha W^\alpha\right) &= \kappa\nabla_\alpha\left(\frac{1}{2}K_M^2W^\alpha\right)\\
&= \quad \frac{1}{2}K_M^2W^\alpha\nabla_\alpha\kappa+\nabla_\alpha\left(\frac{\kappa}{2}K_M^2W^\alpha\right)
\end{aligned}$$

and

$$\begin{aligned}
&\int_{\mathcal{P}}\left\{\nabla_\alpha\left[\kappa K_M\nabla^\alpha W \quad W\nabla^\alpha(\kappa K_M)+\frac{\kappa}{2}K_M^2W^\alpha\right]\right\}\mathrm{d}A\\
&= \int_{\mathcal{P}}n_\alpha\left\{\kappa K_M\nabla^\alpha W \quad W\nabla^\alpha(\kappa K_M)+\frac{\kappa}{2}K_M^2W^\alpha\right\}\mathrm{d}A\\
&= \int_{\mathcal{P}}\left\{\kappa K_M\nabla_\perp W \quad W\nabla_\perp(\kappa K_M)+\frac{\kappa}{2}K_M^2\boldsymbol{W}\cdot\boldsymbol{n}\right\}\mathrm{d}A.
\end{aligned}$$

Thus,

$$\begin{aligned}
\frac{\delta\int_{\mathcal{P}}\frac{\kappa}{n}K_M^2\mathrm{d}A}{\delta\boldsymbol{R}} &= \frac{\delta\int_{\mathcal{P}}\kappa\delta \quad \frac{1}{n}K_M^2\mathrm{d}A)}{\delta\boldsymbol{R}}\\
&= \int_{\mathcal{P}}\left\{\boldsymbol{N}\left[\kappa\left(2K_MK \quad \frac{1}{2}K_M^3\right) \quad \Delta_{\|}(\kappa K_M)\right]+\frac{K_M^2}{2}\nabla_{\|}\kappa\right\}\mathrm{d}A\\
&\quad+\int_{\partial\mathcal{P}}\left\{\boldsymbol{N}\nabla_\perp(\kappa K_M) \quad \left(\frac{\kappa}{2}K_M^2\right)\boldsymbol{n}\right\}\mathrm{d}l. \qquad (3.48)
\end{aligned}$$

Example 3.12. (variation of $\boldsymbol{K_M^n, n>2}$) Consider $f=\frac{\kappa(S)}{n}(K_M)^n$, where n is a positive integer and $\kappa(S)$ is a locally varying bending modulus for the mean curvature and $\delta\kappa=0$.

$$\begin{aligned}
&\delta\int_{\mathcal{P}}\frac{\kappa}{n}K_M^n\mathrm{d}A\\
&= \int_{\mathcal{P}}\left\{\kappa K_M^{n \quad 1}\delta K_M+\frac{\kappa}{n}K_M^n\frac{\delta(\mathrm{d}A)}{\mathrm{d}A}\right\}\mathrm{d}A\\
&= \int_{\mathcal{P}}\left\{\kappa K_M^{n \quad 1}(WB^\alpha_\beta B^\beta_\alpha+\nabla^\alpha\nabla_\alpha W+W^\nu\nabla_\nu K_M)+\frac{\kappa}{n}K_M^n(\nabla_\alpha W^\alpha \quad WK_M)\right\}\mathrm{d}A\\
&= \int_{\mathcal{P}}\left\{\kappa W\left(K_M^{n \quad 1}B^\alpha_\beta B^\beta_\alpha \quad \frac{1}{n}K_M^{n+1}\right)+\kappa K_M^{n \quad 1}\nabla^\alpha\nabla_\alpha W+\kappa\left(W^\alpha K_M^{n \quad 1}\nabla_\alpha K_M+\frac{K_M^n}{n}\nabla_\alpha W^\alpha\right)\right\}\mathrm{d}A\\
&= \int_{\mathcal{P}}\left\{\kappa W\left(\frac{n \quad 1}{n}K_M^{n+1} \quad 2K_M^{n \quad 1}K\right)+W\nabla_\alpha\nabla^\alpha(\kappa K_M^{n \quad 1})\right\}\mathrm{d}A\\
&\quad+\int_{\mathcal{P}}\nabla^\alpha\{\kappa K_M^{n \quad 1}\nabla_\alpha W \quad W\nabla_\alpha(\kappa K_M^{n \quad 1})\}\mathrm{d}A\\
&\quad+\int_{\mathcal{P}}\left\{ \quad \frac{1}{n}K_M^nW^\alpha\nabla_\alpha\kappa+\nabla_\alpha\left(\frac{\kappa}{n}K_M^nW^\alpha\right)\right\}\mathrm{d}A\\
&= \int_{\mathcal{P}}\left\{W\left[\kappa\left(\frac{n \quad 1}{n}K_M^{n+1} \quad 2K_M^{n \quad 1}K\right)+\Delta_{\|}(\kappa K_M^{n \quad 1})\right] \quad \frac{1}{n}W^\alpha K_M^n\nabla_\alpha\kappa\right\}\mathrm{d}A\\
&\quad+\int_{\mathcal{P}}\left\{\nabla_\alpha\left[\kappa K_M^{n \quad 1}\nabla^\alpha W \quad W\nabla^\alpha(\kappa K_M^{n \quad 1})+\frac{\kappa}{n}K_M^nW^\alpha\right]\right\}\mathrm{d}A
\end{aligned}$$

where we used

$$\begin{aligned}
K_M^{n\quad 1}B_\beta^\alpha B_\alpha^\beta \quad \frac{1}{n}K_M^{n+1} &= K_M^{n\quad 1}(K_M^2 \quad 2K) \quad \frac{1}{n}K_M^{n+1} = \frac{n\quad 1}{n}K_M^{n+1} \quad 2K_M^{n\quad 1}K \\
\kappa K_M^{n\quad 1}\nabla^\alpha\nabla_\alpha W &= \quad \nabla^\alpha(\kappa K_M^{n\quad 1})\nabla_\alpha W + \nabla^\alpha(\kappa K_M^{n\quad 1}\nabla_\alpha W) \\
&= W\nabla_\alpha\nabla^\alpha(\kappa K_M^{n\quad 1}) + \nabla^\alpha[\kappa K_M^{n\quad 1}\nabla_\alpha W \quad W\nabla_\alpha \kappa K_M^{n\quad 1}] \\
\kappa\left(W^\alpha K_M^{n\quad 1}\nabla_\alpha K_M + \frac{1}{n}K_M^n\nabla_\alpha W^\alpha\right) &= \kappa\nabla_\alpha\left(\frac{1}{n}K_M^n W^\alpha\right) \\
&= \quad \frac{1}{n}K_M^n W^\alpha\nabla_\alpha\kappa + \nabla_\alpha\left(\frac{\kappa}{n}K_M^n W^\alpha\right)
\end{aligned}$$

and

$$\begin{aligned}
&\int_{\mathcal{P}}\left\{\nabla_\alpha\left[\kappa K_M^{n\quad 1}\nabla^\alpha W \quad W\nabla^\alpha(\kappa K_M^{n\quad 1}) + \frac{\kappa}{n}K_M^n W^\alpha\right]\right\}\mathrm{d}A \\
= &\int_{\mathcal{P}} n_\alpha\left\{\kappa K_M^{n\quad 1}\nabla^\alpha W \quad W\nabla^\alpha(\kappa K_M^{n\quad 1}) + \frac{\kappa}{n}K_M^n W^\alpha\right\}\mathrm{d}A \\
= &\int_{\mathcal{P}}\left\{\kappa K_M^{n\quad 1}\nabla_\perp W \quad W\nabla_\perp(\kappa K_M^{n\quad 1}) + \frac{\kappa}{n}K_M^n \boldsymbol{W}\cdot\boldsymbol{n}\right\}\mathrm{d}A.
\end{aligned}$$

Then

$$\begin{aligned}
\frac{\delta\int_{\mathcal{P}}\frac{\kappa}{n}K_M^n\mathrm{d}A}{\delta\boldsymbol{R}} &= \frac{\delta\int_{\mathcal{P}}\kappa\delta\ \frac{1}{n}K_M^n\mathrm{d}A)}{\delta\boldsymbol{R}} \\
&= \int_{\mathcal{P}}\left\{\boldsymbol{N}\left[\kappa\left(2K_M^{n\quad 1}K \quad \frac{n\quad 1}{n}K_M^{n+1}\right) \quad \Delta_\parallel(\kappa K_M^{n\quad 1})\right] + \frac{K_M^n}{n}\nabla_\parallel\kappa\right\}\mathrm{d}A \\
&\quad + \int_{\partial\mathcal{P}}\left\{\boldsymbol{N}\nabla_\perp(\kappa K_M^{n\quad 1}) \quad \left(\frac{\kappa}{n}K_M^n\right)\boldsymbol{n}\right\}\mathrm{d}l \qquad (3.49)
\end{aligned}$$

Example 3.13. (variation of Gaussian Bending energy)

$$\begin{aligned}
&\delta\int_{\mathcal{P}} K\,\mathrm{d}A \\
= &\int_{\mathcal{P}}\left(\delta K + K\frac{\delta(\mathrm{d}A)}{\mathrm{d}A}\right)\mathrm{d}A \\
= &\int_{\mathcal{P}}\{\delta K + K(\nabla_\alpha W^\alpha \quad WK_M)\}\mathrm{d}A \\
= &\int_{\mathcal{P}}\{K_M\nabla_\beta\nabla^\beta W \quad B^{\nu\beta}\nabla_\nu\nabla_\beta W + \nabla_\gamma K W^\gamma + K\nabla_\alpha W^\alpha\}\mathrm{d}A \\
= &\int_{\mathcal{P}}\{\quad \nabla_\beta K_M\nabla^\beta W + \nabla_\nu B^{\nu\beta}\nabla_\beta W\}\mathrm{d}A + \int_{\mathcal{P}}\nabla_\alpha(K_M\nabla^\alpha W \quad B^{\alpha\beta}\nabla_\beta W + KW^\alpha)\mathrm{d}A \\
= &\int_{\partial\mathcal{P}} n_\alpha(K_M\nabla^\alpha W \quad B^{\alpha\beta}\nabla_\beta W + KW^\alpha)\mathrm{d}l \\
= &\int_{\partial\mathcal{P}}(K_M\nabla_\perp W \quad n_\alpha t_\beta B^{\alpha\beta}\nabla_s W \quad n_\alpha n_\beta B^{\alpha\beta}\nabla_\perp W + Kn_\alpha W^\alpha)\mathrm{d}l \\
= &\int_{\partial\mathcal{P}}(K_M\nabla_\perp W + \nabla_s(n_\alpha t_\beta B^{\alpha\beta})W \quad n_\alpha n_\beta B^{\alpha\beta}\nabla_\perp W + Kn_\alpha W^\alpha)\mathrm{d}l \\
= &\int_{\partial\mathcal{P}}(K_M\nabla_\perp W + \nabla_s\tau_g W \quad n_\alpha n_\beta B^{\alpha\beta}\nabla_\perp W + Kn_\alpha W^\alpha)\mathrm{d}l \\
= &\int_{\partial\mathcal{P}}\{(K_M \quad n_\alpha n_\beta B^{\alpha\beta})\nabla_\perp W + \nabla_s\tau_g W + Kn_\alpha W^\alpha\}\mathrm{d}l \\
= &\int_{\partial\mathcal{P}}\{k_n\nabla_\perp W + \nabla_s\tau_g W + Kn_\alpha W^\alpha\}\mathrm{d}l.
\end{aligned}$$

Thus,

$$\frac{\delta \int_{\mathcal{P}} K \mathrm{d}A}{\delta \boldsymbol{R}} = \oint_{\partial \mathcal{P}} \{\ \ \nabla_s \tau_g \boldsymbol{N} \quad K \boldsymbol{n}\} \mathrm{d}l, \tag{3.50}$$

Example 3.14. (variation of K^n)

$$\begin{aligned}
&\delta \int_{\mathcal{P}} \frac{1}{n} \kappa(s) K^n \,\mathrm{d}A \\
= &\int_{\mathcal{P}} \kappa \left(K^{n\ \ 1} \delta K + \frac{1}{n} K^n \frac{\delta(\mathrm{d}A)}{\mathrm{d}A} \right) \mathrm{dA} \\
= &\int_{\mathcal{P}} \kappa \left\{ K^{n\ \ 1} \delta K + \frac{1}{n} K^n (\nabla_\alpha W^\alpha \quad W B^\alpha_\alpha) \right\} \mathrm{dA} \\
= &\int_{\mathcal{P}} \kappa \left\{ K^{n\ \ 1} (K K_M W + K_M \Delta_{||} W \quad B^{\nu\beta} \nabla_\nu \nabla_\beta W + W^\gamma \nabla_\gamma K) + \frac{K^n}{n} (\nabla_\alpha W^\alpha \quad W K_M) \right\} \mathrm{d}A \\
= &\int_{\mathcal{P}} \left\{ \frac{n \quad 1}{n} \kappa K^n K_M W + \kappa K^{n\ \ 1} K_M \nabla_\beta \nabla^\beta W \quad \kappa K^{n\ \ 1} B^{\nu\beta} \nabla_\nu \nabla_\beta W + \kappa K^{n\ \ 1} W^\gamma \nabla_\gamma K \right\} \mathrm{d}A \\
&+ \int_{\mathcal{P}} \kappa \frac{K^n}{n} \nabla_\alpha W^\alpha \mathrm{d}A \\
= &\int_{\mathcal{P}} W \left\{ \frac{n \quad 1}{n} \kappa K^n K_M + \Delta_{||} (\kappa K^{n\ \ 1} K_M) \quad \nabla_\beta \nabla_\nu (\kappa K^{n\ \ 1} B^{\nu\beta}) \right\} \mathrm{dA} \\
&+ \int_{\mathcal{P}} W^\gamma \left\{ \kappa K^{n\ \ 1} \nabla_\gamma K \quad \nabla_\gamma \left(\kappa \frac{K^n}{n} \right) \right\} dA + \int_{\mathcal{P}} \nabla_\alpha \left(\kappa \frac{K^n}{n} W^\alpha \right) \mathrm{d}A \\
&+ \int_{\mathcal{P}} \nabla_\alpha \{ \kappa K^{n\ \ 1} K_M \nabla^\alpha W \quad W \nabla^\alpha (\kappa K^{n\ \ 1} K_M) \quad \kappa K^{n\ \ 1} B^{\alpha\beta} \nabla_\beta W + W \nabla_\beta (\kappa K^{n\ \ 1} B^{\beta\alpha}) \} \mathrm{d}A
\end{aligned}$$

where we used

$$\begin{aligned}
\kappa K^{n\ \ 1} K_M \nabla_\beta \nabla^\beta W &= \nabla_\beta (\kappa K^{n\ \ 1} K_M \nabla^\beta W) \quad \nabla_\beta (\kappa K^{n\ \ 1} K_M) \nabla^\beta W \\
&= \nabla_\beta [\kappa K^{n\ \ 1} K_M \nabla^\beta W \quad W \nabla^\beta (\kappa K^{n\ \ 1} K_M)] + W \Delta_{||} (\kappa K^{n\ \ 1} K_M) \\
\kappa K^{n\ \ 1} B^{\nu\beta} \nabla_\nu \nabla_\beta W &= \nabla_\nu (\kappa K^{n\ \ 1} B^{\nu\beta} \nabla_\beta W) \quad \nabla_\beta (\kappa K^{n\ \ 1} B^{\beta\nu}) \nabla_\nu W \\
&= \nabla_\nu [\kappa K^{n\ \ 1} B^{\nu\beta} \nabla_\beta W \quad W \nabla_\beta (\kappa K^{n\ \ 1} B^{\beta\nu})] + W \nabla_\nu \nabla_\beta (\kappa K^{n\ \ 1} B^{\beta\nu}) \\
\kappa \frac{K^n}{n} \nabla_\alpha W^\alpha &= \nabla_\alpha \left(\kappa \frac{K^n}{n} W^\alpha \right) \quad W^\alpha \nabla_\alpha \left(\kappa \frac{K^n}{n} \right)
\end{aligned}$$

and the boundary term can be written as

$$\begin{aligned}
&\int_{\mathcal{P}} \nabla_\alpha \{ \kappa K^{n\ \ 1} K_M \nabla^\alpha W \quad W \nabla^\alpha (\kappa K^{n\ \ 1} K_M) \quad \kappa K^{n\ \ 1} B^{\alpha\beta} \nabla_\beta W + W \nabla_\beta (\kappa K^{n\ \ 1} B^{\beta\alpha}) \} \mathrm{d}A \\
= &\int_{\partial \mathcal{P}} \{ \kappa K^{n\ \ 1} K_M \nabla_\perp W \quad W \nabla_\perp (\kappa K^{n\ \ 1} K_M) \quad \kappa K^{n\ \ 1} n_\alpha B^{\alpha\beta} \nabla_\beta W + W n_\alpha \nabla_\beta (\kappa K^{n\ \ 1} B^{\beta\alpha}) \} \mathrm{d}l \\
= &\int_{\partial \mathcal{P}} \{ \kappa K^{n\ \ 1} K_M \nabla_\perp W \quad W \nabla_\perp (\kappa K^{n\ \ 1} K_M) \quad \kappa K^{n\ \ 1} (\tau_g \nabla_s W + B_\perp \nabla_\perp W) \} \mathrm{d}l \\
&+ \int_{\partial \mathcal{P}} \{ W n_\alpha \nabla_\beta (\kappa K^{n\ \ 1} B^{\beta\alpha}) \} \mathrm{d}l \\
= &\int_{\partial \mathcal{P}} \{ \kappa K^{n\ \ 1} (K_M \quad B_\perp) \nabla_\perp W \quad W \nabla_\perp (\kappa K^{n\ \ 1} K_M) + W \nabla_s (\kappa K^{n\ \ 1} \tau_g) \} \mathrm{d}l \\
&+ \int_{\partial \mathcal{P}} W \{ n_\alpha B^{\beta\alpha} \nabla_\beta (\kappa K^{n\ \ 1}) + \kappa K^{n\ \ 1} n_\alpha (\nabla_\beta B^{\beta\alpha}) \} \mathrm{d}l \\
= &\int_{\partial \mathcal{P}} \{ \kappa K^{n\ \ 1} (K_M \quad B_\perp) \nabla_\perp W \} \mathrm{d}l + \\
&\int_{\partial \mathcal{P}} W \{\ \ \nabla_\perp (\kappa K^{n\ \ 1} K_M) + \nabla_s (\kappa K^{n\ \ 1} \tau_g) + \kappa K^{n\ \ 1} \nabla_\perp K_M + \tau_g \nabla_s (\kappa K^{n\ \ 1}) + B_\perp \nabla_\perp (\kappa K^{n\ \ 1}) \} \partial l
\end{aligned}$$

$$
\begin{aligned}
&= \int_{\partial\mathcal{P}} \{\kappa K^{n\ \ 1} k_n \nabla_\perp W\}\partial l + \\
&\quad \int_{\partial\mathcal{P}} W\{\ \ K_M \nabla_\perp(\kappa K^{n\ \ 1}) + \nabla_s(\kappa K^{n\ \ 1}\tau_g) + \tau_g \nabla_s(\kappa K^{n\ \ 1}) + B_\perp \nabla_\perp(\kappa K^{n\ \ 1})\}\mathrm{d}l \\
&= \int_{\partial\mathcal{P}} \{\kappa K^{n\ \ 1} k_n \nabla_\perp W\}\mathrm{d}l + \\
&\quad \int_{\partial\mathcal{P}} W\{\ \ k_n \nabla_\perp(\kappa K^{n\ \ 1}) + \nabla_s(\kappa K^{n\ \ 1}\tau_g) + \tau_g \nabla_s(\kappa K^{n\ \ 1})\}\mathrm{d}l,
\end{aligned}
$$

from which we have the integrand normal component of the bulk term of $\ \ \delta\int(\kappa K^n/n)dA/\delta\boldsymbol{R}$ is

$$
\begin{aligned}
&\boldsymbol{N}\left\{\frac{n\ \ 1}{n}\kappa K^n K_M + \Delta_{||}(\kappa K^{n\ \ 1}K_M)\quad \nabla_\beta\nabla_\nu(\kappa K^{n\ \ 1}B^{\nu\beta})\right\} \\
=\quad &\boldsymbol{N}\left\{\frac{n\ \ 1}{n}\kappa K^n K_M + \Delta_{||}(\kappa K^{n\ \ 1}K_M)\quad B^{\beta\nu}\nabla_\beta\nabla_\nu(\kappa K^{n\ \ 1})\quad \kappa K^{n\ \ 1}\nabla_\beta\nabla_\nu B^{\beta\nu}\right\} \\
=\quad &\boldsymbol{N}\left\{\frac{n\ \ 1}{n}\kappa K^n K_M + \Delta_{||}(\kappa K^{n\ \ 1}K_M)\quad B^{\beta\nu}\nabla_\beta\nabla_\nu(\kappa K^{n\ \ 1})\quad \kappa K^{n\ \ 1}\Delta_{||}K_M\right\} \\
=\quad &\boldsymbol{N}\left\{\frac{n\ \ 1}{n}\kappa K^n K_M + K_M\Delta_{||}(\kappa K^{n\ \ 1})\quad B^{\beta\nu}\nabla_\beta\nabla_\nu(\kappa K^{n\ \ 1})\right\}
\end{aligned}
$$

and the integrand tangential component of the bulk term of $\ \ \delta\int(\kappa K^n/n)dA/\delta\boldsymbol{R}$ is

$$
\left\{\kappa K^{n\ \ 1}\nabla_{||}K\quad \nabla_{||}\left(\kappa\frac{K^n}{n}\right)\right\} = \nabla_{||}\left(\kappa\frac{K^n}{n}\right)\quad \kappa\nabla_{||}\frac{K^n}{n} = \frac{K^n}{n}\nabla_{||}\kappa
$$

and the integrand boundary term of $\ \ \delta\int(\kappa K^n/n)dA/\delta\boldsymbol{R}$ is

$$
\kappa\frac{K^n}{n}\boldsymbol{n}\quad \{\nabla_s(\kappa K^{n\ \ 1}\tau_g) + \tau_g\nabla_s(\kappa K^{n\ \ 1})\quad k_n\nabla_\perp(\kappa K^{n\ \ 1})\}\boldsymbol{N}.
$$

Thus,

$$
\frac{\delta\int(\kappa K^n/n)dA}{\delta\nabla_\perp(\boldsymbol{W}\cdot\boldsymbol{N})} \quad=\quad \int_{\partial\mathcal{P}}\kappa K^{n\ \ 1}k_n\mathrm{d}l, \tag{3.51}
$$

and

$$
\begin{aligned}
&\frac{\delta\int_{\mathcal{P}}\frac{\kappa(S)}{n}K^n\mathrm{d}A}{\delta\boldsymbol{R}} = \frac{\delta\int_{\mathcal{P}}\kappa\delta\ \ \frac{1}{n}K^n\mathrm{d}A)}{\delta\boldsymbol{R}} \\
=\quad &\int_{\mathcal{P}}\left\{\boldsymbol{N}\left[B^{\alpha\beta}\nabla_\alpha\nabla_\beta(\kappa K^{n\ \ 1})\quad \frac{n\ \ 1}{n}\kappa K^n K_M\quad K_M\Delta_{||}(\kappa K^{n\ \ 1})\right] + \frac{K^n}{n}\nabla_{||}\kappa\right\}\mathrm{d}A + \\
&\int_{\partial\mathcal{P}}\left\{\boldsymbol{N}[k_n\nabla_\perp(\kappa K^{n\ \ 1})\quad \nabla_s(\kappa K^{n\ \ 1}\tau_g)\quad \tau_g\nabla_s(\kappa K^{n\ \ 1})]\quad \left(\kappa\frac{K^n}{n}\right)\boldsymbol{n}\right\}\mathrm{d}l,
\end{aligned} \tag{3.52}
$$

from which we see that for vesicles with varying Gaussian bending moduli, which can be a result of phase separation and coexistence or varying lipid and protein species, the elastic forces from the Gaussian curvature energy is generally non-zero.

Example 3.15. (variation of $\boldsymbol{K_M^n K^m}$) Combining the results for K_M^n and K^n, it is straight forward to write down variation of $K_M^n K^m$, where n, m are positive integers. Indeed,

$$
\begin{aligned}
&\frac{\delta\int_{\mathcal{P}}\frac{\kappa}{nm}K_M^n K^m\mathrm{d}A}{\delta\boldsymbol{R}} \\
=\quad &\frac{1}{\delta\boldsymbol{R}}\left\{\int_{\mathcal{P}}\frac{\kappa K^m}{m}\delta\left(\frac{K_M^n}{n}\mathrm{d}A\right) + \int_{\mathcal{P}}\frac{\kappa K_M^n}{n}\delta\left(\frac{K^m}{m}\mathrm{d}A\right)\quad \int_{\mathcal{P}}\frac{\kappa}{nm}K_M^n K^m\delta(\mathrm{d}A)\right\}
\end{aligned} \tag{3.53}
$$

where the first term has the same form as Eq. (3.49) with $\kappa\to\frac{\kappa K^m}{m}$, the second term has the same form as Eq. (3.52) with $\kappa\to\frac{\kappa K_M^m}{m}$ and the last term the same form as Eq. (3.26) with $f\to\frac{\kappa}{nm}K_M^n K^m$.

Example 3.16. Consider $f=\frac{\kappa(S)}{2}\nabla_\alpha K_M\nabla^\alpha K_M$. Then

$$\begin{aligned}
&\delta\int_{\mathcal{P}}\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\mathrm{d}A\\
=\ &\int_{\mathcal{P}}\Big(\kappa\nabla_\alpha K_M\nabla^\alpha\delta K_M+\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\frac{\delta(\mathrm{d}A)}{\mathrm{d}A}\Big)\mathrm{d}A\\
=\ &\int_{\mathcal{P}}\Big\{\nabla^\alpha(\kappa\delta K_M\nabla_\alpha K_M)\quad\delta K_M\nabla^\alpha(\kappa\nabla_\alpha K_M)+\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M(\nabla_\alpha W^\alpha\quad WK_M)\Big\}\mathrm{d}A
\end{aligned}$$

where the first term is

$$\begin{aligned}
&\int_{\mathcal{P}}\nabla^\alpha(\kappa\delta K_M\nabla_\alpha K_M)\mathrm{d}A\\
=\ &\int_{\partial\mathcal{P}}n^\alpha\kappa\delta K_M\nabla_\alpha K_M\mathrm{d}l\\
=\ &\int_{\partial\mathcal{P}}\kappa\{W(K_M^2\quad 2K)+\nabla^\alpha\nabla_\alpha W+W^\nu\nabla_\nu K_M\}\nabla_\perp K_M\mathrm{d}l
\end{aligned}$$

where

$$\begin{aligned}
&\int_{\partial\mathcal{P}}\kappa\nabla_\perp K_M\nabla^\alpha\nabla_\alpha W\,\mathrm{d}l\\
=\ &\int_{\partial\mathcal{P}}\{\kappa\nabla_\perp K_M t^\alpha\nabla_s\nabla_\alpha W+\kappa\nabla_\perp K_M n^\alpha\nabla_\perp\nabla_\alpha W\}\mathrm{d}l\\
=\ &\int_{\partial\mathcal{P}}\{\quad\nabla_s(\kappa\nabla_\perp K_M t^\alpha)\nabla_\alpha W+\kappa\nabla_\perp K_M n^\alpha\nabla_\perp\nabla_\alpha W\}\mathrm{d}l\\
=\ &\int_{\partial\mathcal{P}}\{\quad[\nabla_s(\kappa\nabla_\perp K_M)t^\alpha+\kappa\nabla_\perp K_M\nabla_s t^\alpha]\nabla_\alpha W+\kappa\nabla_\perp K_M n^\alpha\nabla_\perp\nabla_\alpha W\}\mathrm{d}l\\
=\ &\int_{\partial\mathcal{P}}\{\quad\nabla_s(\kappa\nabla_\perp K_M)\nabla_s W\quad\kappa\nabla_\perp K_M k_g\nabla_\perp W+\kappa\nabla_\perp K_M n^\alpha\nabla_\perp\nabla_\alpha W\}\mathrm{d}l\\
=\ &\int_{\partial\mathcal{P}}\{\nabla_s^2(\kappa\nabla_\perp K_M)W\quad\kappa\nabla_\perp K_M k_g\nabla_\perp W+\kappa\nabla_\perp K_M n^\alpha\nabla_\perp\nabla_\alpha W\}\mathrm{d}l
\end{aligned}$$

and the second term is

$$\begin{aligned}
&\int_{\mathcal{P}}\delta K_M\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}A\\
=\ &\int_{\mathcal{P}}\{W(K_M^2\quad 2K)+\nabla_\beta\nabla^\beta W+W^\nu\nabla_\nu K_M\}\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}A
\end{aligned}$$

where

$$\begin{aligned}
&\int_{\mathcal{P}}\nabla_\beta\nabla^\beta W\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}A\\
=\ &\int_{\mathcal{P}}\nabla_\beta\{\nabla^\beta W\nabla^\alpha(\kappa\nabla_\alpha K_M)\}\mathrm{d}A\quad\int_{\mathcal{P}}\nabla^\beta W\nabla_\beta\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}A\\
=\ &\int_{\partial\mathcal{P}}n_\beta\nabla^\beta W\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}l\quad\int_{\mathcal{P}}\nabla^\beta\{W\nabla_\beta\nabla^\alpha(\kappa\nabla_\alpha K_M)\}\mathrm{d}A+\int_{\mathcal{P}}W\nabla^\beta\nabla_\beta\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}A\\
=\ &\int_{\partial\mathcal{P}}\nabla_\perp W\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}l\quad\int_{\partial\mathcal{P}}W\nabla_\perp\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}l+\int_{\mathcal{P}}W\nabla^\beta\nabla_\beta\nabla^\alpha(\kappa\nabla_\alpha K_M)\mathrm{d}A
\end{aligned}$$

and the third term is

$$\begin{aligned}
&\int_{\mathcal{P}}\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M(\nabla_\beta W^\beta\quad WK_M)\mathrm{d}A\\
=\ &\quad\int_{\mathcal{P}}\frac{\kappa}{2}WK_M\nabla_\alpha K_M\nabla^\alpha K_M\mathrm{d}A\\
&+\int_{\mathcal{P}}\nabla_\beta\Big(W^\beta\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\Big)\mathrm{d}A\quad\int_{\mathcal{P}}W^\beta\nabla_\beta\Big(\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\Big)\mathrm{d}A\\
=\ &\quad\int_{\mathcal{P}}\frac{\kappa}{2}WK_M\nabla_\alpha K_M\nabla^\alpha K_M\mathrm{d}A+\int_{\partial\mathcal{P}}n_\beta W^\beta\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\mathrm{d}A\quad\int_{\mathcal{P}}W^\beta\nabla_\beta\Big(\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\Big)\mathrm{d}A.
\end{aligned}$$

Then,

$$\frac{\delta\int_{\mathcal{P}}\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M \mathrm{d}A}{\delta \boldsymbol{R}}$$

$$= \int_{\mathcal{P}}\Big\{\boldsymbol{N}\Big[\quad (K_M^2 \quad 2K)\nabla^\alpha(\kappa\nabla_\alpha K_M) \quad \Delta_\parallel\nabla^\alpha(\kappa\nabla_\alpha K_M) + \frac{\kappa}{2}K_M\nabla_\alpha K_M\nabla^\alpha K_M\Big] + \nabla_\parallel\Big(\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\Big)\Big\}\mathrm{d}A + \int_{\partial\mathcal{P}}\Big\{\boldsymbol{N}[\quad \kappa(K_M^2 \quad 2K)\nabla_\perp K_M \quad \nabla_s^2(\kappa\nabla_\perp K_M) + \nabla_\perp\nabla^\alpha(\kappa\nabla_\alpha K_M)] \quad \nabla_\perp K_M\nabla_\parallel K_M \quad \frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M\boldsymbol{n}\Big\}\mathrm{d}l \tag{3.54}$$

and

$$\frac{\delta\int\frac{\kappa}{2}\nabla_\alpha K_M\nabla^\alpha K_M dA}{\delta\nabla_\perp(\boldsymbol{X}\cdot\boldsymbol{N})} = \kappa k_g\nabla_\perp K_M \quad \nabla^\alpha(\kappa\nabla_\alpha K_M).$$

With all the examples listed in this section, all the variation of the curvature Hamiltonian Eq. (3.1) are obtained.

3.2 Variation of Embedded Curve Geometry

Now we turn to the variation of the line energy Eq. (3.2). The variation of the first two terms of Eq. (3.2) has been obtained by Capovilla [capovilla2002lipid] and then Tu [tu2003lipid] via the method of differential forms. In this section, we shall calculate the variation of all terms in Eq. (3.2) in a covariant form.

3.2.1 Variation of Metrics and Curvatures of Embedded Curves

The variation of the position of the curve is just $\delta\boldsymbol{R}$ restricted to the curve, which can be written as $\delta\boldsymbol{R}=\boldsymbol{W}(U)=W\boldsymbol{N}+W^\alpha\boldsymbol{S}_\alpha=W^i\boldsymbol{Z}_i$, where W^i are components of $\boldsymbol{W}$ in the $\boldsymbol{Z}_i$ direction. Since $\boldsymbol{W}(U)$ is restricted to the curve, $\nabla_\perp\boldsymbol{W}=0$. The variations of the tangent vector is

$$\delta\boldsymbol{U}_\Phi=\delta(\nabla_\Phi\boldsymbol{R})=\nabla_\Phi\delta\boldsymbol{R}=\nabla_\Phi\boldsymbol{W}=\nabla_\Phi W^i\boldsymbol{Z}_i=\varepsilon_\Phi\nabla_s W^i\boldsymbol{Z}_i. \tag{3.55}$$

The variation of the metrics are

$$\begin{aligned}
\delta U_{\Phi\Psi} &= 2\varepsilon_\Phi\varepsilon_\Psi\boldsymbol{t}\cdot\nabla_s\boldsymbol{W}=2U_{\Phi\Psi}\boldsymbol{t}\cdot\nabla_s\boldsymbol{W} &(3.56)\\
\delta U &= e^\Phi e^\Psi\delta U_{\Phi\Psi}=2U\boldsymbol{t}\cdot\nabla_s\boldsymbol{W} &(3.57)\\
\delta\varepsilon^\Phi &= \quad \varepsilon^\Phi\boldsymbol{t}\cdot\nabla_s\boldsymbol{W} &(3.58)\\
\delta\varepsilon_\Phi &= \varepsilon_\Phi\boldsymbol{t}\cdot\nabla_s W. &(3.59)
\end{aligned}$$

The variation of the metrics of the curve is

$$\begin{aligned}
\delta U_{\Phi\Psi} &= \delta\boldsymbol{U}_\Phi\cdot\boldsymbol{U}_\Psi+\boldsymbol{U}_\Phi\cdot\delta\boldsymbol{U}_\Psi=(\varepsilon_\Phi Z_\Psi^i+\varepsilon_\Psi Z_\Phi^i)\nabla_s W_i\\
&= 2\varepsilon_\Phi\varepsilon_\Psi t^i\nabla_s W_i=2U_{\Phi\Psi}t^i\nabla_s W_i &(3.60)\\
\delta U &= e^\Phi e^\Psi\delta U_{\Phi\Psi}=2e^\Phi e^\Psi U_{\Phi\Psi}t^i\nabla_s W_i=2Ut^i\nabla_s W_i &(3.61)\\
\delta\varepsilon^\Phi &= \delta\Big(\frac{e^\Phi}{\sqrt{U}}\Big)=\quad\frac{e^\Phi}{\sqrt{U}}\frac{\delta U}{2U}=\quad\varepsilon^\Phi t^i\nabla_s W_i &(3.62)\\
\delta\varepsilon_\Phi &= \delta(\sqrt{U}e_\Phi)=\sqrt{U}e_\Phi\frac{\delta U}{2U}=\varepsilon_\Phi t^i\nabla_s W_i. &(3.63)
\end{aligned}$$

The variation of the length differential dl is

$$\delta(\mathrm{d}l)=\delta(\sqrt{U}\mathrm{d}U^1)=\frac{\delta U}{2U}\sqrt{U}dU^1=t^i\nabla_s W_i\mathrm{d}l=(\boldsymbol{t}\cdot\nabla_s\boldsymbol{W})\mathrm{d}l. \tag{3.64}$$

The variations of surface geometries such as $\boldsymbol{S}_\alpha,\boldsymbol{N},K_M$ on the curve are still given by Eqs. (3.3-3.15) with the restriction $\nabla_\perp\boldsymbol{W}=0$, which implies

$$\begin{aligned}
\nabla_\alpha\boldsymbol{W} &= t_\alpha\nabla_s\boldsymbol{W} &(3.65)\\
\nabla_\alpha\nabla_\beta\boldsymbol{W} &= t_\alpha\nabla_s(t_\beta\nabla_s\boldsymbol{W})=t_\alpha t_\beta\nabla_s^2\boldsymbol{W}+t_\alpha\nabla_s t_\beta\nabla_s\boldsymbol{W}=t_\alpha t_\beta\nabla_s^2\boldsymbol{W}+t_\alpha n_\beta k_g(\nabla_s\boldsymbol{W}) &(3.66)\\
\nabla_\alpha\nabla^\alpha\boldsymbol{W} &= \nabla_s^2\boldsymbol{W} &(3.67)
\end{aligned}$$

Then from Eqs. (3.3-3.15) we have,

$$\begin{aligned}
\delta \boldsymbol{S}_\alpha &= (t_\alpha \nabla_s W + W^\beta B_{\alpha\beta})\boldsymbol{N} + (t_\alpha \nabla_s W^\beta \quad W B_\alpha^\beta)\boldsymbol{S}_\beta && (3.68)\\
\delta \boldsymbol{N} &= \quad (B_{\alpha\beta} W^\beta + t_\alpha \nabla_s W)\boldsymbol{S}^\alpha = \quad B_{\alpha\beta} W^\beta \boldsymbol{S}^\alpha \quad (\nabla_s W)\boldsymbol{t} && (3.69)\\
\delta B_{\alpha\beta} &= t_\alpha \nabla_s B_{\beta\nu} W^\nu + B_{\alpha\nu} t_\beta \nabla_s W^\nu + B_{\beta\nu} t_\alpha \nabla_s W^\nu \quad B_{\beta\gamma} B_\alpha^\gamma W + t_\alpha t_\beta \nabla_s^2 W + t_\alpha n_\beta k_g \nabla_s W && (3.70)\\
\delta K_M &= W(K_M^2 \quad 2K) + W^\nu \nabla_\nu K_M + \nabla_s^2 W && (3.71)
\end{aligned}$$

The variations of the unit tangent $\boldsymbol{t}$ and unit normal $\boldsymbol{n}$ are

$$\begin{aligned}
\delta \boldsymbol{t} &= \delta(\varepsilon^\Phi \boldsymbol{U}_\Phi) = \varepsilon^\Phi \delta \boldsymbol{U}_\Phi + \delta\varepsilon^\Phi \boldsymbol{U}_\Phi = \nabla_s W^i \boldsymbol{Z}_i \quad t_i \nabla_s W^i \boldsymbol{t} = (n_i \boldsymbol{n} + N_i \boldsymbol{N})\nabla_s W^i \\
&= (\mathbb{1} \quad \boldsymbol{t}\otimes\boldsymbol{t})\cdot\nabla_s \boldsymbol{W} = (\boldsymbol{n}\otimes\boldsymbol{n} + \boldsymbol{N}\otimes\boldsymbol{N})\cdot\nabla_s \boldsymbol{W} && (3.72)\\
\delta \boldsymbol{n} &= \quad (\delta\boldsymbol{t}\cdot\boldsymbol{n})\boldsymbol{t} \quad (\delta\boldsymbol{N}\cdot\boldsymbol{n})\boldsymbol{N} = \quad n_i \nabla_s W^i \boldsymbol{t} + (B_{\alpha\beta} W^\beta + t_\alpha \nabla_s W) n^\alpha \boldsymbol{N} \\
&= \quad n_i \nabla_s W^i \boldsymbol{t} + B_{\alpha\beta} W^\beta n^\alpha \boldsymbol{N} = \quad (\boldsymbol{n}\cdot\nabla_s \boldsymbol{W})\boldsymbol{t} + (n^\alpha W^\beta B_{\alpha\beta})\boldsymbol{N}. && (3.73)
\end{aligned}$$

To compute the variation of $k_g = \boldsymbol{n}\cdot\nabla_s \boldsymbol{t}, k_n = \boldsymbol{N}\cdot\nabla_s \boldsymbol{t}, \tau_g = \quad \boldsymbol{n}\cdot\nabla_s \boldsymbol{N}$, we first compute $\delta(\nabla_s \boldsymbol{t})$ and $\delta(\nabla_s \boldsymbol{N})$, which are straightforward to calculate. Indeed,

$$\begin{aligned}
\nabla_s \delta \boldsymbol{t} &= \nabla_s(\nabla_s W^i \boldsymbol{Z}_i \quad t^i \nabla_s W_i \boldsymbol{t}) = \nabla_s^2 W^i \boldsymbol{Z}_i \quad \nabla_s t^i \nabla_s W_i \boldsymbol{t} \quad t^i \nabla_s^2 W_i \boldsymbol{t} \quad t^i \nabla_s W_i \nabla_s \boldsymbol{t} && (3.74)\\
\delta(\nabla_s \boldsymbol{t}) &= \delta(\varepsilon^\Phi \nabla_\Phi \boldsymbol{t}) = \delta\varepsilon^\Phi \nabla_\Phi \boldsymbol{t} + \varepsilon^\Phi \nabla_\Phi \delta \boldsymbol{t} \\
&= \quad t^i \nabla_s W_i \nabla_s \boldsymbol{t} + \nabla_s^2 W^i \boldsymbol{Z}_i \quad \nabla_s t^i \nabla_s W_i \boldsymbol{t} \quad t^i \nabla_s^2 W_i \boldsymbol{t} \quad t^i \nabla_s W_i \nabla_s \boldsymbol{t} \\
&= \nabla_s^2 W^i \boldsymbol{Z}_i \quad \nabla_s t^i \nabla_s W_i \boldsymbol{t} \quad t^i \nabla_s^2 W_i \boldsymbol{t} \quad 2t^i \nabla_s W_i \nabla_s \boldsymbol{t} && (3.75)
\end{aligned}$$

and

$$\begin{aligned}
\nabla_s(\delta \boldsymbol{N}) &= \quad \nabla_s(B_{\alpha\beta} W^\beta + t_\alpha \nabla_s W)\boldsymbol{S}^\alpha \quad (B_{\alpha\beta} W^\beta + t_\alpha \nabla_s W) t^\gamma B_\gamma^\alpha \boldsymbol{N} \\
&= \quad \nabla_s(B_{\alpha\beta} W^\beta + t_\alpha \nabla_s W)\boldsymbol{S}^\alpha \quad (B_{\alpha\beta} W^\beta t^\gamma B_\gamma^\alpha + k_n \nabla_s W)\boldsymbol{N} \\
\delta \nabla_s \boldsymbol{N} &= \delta(\varepsilon^\Phi \nabla_\Phi \boldsymbol{N}) = \delta\varepsilon^\Phi \nabla_\Phi \boldsymbol{N} + \varepsilon^\Phi \delta \nabla_\Phi \boldsymbol{N} = \delta\varepsilon^\Phi \nabla_\Phi \boldsymbol{N} + \nabla_s \delta \boldsymbol{N} \\
&= \quad t^i \nabla_s W_i \nabla_s \boldsymbol{N} \quad \nabla_s(B_{\alpha\beta} W^\beta + t_\alpha \nabla_s W)\boldsymbol{S}^\alpha \quad (B_{\alpha\beta} W^\beta t^\gamma B_\gamma^\alpha + k_n \nabla_s W)\boldsymbol{N}. && (3.76)
\end{aligned}$$

The variation of the geodesic curvature k_g is

$$\begin{aligned}
\delta k_g &= \delta(\boldsymbol{n}\cdot\nabla_s \boldsymbol{t}) = \boldsymbol{n}\cdot\delta\nabla_s \boldsymbol{t} + \nabla_s \boldsymbol{t}\cdot\delta\boldsymbol{n} = \boldsymbol{n}\cdot\delta\nabla_s \boldsymbol{t} + (k_g \boldsymbol{n} + k_n \boldsymbol{N})\cdot\delta\boldsymbol{n} \\
&= \boldsymbol{n}\cdot\delta\nabla_s \boldsymbol{t} + k_n \boldsymbol{N}\cdot\delta\boldsymbol{n} = n_i \nabla_s^2 W^i \quad 2k_g t^i \nabla_s W_i + k_n n^\alpha W^\beta B_{\alpha\beta} && (3.77)
\end{aligned}$$

The variation of the normal curvature k_n is

$$\begin{aligned}
\delta k_n &= \delta(\boldsymbol{N}\cdot\nabla_s \boldsymbol{t}) = \delta\boldsymbol{N}\cdot\nabla_s \boldsymbol{t} + \boldsymbol{N}\cdot\delta(\nabla_s \boldsymbol{t}) = \delta\boldsymbol{N}\cdot(k_g \boldsymbol{n} + k_n \boldsymbol{N}) + \boldsymbol{N}\cdot\delta(\nabla_s \boldsymbol{t}) \\
&= k_g \boldsymbol{n}\cdot\delta\boldsymbol{N} + \boldsymbol{N}\cdot\delta(\nabla_s \boldsymbol{t}) = \quad k_g B_{\alpha\beta} W^\beta n^\alpha + N_i \nabla_s^2 W^i \quad 2k_n t^i \nabla_s W_i && (3.78)
\end{aligned}$$

The variation of $B_\perp$ is

$$\begin{aligned}
\delta B_\perp &= \delta K_M \quad \delta k_n \\
&= W(K_M^2 \quad 2K) + W^\nu \nabla_\nu K_M + \nabla_s^2 W \quad (\quad k_g B_{\alpha\beta} W^\beta n^\alpha + N_i \nabla_s^2 W^i \quad 2k_n t^i \nabla_s W_i) \\
&= W(K_M^2 \quad 2K) + W^\nu \nabla_\nu K_M + \nabla_s^2 W + k_g B_{\alpha\beta} W^\beta n^\alpha \quad N_i \nabla_s^2 W^i + 2k_n t^i \nabla_s W_i \\
&= W(K_M^2 \quad 2K) + W^\beta(\nabla_\beta K_M + k_g B_{\alpha\beta} n^\alpha) + \nabla_s^2 W \quad N_i \nabla_s^2 W^i + 2k_n t^i \nabla_s W_i \\
&= W(K_M^2 \quad 2K) + W^\beta(\nabla_\beta K_M + k_g B_{\alpha\beta} n^\alpha) + 2\nabla_s \boldsymbol{N}\cdot\nabla_s \boldsymbol{W} + \boldsymbol{W}\cdot\nabla_s^2 \boldsymbol{N} + 2k_n t^i \nabla_s W_i \\
&= W(K_M^2 \quad 2K) + W^\beta(\nabla_\beta K_M + k_g B_{\alpha\beta} n^\alpha) + 2(\quad k_n \boldsymbol{t} \quad \tau_g \boldsymbol{n})\cdot\nabla_s \boldsymbol{W} + \boldsymbol{W}\cdot\nabla_s^2 \boldsymbol{N} + 2k_n t^i \nabla_s W_i \\
&= W(K_M^2 \quad 2K) + W^\beta(\nabla_\beta K_M + k_g B_{\alpha\beta} n^\alpha) \quad 2\tau_g \boldsymbol{n}\cdot\nabla_s \boldsymbol{W} + \boldsymbol{W}\cdot\nabla_s^2 \boldsymbol{N} && (3.79)
\end{aligned}$$

where we used

$$\nabla_s^2 W = \nabla_s(\nabla_s(\boldsymbol{W}\cdot\boldsymbol{N})) = 2\nabla_s \boldsymbol{N}\cdot\nabla_s \boldsymbol{W} + \boldsymbol{N}\cdot\nabla_s^2 \boldsymbol{W} + \boldsymbol{W}\cdot\nabla_s^2 \boldsymbol{N}. \qquad (3.80)$$

The variation of τ_g is

$$\begin{aligned}
\delta \tau_g &= \quad \delta(\boldsymbol{n}\cdot\nabla_s \boldsymbol{N}) = \quad \delta\boldsymbol{n}\cdot\nabla_s \boldsymbol{N} \quad \boldsymbol{n}\cdot\delta(\nabla_s \boldsymbol{N}) = \quad \delta\boldsymbol{n}\cdot(\quad k_n \boldsymbol{t} \quad \tau_g \boldsymbol{n}) \quad \boldsymbol{n}\cdot\delta(\nabla_s \boldsymbol{N}) \\
&= k_n \boldsymbol{t}\cdot\delta\boldsymbol{n} \quad \boldsymbol{n}\cdot\delta(\nabla_s \boldsymbol{N}) = \quad k_n n_i \nabla_s W^i \quad \tau_g t^i \nabla_s W_i + \nabla_s(B_{\alpha\beta} W^\beta + t_\alpha \nabla_s W) n^\alpha. && (3.81)
\end{aligned}$$

3.2.2 Variation of Curve Hamiltonian

Example 3.17. (variation of a curve functional) Consider an energy functional defined as $\oint_{\partial\mathcal{P}} f \mathrm{d}l$. Then,

$$\delta\oint_{\partial\mathcal{P}} f\mathrm{d}l \;=\; \oint_{\partial\mathcal{P}}\left(\delta f + f\frac{\delta(\mathrm{d}l)}{\mathrm{d}l}\right)\mathrm{d}l = \oint_{\partial\mathcal{P}}(\delta f + f t^i\nabla_s W_i)\mathrm{d}l = \oint_{\partial\mathcal{P}}\{\delta f \quad \boldsymbol{W}\cdot\nabla_s(f\boldsymbol{t})\}\mathrm{d}l \tag{3.82}$$

and

$$\frac{\delta\oint_{\partial\mathcal{P}} f\mathrm{d}l}{\delta\boldsymbol{R}} = \oint_{\partial\mathcal{P}}\left\{\quad\frac{\delta f}{\delta\boldsymbol{R}} + \nabla_s(f\boldsymbol{t})\right\}\mathrm{d}l. \tag{3.83}$$

Example 3.18. (variation of curve length) The length of the boundary curve $\partial\mathcal{P}$ is $l = \oint_{\partial\mathcal{P}}\mathrm{d}l$. Then setting $f = 1$ in Eq. (3.83) gives

$$\frac{\delta l}{\delta\boldsymbol{R}} = \oint_{\partial\mathcal{P}}(\nabla_s\boldsymbol{t})\mathrm{d}l = \oint_{\partial\mathcal{P}}(k_g\boldsymbol{n} + k_n\boldsymbol{N})\mathrm{d}l. \tag{3.84}$$

From Eq. (3.84), the line energy σl will give rise to line tension $\sigma(k_g\boldsymbol{n} + k_n\boldsymbol{N})$ along the boundary curve.

Example 3.19. (variation of the geodesic curvature energy) Consider $f = k_g,$. Then,

$$\begin{aligned}
\delta\oint k_g\mathrm{d}l \;&=\; \oint\left(\delta k_g + k_g\frac{\delta(\mathrm{d}l)}{\mathrm{d}l}\right)\mathrm{d}l \\
&=\; \oint(n_i\nabla_s^2 W^i \quad 2k_g t^i\nabla_s W_i + k_n n^\alpha W^\beta B_{\alpha\beta} + k_g t^i\nabla_s W_i)\mathrm{d}l \\
&=\; \oint(n_i\nabla_s^2 W^i \quad k_g t^i\nabla_s W_i + k_n n^\alpha W^\beta B_{\alpha\beta})\mathrm{d}l \\
&=\; \oint(\nabla_s^2 n_i W^i + \nabla_s(k_g t^i)W_i + k_n n^\alpha B_{\alpha\beta}S^\beta)\mathrm{d}l
\end{aligned} \tag{3.85}$$

and

$$\frac{\delta\oint k_g\mathrm{d}l}{\delta\boldsymbol{R}} = \quad \oint(\nabla_s^2\boldsymbol{n} + \nabla_s(k_g\boldsymbol{t}) + k_n n^\alpha B_{\alpha\beta}\boldsymbol{S}^\beta)\mathrm{d}l = \quad \oint(\nabla_s\tau_g\boldsymbol{N} + K\boldsymbol{n})\mathrm{d}l \tag{3.86}$$

where we used

$$\begin{aligned}
&\nabla_s^2\boldsymbol{n} + \nabla_s(k_g\boldsymbol{t}) + k_n n^\alpha B_{\alpha\beta}\boldsymbol{S}^\beta = \nabla_s(\tau_g\boldsymbol{N}) + k_n n^\alpha B_{\alpha\beta}(t^\beta\boldsymbol{t} + n^\beta\boldsymbol{n}) \\
=\;& \nabla_s\tau_g\boldsymbol{N} + \tau_g(\quad k_n\boldsymbol{t} \quad \tau_g\boldsymbol{n}) + k_n(\tau_g\boldsymbol{t} + B_\perp\boldsymbol{n}) = \nabla_s\tau_g\boldsymbol{N} + (k_n B_\perp \quad \tau_g^2)\boldsymbol{n} = \nabla_s\tau_g\boldsymbol{N} + K\boldsymbol{n}
\end{aligned}$$

As expected from the Gauss-Bonnet theorem $\int_{\mathcal{P}} K\mathrm{d}A \quad \oint_{\partial\mathcal{P}} k_g\mathrm{d}l = 2\pi\chi(\mathcal{P})$, where $\chi(\mathcal{P})$ is the Euler characteristic of $\mathcal{P}$, the right hand sides of Eq. (3.50) and Eq. (3.86) are the same. Eq. (3.86) agrees with the result from Capovilla [capovilla2002lipid] and Tu [tu2003lipid].

Example 3.20. (variation of the geodesic curvature energy) Consider $f = \frac{\kappa}{n}k_g^n$, where n is a positive integer and $\kappa(U)$ can vary along the curve. Then,

$$\begin{aligned}
&\delta\oint\frac{\kappa}{n}k_g^n\mathrm{d}l \\
=\;& \oint\left(\kappa k_g^{n\quad 1}\delta k_g + \frac{\kappa}{n}k_g^n\frac{\delta(\mathrm{d}l)}{\mathrm{d}l}\right)\mathrm{d}l \\
=\;& \oint\left\{\kappa k_g^{n\quad 1}(n_i\nabla_s^2 W^i \quad 2k_g t^i\nabla_s W_i + k_n n^\alpha W^\beta B_{\alpha\beta}) + \frac{\kappa}{n}k_g^n t^i\nabla_s W_i\right\}\mathrm{d}l \\
=\;& \oint\left\{\kappa k_g^{n\quad 1}n_i\nabla_s^2 W^i \quad \frac{2n\quad 1}{n}\kappa k_g^n t^i\nabla_s W_i + \kappa k_g^{n\quad 1}k_n n^\alpha W^\beta B_{\alpha\beta}\right\}\mathrm{d}l \\
=\;& \oint\left\{\nabla_s^2(\kappa k_g^{n\quad 1}n_i)W^i + \nabla_s\left(\frac{2n\quad 1}{n}\kappa k_g^n t^i\right)W_i + \kappa k_g^{n\quad 1}k_n n^\alpha W^\beta B_{\alpha\beta}\right\}\mathrm{d}l
\end{aligned}$$

Thus,

$$\frac{\delta\oint_{\partial\mathcal{P}}\frac{\kappa}{n}k_g^n\mathrm{d}l}{\delta\boldsymbol{R}}=\oint_{\partial\mathcal{P}}\Big\{\quad\nabla_s^2(\kappa k_g^{n\quad 1}\boldsymbol{n})\quad\nabla_s\Big(\frac{2n\quad 1}{n}\kappa k_g^n\boldsymbol{t}\Big)\quad\kappa k_g^{n\quad 1}k_n n^\alpha\boldsymbol{S}^\beta B_{\alpha\beta}\Big\}\mathrm{d}l \tag{3.87}$$

Example 3.21. (variation of the normal curvature energy) Consider $f=k_n$, then,

$$\begin{aligned}\delta\oint k_n\mathrm{d}l &= \oint\Big(\delta k_n+k_n\frac{\delta(\mathrm{d}l)}{\mathrm{d}l}\Big)\mathrm{d}l\\ &= \oint(\quad k_gB_{\alpha\beta}W^\beta n^\alpha+N_i\nabla_s^2W^i\quad 2k_nt^i\nabla_sW_i+k_nt^i\nabla_sW_i)\mathrm{d}l\\ &= \oint(\quad k_gB_{\alpha\beta}W^\beta n^\alpha+N_i\nabla_s^2W^i\quad k_nt^i\nabla_sW_i)\mathrm{d}l\\ &= \oint(\quad k_gB_{\alpha\beta}W^\beta n^\alpha+\nabla_s^2N_iW^i+\nabla_s(k_nt^i)W_i)\mathrm{d}l\end{aligned} \tag{3.88}$$

and

$$\frac{\delta\oint_{\partial\mathcal{P}}k_n\mathrm{d}l}{\delta\boldsymbol{R}}=\quad\oint(\quad k_gB_{\alpha\beta}\boldsymbol{S}^\beta n^\alpha+\nabla_s^2\boldsymbol{N}+\nabla_s(k_n\boldsymbol{t}))\mathrm{d}l=\oint((k_gB_\perp+\nabla_s\tau_g)\boldsymbol{n}+\tau_g^2\boldsymbol{N})\mathrm{d}l \tag{3.89}$$

where we used

$$\begin{aligned}&\quad k_gB_{\alpha\beta}n^\alpha\boldsymbol{S}^\beta+\nabla_s^2\boldsymbol{N}+\nabla_s(k_n\boldsymbol{t})=\quad k_gB_{\alpha\beta}n^\alpha\boldsymbol{S}^\beta+\nabla_s(\quad\tau_g\boldsymbol{n})\\ =&\quad k_gB_{\alpha\beta}n^\alpha(t^\beta\boldsymbol{t}+n^\beta\boldsymbol{n})\quad\nabla_s\tau_g\boldsymbol{n}\quad\tau_g(\quad k_g\boldsymbol{t}+\tau_g\boldsymbol{N})\\ =&\quad k_g(\tau_g\boldsymbol{t}+B_\perp\boldsymbol{n})\quad\nabla_s\tau_g\boldsymbol{n}\quad\tau_g(\quad k_g\boldsymbol{t}+\tau_g\boldsymbol{N})=\quad(k_gB_\perp+\nabla_s\tau_g)\boldsymbol{n}\quad\tau_g^2\boldsymbol{N}.\end{aligned}$$

Example 3.22. (variation of the normal curvature energy) Consider $f=\frac{\kappa}{n}(k_n)^n$, where n is a positive integer and $\kappa(U)$ can vary along the curve. Then,

$$\begin{aligned}&\delta\oint\frac{\kappa}{n}(k_n)^n\mathrm{d}l\\ =&\oint\Big\{\kappa(k_n)^{n\quad 1}\delta k_n+\frac{\kappa}{n}(k_n)^n\frac{\delta(\mathrm{d}l)}{\mathrm{d}l}\Big\}\mathrm{d}l\\ =&\oint\Big\{\kappa(k_n)^{n\quad 1}(\quad\kappa_gB_{\alpha\beta}W^\beta n^\alpha+N_i\nabla_s^2W^i\quad 2k_nt^i\nabla_sW_i)+\frac{\kappa}{n}(k_n)^nt^i\nabla_sW_i\Big\}\mathrm{d}l\\ =&\oint\Big\{\quad\kappa(k_n)^{n\quad 1}\kappa_gB_{\alpha\beta}W^\beta n^\alpha+\kappa(k_n)^{n\quad 1}N_i\nabla_s^2W^i\quad 2\kappa(k_n)^{n\quad 1}k_nt^i\nabla_sW_i+\frac{\kappa}{n}(k_n)^nt^i\nabla_sW_i\Big\}\mathrm{d}l\\ =&\oint\Big\{\quad\kappa(k_n)^{n\quad 1}\kappa_gB_{\alpha\beta}W^\beta n^\alpha+\kappa(k_n)^{n\quad 1}N_i\nabla_s^2W^i\quad\frac{2n\quad 1}{n}\kappa(k_n)^nt^i\nabla_sW_i\Big\}\mathrm{d}l\\ =&\oint\Big\{\quad\kappa(k_n)^{n\quad 1}\kappa_gB_{\alpha\beta}W^\beta n^\alpha+\nabla_s^2(\kappa(k_n)^{n\quad 1}N_i)W^i+\nabla_s\Big(\frac{2n\quad 1}{n}\kappa(k_n)^nt^i\Big)W_i\Big\}\mathrm{d}l\end{aligned}$$

and

$$\frac{\delta\oint_{\partial\mathcal{P}}\frac{\kappa}{n}(k_n)^n\mathrm{d}l}{\delta\boldsymbol{R}}=\oint_{\partial\mathcal{P}}\Big\{\kappa(k_n)^{n\quad 1}k_gB_{\alpha\beta}\boldsymbol{S}^\beta n^\alpha\quad\nabla_s^2(\kappa(k_n)^{n\quad 1}\boldsymbol{N})\quad\nabla_s\Big(\frac{2n\quad 1}{n}\kappa(k_n)^n\boldsymbol{t}\Big)\Big\}\mathrm{d}l. \tag{3.90}$$

Example 3.23. Let us calculate the variation of $\oint K_M\mathrm{d}l$.

$$\begin{aligned}\delta\oint K_M\mathrm{d}l &= \oint\Big(\delta K_M+K_M\frac{\delta(\mathrm{d}l)}{\mathrm{d}l}\Big)\mathrm{d}l\\ &= \oint(W(K_M^2\quad 2K)+W^\nu\nabla_\nu K_M+\nabla_s^2W+K_M(t^i\nabla_sW_i))\mathrm{d}l\\ &= \oint(W(K_M^2\quad 2K)+W^\nu\nabla_\nu K_M\quad W_i\nabla_s(t^iK_M))\mathrm{d}l\end{aligned}$$

and

$$\begin{aligned}\frac{\delta\oint K_M \mathrm{d}l}{\delta \boldsymbol{R}} &= \oint((K_M^2 \quad 2K)\boldsymbol{N}+\nabla_{\|}K_M \quad \nabla_s(\boldsymbol{t}K_M))\mathrm{d}l\\ &= \oint((2K \quad K_M B_\perp)\boldsymbol{N}+(K_M k_g \quad \nabla_\perp K_M)\boldsymbol{n})\mathrm{d}l \qquad (3.91)\end{aligned}$$

where we used

$$\begin{aligned}&(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_{\|}K_M \quad \nabla_s(\boldsymbol{t}K_M)\\ =\ &(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_{\|}K_M \quad \nabla_s K_M\boldsymbol{t} \quad K_M\nabla_s\boldsymbol{t}\\ =\ &(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_\perp K_M\boldsymbol{n} \quad K_M(k_g\boldsymbol{n}+k_n\boldsymbol{N})\\ =\ &(K_M^2 \quad 2K \quad K_M k_n)\boldsymbol{N}+(\nabla_\perp K_M \quad K_M k_g)\boldsymbol{n}\\ =\ &(K_M B_\perp \quad 2K)\boldsymbol{N}+(\nabla_\perp K_M \quad K_M k_g)\boldsymbol{n}\end{aligned}$$

Example 3.24. (variation of $\boldsymbol{B_\perp}$) Consider $f=B_\perp$, then

$$\begin{aligned}&\delta\oint B_\perp \mathrm{d}l\\ =\ &\oint\left(\delta B_\perp + B_\perp\frac{\delta(\mathrm{d}l)}{\delta l}\right)\mathrm{d}l\\ =\ &\oint(W(K_M^2 \quad 2K)+W^\beta(\nabla_\beta K_M+k_g B_{\alpha\beta}n^\alpha) \quad 2\tau_g\boldsymbol{n}\cdot\nabla_s\boldsymbol{W}+\boldsymbol{W}\cdot\nabla_s^2\boldsymbol{N}+B_\perp(t^i\nabla_s W_i))\mathrm{d}l\\ =\ &\oint(W(K_M^2 \quad 2K)+W^\beta(\nabla_\beta K_M+k_g B_{\alpha\beta}n^\alpha)+\boldsymbol{W}\cdot\nabla_s(2\tau_g\boldsymbol{n})+\boldsymbol{W}\cdot\nabla_s^2\boldsymbol{N} \quad W_i\nabla_s(B_\perp t^i))\mathrm{d}l\end{aligned}$$

and

$$\begin{aligned}&\frac{\delta\oint B_\perp \mathrm{d}l}{\delta\boldsymbol{R}}\\ =\ &\oint((K_M^2 \quad 2K)\boldsymbol{N}+\boldsymbol{S}^\beta(\nabla_\beta K_M+k_g B_{\alpha\beta}n^\alpha)+\nabla_s(2\tau_g\boldsymbol{n})+\nabla_s^2\boldsymbol{N} \quad \nabla_s(B_\perp\boldsymbol{t}))\mathrm{d}l\\ =\ &\oint((2K \quad K_M B_\perp \quad \tau_g^2)\boldsymbol{N}+(k_g k_n \quad \nabla_\perp K_M \quad \nabla_s\tau_g)\boldsymbol{n})\,\mathrm{d}l \qquad (3.92)\end{aligned}$$

where we used

$$\begin{aligned}&(K_M^2 \quad 2K)\boldsymbol{N}+\boldsymbol{S}^\beta(\nabla_\beta K_M+k_g B_{\alpha\beta}n^\alpha)+\nabla_s(2\tau_g\boldsymbol{n})+\nabla_s^2\boldsymbol{N} \quad \nabla_s(B_\perp\boldsymbol{t})\\ =\ &(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_{\|}K_M+k_g B_{\alpha\beta}n^\alpha\boldsymbol{S}^\beta+\nabla_s(2\tau_g\boldsymbol{n}+\nabla_s\boldsymbol{N} \quad B_\perp\boldsymbol{t})\\ =\ &(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_{\|}K_M+k_g B_{\alpha\beta}n^\alpha(t^\beta\boldsymbol{t}+n^\beta\boldsymbol{n})+\nabla_s(\tau_g\boldsymbol{n} \quad k_n\boldsymbol{t} \quad B_\perp\boldsymbol{t})\\ =\ &(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_{\|}K_M+k_g(\tau_g\boldsymbol{t}+B_\perp\boldsymbol{n})+\nabla_s(\tau_g\boldsymbol{n} \quad K_M\boldsymbol{t})\\ =\ &(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_{\|}K_M+k_g(\tau_g\boldsymbol{t}+B_\perp\boldsymbol{n})+\nabla_s\tau_g\boldsymbol{n}+\tau_g\nabla_s\boldsymbol{n} \quad \nabla_s K_M\boldsymbol{t} \quad K_M\nabla_s\boldsymbol{t}\\ =\ &(K_M^2 \quad 2K)\boldsymbol{N}+\nabla_\perp K_M\boldsymbol{n}+k_g(\tau_g\boldsymbol{t}+B_\perp\boldsymbol{n})+\nabla_s\tau_g\boldsymbol{n}+\tau_g(\quad k_g\boldsymbol{t}+\tau_g\boldsymbol{N}) \quad K_M(k_g\boldsymbol{n}+k_n\boldsymbol{N})\\ =\ &(K_M^2 \quad 2K+\tau_g^2 \quad K_M k_n)\boldsymbol{N}+(\nabla_\perp K_M+k_g B_\perp+\nabla_s\tau_g \quad K_M k_g)\boldsymbol{n}\\ =\ &(K_M B_\perp \quad 2K+\tau_g^2)\boldsymbol{N}+(\nabla_\perp K_M \quad k_g k_n+\nabla_s\tau_g)\boldsymbol{n}.\end{aligned}$$

We can also use $B_\perp = K_M \quad k_n$ to verify the above result

$$\begin{aligned}\frac{\delta\oint B_\perp \mathrm{d}l}{\delta\boldsymbol{R}} &= \frac{\delta\oint(K_M \quad k_n)\mathrm{d}l}{\delta\boldsymbol{R}}\\ &= \oint((2K \quad K_M B_\perp)\boldsymbol{N}+(K_M k_g \quad \nabla_\perp K_M)\boldsymbol{n})\mathrm{d}l \quad \oint((k_g B_\perp+\nabla_s\tau_g)\boldsymbol{n}+\tau_g^2\boldsymbol{N})\mathrm{d}l\\ &= \oint((2K \quad K_M B_\perp \quad \tau_g^2)\boldsymbol{N}+(K_M k_g \quad \nabla_\perp K_M \quad k_g B_\perp \quad \nabla_s\tau_g)\boldsymbol{n})\mathrm{d}l\\ &= \oint((2K \quad K_M B_\perp \quad \tau_g^2)\boldsymbol{N}+(k_n k_g \quad \nabla_\perp K_M \quad \nabla_s\tau_g)\boldsymbol{n})\mathrm{d}l.\end{aligned}$$

Example 3.25. (variation of the geodesic torsion energy) Consider $f = \tau_g$, then

$$
\begin{aligned}
\delta\oint \tau_g \mathrm{d}l &= \oint\left(\delta\tau_g + \tau_g\frac{\delta(dl)}{dl}\right)\mathrm{d}l \\
&= \oint(\quad k_n n_i \nabla_s W^i \quad \tau_g t^i \nabla_s W_i + \nabla_s(B_{\alpha\beta}W^\beta + t_\alpha \nabla_s W)n^\alpha + \tau_g t^i \nabla_s W_i)\mathrm{d}l \\
&= \oint(\quad k_n n_i \nabla_s W^i + \nabla_s(B_{\alpha\beta}W^\beta + t_\alpha \nabla_s W)n^\alpha)\mathrm{d}l \\
&= \oint(\nabla_s(k_n n_i)W^i \quad (B_{\alpha\beta}W^\beta + t_\alpha \nabla_s W)\nabla_s n^\alpha)\mathrm{d}l \\
&= \oint(\nabla_s(k_n n_i)W^i + k_g(B_{\alpha\beta}W^\beta + t_\alpha \nabla_s W)t^\alpha)\mathrm{d}l \\
&= \oint(\nabla_s(k_n n_i)W^i + k_g t^\alpha B_{\alpha\beta}W^\beta + k_g \nabla_s W)\mathrm{d}l \\
&= \oint(\nabla_s(k_n n_i)W^i + k_g t^\alpha B_{\alpha\beta}W^\beta \quad W\nabla_s k_g)\mathrm{d}l \\
&= \oint[\{\nabla_s(k_n n_i) + k_g t^\alpha B_{\alpha\beta}Z^\beta_i\}W^i \quad W\nabla_s k_g]\mathrm{d}l
\end{aligned}
$$

and

$$
\begin{aligned}
\frac{\delta\oint \frac{\kappa}{n}\tau_g^n \mathrm{d}l}{\delta\boldsymbol{R}} &= \oint[\{\nabla_s(k_n\boldsymbol{n}) + k_g t^\alpha B_{\alpha\beta}\boldsymbol{S}^\beta\} \quad \boldsymbol{N}\nabla_s k_g]\mathrm{d}l \\
&= \oint\{\quad (\nabla_s k_n + k_g\tau_g)\boldsymbol{n} + (\nabla_s k_g \quad k_n\tau_g)\boldsymbol{N}\}\mathrm{d}l \qquad (3.93)
\end{aligned}
$$

where we used

$$
\begin{aligned}
& \nabla_s(k_n\boldsymbol{n}) + k_g t^\alpha B_{\alpha\beta}\boldsymbol{S}^\beta \quad \nabla_s k_g \boldsymbol{N} \\
= \ & \nabla_s k_n \boldsymbol{n} + k_n(\quad k_g\boldsymbol{t} + \tau_g\boldsymbol{N}) + k_g(k_n\boldsymbol{t} + \tau_g\boldsymbol{n}) \quad \nabla_s k_g\boldsymbol{N} \\
= \ & \nabla_s k_n\boldsymbol{n} + k_n\tau_g\boldsymbol{N} \quad \nabla_s k_g\boldsymbol{N} + k_g\tau_g\boldsymbol{n} \\
= \ & (\nabla_s k_n + k_g\tau_g)\boldsymbol{n} + (k_n\tau_g \quad \nabla_s k_g)\boldsymbol{N}
\end{aligned}
$$

Example 3.26. (variation of the geodesic torsion energy) Consider $f = \frac{\kappa}{n}(\tau_g)^n$, where n is a positive integer and $\kappa(U)$ can vary along the curve. Then,

$$
\begin{aligned}
& \oint \frac{\kappa}{n}\tau_g^n \mathrm{d}l \\
= \ & \oint\left\{\kappa\tau_g^{n \quad 1}\delta\tau_g + \frac{\kappa}{n}\tau_g^n\frac{\delta(\mathrm{d}l)}{\mathrm{d}l}\right\}\mathrm{d}l \\
= \ & \oint\left\{\kappa\tau_g^{n \quad 1}[\quad k_n n_i\nabla_s W^i \quad \tau_g t^i\nabla_s W_i + \nabla_s(B_{\alpha\beta}W^\beta + t_\alpha\nabla_s W)n^\alpha] + \frac{\kappa}{n}\tau_g^n t^i \nabla_s W_i\right\}\mathrm{d}l \\
= \ & \oint\left\{\quad \kappa\tau_g^{n \quad 1}k_n n_i \nabla_s W^i \quad \frac{n \quad 1}{n}\kappa\tau_g^n t^i\nabla_s W_i + \kappa\tau_g^{n \quad 1}\nabla_s(B_{\alpha\beta}W^\beta + t_\alpha\nabla_s W)n^\alpha\right\}\mathrm{d}l \\
= \ & \oint\left\{W^i\nabla_s\left(\kappa\tau_g^{n \quad 1}k_n n_i + \frac{n \quad 1}{n}\kappa\tau_g^n t^i\right) \quad (B_{\alpha\beta}W^\beta + t_\alpha\nabla_s W)\nabla_s(\kappa\tau_g^{n \quad 1}n_\alpha)\right\}\mathrm{d}l \\
= \ & \oint\left\{W^i\nabla_s\left(\kappa\tau_g^{n \quad 1}k_n n_i + \frac{n \quad 1}{n}\kappa\tau_g^n t^i\right) \quad B_{\alpha\beta}W^\beta\nabla_s(\kappa\tau_g^{n \quad 1}n_\alpha) \quad t_\alpha\nabla_s W\nabla_s(\kappa\tau_g^{n \quad 1}n_\alpha)\right\}\mathrm{d}l \\
= \ & \oint\left\{W^i\nabla_s\left(\kappa\tau_g^{n \quad 1}k_n n_i + \frac{n \quad 1}{n}\kappa\tau_g^n t^i\right) \quad B_{\alpha\beta}W^\beta\nabla_s(\kappa\tau_g^{n \quad 1}n_\alpha) + W\nabla_s[t_\alpha\nabla_s(\kappa\tau_g^{n \quad 1}n_\alpha)]\right\}\mathrm{d}l
\end{aligned}
$$

and

$$
\begin{aligned}
& \frac{\delta\oint_{\partial\mathcal{P}}\frac{\kappa}{n}\tau_g^n \mathrm{d}l}{\delta\boldsymbol{R}} = \\
& \oint_{\partial\mathcal{P}}\left\{\quad \nabla_s\left[\kappa\tau_g^{n \quad 1}\left(k_n\boldsymbol{n} + \frac{n \quad 1}{n}\tau_g\boldsymbol{t}\right)\right] + B_{\alpha\beta}\boldsymbol{S}^\beta\nabla_s(\kappa\tau_g^{n \quad 1}n_\alpha) \quad \boldsymbol{N}\nabla_s[t_\alpha\nabla_s(\kappa\tau_g^{n \quad 1}n_\alpha)]\right\}\mathrm{d}l \qquad (3.94)
\end{aligned}
$$

With Eqs. (3.84-3.94), all the variations of the line energy Eq. (3.2) are obtained. We can therefore calculate forces from a Hamiltonian consisting of line curvature scalars. While we will not explore the numerical impacts of all terms in the Hamiltonian Eq. (3.1) and Eq. (3.2), we have provided useful theoretical calculations for a model beyond the canonical curvature squared Helfrich theory. These calculations are decoupled from the numerical schemes presented in the rest of the thesis and can be used in any kinds of applications that adopts a more general description of elastic bilayers.

Chapter 4

The Level Set Framework and High Order Schemes for Reinitialization and Extrapolation

A wide range of natural phenomena involves the motion of dynamic interfaces. When an elastic ball hits a rigid wall, its shape deforms. A fish undulates in order to swim. The surface of a soap bubble quivers from the air currents that keep it afloat. Beams bend, flags flutter, and seas swell. Even the membranes of the cells that comprise our bodies ruffle, protrude, invaginate, and pinch off. In some cases, the motions of these interfaces involve the dynamics of surface bound components, such as the proteins and lipids that diffuse and react on cell membranes. The **level set method**, introduced by Osher and Sethian [osher1988fronts], is an extremely simple and elegant numerical framework for these kinds of problems [salac2011level, laadhari2017fully].

4.1 The Level Set Method

4.1.1 The Level Set Equation, Reinitialization Equation and Extrapolation Equation

In practice, a moving interface is usually represented implicitly as the zero level set $\ (t)$ of a signed distance function $\phi(\boldsymbol{x},t)$ in the embedding space $\mathbb{R}^n$ ($n=2$ for a curve and $n=3$ for a surface), i.e., $\ (t)=\{\boldsymbol{x}|\,\phi(\boldsymbol{x},t)=0,\boldsymbol{x}\in\mathbb{R}^n\}$. The dynamics of $\ (t)$ under a velocity field $\boldsymbol{V}$ is then captured by the **level set equation** [osher1988fronts]:

$$\frac{\partial\phi}{\partial t}+\boldsymbol{V}\cdot\nabla\phi=0. \tag{4.1}$$

Due to the embedding, requirement on computational storage and intensity is an order of magnitude higher than the usual Lagrangian method (N^2 compared to N for a one dimensional surface and N^3 compared to N^2 for a two dimensional surface). To ameliorate this problem, Adalesteinsson and Sethian [adalsteinsson1995fast] introduced a variant called **narrow banded level set method** where only grids near the interface were updated. Even higher efficiency can be achieved by the use of QuadTree or OctTree data structures [strain1999fast].

Under a general velocity field, $\phi(\boldsymbol{x},t)$ will not remain a signed distance function. For the sake of numerical accuracy and stability, $\phi(\boldsymbol{x},t)$ needs to to restored to a signed distance function from time to time. This is called **reinitialization**, an idea introduced by Chopp [chopp1991computing] to remedy numerical stability and accuracy issues with the level set method. The issue of reinitialization has been extensively investigated since the birth of level set method. One way to do this is to solve the **reinitialization equation** by Sussman [sussman1994level]

$$\frac{\partial\phi}{\partial\tau}+\mathrm{sign}(\phi_0)(|\nabla\phi|\quad 1)=0 \tag{4.2}$$

in a pseudo time domain. Eq. (4.2) restores ϕ to a signed distance function when iterated to equilibrium. Improved numerical schemes by Russo [russo2000remark] and Min [min2007second, min2010reinitializing] solves the problem of motion of interfaces and loss of volume during reinitialization. Another approach is offered by the **fast marching method** [sethian1996fast] to solve the level set equation for monotonically advancing fronts, i.e. fronts with positive normal speed. The fast marching method solves the **Eikonal equation** $|\nabla\phi|=1$ in $N\log N$ time where N is the number of grid points of the domain of ϕ. An recent implementation of this method can be found in [chopp2001some, chopp2009another].

A quite different approach towards the problem of reinitialization is to extend velocities off the front such that it remains constant along the normal direction [zhao1996variational, adalsteinsson1999fast, peng1999pde]. This idea of extending fields off the interface is critical to solve PDEs on a moving surface represented implicitly [xu2003eulerian, adalsteinsson2003transport, bertalmio2001variational]. The so-called **closed point method** has a similar spirit and a slightly different implementation [ruuth2008simple, macdonald2009implicit, macdonald2008level]. We shall adopt this approach to embed surface PDEs in space. To extrapolate surface fields away from the surface, a hyperbolic PDE advecting c in the normal direction is solved [peng1999pde]:

$$\frac{\partial c}{\partial \tau} + \mathrm{Sign}(\phi)\boldsymbol{N}\cdot\nabla c = 0 \tag{4.3}$$

where $\boldsymbol{N}\equiv\nabla\phi/|\nabla\phi|$ is the normal of the interface and c represents the scalar field that is being extended away from the surface.

4.1.2 The Level Set Method for Implicit Curves

Dynamics of codimensional two objects in three dimensional Euclidean space, i.e. space curves or curves embedded in a surface, can be represented with systems of level set equations [burchard2001motion]. We follow the methodology developed in [burchard2001motion, cheng2002motion] to evolve a curve embedded in the surface which in our model will represent phase boundaries.

Under a given velocity field $\boldsymbol{V}$, the dynamics of the curve is governed by two level set equations evolving both ϕ and ψ

$$\begin{aligned} \frac{\partial\phi}{\partial t} + \boldsymbol{V}\cdot\nabla\phi &= 0 \qquad (4.4)\\ \frac{\partial\psi}{\partial t} + \boldsymbol{V}\cdot\nabla\psi &= 0. \qquad (4.5)\end{aligned}$$

For the sake of numerical accuracy and stability, ψ needs also to be reinitialized from time to time [burchard2001motion, cheng2002motion] by first solving

$$\frac{\partial\psi}{\partial\tau} + \mathrm{sign}(\psi)(|\nabla_{||}\psi| \quad 1) = 0 \tag{4.6}$$

and then solving

$$\frac{\partial\psi}{\partial\tau} + \mathrm{sign}(\phi)\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\psi = 0 \tag{4.7}$$

where the first equation makes ψ a signed distance function on the level surface $\phi=0$ and the second one makes level sets of ψ orthogonal to the surface. Those properties will prove important when we need to discretize delta function of a codimension two object [towers2009discretizing]. Note that the advection velocity for the curve and surface need not to be the same if they have different physical origins. Suppose the relative motion of the curve on the surface is dictated by a velocity field $\boldsymbol{V}$ and that of a surface by $\boldsymbol{U}$. Then the dynamics of the system is

$$\begin{aligned} \frac{\partial\phi}{\partial t} + \boldsymbol{U}\cdot\nabla\phi &= 0 \qquad (4.8)\\ \frac{\partial\psi}{\partial t} + (\boldsymbol{U}+\boldsymbol{V})\cdot\nabla\psi &= 0, \qquad (4.9)\end{aligned}$$

which says that the curve also moves along with the surface [cheng2002motion]. Equations like those also appear in the region tracking algorithms [bertalmio1999region].

4.2 Discretization of Geometries in the Level Set Framework

With the level set method, a two dimensional surface $\mathcal{P}$ is represented implicitly as the zero level set of the level set function $\phi(Z)$ in the three dimensional space

$$\mathcal{P} = \{\boldsymbol{R}(Z)|\phi(Z)=0\}. \tag{4.10}$$

A strength of the level set formalism is that it allows irregular surfaces to be defined implicitly on a regular grid, such as a simple, rectangular Cartesian grid. Computing geometric properties of an implicit interface represented by the level set function $\phi(x, y, z)$ is then much easier than that for a triangulated surface. Most of the geometrical information about the interface is encoded in the gradient $\nabla\phi$ and Hessian $H(\phi)$ of ϕ, which in Cartesian coordinates are defined as

$$\nabla\phi = \begin{pmatrix} \phi_x & \phi_y & \phi_z \end{pmatrix} \tag{4.11}$$

$$\text{Hessian}(\phi) = \begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix}. \tag{4.12}$$

4.2.1 Surface Normals and Surface Curvature

Suppose that the surface is represented implicitly as the zero level set contour of the signed distance function $\phi(Z^1, Z^2, Z^3)$ and $\phi < 0$ inside. Then by this convention, we choose the normal to the surface to point into the region where $\phi > 0$. The orientation of surface coordinates is then chosen such that $\boldsymbol{N} \cdot (\boldsymbol{S}_1 \times \boldsymbol{S}_2) > 0$. By this convention, the normal of the surface can be represented by

$$\boldsymbol{N} = \frac{\nabla\phi}{|\nabla\phi|}, N^i = \frac{\nabla^i\phi}{\sqrt{\nabla^j\phi\nabla_j\phi}}. \tag{4.13}$$

The tensor product of ∇ and $\boldsymbol{N}$ encodes curvature information,

$$\begin{aligned} \nabla^j N_i &= \nabla^j\left(\frac{\nabla_i\phi}{|\nabla\phi|}\right) = \frac{\nabla^j\nabla_i\phi}{|\nabla\phi|} \quad \frac{\nabla_i\phi}{2|\nabla\phi|^3}\nabla^j(\nabla^k\phi\nabla_k\phi) = \frac{\nabla^j\nabla_i\phi}{|\nabla\phi|} \quad \frac{\nabla_i\phi}{|\nabla\phi|^3}\nabla^j\nabla^k\phi\nabla_k\phi \\ &= \frac{1}{|\nabla\phi|^3}(|\nabla\phi|^2\nabla^j\nabla_i\phi \quad \nabla_i\phi\nabla^k\phi\nabla^j\nabla_k\phi) \\ &= (\delta_k^m\delta_i^n \quad \delta_i^m\delta_k^n)\frac{\nabla_m\phi\nabla^k\phi\nabla^j\nabla_n\phi}{|\nabla\phi|^3} = \delta_{ki}^{mn}\frac{\nabla_m\phi\nabla^k\phi\nabla^j\nabla_n\phi}{|\nabla\phi|^3} \end{aligned} \tag{4.14}$$

and the vector (matrix) form is

$$\begin{aligned} &\nabla \otimes \boldsymbol{N} \\ &= \frac{\text{Hessian}(\phi)}{|\nabla\phi|} \quad \frac{1}{|\nabla\phi|^3}\text{Hessian}(\phi) \otimes (\nabla\phi)^T \otimes \nabla\phi \qquad (4.15) \\ &= \frac{1}{|\nabla\phi|}\begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix} \quad \frac{1}{|\nabla\phi|^3}\begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix}\begin{pmatrix} \phi_x \\ \phi_y \\ \phi_z \end{pmatrix}\begin{pmatrix} \phi_x & \phi_y & \phi_z \end{pmatrix} \\ &= \frac{1}{|\nabla\phi|}\begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix} \quad \frac{1}{|\nabla\phi|^3}\begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix}\begin{pmatrix} \phi_x\phi_x & \phi_x\phi_y & \phi_x\phi_z \\ \phi_y\phi_x & \phi_y\phi_y & \phi_y\phi_z \\ \phi_z\phi_x & \phi_z\phi_y & \phi_z\phi_z \end{pmatrix} \\ &= \frac{1}{|\nabla\phi|}\begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix} \quad \frac{1}{|\nabla\phi|^3} \times \end{aligned}$$

$$\begin{pmatrix} \phi_{xx}\phi_x^2 + \phi_{xy}\phi_y\phi_x + \phi_{xz}\phi_z\phi_x & \phi_{xx}\phi_x\phi_y + \phi_{xy}\phi_y^2 + \phi_{xz}\phi_z\phi_y & \phi_{xx}\phi_x\phi_z + \phi_{xy}\phi_y\phi_z + \phi_{xz}\phi_z^2 \\ \phi_{yx}\phi_x^2 + \phi_{yy}\phi_y\phi_x + \phi_{yz}\phi_z\phi_x & \phi_{yx}\phi_x\phi_y + \phi_{yy}\phi_y^2 + \phi_{yz}\phi_z\phi_y & \phi_{yx}\phi_x\phi_z + \phi_{yy}\phi_y\phi_z + \phi_{yz}\phi_z^2 \\ \phi_{zx}\phi_x^2 + \phi_{zy}\phi_y\phi_x + \phi_{zz}\phi_z\phi_x & \phi_{zx}\phi_x\phi_y + \phi_{zy}\phi_y^2 + \phi_{zz}\phi_z\phi_y & \phi_{zx}\phi_x\phi_z + \phi_{zy}\phi_y\phi_z + \phi_{zz}\phi_z^2 \end{pmatrix}.$$

Remark 4.1. In this embedding,

$$\begin{aligned} N_j\nabla^j N_i &= \delta_{ki}^{mn}\frac{\nabla_m\phi\nabla^k\phi\nabla_j\phi\nabla^j\nabla_n\phi}{|\nabla\phi|^4} = \delta_{ki}^{mn}\frac{\nabla_m\phi\nabla^k\phi\nabla_j\phi\nabla_n\nabla^j\phi}{|\nabla\phi|^4} \\ &= \delta_{ki}^{mn}\frac{\nabla_m\phi\nabla^k\phi\nabla_n(\nabla^j\phi\nabla_j\phi)}{2|\nabla\phi|^4} = \delta_{ki}^{mn}\frac{\nabla_m\phi\nabla^k\phi\nabla_n|\nabla\phi|^2}{2|\nabla\phi|^4} \end{aligned}$$

Thus $\partial N_i/\partial N$ will be zero if $|\nabla\phi|$ is a constant. In particular, if ϕ is a signed distance map, the normal derivative of the components of the surface normal vector is zero.

To find the expression for mean curvature and Gaussian curvature, we first note that the curvature tensor B^{α}_{β} can be written as

$$B^{\alpha}_{\beta} \;=\; \quad Z^i_{\beta}\nabla^{\alpha}N_i = \quad Z^i_{\beta}Z^{\alpha}_j\nabla^j N_i. \tag{4.16}$$

The mean curvature is

$$\begin{aligned} K_M &= B^{\alpha}_{\alpha} = \quad Z^{\alpha}_i Z^j_{\alpha}\nabla_j N^i = (N^j N_i \quad \delta^j_i)\nabla_j N^i = \quad \nabla_i N^i = \quad \delta^{mn}_{ki}\frac{\nabla_m\phi\nabla^k\phi\nabla^i\nabla_n\phi}{|\nabla\phi|^3} & (4.17)\\ &= \frac{1}{|\nabla\phi|^3}(|\nabla\phi|^2\nabla^i\nabla_i\phi \quad \nabla_i\phi\nabla^k\phi\nabla^i\nabla_k\phi). & (4.18)\end{aligned}$$

The Gaussian curvature is $2K = (K_M)^2 \quad B^{\beta}_{\alpha}B^{\alpha}_{\beta}$. Thus we need to calculate $B^{\beta}_{\alpha}B^{\alpha}_{\beta}$:

$$\begin{aligned} B^{\beta}_{\alpha}B^{\alpha}_{\beta} &= Z^m_{\alpha}Z^{\beta}_n\nabla^n N_m Z^i_{\beta}Z^{\alpha}_j\nabla^j N_i = Z^m_{\alpha}Z^{\alpha}_j Z^{\beta}_n Z^i_{\beta}\nabla^n N_m\nabla^j N_i\\ &= (\delta^m_j \quad N^m N_j)(\delta^i_n \quad N^i N_n)\nabla^n N_m\nabla^j N_i\\ &= (\delta^m_j \quad N^m N_j)\nabla^i N_m\nabla^j N_i = \nabla^i N_j\nabla^j N_i. & (4.19)\end{aligned}$$

Therefore,

$$\begin{aligned} & K\\ =\; & \frac{1}{2!}(\nabla^i N_i\nabla^j N_j \quad \nabla^j N_i\nabla^i N_j) = \frac{1}{2!}(\delta^m_j\delta^n_i \quad \delta^m_i\delta^n_j)\nabla^j N_m\nabla^i N_n = \frac{1}{2!}\delta^{mn}_{ji}\nabla^j N_m\nabla^i N_n\\ =\; & \frac{1}{2!}\delta^{mn}_{ji}\delta^{rs}_{km}\frac{\nabla_r\phi\nabla^k\phi\nabla^j\nabla_s\phi}{|\nabla\phi|^3}\delta^{uv}_{ln}\frac{\nabla_u\phi\nabla^l\phi\nabla^i\nabla_v\phi}{|\nabla\phi|^3}\\ =\; & \frac{1}{2!}\frac{1}{|\nabla\phi|^6}\delta^{mn}_{ji}\delta^{rs}_{km}\delta^{uv}_{ln}\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\\ =\; & \frac{1}{2!}\frac{1}{|\nabla\phi|^6}(\delta^m_j\delta^n_i \quad \delta^m_i\delta^n_j)\delta^{rs}_{km}\delta^{uv}_{ln}\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\\ =\; & \frac{1}{2!}\frac{1}{|\nabla\phi|^6}(\delta^{rs}_{kj}\delta^{uv}_{li} \quad \delta^{rs}_{ki}\delta^{uv}_{lj})\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\\ =\; & \frac{\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi}{2|\nabla\phi|^6}[(\delta^r_k\delta^s_j \quad \delta^r_j\delta^s_k)(\delta^u_l\delta^v_i \quad \delta^u_i\delta^v_l) \quad (\delta^r_k\delta^s_i \quad \delta^r_i\delta^s_k)(\delta^u_l\delta^v_j \quad \delta^u_j\delta^v_l)]\\ =\; & \frac{\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi}{2|\nabla\phi|^6}[\delta^r_k(\delta^s_j\delta^{uv}_{li} \quad \delta^s_i\delta^{uv}_{lj}) + \delta^u_l(\delta^v_j\delta^r_i\delta^s_k \quad \delta^v_i\delta^r_j\delta^s_k) + \delta^r_j\delta^s_k\delta^u_i\delta^v_l \quad \delta^r_i\delta^s_k\delta^u_j\delta^v_l]\\ =\; & \frac{\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi}{2|\nabla\phi|^6}[\delta^r_k(\delta^s_j\delta^{uv}_{li} \quad \delta^s_i\delta^{uv}_{lj}) + \delta^r_k(\delta^v_j\delta^u_i\delta^s_l \quad \delta^v_i\delta^u_j\delta^s_l) + \delta^r_j\delta^s_k\delta^u_i\delta^v_l \quad \delta^r_i\delta^s_k\delta^u_j\delta^v_l]\\ =\; & \frac{1}{2|\nabla\phi|^6}\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\delta^r_k\delta^{suv}_{jli}\\ =\; & \frac{1}{2|\nabla\phi|^4}\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\delta^{suv}_{jli}\\ =\; & \frac{1}{|\nabla\phi|^4}\nabla_u\phi\nabla^l\phi\left(\frac{1}{2!}\delta^{svu}_{jil}\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\right) & (4.20)\end{aligned}$$

where we used

$$\begin{aligned} & \delta^r_k(\delta^s_j\delta^{uv}_{li} \quad \delta^s_i\delta^{uv}_{lj}) + \delta^r_k(\delta^v_j\delta^u_i\delta^s_l \quad \delta^v_i\delta^u_j\delta^s_l) = \delta^r_k(\delta^s_j\delta^{uv}_{li} \quad \delta^s_i\delta^{uv}_{lj}) + \delta^r_k\delta^s_l\delta^{vu}_{ji}\\ =\; & \delta^r_k(\delta^s_j\delta^{uv}_{li} \quad \delta^s_i\delta^{uv}_{lj} + \delta^s_l\delta^{vu}_{ji}) = \delta^r_k(\delta^s_j\delta^{uv}_{li} \quad \delta^s_l\delta^{uv}_{ji} + \delta^s_i\delta^{uv}_{jl}) = \delta^r_k\delta^{suv}_{jli}\end{aligned}$$

and

$$\begin{aligned} & \frac{\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi}{2|\nabla\phi|^6}(\delta^r_j\delta^s_k\delta^u_i\delta^v_l \quad \delta^r_i\delta^s_k\delta^u_j\delta^v_l)\\ =\; & \frac{\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi}{2|\nabla\phi|^6}\delta^r_j\delta^s_k\delta^u_i\delta^v_l \quad \frac{\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi}{2|\nabla\phi|^6}\delta^r_i\delta^s_k\delta^u_j\delta^v_l\\ =\; & \frac{\nabla_r\phi\nabla^k\phi\nabla_u\phi\nabla^l\phi\nabla^r\nabla_k\phi\nabla^u\nabla_l\phi}{2|\nabla\phi|^6} \quad \frac{\nabla_i\phi\nabla^s\phi\nabla_j\phi\nabla^v\phi\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi}{2|\nabla\phi|^6} = 0.\end{aligned}$$

Summarily,

$$\boldsymbol{N} \;=\; \frac{\nabla^i\phi}{\sqrt{\nabla^j\phi\nabla_j\phi}}\boldsymbol{Z}_i \tag{4.21}$$

$$K_M \;=\; \nabla_i N^i = \frac{1}{|\nabla\phi|^3}\delta^{mn}_{ki}\nabla_m\phi\nabla^k\phi\nabla^i\nabla_n\phi = \frac{1}{|\nabla\phi|^3}(|\nabla\phi|^2\nabla^i\nabla_i\phi \quad \nabla_i\phi\nabla^k\phi\nabla^i\nabla_k\phi) \tag{4.22}$$

$$K \;=\; \frac{1}{2!}\delta^{mn}_{ji}\nabla^j N_m\nabla^i N_n = \frac{1}{|\nabla\phi|^4}\nabla_u\phi\nabla^l\phi\left(\frac{1}{2!}\delta^{svu}_{jil}\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\right). \tag{4.23}$$

Note that $\frac{1}{2!}\delta^{svu}_{jil}\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi)$ is the cofactor matrix of $\nabla^i\nabla_j\phi$. If $Z^i = \{x,\, y,\, z\}$, then the above covariant formula for K_M and K reduces to the expression given by Goldman [goldman2005curvature]. Since our formula is covariant, it applies to general curvilinear coordinates as well. This facilitates applications with spherical or cylindrical symmetries.

In Cartesian coordinates, the normal is

$$\boldsymbol{N} \;=\; \frac{\nabla\phi}{|\nabla\phi|} = \frac{1}{|\nabla\phi|}(\ \phi_x \quad \phi_y \quad \phi_z\ ). \tag{4.24}$$

We also need the cofactor matrix of the Hessian

$$H^*(\phi) \;=\; \begin{pmatrix} \phi_{yy}\phi_{zz} & \phi_{zy}\phi_{yz} & \phi_{zx}\phi_{yz} & \phi_{yx}\phi_{zz} & \phi_{yx}\phi_{zy} & \phi_{zx}\phi_{yy} \\ \phi_{zy}\phi_{xz} & \phi_{xy}\phi_{zz} & \phi_{xx}\phi_{zz} & \phi_{zx}\phi_{xz} & \phi_{zx}\phi_{xy} & \phi_{xx}\phi_{zy} \\ \phi_{xy}\phi_{yz} & \phi_{yy}\phi_{xz} & \phi_{xz}\phi_{yx} & \phi_{xx}\phi_{yz} & \phi_{xx}\phi_{yy} & \phi_{yx}\phi_{xy} \end{pmatrix} \tag{4.25}$$

The mean curvature K_M in Cartesian coordinates is

$$\begin{aligned} K_M \;&=\; \frac{1}{|\nabla\phi|^3}(|\nabla\phi|^2\nabla^i\nabla_i\phi \quad \nabla_i\phi\nabla^k\phi\nabla^i\nabla_k\phi) \\ &=\; \frac{|\nabla\phi|^2\mathrm{Trace}(\mathrm{Hessian}(\phi)) + \nabla\phi\otimes\mathrm{Hessian}(\phi)\otimes(\nabla\phi)^T}{|\nabla\phi|^3} \qquad (4.26) \\ &=\; \frac{1}{|\nabla\phi|^3}\left[\ |\nabla\phi|^2(\phi_{xx}+\phi_{yy}+\phi_{zz}) + (\ \phi_x \quad \phi_y \quad \phi_z\ )\begin{pmatrix} \phi_{xx} & \phi_{xy} & \phi_{xz} \\ \phi_{yx} & \phi_{yy} & \phi_{yz} \\ \phi_{zx} & \phi_{zy} & \phi_{zz} \end{pmatrix}\begin{pmatrix} \phi_x \\ \phi_y \\ \phi_z \end{pmatrix}\right] \qquad (4.27) \end{aligned}$$

which gives 2 for a unit sphere. Note that $K_M|\nabla\phi|$ can be split into a linear operator on ϕ and a nonlinear operator on ϕ. This operator splitting can play a crucial role in designing a stable scheme for curvature involved surface propagation [smereka2003semi, xu2003eulerian, duchemin2014explicit].

The Gaussian curvature K is

$$\begin{aligned} K \;&=\; \frac{1}{|\nabla\phi|^4}\nabla_u\phi\nabla^l\phi\left(\frac{1}{2!}\delta^{svu}_{jil}\nabla^j\nabla_s\phi\nabla^i\nabla_v\phi\right) = \frac{\nabla\phi\otimes H^*(\phi)\otimes(\nabla\phi)^T}{|\nabla\phi|^4} \qquad (4.28) \\ &=\; \frac{1}{|\nabla\phi|^4}(\ \phi_x \quad \phi_y \quad \phi_z\ )\begin{pmatrix} \phi_{yy}\phi_{zz} & \phi_{zy}\phi_{yz} & \phi_{zx}\phi_{yz} & \phi_{yx}\phi_{zz} & \phi_{yx}\phi_{zy} & \phi_{zx}\phi_{yy} \\ \phi_{zy}\phi_{xz} & \phi_{xy}\phi_{zz} & \phi_{xx}\phi_{zz} & \phi_{zx}\phi_{xz} & \phi_{zx}\phi_{xy} & \phi_{xx}\phi_{zy} \\ \phi_{xy}\phi_{yz} & \phi_{yy}\phi_{xz} & \phi_{xz}\phi_{yx} & \phi_{xx}\phi_{yz} & \phi_{xx}\phi_{yy} & \phi_{yx}\phi_{xy} \end{pmatrix}\begin{pmatrix} \phi_x \\ \phi_y \\ \phi_z \end{pmatrix} \end{aligned}$$

4.2.2 Surface Integral and Volume Integral

Since the surface is represented implicitly, all surface integrals need to be calculated as volume integrals. Let us use (S^1, S^2, ϕ) as the coordinates for the three dimensional Euclidean space near the surface. Then, with the help of the Dirac delta function $\delta(\phi)$, a surface integral over a function f can be written as

$$I \;=\; \int f(S)\mathrm{d}A = \int f(S)\delta(\phi)\mathrm{d}A\,\mathrm{d}\phi. \tag{4.29}$$

Note that the first integral is taken over the coordinate space (S^1, S^2) which is mapped to the surface and the second integral is take over the the coordinate space (S^1, S^2, ϕ) where $\phi = 0$ is mapped to the surface and we assume that the mapping $\{S^1, S^2, \phi\} \to \mathbb{R}^3$ is smooth and that coordinate lines $S^\alpha = \text{constant}$ is perpendicular to level set surface of constant ϕ. Now consider the coordinate lines $S^\alpha = \text{const}$ parametrized by ϕ. We can make a change of variable from ϕ to arclength s and write

$$I \;=\; \int f(S)\delta(\phi)\frac{\mathrm{d}\phi}{\mathrm{d}s}\mathrm{d}A\mathrm{d}s = \int f(S)\delta(\phi)\boldsymbol{N}\cdot\nabla\phi\mathrm{d}V = \int f(S)\delta(\phi)|\nabla\phi|\mathrm{d}V \tag{4.30}$$

where we see that $\mathrm{d}\phi/\mathrm{d}s = \nabla_s\phi = \boldsymbol{N}\cdot\nabla\phi = |\nabla\phi|$ is just the invariant derivative of ϕ along the s axis and $\mathrm{d}V$ is the invariant volume element. Therefore we may write

$$\int_{\mathcal{P}} f(S)\mathrm{d}A \;=\; \int_{\Omega} F(\boldsymbol{R})\delta(\phi)|\nabla\phi|\mathrm{d}V \tag{4.31}$$

where $\mathcal{P} = \{\boldsymbol{R}\in\Omega|\phi(\boldsymbol{R}) = 0\}$ is a two dimensional subset of $\Omega\subset\mathbb{R}^3$ such that $\phi(\boldsymbol{R}\in\mathcal{P}) = 0$ and the restriction of F to $\mathcal{P}$ agrees with the value of f on $\mathcal{P}$. The Hausdorff measure (surface area) and Lesbegue measure (volume) are

$$A = |\;\;(t)| \;=\; \int\delta(\phi)|\nabla\phi|\mathrm{d}V \tag{4.32}$$

$$V = |\Omega(t)| \;=\; \int H(\;\;\phi)\mathrm{d}V, \tag{4.33}$$

where $H(x)$ is the Heaviside function. Now suppose that $f(S)$ represents surface density of something (protein, lipids, force etc.), then $F(\boldsymbol{R})\delta(\phi)|\nabla\phi|$ will represent volume density, which tells us how to embedded surface quantities in space numerically, for example, embedding boundary forces in bulk. Note that if ϕ is replaced by another function $\tilde{\phi}$ such that $\phi'(\tilde{\phi}) > 0$ and that they have the same zero level set contour, then because $\delta(\phi)|\nabla\phi| = \delta(\tilde{\phi})|\nabla\tilde{\phi}|$, $F(\boldsymbol{R})\delta(\phi)|\nabla\phi|$ will remain unchanged. This is a necessary requirement for the use of reinitialization scheme in the level set method, which replaces ϕ with a signed distance function having the same zero level set. In the extension step, F is replaced by another $\tilde{F}$ having a vanishing normal derivative. This also keeps $F\delta(\phi)|\nabla\phi|$ unchanged.

4.2.2.1 Smeared Dirac-Delta Function and Heaviside Function

The common practice in the level set method to discretize the Dirac-Delta function and Heaviside function is to smear them out to a bandwise of $\epsilon = 1.5\Delta x$ around the interface [osher2006level]:

$$H(\phi) \;=\; \begin{cases} 0 & \phi < \;\;\epsilon \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\epsilon}\right) & \;\;\epsilon \le \phi \le \epsilon \\ 1 & \epsilon < \phi \end{cases} \tag{4.34}$$

and

$$\delta(\phi) \;=\; \begin{cases} 0 & \phi < \;\;\epsilon \\ \frac{1}{2\epsilon} + \frac{1}{2\epsilon}\cos\left(\frac{\pi\phi}{\epsilon}\right) & \;\;\epsilon \le \phi \le \epsilon\;. \\ 0 & \epsilon < \phi. \end{cases} \tag{4.35}$$

This discretization, however, can lead to systematic $\mathcal{O}(1)$ errors in trivial numerical experiments such as computing length of a straight line titled to the grid axis with some angle [tornberg2004numerical]. Several numerical schemes for approximating $\delta(\phi)$ and $H(\phi)$ were then proposed to address this convergence issue. Engquist proposed to use information in local gradient of the level set function to modify ϵ [engquist2005discretization]. Smereka borrowed a technique from Green's function theory to discretize the delta function [smereka2006numerical]. We shall adopt a finite difference method developed by Towers in a series of papers [towers2007two, towers2008convergence, towers2009discretizing, towers2009finite] to discretize $\delta(\phi)$ and $H(\phi)$.

Following [towers2007two], we define $I(\phi) = \int_0^{\phi} H(\zeta) d\zeta = \max(\phi, 0)$ on the grid. Then,

$$\nabla I(\phi) = H(\phi)\nabla\phi \tag{4.36}$$
$$\Delta I(\phi) = \nabla H(\phi)\cdot\nabla\phi + H(\phi)\Delta\phi = \delta(\phi)|\nabla\phi|^2 + H(\phi)\Delta\phi. \tag{4.37}$$

and

$$\Delta H = \nabla\cdot\nabla H = \nabla\cdot(\nabla\phi\delta(\phi)) = \delta(\phi)\Delta\phi + \nabla\phi\cdot\nabla\delta(\phi) \tag{4.38}$$

Solving for $H(\phi)$ and $\delta(\phi)$ from the above relations gives

$$H(\phi) = \frac{\nabla I\cdot\nabla\phi}{|\nabla\phi|^2} \tag{4.39}$$
$$\delta(\phi) = \frac{\Delta I(\phi) \quad H(\phi)\Delta\phi}{|\nabla\phi|^2} = \frac{\Delta I(\phi)}{|\nabla\phi|^2} \quad \frac{(\nabla I\cdot\nabla\phi)\Delta\phi}{|\nabla\phi|^4}. \tag{4.40}$$
$$\frac{\nabla\phi}{|\nabla\phi|}\cdot\nabla\delta(\phi) = \frac{\Delta H \quad \delta(\phi)\Delta\phi}{|\nabla\phi|} \tag{4.41}$$

Since $I(\phi)$ is way more regular than $H(\phi)$ and $\delta(\phi)$, we do not need to smear it and simple finite difference schemes can be employed to calculate all the derivatives. Far away from the interface, $H(\phi)$ and $\delta(\phi)$ becomes trivial and we can use sign of the signed distance function to determine its value. Note that in [towers2009finite], the primitive of I is used to calculate $I(\phi)$ and $H(\phi)$.

The above ideas can be generalized to calculate multi-dimensional delta functions [towers2009discretizing], which appears in integrals on a codimension $d \quad m$ manifold represented by the intersection of the zero level sets of m level set functions $\phi^i, i = 1, 2\cdots m$ in a d-dimensional space parametrized by $Z^i, i = 1, 2\cdots d$

$$\int f(S)\mathrm{d}S = \int_{\Omega} \hat{f}(Z)\Pi_{i=1}^{m}\delta(\phi^i(Z))|\wedge_m\nabla\phi^i|\mathrm{d}\Omega \tag{4.42}$$

where $\mathrm{d}\Omega$ is the invariant volume element in d-dimensional space, S is some parametrization of , $\mathrm{d}S$ is the invariant volume element in , $\hat{f}(Z)$ is a function defined over Ω which when restricted to agrees with f and $\wedge$ is the wedge product. Consider the case $m = 2, d = 1$. Let us denote the two level set functions by ϕ and ψ. Then [towers2009discretizing]

$$\nabla H(\phi)\wedge\nabla H(\psi) = \delta(\phi)\delta(\psi)\nabla\phi\wedge\nabla\psi \tag{4.43}$$

and

$$\delta(\phi)\delta(\psi) = \frac{(\nabla\phi\wedge\nabla\psi)\cdot(\nabla H(\phi)\wedge\nabla H(\psi))}{|\nabla\phi\wedge\nabla\psi|^2} \tag{4.44}$$

where $\wedge$ becomes vector product for two arguments and $H(\phi)$ can be either a smeared version or calculated from I and possibly $J(\phi) = \int_0^{\phi} I(\zeta)d\zeta = \begin{cases} \phi^2/2 & \phi \geq 0 \\ 0 & \phi < 0 \end{cases}$. Formula for other cases can be generalized easily. In [towers2009discretizing], it was shown that to obtain convergence ϕ, ψ should be signed distance functions and their zero level sets should be orthogonal to each other and their normals should be parallel to the grid axis. The last condition can be hard to satisfy but the first two can be enforced.

4.2.2.2 Discretization of Embedded Curve Geometry

Phase boundaries and open edges of membranes can be numerically described as the zero level set of a vector function $(\phi(Z), \psi(Z))$ [burchard2001motion]. This idea is then borrowed by Wang with a phase field approach to the simulation of multi-component lipid membranes [wang2008modelling]. Their Hamiltonian for the phase boundary, however, contains only the zeroth order term from the more general Hamiltonian Eq. (3.2) and is therefore unable to describe more realistic effects of the boundary energy.

Following [burchard2001motion], a closed curve embedded in the surface $\mathcal{P}=\{\boldsymbol{R}(Z)|\phi(Z)=0\}$ is represented by $=\{\boldsymbol{R}(Z)|\phi(Z)=0,\psi(Z)=0\}$. Then separates $\mathcal{P}$ into two regions corresponding to $\psi>0$ and $\psi<0$. We shall choose the orientation of such that as we move along the tangential direction $\boldsymbol{t}$ of with our head pointing to the $\boldsymbol{N}$ direction, the $\psi<0$ region is to our left. Furthermore, we choose the direction of the normal $\boldsymbol{n}$ to be pointing into the $\psi>0$ region. With this convention, $\{\boldsymbol{n},\boldsymbol{t},\boldsymbol{N}\}$ forms a right-handed coordinate system. The geometries of can then be expressed in terms of $\phi(Z)$ and $\psi(Z)$.

The tangent vector is

$$\boldsymbol{T} \;=\; \nabla\phi\times\nabla\psi=T\boldsymbol{t}, \tag{4.45}$$

where $T=|\boldsymbol{T}|$ and $\boldsymbol{t}$ is the unit tangent

$$\boldsymbol{t} \;=\; \frac{\nabla\phi\times\nabla\psi}{|\nabla\phi\times\nabla\psi|}. \tag{4.46}$$

The unit normal $\boldsymbol{n}$ is

$$\boldsymbol{n} \;=\; \boldsymbol{t}\times\boldsymbol{N}. \tag{4.47}$$

The invariant derivatives ∇_s and $\nabla_\perp$ are

$$\nabla_s \;=\; \boldsymbol{t}\cdot\nabla=t^i\nabla_i \tag{4.48}$$
$$\nabla_\perp \;=\; \boldsymbol{n}\cdot\nabla=n^i\nabla_i. \tag{4.49}$$

The geodesic curvature is

$$k_g \;=\; \boldsymbol{n}\cdot\nabla_s\boldsymbol{t}=n^i\nabla_s t_i=n^it^j\nabla_j t_i \qquad (4.50).$$

The normal curvature is

$$\begin{aligned} k_n &= \boldsymbol{N}\cdot\nabla_s\boldsymbol{t}=N^i\nabla_s t_i=N^it^j\nabla_j t_i\\ &= \quad \boldsymbol{t}\cdot\nabla_s\boldsymbol{N}= \quad t^i\nabla_s N_i= \quad t^it^j\nabla_j N_i. \end{aligned} \tag{4.51}$$

The geodesic torsion is

$$\tau_g \;=\; \quad \boldsymbol{t}\cdot\nabla_\perp\boldsymbol{N}= \quad t^i\nabla_\perp N_i= \quad t^in^j\nabla_j N_i. \tag{4.52}$$

$B_\perp$ is

$$B_\perp \;=\; \quad \boldsymbol{n}\cdot\nabla_\perp\boldsymbol{N}= \quad n^i\nabla_\perp N_i= \quad n^in^j\nabla_j N_i. \tag{4.53}$$

As can be seen, $k_n,\tau_g,B_\perp$ are components of $\nabla\otimes\boldsymbol{N}$ in the Darboux frame. For numerical computation of k_g, we first calculate

$$\nabla_s\boldsymbol{t} \;=\; \nabla_s\frac{\boldsymbol{T}}{T}=\frac{\nabla_s\boldsymbol{T}}{T}+\boldsymbol{T}\nabla_s\left(\frac{1}{T}\right). \tag{4.54}$$

Then

$$k_g=\boldsymbol{n}\cdot\nabla_s\boldsymbol{t}=\frac{1}{T}\boldsymbol{n}\cdot\nabla_s\boldsymbol{T}=\frac{1}{T}\boldsymbol{n}\cdot(\nabla_s\nabla\phi\times\nabla\psi+\nabla\phi\times\nabla_s\nabla\psi), \tag{4.55}$$

where $\nabla_s\nabla\phi$ and $\nabla_s\nabla\psi$ should be calculated as $\nabla_s\nabla\phi=\nabla_s\nabla^i\phi\boldsymbol{Z}_i=t^j\nabla_j\nabla^i\phi\boldsymbol{Z}_i$, $\nabla_s\nabla\psi=\nabla_s\nabla^i\psi\boldsymbol{Z}_i=t^j\nabla_j\nabla^i\psi\boldsymbol{Z}_i$ so that the second derivatives of ϕ,ψ are directly discretized.

4.2.3 Curve Integral and Curve Length

The integral of a scalar field F over can be computed as a volume integral

$$\int F(s)\mathrm{d}l=\int_\Omega\hat{F}(\boldsymbol{R})\delta(\phi)\delta(\psi)|\nabla\phi\times\nabla\psi|\mathrm{d}V. \tag{4.56}$$

The length of the curve can be written as [burchard2001motion]

$$L(\phi,\psi) \;=\; \int_\Omega \delta(\phi)\delta(\psi)|\nabla_{||}\psi||\nabla\phi|d\Omega = \int_\Omega \delta(\phi)\delta(\psi)|\nabla\phi\times\nabla\psi|d\Omega, \tag{4.57}$$

where we used

$$\begin{aligned} |\nabla_{||}\psi|^2 &= |\nabla\psi \quad \boldsymbol{N}\boldsymbol{N}\cdot\nabla\psi|^2 = |\nabla\psi|^2 + |\boldsymbol{N}\cdot\nabla\psi|^2 \quad 2|\boldsymbol{N}\cdot\nabla\psi|^2 \\ &= |\nabla\psi|^2|\boldsymbol{N}|^2 \quad |\boldsymbol{N}\cdot\nabla\psi|^2 = |\boldsymbol{N}\times\nabla\psi|^2 = \left(\frac{|\nabla\phi\times\nabla\psi|}{|\nabla\phi|}\right)^2. \end{aligned} \tag{4.58}$$

4.3 Sixth-Order Accurate Schemes for Reinitialization and Extrapolation

Mathematically modeling dynamics in these problems can be challenging when high-order, non-linear partial differential equations are involved. For instance, the force density for a thin elastic surface with bending rigidity is proportional to the Laplacian of the mean curvature of the surface [helfrich1973elastic]. Simulation of thin surfaces, such as cell membranes, then requires calculation of second order derivatives of the surface mean curvature, which includes fourth order derivatives of the surface position [helfrich1973elastic]. The Cahn-Hillard equation describing phase separation on a surface, as occurs in the formation of lipid rafts [sezgin2017], leads to fourth order derivatives of an order parameter [greer2006fourth]. To accurately capture these dynamics in the level set framework, convergent methods for fourth order derivatives of the level set function and the surface fields are necessary, which requires the level set function to be locally smooth and the extrapolation scheme to be very accurate. Existing methods do not provide sufficient accuracy to address these types of problems [du2008second, guckenberger2016bending]. Therefore, we develop a method here that can be used accurately preserve geometric properties, such as high order derivatives of the shape, of dynamic interfaces and can also be used to extrapolate information defined on those surfaces.

Both Eq. (4.2) and Eq. (4.3) are Hamilton-Jacobi equations with initial and boundary conditions

$$\frac{\partial\psi}{\partial\tau} + H(\psi,\nabla\psi) = 0 \tag{4.59}$$

where H is the corresponding Hamiltonian. Successful techniques for solving Hamilton-Jacobi equations depend on construction of the numerical Hamiltonian [bardi1991nonconvex], time discretization [shu1988efficient] and space discretization [jiang1996efficient, jiang2000weighted], and accurate methods have been developed and extensively tested numerically. For spatial discretization, the weighted, essentially-non-oscillatory (WENO) scheme is often used [jiang2000weighted]. This method uses divided differences to determine the smoothness of various approximations to the first derivative, and then weights the various approximations to achieve a smooth, and potentially fifth-order accurate, estimate of the derivative. While this scheme works well when solving the level set equation (Eq. (4.1)), application of the WENO scheme to the reinitialization equation (Eq. (4.2)) can yield poor results [jiang2000weighted]. The reason for the poor accuracy was pointed out by Russo and Smereka [russo2000remark], and they developed a second order accurate remedy by modifying the treatment of Eq. (4.2) near the boundary. Based on the results of Russo and Smereka, Chéné and Min [du2008second] obtained a fourth order accurate reinitialization scheme by adopting a cubic ENO interpolation near the boundary and an HJ-WENO scheme away from the boundary.

In this section, we further develop the ideas in [russo2000remark] and [du2008second] to design a novel sixth-order accurate scheme for Hamilton-Jacobi equations with boundary condition specified at the zero of a level set function. Note that because Eqs. (2-3) are hyperbolic, information defined at the zero level set is propagated away from the implicit boundary and an additional boundary condition at the edge of the computational domain is not needed.

Hamilton-Jacobi equations (Eq. (4.59)) involve directionality in the flow of information that carries the function ψ at one time to its values at later times. As such, it is crucial to use an upwind spatial discretization along with an accurate time stepping routine. In this section, we describe the standard spatial and temporal discretization schemes that are used with the level set method and then provide a detailed description of our modification to the ENO scheme for non-uniform grids that was originally developed in [du2008second].

All of our computations are carried out on three dimensional Cartesian grids, with the level set function and its geometry fields (normals, curvatures etc.) defined at the nodes of the grid. Fourth order finite difference schemes are used to compute derivatives in Eq. (4.11-4.55). For instance [chopp1999motion],

$$\begin{aligned}
\phi_x &= \frac{1}{12\Delta x}(\;\;\phi_{i+2,j,k}+8\phi_{i+1,j,k}\;\;\;8\phi_{i\;\;1,j,k}+\phi_{i\;\;2,j,k}) && (4.60)\\
\phi_{xx} &= \frac{1}{12(\Delta x)^2}(\;\;\phi_{i+2,j,k}+16\phi_{i+1,j,k}\;\;\;30\phi_{i,j,k}+16\phi_{i\;\;1,j,k}\;\;\;\phi_{i\;\;2,j,k}) && (4.61)\\
\phi_{xy} &= \frac{1}{48\Delta x\Delta y}\begin{pmatrix} \;\;\phi_{i+2,j+2,k}+16\phi_{i+1,j+1,k}+\phi_{i\;\;2,j+2,k}\;\;\;16\phi_{i\;\;1,j+1,k} \\ +\phi_{i+2,j\;\;2,k}\;\;\;16\phi_{i+1,j\;\;1,k}\;\;\;\phi_{i\;\;2,j\;\;2,k}+16\phi_{i\;\;1,j\;\;1,k}\end{pmatrix}, && (4.62)
\end{aligned}$$

with similar constructions for the derivatives along the other directions.

4.3.1 Spatial Discretization

Eq. (4.59) can be written in a semi-discrete form as

$$\frac{\partial\psi}{\partial\tau}+\hat{H}(D_x\,\psi,D_x^+\psi;D_y\,\psi,D_y^+\psi;D_z\,\psi,D_z^+\psi)=0 \tag{4.63}$$

where $D_x^\pm\psi,D_y^\pm\psi,D_z^\pm\psi$ are the one-sided derivatives of ψ and $\hat{H}(D_x\,\psi,D_x^+\psi;\dots)$ is a numerical approximation of $H(\psi,\nabla\psi)$. Among all monotone schemes to construct $\hat{H}$, Godunov schemes introduce the least numerical diffusion and will be used for our solution. Godunov schemes, however, can be difficult to implement numerically for a general Hamiltonian, with specific Hamiltonians requiring their own specific treatment. For the reinitialization equation (Eq. (4.2)),

$$H(\phi,\nabla\phi)=\mathrm{Sign}(\phi^0)(|\nabla\phi|\;\;\;1)$$

and

$$\hat{H}(D_x\,\psi,D_x^+\psi;D_y\,\psi,D_y^+\psi;D_z\,\psi,D_z^+\psi)=\mathrm{Sign}(\phi^0)\left\{\sqrt{\phi_x^2+\phi_y^2+\phi_z^2}\;\;\;1\right\} \tag{4.64}$$

where

$$\phi_x^2\equiv\max\left(\max\left(D_x\,\phi,0\right)^2,\min\left(D_x^+\phi,0\right)^2\right) \tag{4.65}$$

when $\mathrm{Sign}(\phi^0)>0$ and

$$\phi_x^2\equiv\max\left(\min\left(D_x\,\phi,0\right)^2,\max\left(D_x^+\phi,0\right)^2\right) \tag{4.66}$$

when $\mathrm{Sign}(\phi^0)\le 0$. ϕ_y^2 and ϕ_z^2 are defined similarly. For the extrapolation equation (Eq. (4.3)),

$$H(\nabla c)=\mathrm{Sign}(\phi)\boldsymbol{N}\cdot\nabla c \tag{4.67}$$

and

$$\hat{H}(D_x\,c,D_x^+c;D_y\,c,D_y^+c;D_z\,c,D_z^+c)=\sum_{i=x,y,z}\{\min\left(V_i,0\right)D_i^+c+\max\left(V_i,0\right)D_i\;\;c\}. \tag{4.68}$$

where we define $\boldsymbol{V}\equiv\mathrm{Sign}(\phi)\boldsymbol{N}$. Numerically, $\mathrm{Sign}(\phi)$ can take a value among 1, $\;\;$1 and 0. We set $\mathrm{Sign}(\phi)$ to be zero when $\mathrm{abs}(\phi)<10\;\;^{6}\times\Delta x$, where Δx is the grid spacing.

4.3.2 One-Sided WENO Derivatives on a Nonuniform Grid

Solving the reinitialization equation (Eq. (4.2)) or extension equation (Eqs. (4.3)), requires forward and backward one-sided derivatives. For nodes not immediately next to the boundary, the standard WENO schemes from Jiang [jiang2000weighted, jiang1996efficient] can be applied to calculate these derivatives. However, for nodes immediately next to the boundary, special treatments are usually needed [russo2000remark, du2008second]. In [du2008second], the subcell fix from [russo2000remark] was augmented using general ENO reconstructions to achieve 4th order accuracy. Here, we utilize the conceptual method for converting from ENO to WENO derivatives in order to further improve the scheme of [du2008second], such that our method can achieve an optimal 6th order accuracy when solving HJ equations involving level set defined boundaries conditions. The key point is to develop WENO schemes for non-uniform grids and then apply this more general WENO scheme to calculate one-sided derivatives for boundary nodes.

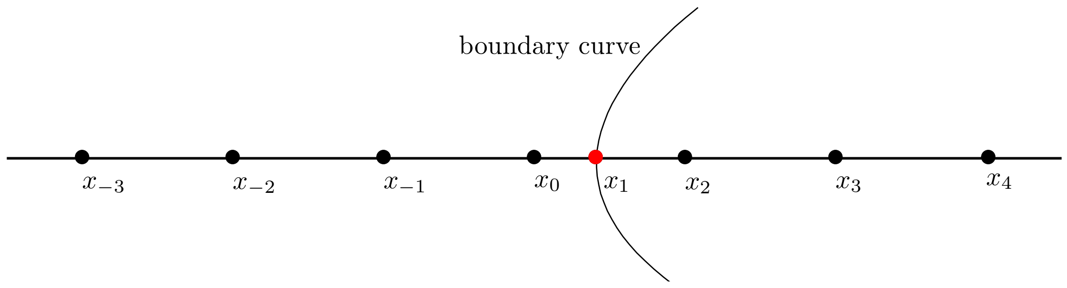

Figure 4.1. When the boundary passes between nodes x_0 and x_2, an accurate estimate for the off-grid location of the boundary x_1 is used when calculating the one-sided WENO derivatives for x_0 and x_2.

4.3.2.1 Computation of the Location of the Boundary

In three dimensional space, the boundary defined by the zero level set of $\phi(x, y, z)$ is a two dimensional surface. To compute the one-sided WENO derivatives near the boundary with sufficient accuracy, we need to locate the points where the zero contour crosses between grid nodes. We define a grid node at point (x_0, y_0, z_0) to be a boundary node if $\phi(x_0, y_0, z_0)$ differs in sign from any of its six nearest neighbors on the Cartesian grid.

For instance, if $\phi(x_0, y_0, z_0)\phi(x_0 + \Delta x, y_0, z_0) < 0$, both (x_0, y_0, z_0) and $(x_0 + \Delta x, y_0, z_0)$ are boundary nodes. To compute $D_x^{\pm}\phi$ at (x_0, y_0, z_0) and $(x_0 + \Delta x, y_0, z_0)$, we begin by localizing the crossing between the boundary and the line segment $(x_0 + \xi\Delta x, y_0, z_0), \xi \in (0, 1)$. This is illustrated in Figure 4.1, where the boundary passes between x_0 and x_2 at the location defined as x_1. Note that all the nodes shown are grid nodes except for x_1. For this calculation, we are only concerned with the crossing point along the x direction. Crossing points along the y and z directions are handled in an identical manner.

When $\phi(x_0)\phi(x_{\ 1}) < 0$ or $\phi(x_2)\phi(x_3) < 0$, there is a kink near x_0 or x_2. We then use $(x_{\ 1}, \phi_{\ 1})$, (x_0, ϕ_0), (x_2, ϕ_2), (x_3, ϕ_3) to construct a quadratic ENO polynomial of ϕ and the root x_1 of this polynomial approximates the boundary location [min2010reinitializing]. In particular [min2010reinitializing],

$$x_1 = \begin{cases} x_0 + \Delta x\left(\frac{1}{2} + \frac{\phi_0 \quad \phi_2 \quad \text{sign}(\phi_0 \quad \phi_2)\sqrt{D}}{\phi_{xx}^0}\right) & \text{if } |\phi_{xx}^0| > 10^{\quad 10} \\ x_0 + \Delta x \frac{\phi^0}{\phi^0 \quad \phi^2} & \text{else} \end{cases} \tag{4.69}$$

where $\phi_{xx}^0 = \text{minmod}(\phi_{\ 1} \quad 2\phi_0 + \phi_2, \phi_0 \quad 2\phi_2 + \phi_3)$, $D = (\phi_{xx}^0/2 \quad \phi_0 \quad \phi_2)^2 \quad 4\phi_0\phi_2$ and the minmod function is defined as

$$\operatorname{minmod}(\alpha,\beta)=\begin{cases}0 & \text{if}\ \alpha\beta\leqslant 0\\ \alpha & \text{if}\ \alpha\beta>0\ \text{and}\ |\alpha|\leqslant|\beta|\\ \beta & \text{if}\ \alpha\beta>0\ \text{and}\ |\alpha|>|\beta|\end{cases}. \tag{4.70}$$

When a kink is not present near x_0 and x_2, we construct a 5th order Lagrange polynomial that interpolates through $(x_i,\phi_i), i\in\{\ \ 2,\ \ 1,0,2,3,4\}$ and use Brent's method to find the root x_1 that lies between x_0 and x_2. Then $(x_1,\phi_1=0)$ is included in the stencil to construct $D_x^{\pm}\phi$ at x_0 and x_2. For later use, we define $\Delta x^{+}=x_1\ \ \ x_0$, which is the distance between node (x_0,y_0,z_0) and the boundary along the positive x direction. If the boundary is to the left of x_0 in Figure 4.1, this distance is then denoted as $\Delta x\ \ $, which will be the distance between node (x_0,y_0,z_0) and the boundary along the negative x direction. In a similar way, we can define $\Delta y^{\pm}$ and $\Delta z^{\pm}$ for the boundary nodes.

For an advected field ψ other than the level set function ϕ, a Lagrange polynomial $l_5(x)$ interpolating through $(x_i,\psi_i), i\in\{\ \ 2,\ \ 1,0,2,3,4\}$ is constructed and $(x_1,l_5(x_1))$ will be included in the stencil to construct $D_x^{\pm}\psi$ at x_0 and x_2.

In our analysis, we assumed that the zero level set is sufficiently far off from the boundary of the grid mesh so that we can always find enough grid points near the boundary to do the interpolation mentioned above. The accuracy of the boundary location procedure depends on the smoothness of the boundary and the advected field. Near a kink, the boundary location can only be determined with first order accuracy and the algorithm is then only first order accurate [min2010reinitializing].

The computation of boundary locations and the interpolation of other advected fields is carried out in a dimension by dimension manner. This process applies in the same way, regardless of the dimensionality of the problem.

4.3.2.2 Computation of WENO Derivatives

To compute the one-sided derivatives at node x_0, we begin by defining a seven-point stencil about this node (Figure 4.2). Note that when x_0 is a boundary node as is shown in Figure 4.1, the node (x_1,ψ_1) on the boundary should be included in the stencil, in which case the stencil will be nonuniform. In all other cases, the stencil is uniform. We maintain full generality by allowing all of the nodes to be non-uniformly spaced.

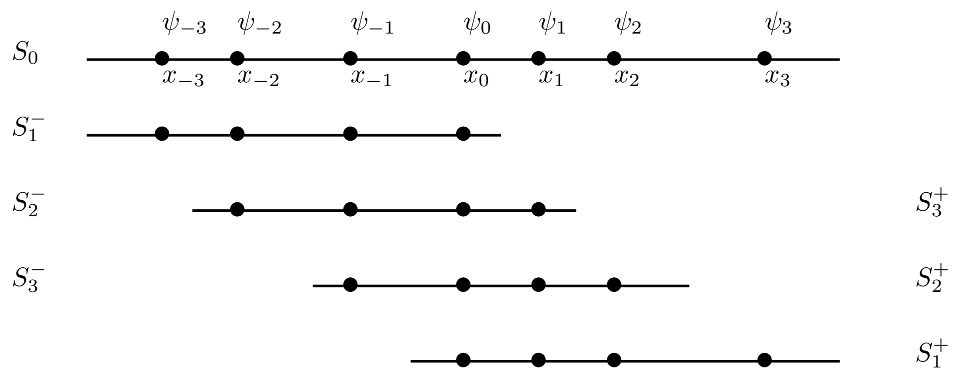

Figure 4.2. A general, nonuniform, seven-point stencil, S_0, is used to find backward and forward derivatives at x_0. In the WENO scheme, the full stencil is broken into substencils ($S_j^{\pm}$), and the derivative is approximated on each of these substencils. A weighted average of these approximations is then used to define the forward and backward WENO derivatives.

To compute $D_x\,\psi(x_0)$, a left biased stencil $S\ \ = \{(x_i, \psi_i) | i \in \{\ \ 3,\ \ 2,\ \ 1, 0, 1, 2\}\}$ should be used. The first step is to break S into three candidate ENO (essentially non-oscillatory) stencils $S_i\ , i = 1, 2, 3$ and approximate $\psi(x)$ on those stencils with polynomial interpolations $p_i\ (x), i = 1,$ 2, 3. For instance, $p_2\,(x)$ will be a third order polynomial interpolating through all points in $S_2\ = \{(x_i, \psi_i) | i \in \{\ \ 2,\ \ 1, 0, 1\}\}$ and $u_2\ \equiv \frac{dp_2\,(x)}{dx}|_{x=x_0}$ will be a candidate for $D_x\,\psi(x_0)$ in the ENO scheme. It is straightforward to compute all $u_i\ , i = 1, 2, 3$ and the results are [smit2005grid]

$$
\begin{aligned}
u_1 &= \bar{u}_{\ \frac{3}{2}} + \left(\frac{x_0\ \ \ x_{\ 1}}{x_{\ 1}\ \ \ x_{\ 3}}\right)\left(\frac{x_0\ \ \ x_{\ 2}}{x_0\ \ \ x_{\ 3}}\right)\left(\bar{u}_{\ \frac{5}{2}}\ \ \ \bar{u}_{\ \frac{3}{2}}\right) + \left(1 + \frac{x_0\ \ \ x_{\ 1}}{x_0\ \ \ x_{\ 2}} + \frac{x_0\ \ \ x_{\ 1}}{x_0\ \ \ x_{\ 3}}\right)\left(\bar{u}_{\ \frac{1}{2}}\ \ \ \bar{u}_{\ \frac{3}{2}}\right)\\
u_2 &= \bar{u}_{\ \frac{1}{2}} + \left(\frac{x_0\ \ \ x_{\ 1}}{x_1\ \ \ x_{\ 2}}\right)\left(\frac{x_0\ \ \ x_{\ 2}}{x_1\ \ \ x_{\ 1}}\right)\left(\bar{u}_{\frac{1}{2}}\ \ \ \bar{u}_{\ \frac{1}{2}}\right)\ \ \ \left(\frac{x_0\ \ \ x_{\ 1}}{x_1\ \ \ x_{\ 2}}\right)\left(\frac{x_1\ \ \ x_0}{x_0\ \ \ x_{\ 2}}\right)\left(\bar{u}_{\ \frac{3}{2}}\ \ \ \bar{u}_{\ \frac{1}{2}}\right)\\
u_3 &= \bar{u}_{\frac{1}{2}} + \left(\frac{x_1\ \ \ x_0}{x_2\ \ \ x_{\ 1}}\right)\left(\frac{x_2\ \ \ x_0}{x_1\ \ \ x_{\ 1}}\right)\left(\bar{u}_{\ \frac{1}{2}}\ \ \ \bar{u}_{\frac{1}{2}}\right)\ \ \ \left(\frac{x_1\ \ \ x_0}{x_2\ \ \ x_{\ 1}}\right)\left(\frac{x_0\ \ \ x_{\ 1}}{x_2\ \ \ x_0}\right)\left(\bar{u}_{\frac{3}{2}}\ \ \ \bar{u}_{\frac{1}{2}}\right)
\end{aligned}
\tag{4.71}
$$

where $\bar{u}_{r+\frac{1}{2}}, r \in \{\ \ 3,\ \ 2,\ \ 1, 0, 1, 1\}$ are first Newton divided differences defined as

$$\bar{u}_{r+\frac{1}{2}} = \frac{\psi_{r+1}\ \ \ \psi_r}{x_{r+1}\ \ \ x_r}. \tag{4.72}$$

The expressions for u_j^+ can be obtained by the following reflection transformation:

$$x_r \to x_{\ \ r}, \bar{u}_{r+\frac{1}{2}} \to \bar{u}_{\ \ \left(r+\frac{1}{2}\right)}. \tag{4.73}$$

In what follows, we will only list formula with superscript since those with superscript + can be obtained from reflection using Eq. (4.73).

In the ENO schemes introduced by Harten and Osher [harten1987uniformly], $D_x\,\psi(x_0)$ is approximated by the u_i from Eq. (4.71), which corresponds to the substencil S_i where $\psi(x)$ varies most smoothly. Consequently, ENO schemes approximate $D_x\,\psi(x_0)$ with only third order accuracy on a six point stencil S , since information from the other substencils are not used. WENO schemes use the same substencil approximations to the first derivatives, $u_1\ , u_2\ , u_3\ ,$ but employ a convex combination $\sum_{i=1}^{3} \omega_i\ u_i$ of the substencil derivatives to approximate $D_x\,\psi(x_0)$, thereby increasing the accuracy up to potentially 5th order [liu1994weighted].

When $\psi(x)$ is smooth over all stencils, the weights ω_i should approximately cancel truncation errors in u_i to achieve optimal accuracy. Let us denote by C_i weights that will give the optimal fifth order accuracy for $D_x\,\psi(x_0)$. In other words, if a fifth order polynomial $p\ \ (x)$ is constructed to interpolate through all points in S , the optimal weights C_i should satisfy

$$\frac{dp\ \ (x)}{dx}|_{x=x_0} = C_1\ u_1\ + C_2\ u_2\ + C_3\ u_3\ , \tag{4.74}$$

which gives uniquely [smit2005grid]

$$
\begin{aligned}
C_1 &= \left(\frac{x_1\ \ \ x_0}{x_2\ \ \ x_{\ 3}}\right)\left(\frac{x_2\ \ \ x_0}{x_1\ \ \ x_{\ 3}}\right)\\
C_2 &= \left(\frac{x_0\ \ \ x_{\ 3}}{x_2\ \ \ x_{\ 3}}\right)\left(\frac{x_2\ \ \ x_0}{x_2\ \ \ x_{\ 2}} + \frac{x_2\ \ \ x_0}{x_1\ \ \ x_{\ 3}}\right)\\
C_3 &= \left(\frac{x_0\ \ \ x_{\ 3}}{x_2\ \ \ x_{\ 3}}\right)\left(\frac{x_0\ \ \ x_{\ 2}}{x_2\ \ \ x_{\ 2}}\right).
\end{aligned}
\tag{4.75}
$$

When $\psi(x)$ is not smooth over the full stencil, non-smooth substencils should be given smaller weights. The smoothness indicator IS_i for stencil S_i is a weighted measure of the integrated square of the 2nd and 3rd derivatives over the substencil, which is given by [jiang1996efficient],

$$\mathrm{IS}_i\ = \sum_{l=1}^{2} \int_{x_{\ 1}}^{x_0} (x_0\ \ \ x_{\ 1})^{2l\ \ 1} \left(\frac{d^{l+1} p_i\ (x)}{d^{l+1} x}\right)^2 \mathrm{d}x, \tag{4.76}$$

leading to [smit2005grid]

$$\begin{aligned}
\mathrm{IS}_1 &= 40(x_0 \quad x_{\;1})^4 \;\; \bar w_{\;\frac{3}{2}})^2 + 4\left(\frac{x_0 \quad x_{\;1}}{x_0 \quad x_{\;3}}\right)^2 \left(\begin{array}{c} \{(x_0 \quad x_{\;2})\bar v_{\;2} \quad (2x_0 \quad x_{\;2} \quad x_{\;3})\bar v_{\;1}\}\times \\ \{(x_{\;1} \quad x_{\;2})\bar v_{\;2} \quad (x_0 + x_{\;1} \quad x_{\;2} \quad x_{\;3})\bar v_{\;1}\} \end{array}\right) \\
\mathrm{IS}_2 &= 40(x_0 \quad x_{\;1})^4 \;\; \bar w_{\;\frac{1}{2}})^2 + 4\left(\frac{x_0 \quad x_{\;1}}{x_1 \quad x_{\;2}}\right)^2 \left(\begin{array}{c} \{(x_0 \quad x_1)\bar v_{\;1} \quad (x_0 \quad x_{\;2})\bar v_0\} \\ \times\{(x_{\;1} \quad x_1)\bar v_{\;1} \quad (x_{\;1} \quad x_{\;2})\bar v_0\} \end{array}\right) \\
\mathrm{IS}_3 &= 40(x_0 \quad x_{\;1})^4 \;\; \bar w_{\frac{1}{2}})^2 + 4\left(\frac{x_0 \quad x_{\;1}}{x_2 \quad x_{\;1}}\right)^2 \left(\begin{array}{c} \{(2x_{\;1} \quad x_1 \quad x_2)\bar v_0 \quad (x_{\;1} \quad x_1)\bar v_1\}\times \\ \{(x_0 + x_{\;1} \quad x_1 \quad x_2)\bar v_0 \quad (x_0 \quad x_1)\bar v_1\} \end{array}\right),
\end{aligned} \tag{4.77}$$

where $\bar v_i$ and $\bar w_i$ are second and third Newton divided differences defined by

$$\bar v_i \equiv \frac{1}{x_{i+1} \quad x_{i \;\; 1}}\left(\bar u_{i+\frac{1}{2}} \quad \bar u_{i \;\; \frac{1}{2}}\right), \bar w_{i+\frac{1}{2}} \equiv \frac{1}{x_{i+2} \quad x_{i \;\; 1}}(\bar v_{i+1} \quad \bar v_i). \tag{4.78}$$

Weights that approximate the optimal weights C_i in smooth regions while suppressing oscillations in non-smooth regions are then defined by [liu1994weighted]

$$\omega_j = \frac{\alpha_j}{\sum_{k=1}^{3}\alpha_k}, \quad \alpha_j = \frac{C_j}{(\epsilon + \mathrm{IS}_j)^2}, \quad \epsilon = 10^{\;\;6}. \tag{4.79}$$

The WENO derivatives $D_x\,\psi(x_0)$ are then defined as

$$D_x\,\psi(x_0) = \sum_{j=1}^{3}\omega_j\, u_j\,. \tag{4.80}$$

When the grid is uniform, Eqs. (4.71,4.75,4.77) reduce to

$$\begin{aligned}
u_1 &= \frac{1}{3}\bar u_{\;\frac{5}{2}} \quad \frac{7}{6}\bar u_{\;\frac{3}{2}} + \frac{11}{6}\bar u_{\;\frac{1}{2}}, C_1 = 0.1 && (4.81)\\
u_2 &= \quad \frac{1}{6}\bar u_{\;\frac{3}{2}} + \frac{5}{6}\bar u_{\;\frac{1}{2}} + \frac{1}{3}\bar u_{\frac{1}{2}}, C_2 = 0.6 && (4.82)\\
u_3 &= \frac{1}{3}\bar u_{\;\frac{1}{2}} + \frac{5}{6}\bar u_{\frac{1}{2}} \quad \frac{1}{6}\bar u_{\frac{3}{2}}, C_3 = 0.3 && (4.83)
\end{aligned}$$

and

$$\begin{aligned}
\mathrm{IS}_1 &= \frac{13}{12}\left(\bar u_{\;\frac{5}{2}} \quad 2\bar u_{\;\frac{3}{2}} + \bar u_{\;\frac{1}{2}}\right)^2 + \frac{1}{4}\left(\bar u_{\;\frac{5}{2}} \quad 4\bar u_{\;\frac{3}{2}} + 3\bar u_{\;\frac{1}{2}}\right)^2 && (4.84)\\
\mathrm{IS}_2 &= \frac{13}{12}\left(\bar u_{\;\frac{3}{2}} \quad 2\bar u_{\;\frac{1}{2}} + \bar u_{\frac{1}{2}}\right)^2 + \frac{1}{4}\left(\bar u_{\;\frac{3}{2}} \quad \bar u_{\frac{1}{2}}\right)^2 && (4.85)\\
\mathrm{IS}_3 &= \frac{13}{12}\left(\bar u_{\;\frac{1}{2}} \quad 2\bar u_{\frac{1}{2}} + \bar u_{\frac{3}{2}}\right)^2 + \frac{1}{4}\left(3\bar u_{\;\frac{1}{2}} \quad 4\bar u_{\frac{1}{2}} + \bar u_{\frac{3}{2}}\right)^2, && (4.86)
\end{aligned}$$

which are the canonical formulae for the WENO scheme [jiang2000weighted]. We refer the reader to [jiang1996efficient] for a more thorough discussion of WENO schemes and to [smit2005grid] for WENO schemes on non-uniform grids.

4.3.3 Time Discretization

A popular time discretization for Eq. (4.63) is the third order TVD Runge-Kutta scheme [shu1988efficient] with the following Euler steps:

$$\begin{aligned}
\tilde\psi^{n+1} &= \psi^n \quad \Delta\tau\hat H(D_x\,\psi^n, D_x^+\psi^n; D_y\,\psi^n, D_y^+\psi^n; D_z\,\psi^n, D_z^+\psi^n) \\
\tilde\psi^{n+2} &= \tilde\psi^{n+1} \quad \Delta\tau\hat H(D_x\,\tilde\psi^{n+1}, D_x^+\tilde\psi^{n+1}; D_y\,\tilde\psi^{n+1}, D_y^+\tilde\psi^{n+1}; D_z\,\tilde\psi^{n+1}, D_z^+\tilde\psi^{n+1}) \\
\tilde\psi^{n+\frac{1}{2}} &= \frac{3}{4}\psi^n + \frac{1}{4}\tilde\psi^{n+2} \\
\tilde\psi^{n+\frac{3}{2}} &= \tilde\psi^{n+\frac{1}{2}} \quad \Delta\tau\hat H\left(D_x\,\tilde\psi^{n+\frac{1}{2}}, D_x^+\tilde\psi^{n+\frac{1}{2}}; D_y\,\tilde\psi^{n+\frac{1}{2}}, D_y^+\tilde\psi^{n+\frac{1}{2}}; D_z\,\tilde\psi^{n+\frac{1}{2}}, D_z^+\tilde\psi^{n+\frac{1}{2}}\right) \\
\tilde\psi^{n+1} &= \frac{1}{3}\psi^n + \frac{2}{3}\tilde\psi^{n+\frac{3}{2}}.
\end{aligned}$$

One benefit of this Runge-Kutta scheme is that relatively large CFL numbers can be used with the WENO scheme. We use 0.3 as our CFL number. In our numerical experiments, $\Delta\tau$ varies locally and at grid (x, y, z) is defined by [min2010reinitializing]

$$\Delta\tau = 0.3 \cdot \min\left(\Delta x^{+}, \Delta x\ \ , \Delta y^{+}, \Delta y\ \ , \Delta z^{+}, \Delta z\ \ \right), \tag{4.87}$$

where $\Delta x^{\pm}, \Delta y^{\pm}, \Delta z^{\pm}$ are taken to be $\Delta x, \Delta y, \Delta z$ for non-boundary nodes.

In the next chapter, numerical experiments will be presented to verify order of accuracy for our

method.

Chapter 5

Numerical Results: Sixth-Order Accurate Schemes for Reinitialization, Extrapolation and Computation of Geodesics

In this chapter, we present numerical experiments showing that our scheme for HJ equations with a level set defined boundary condition is indeed sixth-order accurate.

5.1 Computation of Interfacial Curvature and Bending Forces in 3D

A primary goal for developing a high order accurate implementation of the reinitialization equation is to preserve the quality of geometric information stored in the distance map at a sufficient level that the forces derived from the shape remain accurate. In problems involving elastic surfaces, bending forces depend on the curvatures (mean and Gaussian) and the surface Laplacian of the mean curvature. The simplest form for this force density (up to a multiplication constant) is $f = \Delta_{||}K_M + K_M^3 \quad 2K_M K$ [zhong1989bending]. In this section, we test our method's ability to preserve the accuracy of these quantities when an initial level set function (that is not necessarily an exact signed distance map) is reinitialized.

As a first test, we consider a spherical surface defined by an initial level set function $\phi(x,y,z) = x^2+y^2+z^2 \quad (0.6)^2$ in the $[\ \ 1,1]^3$ domain. This initial exact distance map is used as the seed for the reinitialization method described in Sec. 4.3. As shown in Table 5.1 the reinitialized ϕ maintains sixth-order accuracy. Using the reinitialized ϕ, we then compute the interfacial mean curvature K_M and Gaussian curvature K using fourth order accurate approximations to the first and second derivatives, as described in Sec. 5.3.1 of [chopp1999motion]. We compute the L_1 and L_2 errors in these quantities at the interface by integrating the error over the surface using the numerical integration described in Sec. 4.2. The L_∞ error is also determined for nodes at the boundary. Tables 5.2 and 5.3 demonstrate fourth-order accuracy for the curvatures. Finally, using the same approximations for the derivatives, we compute the surface Laplacian of the mean curvature, which is found to be second-order accurate (Table 5.4). As a comparison, Table 5.5 shows accuracy results for $\Delta_{||}K_M$ using the fourth order accurate scheme from [du2008second]. Comparing these results from Table 5.5 with Table 9 and Table 12 in [du2008second] shows that much higher accuracy can be achieved on a much smaller grid with our sixth-order accurate scheme.

	$\|\phi \quad \phi_h\|_1$	order	$\|\phi \quad \phi_h\|_2$	order	$\|\phi \quad \phi_h\|_\infty$	order
16^3	$7.122\times10^{\ 6}$	—	$5.283\times10^{\ 6}$	—	$8.013\times10^{\ 6}$	—
32^3	$9.772\times10^{\ 8}$	6.19	$6.512\times10^{\ 8}$	6.34	$8.995\times10^{\ 8}$	6.48
64^3	$2.761\times10^{\ 9}$	5.15	$1.684\times10^{\ 9}$	5.27	$2.252\times10^{\ 9}$	5.32
128^3	$4.384\times10^{\ 11}$	5.98	$2.659\times10^{\ 11}$	5.98	$3.566\times10^{\ 11}$	5.98

Table 5.1. Accuracy results for the reinitialized level set function ϕ for example in Sect.5.1 after 1000 iterations.

	$\|K_M \quad (K_M)_h\|_1$	order	$\|K_M \quad (K_M)_h\|_2$	order	$\|K_M \quad (K_M)_h\|_\infty$	order
16^3	$7.100\times 10^{\;\;3}$	——	$3.754\times 10^{\;\;3}$	——	$3.139\times 10^{\;\;3}$	——
32^3	$5.066\times 10^{\;\;4}$	3.81	$3.183\times 10^{\;\;4}$	3.56	$3.782\times 10^{\;\;4}$	3.05
64^3	$2.432\times 10^{\;\;5}$	4.38	$1.509\times 10^{\;\;5}$	4.40	$2.583\times 10^{\;\;5}$	3.87
128^3	$1.510\times 10^{\;\;6}$	4.01	$9.700\times 10^{\;\;7}$	3.96	$1.633\times 10^{\;\;6}$	3.98

Table 5.2. Accuracy results for the computation of interface mean curvature K_M.

	$\|K \quad K_h\|_1$	order	$\|K \quad K_h\|_2$	order	$\|K \quad K_h\|_\infty$	order
16^3	$2.176\times 10^{\;\;2}$	——	$1.285\times 10^{\;\;2}$	——	$1.103\times 10^{\;\;2}$	——
32^3	$1.141\times 10^{\;\;3}$	4.25	$8.449\times 10^{\;\;4}$	3.93	$1.142\times 10^{\;\;3}$	3.27
64^3	$5.559\times 10^{\;\;5}$	4.36	$3.722\times 10^{\;\;5}$	4.50	$7.691\times 10^{\;\;5}$	3.89
128^3	$3.433\times 10^{\;\;6}$	4.02	$2.451\times 10^{\;\;6}$	3.92	$4.902\times 10^{\;\;6}$	3.97

Table 5.3. Accuracy results for the computation of interface Gaussian curvature K.

	L_1-Error	order	L_2-Error	order	L_∞-Error	order
16^3	$9.575\times 10^{\;\;1}$	——	$5.311\times 10^{\;\;1}$	——	$6.307\times 10^{\;\;1}$	——
32^3	$1.675\times 10^{\;\;1}$	2.52	$8.952\times 10^{\;\;2}$	2.57	$8.912\times 10^{\;\;2}$	2.82
64^3	$4.513\times 10^{\;\;2}$	1.89	$2.898\times 10^{\;\;2}$	1.63	$5.020\times 10^{\;\;2}$	0.82
128^3	$9.914\times 10^{\;\;3}$	2.19	$6.825\times 10^{\;\;3}$	2.09	$1.341\times 10^{\;\;2}$	1.90

Table 5.4. Accuracy results for the computation of the surface Laplacian of the mean curvature $\Delta_{||}K_M$. K_M field is extended before used to compute $\Delta_{||}K_M$. The error is computed as $\|\Delta_{||}K_M \quad (\Delta_{||}K_M)_h\|$.

	L_1-Error	order	L_2-Error	order	L_∞-Error	order
16^3	3.197×10^{0}	——	1.949×10^{0}	——	1.916×10^{0}	——
32^3	3.382×10^{0}	0.08	1.965×10^{0}	0.01	1.853×10^{0}	0.05
64^3	5.763×10^{0}	0.77	5.895×10^{0}	1.58	1.773×10^{1}	3.26
128^3	5.609×10^{0}	0.04	5.574×10^{0}	0.08	2.997×10^{1}	0.76

Table 5.5. Accuracy results for the computation of surface Laplacian of the mean curvature using fourth-order accurate scheme from [du2008second]. The error is computed as $\|\Delta_{||}K_M \quad (\Delta_{||}K_M)_h\|$.

As another more complicated example, we consider the red blood cell shape [guckenberger2016bending] given by the zero level set of

$$\phi(x,y,z)=\left(\frac{2z}{R}\right)^2 \quad \left(1 \quad \frac{x^2+y^2}{R^2}\right)\left(C_0+C_1\frac{x^2+y^2}{R^2}+C_2\frac{(x^2+y^2)^2}{R^4}\right)^2 \tag{5.1}$$

where $C_0=0.2072, C_1=2.0026, C_2= \quad 1.1228$ and R is the length of the large half-axis of the RBC and is taken to be 1 here. Note that this function is not an exact signed distance function. The zero level set for this ϕ is shown in Figure 5.1. After initializing ϕ according to Eq. (5.1), we reinitialize it to be a signed distance function and calculate the mean curvature K_M, Gaussian curvature K and $\Delta_{||}K_M$ on the surface. On this more realistic shape, our scheme still produces fourth order accuracy for the computation of interfacial mean and Gaussian curvatures (Tables 5.6 and 5.7) and can compute $\Delta_{||}K_M$ with second order accuracy in both the L_2 norm and maximum norm (Table 5.8 and Table 5.9), whereas the scheme from [du2008second] does not provide convergence for the computation of $\Delta_{||}K_M$. It is also important to note that schemes that triangulate the surface are not convergent for the computation of $\Delta_{||}K_M$ in the maximum norm [guckenberger2016bending]. As previously mentioned, the surface Laplacian of the interface mean curvature is important in the force density of an elastic surface. Many of the existing methods will produce large errors when computing this force. Therefore, high order schemes, such as the one proposed here, are necessary for these problems. We note that a fourth order scheme for the computation of curvature in the level set framework in two dimension has been proposed in [coquerelle2016fourth]. This approach is based on the osculatory circle approximation, which is difficult to implement in three dimensions and fails whenever the curvature is zero. Our approach, however, is straightforward to implement in two or three dimensions, and simple formulae for mean and Gaussian curvatures can be used [goldman2005curvature]. Thus our sixth order scheme provides a significant improvement that can enable accurate simulations of three-dimensional soft objects, such as vesicles and biological cells [guckenberger2016bending].

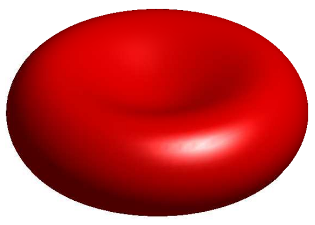

Figure 5.1. The red blood cell shape defined by the level set function Eq. (5.1).

	L_1-Error	order	L_2-Error	order	L_∞-Error	order
16^3	1.678×10^{0}	—	$7.923\times10^{\ 1}$	—	$8.740\times10^{\ 1}$	—
32^3	$2.856\times10^{\ 1}$	2.55	$1.490\times10^{\ 1}$	2.41	$2.511\times10^{\ 1}$	1.80
64^3	$1.998\times10^{\ 2}$	3.84	$1.691\times10^{\ 2}$	3.14	$4.728\times10^{\ 2}$	2.41
128^3	$1.176\times10^{\ 3}$	4.09	$8.532\times10^{\ 4}$	4.31	$3.506\times10^{\ 3}$	3.75

Table 5.6. Accuracy results for the computation of interface mean curvature for the red blood cell shape using our sixth order accurate scheme, where the error is defined as $\|K_M \quad (K_M)_h\|$.

	$\|K \quad K_h\|_1$	order	$\|K \quad K_h\|_2$	order	$\|K \quad K_h\|_\infty$	order
16^3	2.657×10^{0}	—	1.266×10^{0}	—	1.296×10^{0}	—
32^3	$7.583\times10^{\ 1}$	1.81	$5.456\times10^{\ 1}$	1.21	$9.430\times10^{\ 1}$	0.46
64^3	$7.225\times10^{\ 2}$	3.39	$8.023\times10^{\ 2}$	2.77	$2.296\times10^{\ 1}$	2.04
128^3	$3.924\times10^{\ 3}$	4.20	$3.519\times10^{\ 3}$	4.51	$1.777\times10^{\ 2}$	3.69

Table 5.7. Accuracy results for the computation of interface Gaussian curvature for the red blood cell shape using our sixth order accurate scheme.

	L_1-Error	order	L_2-Error	order	L_∞-Error	order
16^3	2.549×10^{2}	—	1.094×10^{2}	—	9.351×10^{2}	—
32^3	1.680×10^{2}	0.60	7.791×10^{1}	0.49	8.049×10^{1}	0.22
64^3	2.884×10^{1}	2.54	2.133×10^{1}	1.87	6.434×10^{1}	0.32
128^3	4.610×10^{0}	2.65	3.449×10^{0}	2.63	1.297×10^{1}	2.31

Table 5.8. Accuracy results for the computation of surface Laplacian of the interface mean curvature for the red blood cell shape using our sixth order accurate scheme. The error is computed as $\|\Delta_{||}K_M \quad (\Delta_{||}K_M)_h\|$.

	L_1-Error	order	L_2-Error	order	L_∞-Error	order
16^3	2.672×10^{2}	—	1.156×10^{2}	—	9.802×10^{1}	—
32^3	2.203×10^{2}	0.28	1.096×10^{2}	0.08	1.270×10^{2}	0.37
64^3	1.333×10^{2}	0.72	7.319×10^{1}	0.58	1.131×10^{2}	0.17
128^3	1.338×10^{2}	0.01	7.785×10^{1}	0.09	1.296×10^{2}	0.20

Table 5.9. Accuracy results for the computation of surface Laplacian of the interface mean curvature for the red blood cell shape using fourth order accurate scheme from [du2008second]. The error is computed as $\|\Delta_{||}K_M \quad (\Delta_{||}K_M)_h\|$.

As a final test, we consider a surface with a kink. We initialize our level set function $\phi(x,y,z)$ to be two merging spheres:

$$\phi(x,y,z)=\min\left((x^2+y^2+(z \quad z_0)^2 \quad r^2),(x^2+y^2+(z+z_0)^2 \quad r^2)\right), z_0=0.3, r=0.6.$$

Then we reinitialize $\phi(x,y,z)$ to be a signed distance map on different grids and plot $\log_{10}|\phi_{\text{exact}} \quad \phi_{\text{numerical}}|$ on the surface, where ϕ_{exact} is the exact signed distance map and $\phi_{\text{numerical}}$ is the reinitialized ϕ. As is shown in figure 5.2, near the kink, accuracy is greatly affected, but the rate of convergence is not affected away from the kink. Moreover, as finer grids are used, the affected region shrinks, as the inaccuracy arises due to not having a sufficient number of nodes to properly resolve the distance map with our interpolation scheme. As the grid spacing decreases, it is possible to resolve a larger fraction of the shape away from the kink. Figure 5.3 demonstrates the deterioration of accuracy for two approaching spheres. In the left graph of Figure 5.3, the maximum error $||\phi \quad \phi_h||_\infty$ is plotted as a function of the distance $d=2(z_0 \quad r)$ between the two spheres on different grids. A visible jump in error can be seen as the two spheres approach each other. The right graph of Figure 5.3 shows that the order of accuracy for $||\phi \quad \phi_h||_\infty$ decreases to first order, which is expected in the presence of kinks [min2010reinitializing]. Adaptive mesh refinement or finite element methods [lipnikov2019high] might help in the presence of kinks, but this is beyond the scope of this thesis.

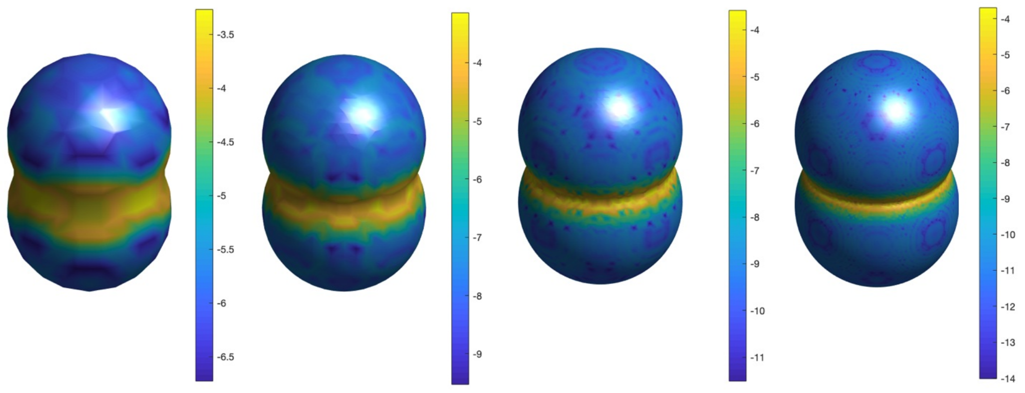

Figure 5.2. Distribution of numerical error $\log_{10}|\phi \quad \phi_h|$ of the level set function on different grids. Grid size from left to right: $16 \times 16 \times 32, 32 \times 32 \times 64, 64 \times 64 \times 128, 128 \times 128 \times 256$.

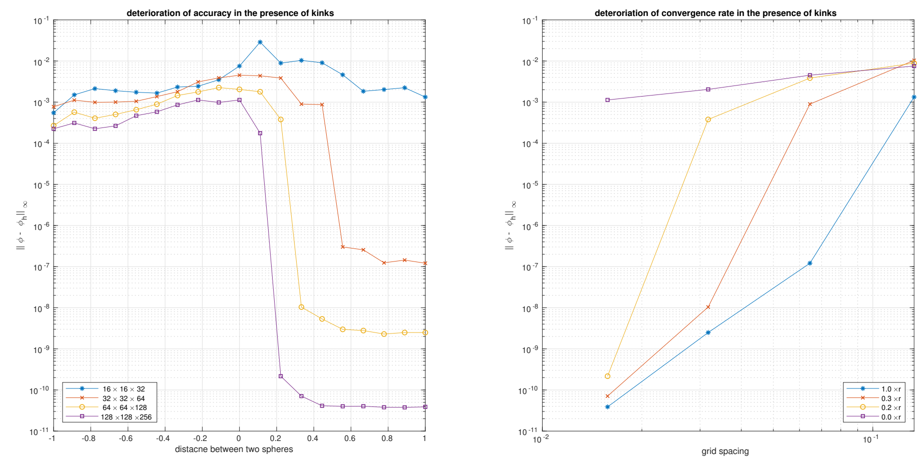

Figure 5.3. Left graph: $\|\phi \quad \phi_h\|_\infty$ and distance $d = 2z_0 \quad 2r$ for different grids. Convergence of $\|\phi \quad \phi_h\|_\infty$ for different distances $d = r, 0.3r, 0.2r, 0$.

5.2 Extrapolation of Surface Scalar Fields in 3D

Extrapolation of fields living on a surface to the embedding space, i.e. solving Eq. (4.3), is of great importance in many applications. In the level set method, it is common practice to extend velocity fields defined only on the surface away from the interface in the normal direction. When solving PDEs on implicit surfaces represented by level set functions [xu2003eulerian, adalsteinsson2003transport], dynamical fields have to be extrapolated to embed those PDEs in space. When those embedded PDEs are of high orders, it becomes crucial to accurately extrapolate those surface fields [greer2006fourth]. However, this extrapolation is seldom done in an accurate way because boundary conditions on the implicit interface are seldom treated appropriately. Often the sign function is smoothed out in some fashion [xu2003eulerian, greer2006fourth], but this regularization is not enough to prevent information from flowing across the interface, affecting the accuracy of the scheme globally. Note that in [greer2006fourth], the author shows that by using $\mathrm{Sign}(\phi)=\phi/\sqrt{\phi^2+h^{1/2}}$ and the WENO scheme relatively small errors are obtained on a 400^2 grid. However, they did not show the convergence of their extrapolation scheme.

Here we test our ability to accurately extend fields away from a given surface. We consider a surface represented by the signed distance function $\phi(x,y,z)=\sqrt{x^2+y^2+z^2}\quad 0.5$ on a $[\ \ 1,1]^3$ domain. Let a surface field ψ be the z coordinate of the position vector. We initialize ψ to be z at all points in space, and then solve the extension velocity equation (Eq. (4.63) with the Hamiltonian defined in Eq. (4.68)). Figure 5.4 shows isosurfaces of ψ before and after extrapolation. The extrapolated surface field is determined to sixth-order accuracy (Table 5.10). Likewise, the Laplacian of ψ is fourth-order accurate (Table 5.11) and the Bilaplacian of the ψ is accurate to second-order (Table 5.12). If, instead, the ENO scheme from [du2008second] is used, the computation of $\Delta^2\psi$ does not converge (Table 5.13), which clearly shows the necessity of using a high order scheme to extrapolate any surface field whose dynamics depends on differential operators such as the Laplacian and the bilaplacian. Indeed, the scheme presented here will be useful for a wide range of applications involving PDEs on curved surfaces.

	$\|\psi \quad \psi_h\|_1$	order	$\|\psi \quad \psi_h\|_2$	order	$\|\psi \quad \psi_h\|_\infty$	order
16^3	$1.794\times10^{\ \ 4}$	—	$1.138\times10^{\ \ 4}$	—	$1.324\times10^{\ \ 4}$	—
32^3	$4.843\times10^{\ \ 6}$	5.21	$4.014\times10^{\ \ 6}$	4.82	$6.734\times10^{\ \ 6}$	4.30
64^3	$3.517\times10^{\ \ 8}$	7.11	$3.341\times10^{\ \ 8}$	6.91	$2.074\times10^{\ \ 7}$	5.02
128^3	$5.611\times10^{\ \ 10}$	5.97	$5.345\times10^{\ \ 10}$	5.97	$2.932\times10^{\ \ 9}$	6.14

Table 5.10. Accuracy results for the extended surface field.

	$\|\Delta\psi \quad (\Delta\psi)_h\|_1$	order	$\|\Delta\psi \quad (\Delta\psi)_h\|_2$	order	$\|\Delta\psi \quad (\Delta\psi)_h\|_\infty$	order
16^3	$2.482\times10^{\ \ 2}$	—	$1.605\times10^{\ \ 1}$	—	$1.539\times10^{\ \ 1}$	—
32^3	$1.423\times10^{\ \ 3}$	4.12	$1.152\times10^{\ \ 2}$	3.80	$2.231\times10^{\ \ 2}$	2.73
64^3	$8.935\times10^{\ \ 4}$	3.99	$7.101\times10^{\ \ 4}$	4.02	$1.917\times10^{\ \ 3}$	3.60
128^3	$5.617\times10^{\ \ 5}$	3.99	$4.644\times10^{\ \ 5}$	3.93	$1.245\times10^{\ \ 4}$	3.94

Table 5.11. Accuracy results for the Laplacian of the extended surface field.

	$\|\Delta^2\psi \quad (\Delta^2\psi)_h\|_1$	order	$\|\Delta^2\psi \quad (\Delta^2\psi)_h\|_2$	order	$\|\Delta^2\psi \quad (\Delta^2\psi)_h\|_\infty$	order
16^3	3.511×10^{1}	—	2.369×10^{1}	—	2.675×10^{1}	—
32^3	1.320×10^{1}	1.41	1.035×10^{1}	1.19	2.236×10^{1}	0.26
64^3	1.810×10^{0}	2.87	1.612×10^{0}	2.68	4.035×10^{0}	2.47
128^3	$2.985\times10^{\ \ 1}$	2.60	$3.226\times10^{\ \ 1}$	2.32	1.288×10^{0}	1.65

Table 5.12. Accuracy results for the BiLaplacian of the extended surface field.

	$\|\Delta^2\psi \quad (\Delta^2\psi)_h\|_1$	order	$\|\Delta^2\psi \quad (\Delta^2\psi)_h\|_2$	order	$\|\Delta^2\psi \quad (\Delta^2\psi)_h\|_\infty$	order
16^3	3.627×10^{1}	—	2.390×10^{1}	—	2.765×10^{1}	—
32^3	1.694×10^{1}	1.10	1.671×10^{1}	0.52	4.017×10^{1}	0.54
64^3	1.744×10^{1}	0.04	1.634×10^{1}	0.03	4.005×10^{1}	0.00
128^3	1.449×10^{1}	0.27	1.675×10^{1}	0.04	6.863×10^{1}	0.78

Table 5.13. Accuracy results for the BiLaplacian of the extended surface field if ENO scheme from [du2008second] is used to treat boundary terms.

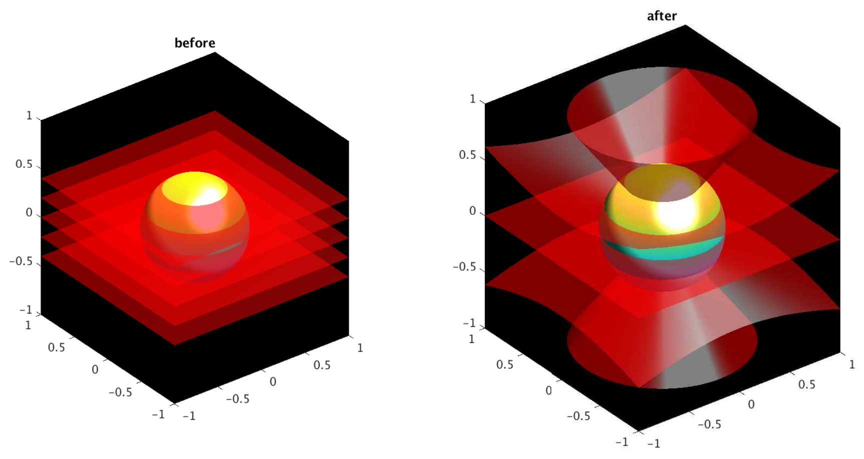

Figure 5.4. The $(0, \pm 0.2, \pm 0.4)$ isosurfaces of ψ before and after extrapolation.

5.3 Computation of Geodesic Distances on an Implicit Surface

Computation of geodesics is of interest in many applications [crane2013geodesics, cheng2002motion]. A geodesic, though, is just a distance map defined on a surface. Solving for geodesics can therefore be regarded as a generalization of the reinitialization process to a non-Euclidean space. For example, consider a curve represented by the intersection of two level set functions ϕ and ψ, where the zero level set of ϕ represents the surface confining motion of the curve [cheng2002motion]. To compute the signed distance function on the surface for this curve, we need to iterate to equilibrium the following PDE on the surface [cheng2002motion]

$$\frac{\partial \psi}{\partial \tau} + \mathrm{Sign}(\psi^0)(|\nabla_{||}\psi| \quad 1) = 0 \tag{5.2}$$

where $\nabla_{||}$ is the differential operator defined on the surface. Eq. (5.2) is still a Hamilton-Jacobi equation. Due to the surface differential operator it is no longer feasible to use Godunov's scheme to construct the numerical Hamiltonian. In [cheng2002motion], Local Lax-Friedrichs (LLF) schemes were used instead, and the author claims to find first-order accuracy in one dimension due to motion of the curve position during the iteration process. There are two sources of numerical error. First, the LLF scheme is more dissipative than Godunov's scheme and will perturb the curve position. Second, the treatment of the boundary condition is not appropriately implemented, and information flows across curve. There is no easy way to remedy these problems, as far as we know.

Another way to solve Eq. (5.2) is to replace the surface differential operator $\nabla_{||}$ by the 3D Cartesian differential operator ∇, using a closest point representation of ψ [macdonald2008level]. The closest point representation can be obtained by constraining the level sets of the distance map that defines the curve, ψ, to be perpendicular to the zero level set of ϕ. That is, we want to solve the 3D reinitialization equation for ψ, subject to the condition that $\nabla\phi \cdot \nabla\psi = 0$, which is the same condition as for extending a field away from the surface. Therefore, we can simultaneously solve the extension equation (Eq. (4.3)) and the reinitialization equation (Eq. (4.2)) in order to determine the geodesic distance map ψ. We do this by iterating the extension equation (Eq. (4.3)) before each stage of the Runge-Kutta scheme in the reinitialization time stepping. In this way, we can use the standard reinitialization equation (Eq. (4.2)) in place of the more complicated non-Euclidean reinitialization equation (Eq. (5.2)).

Consider a surface represented by the signed distance function $\phi(x,y,z)=\sqrt{x^2+y^2+z^2}\quad 0.6$ on a $[\quad 1,1]^3$ domain. Let a closed curve   on this surface be the intersection of the zero level set of ϕ and the zero of the function $\psi(x,y,z)=e^{2z}\quad 1$. We then reinitialize ψ using the procedure just described. Figure 5.5 shows the level curves of ψ before and after this redistancing. As shown in Table 5.14, our method provides sixth order accuracy for the computation of the geodesic distance between   and other points on the surface. In addition, we are able to calculate the geodesic curvature k_g of the level curves of ψ with fourth order accuracy (Table 5.15). Note that the order of accuracy for ψ seems to degenerate to fifth order as grids are refined (Table 5.14). This decrease in accuracy might result from accumulation of numerical errors from the extrapolation of ψ during each reinitialization step. This will be a subject of future research.

	$\|\psi\quad\psi_h\|_1$	order	$\|\psi\quad\psi_h\|_2$	order	$\|\psi\quad\psi_h\|_\infty$	order
16^3	$3.166\times 10^{\ 3}$	—	$2.221\times 10^{\ 3}$	—	$3.320\times 10^{\ 3}$	—
32^3	$4.246\times 10^{\ 5}$	6.02	$3.327\times 10^{\ 5}$	6.06	$5.229\times 10^{\ 5}$	5.99
64^3	$1.310\times 10^{\ 6}$	5.02	$9.151\times 10^{\ 7}$	5.18	$1.448\times 10^{\ 6}$	5.17
128^3	$4.312\times 10^{\ 8}$	4.93	$2.928\times 10^{\ 8}$	4.97	$6.897\times 10^{\ 8}$	4.39

Table 5.14. Accuracy results for the signed distance function on a curved surface.

	$\|k_g\quad(k_g)_h\|_1$	order	$\|k_g\quad(k_g)_h\|_2$	order	$\|k_g\quad(k_g)_h\|_\infty$	order
16^3	$2.772\times 10^{\ 1}$	—	$1.851\times 10^{\ 1}$	—	$2.147\times 10^{\ 1}$	—
32^3	$2.723\times 10^{\ 2}$	3.35	$1.987\times 10^{\ 2}$	3.22	$2.655\times 10^{\ 2}$	3.02
64^3	$8.437\times 10^{\ 4}$	5.01	$9.341\times 10^{\ 4}$	4.41	$3.100\times 10^{\ 3}$	3.10
128^3	$3.035\times 10^{\ 5}$	4.80	$4.190\times 10^{\ 5}$	4.48	$3.867\times 10^{\ 4}$	3.00

Table 5.15. Accuracy results for the geodesic curvature for the level sets of ψ.

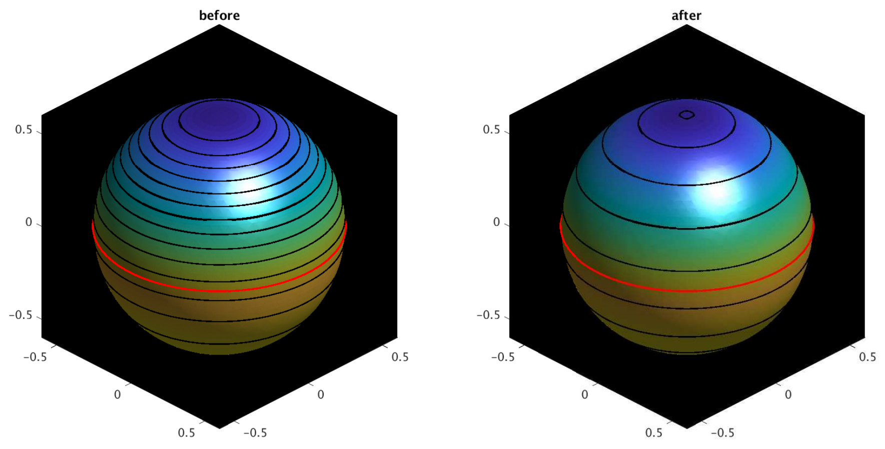

Figure 5.5. Level curves of ψ before and after redistancing. The red curve is the zero level curve of ψ. The spacing between neighboring black curves is $\frac{20}{99}$.

As a more stringent test, consider another example where the surface is still given by the signed distance function $\phi(x,y,z)=\sqrt{x^2+y^2+z^2}\quad 0.6$ on a $[\quad 1,1]^3$ domain, but the curve   is the intersection of the zero level set of ϕ and the zero level set of $\psi(x,y,z) = \exp\left[1.2\left(\operatorname{asin}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right) + \frac{\pi}{12}\right)\right]$. Figure 5.6 shows the level curves of ψ before and after redistancing. Tables 5.16 and 5.17 show the results of our convergence tests for the geodesics and the geodesic curvature, respectively. It seems that the order of convergence degrades as the grid is refined. The convergence rate, however, is still better than that in [cheng2002motion]. This loss of convergence rate is beyond the scope of current work and requires further investigation of our implementation of the closest point method [macdonald2008level]. Still ,this scheme can be useful in applications dealing with the motion of curves embedded in a surface driven by geodesic curvatures [cheng2002motion].

	$\|\psi \quad \psi_h\|_1$	order	$\|\psi \quad \psi_h\|_2$	order	$\|\psi \quad \psi_h\|_\infty$	order
16^3	$9.154\times 10^{\ 4}$	—	$7.693\times 10^{\ 4}$	—	$2.620\times 10^{\ 3}$	—
32^3	$8.714\times 10^{\ 5}$	3.39	$7.777\times 10^{\ 5}$	3.31	$1.400\times 10^{\ 4}$	4.23
64^3	$8.170\times 10^{\ 6}$	3.41	$7.002\times 10^{\ 6}$	3.47	$1.415\times 10^{\ 5}$	3.31
128^3	$2.549\times 10^{\ 6}$	1.68	$2.226\times 10^{\ 6}$	1.65	$4.114\times 10^{\ 6}$	1.78

Table 5.16. Accuracy results for the signed distance function on a curved surface.

	$\|k_g \quad (k_g)_h\|_1$	order	$\|k_g \quad (k_g)_h\|_2$	order	$\|k_g \quad (k_g)_h\|_\infty$	order
16^3	$1.337\times 10^{\ 1}$	—	$1.233\times 10^{\ 1}$	—	$1.885\times 10^{\ 1}$	—
32^3	$2.500\times 10^{\ 2}$	2.42	$2.089\times 10^{\ 2}$	2.56	$3.548\times 10^{\ 2}$	2.41
64^3	$1.217\times 10^{\ 2}$	1.04	$1.195\times 10^{\ 2}$	0.81	$2.617\times 10^{\ 2}$	0.44
128^3	$8.174\times 10^{\ 3}$	0.57	$7.740\times 10^{\ 3}$	0.64	$2.245\times 10^{\ 2}$	0.22

Table 5.17. Accuracy results for the geodesic curvature for the level sets of ψ.

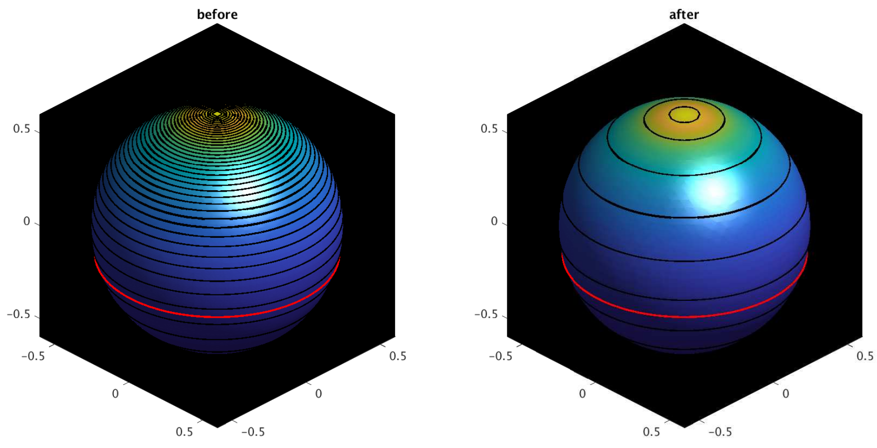

Figure 5.6. Level curves of ψ before and after redistancing. The red curve is the zero level curve of ψ. The spacing between neighboring black curves is $\frac{20}{99}$.

As a final test, we compute geodesic distances on the Stanford bunny [stanfordbunny] (Figure 5.7). The level set representation of the bunny is obtained by the radial function method [fasshauer2007meshfree] on a 96^3 grid and is then smoothed by diffusion. The function $\psi(x,y,z) = \exp(x^2+y^2+(z \quad 0.05)^2) \quad \exp(0.0009)$ is used to initialize curve position on the bunny. This example shows that we can compute geodesic distances on highly curved surfaces.

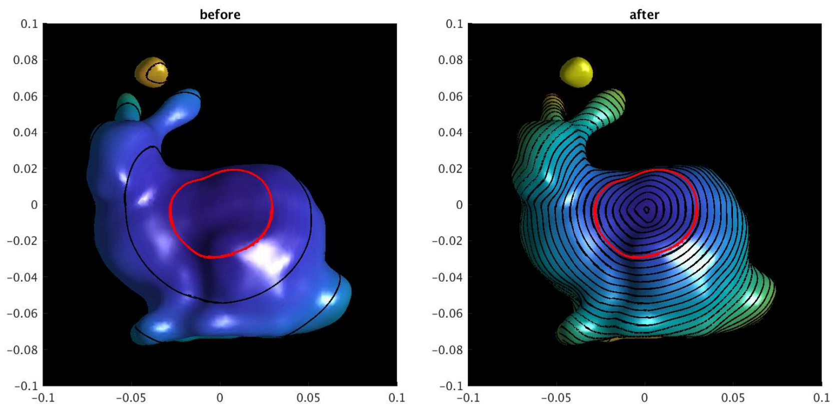

Figure 5.7. Geodesic distances computed on the Stanford bunny.The left panel shows the equipotential intersections of ψ before redistancing, and the right panel shows the geodesics computed by our redistancing procedure. The red curve is the intersection of the zero level set of ψ with the bunny whose position is preserved during the redistancing procedure. The spacing between neighborhood black curves is $\frac{1}{245}$.

5.4 Conclusion

In many applications of science and engineering, it is important to accurately compute surface curvature and its derivatives in three dimensional space. In problems involving elastic surfaces, second derivatives of surface curvatures such as $\Delta_{\|}K_M$ are needed. Many widely used schemes are unable to compute this geometric quantity accurately. For instance, all schemes reviewed in [guckenberger2016bending] are based on surface triangulation and fail to compute $\Delta_{\|}K_M$ convergently. Other level set based schemes such as those proposed in [coquerelle2016fourth] are difficult to generalize to three dimensional cases. We approached this problem in the level set framework by maintaining accuracy of the level set function in the reinitialization process.

In particular, we have presented a sixth-order accurate numerical scheme for Hamilton-Jacobi equations with a level set-defined boundary condition. This work builds on the work of Chéné and Min [du2008second]. We showed that our method can solve the reinitialization equation of Sussman [sussman1994level] and the extrapolation equation (4.3) to sixth order accuracy in the L_1, L_2, and L_∞ norms for smooth surfaces. Our numerical experiments also show that this method leads to an interface curvature that is accurate to fourth order, which results in second order accuracy of bending forces calculated for elastic surfaces. These schemes thus make possible an accurate simulation of dynamical elastic surfaces in the level set framework and of other applications in physics and engineering that require accurate computation of interface curvature. Also, a sixth order accurate extrapolation scheme for surface fields is proposed, which allows a very accurate closest point representation [macdonald2008level] of surface fields defined near the interface, which can be a crucial component for embedding high order PDEs on a non-Euclidean surface in space [greer2006fourth] and in solving the level set equation [zhao1996variational]. Finally, we presented a convergent method for computing geodesics on an implicit surface. The combination of techniques presented here can greatly improve the accuracy of simulations involving elastic surfaces and also makes possible the simulation of the dynamics of biomembranes coupled to concentrations of interacting, surface-bound proteins, as occurs in processes such as endo- and exocytosis, cell division, and cell motility.

Chapter 6
Numerical Experiments: Shape of Vesicles

In this chapter, we apply results from the theoretical calculations in Chapter 3 and the level set discretization in Chapter 4 to the numerical simulation of the dynamics of vesicles with different models.

6.1 Shape of Single Phase Vesicles

We first investigate shapes and dynamics of single phase vesicles as a test of the convergence and accuracy of our algorithm. Here we adopt the area-difference elasticity model [miao1994budding] for single phase vesicles where the effect of a relative areal stretching of the two monolayers are taken into account. The Hamiltonian for a closed fluid-bilayer vesicle can be written as

$$H=\int\frac{\kappa}{2}(K_M\quad C)^2\mathrm{d}A+\sigma\left(\int\mathrm{d}A\quad A_p\right)\quad P\left(\int\frac{\boldsymbol{R}\cdot\boldsymbol{N}}{3}\mathrm{d}A\quad V_p\right)+\sigma_{\Delta A}\left(h\int K_M\mathrm{d}A\quad \Delta A_p\right)\tag{6.1}$$

where the first integral accounts for the Helfrich bending energy, the second term and the third term constrains total area and total volume of the vesicle, the last term constrains the area difference between the two layers, C is the constant spontaneous curvature, κ is a constant bending moduli, A_p is the prescribed area, V_p is the prescribed volume, ΔA_p is the prescribed area difference, h is the thickness of the bilayer and $\sigma, P, \sigma_{\Delta A}$ are Lagrange multipliers. Under a variation of the position of the vesicle $\boldsymbol{R}\to\boldsymbol{R}+\delta\boldsymbol{R}$, the variation of the Hamiltonian Eq. (6.1) is

$$\delta H=\int\mathrm{d}A\left\{\kappa\left(\Delta_{\|}K_M+\frac{1}{2}K_M^3\quad 2K_MK+2CK\quad \frac{1}{2}C^2K_M\right)\quad \sigma K_M\quad P\quad 2\sigma_{\Delta A}hK\right\}\boldsymbol{N}\cdot\delta\boldsymbol{R},\tag{6.2}$$

which can be easily obtained from results in Chapter 3. The elastic bending force density from Eq. (6.1) is thus

$$\boldsymbol{f}=\quad\left\{\kappa\left(\Delta_{\|}K_M+\frac{1}{2}K_M^3\quad 2K_MK+2CK\quad \frac{1}{2}C^2K_M\right)\quad \sigma K_M\quad P\quad 2\sigma_{\Delta A}hK\right\}\boldsymbol{N}.\tag{6.3}$$

The velocity $\boldsymbol{v}$ of the vesicle is obtained from the fact that, the elastic bending force $\boldsymbol{f}$ will be balanced by the viscous force $\zeta\boldsymbol{v}$ in an environment of low Reynolds number, i.e.

$$\boldsymbol{f}=\zeta\boldsymbol{v},\tag{6.4}$$

where ζ will be taken to be one for simplicity. When the vesicle deforms with a velocity field $\boldsymbol{v}$, the time rate change of the the total volume $V=\frac{1}{3}\int\boldsymbol{R}\cdot\boldsymbol{N}\mathrm{d}A$, total area $A=\int\mathrm{d}A$ and the area difference $\Delta A=h\int K_M\mathrm{d}A$ are

$$\frac{\mathrm{d}V}{\mathrm{d}t}=\int\boldsymbol{v}\cdot\boldsymbol{N}\mathrm{d}A\tag{6.5}$$

$$\frac{\mathrm{d}A}{\mathrm{d}t}=\int(\quad K_M)(\boldsymbol{v}\cdot\boldsymbol{N})\mathrm{d}A\tag{6.6}$$

$$\frac{\mathrm{d}(\Delta A)}{\mathrm{d}t}=h\int(\quad 2K)(\boldsymbol{v}\cdot\boldsymbol{N})\mathrm{d}A\tag{6.7}$$

Eqs. (6.5-6.7) is a system of linear equations for $\sigma, P, \sigma_{\Delta A}$. By specifying the time rate of change for $V, A, \Delta A$, the Lagrange multipliers can be easily solved to either preserve volume, area and area difference or to evolve those parameters in some prescribed manner [du2006simulating]. This will facilitate our study of the effects of volume and area difference on the shape and dynamics of vesicles. The dimensionless version of volume and area difference is defined by

$$v=\frac{V}{\frac{4}{3}\pi R_s^3},\ \Delta a=\frac{\Delta A}{8\pi hR_s},\ R_s=\sqrt{\frac{A}{4\pi}}\tag{6.8}$$

where v is known as the *reduced volume*, Δa is known as the *reduced area difference*, $8\pi h R_s$ is the area difference of a spherical bilayer of radius R_s and thickness h.

The dynamics of the vesicle is then captured by the level set equation

$$\frac{\partial \phi}{\partial t} + \boldsymbol{v} \cdot \nabla \phi = 0. \tag{6.9}$$

Since $\boldsymbol{v}$ involves fourth order derivatives of ϕ, Eq. (6.9) is very stiff, which allows a very small time step when explicit methods are used. Here we adopt the semi-implicit level set method from [smereka2003semi] to advance Eq. (6.9). With this semi-implicit approach, Eq. (6.9) is discretized in time as:

$$\frac{\phi^{n+1} \quad \phi^n}{\Delta t} = \alpha\{(\Delta^2\phi)^n \quad (\Delta^2\phi)^{n+1}\} \quad (\boldsymbol{v} \cdot \nabla \phi)^n \tag{6.10}$$

which can be rewritten as

$$\phi^{n+1} = \phi^n \quad \Delta t(\mathbb{1} + \alpha \Delta t \Delta^2) \;\; ^1(\boldsymbol{v} \cdot \nabla \phi)^n, \tag{6.11}$$

where α is a positive constant (taken to be 0.5 in our simulation) and the inversion $(\mathbb{1} + \alpha \Delta t \Delta^2) \;\; ^1$ can be efficiently computed with FFT (Fast Fourier Transform) method. A more accurate inversion can be obtained with the GMRES (generalized minimal residual) method preconditioned by the FFT solver. The GMRES solver gives slightly better stability and accuracy, but the FFT method is much more computationally efficient. We will use the FFT solver in our simulation.

The algorithm to evolve shapes of the single phase vesicle is then as follows:

1. Initialize the level set function ϕ on a three dimensional cartesian grid. Usually ϕ is initialized to be an oblate or a prolate with a prescribed reduced volume and then reinitialized to be a signed distance map. The ellipsoid is usually taken to be cylindrically symmetric, the reduced volume of which is then a function of the ratio between the lengths of the two distinct semi-axis.
2. Repeat until maximum time or the maximum number of iterations is reached:
   i. Calculate K_M, K in all grid points and extend values of K_M, K away from the interface.
   ii. Calculate $\Delta_{\|} K_M$ in all grid points and extend its value away from the interface. Note that when K_M is constant along the surface $\boldsymbol{N}$, $\Delta_{\|} K_M = \Delta K_M$ and the cartesian spatial diffusion operator Δ can be used to compute the surface diffusion of K_M.
   iii. Calculate $\boldsymbol{v}$ from Eqs. (6.3-6.7) and extend its value away from the interface, which will maintain ϕ as a signed distance map [zhao1996variational].
   iv. Find the maximum $v_{\max}$ value of $|\boldsymbol{v}|$ near the interface and use $v_{\max}\Delta t = N_{\text{CFL}}\Delta s$ to determine time step Δt, where N_{CFL} is the CFL number for the current step and Δs is the grid spacing in each dimension. The CFL number is taken to be 0.1 for the sake of stability and accuracy although larger values can also be used.
   v. Update ϕ with Eq. (6.11).
   vi. Reinitialize ϕ to be a signed distance function if necessary.

6.1.1 Relaxation Under Constant Volume and Area without Spontaneous Curvature

We first show that our algorithm will reproduce the discocyte and gourd shape for axis-symmetric ellipsoids with zero spontaneous curvature. In this case, we set at each time step [du2006simulating]

$$C = 0, \frac{\mathrm{d}V}{\mathrm{d}t} = \frac{V_p \quad V(t)}{\Delta t}, \frac{\mathrm{d}A}{\mathrm{d}t} = \frac{A_p \quad A(t)}{\Delta t}, \sigma_{\Delta A} = 0, \tag{6.12}$$

where $V(t), A(t)$ is the current volume and current area of the vesicle, V_p and A_p is taken to be the initial volume and initial area of the ellipsoid. Requirement Eq. (6.12) combined with Eqs. (6.5-6.7) gives σ, P that will keep the the volume and area of the vesicle constant during the simulation. Figure 6.1 shows the relaxation of an oblate to a discocyte and the relaxation of a prolate to a gourd. The reduced volume v for all shapes in figure 6.1 is 0.6.

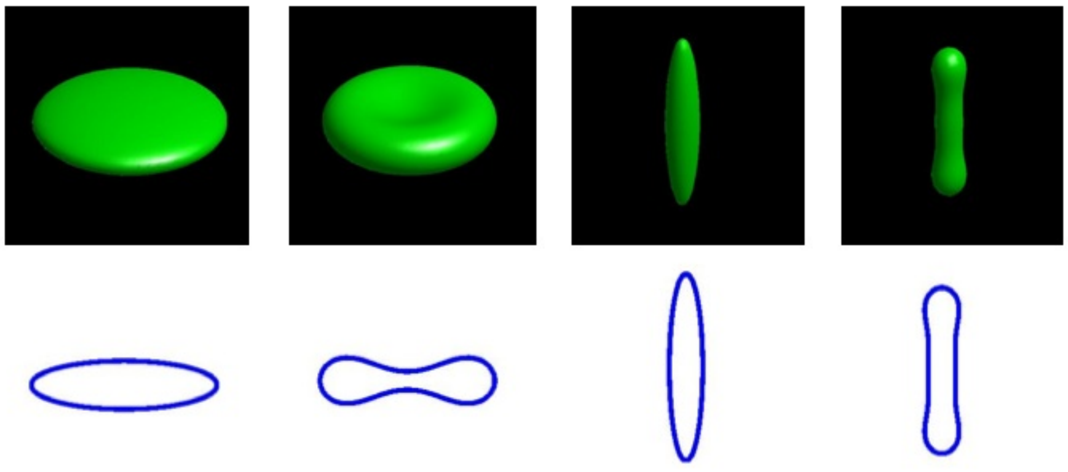

Figure 6.1. Relaxation of an oblate (top left, $v = 0.6$) to a discocyte (first row, second from left) and relaxation of a prolate (first row, third from left, $v = 0.6$) to a gourd (top right). The second row is a cross section view of the first row.

6.1.2 Convergence verification for spatial and time discretization

To verify convergence in space discretization, we relax an oblate with $v = 0.6$ on girds of different sizes and plot bending energy, Lagrange multipliers P and σ, the relative error in volume and area as a function of time. In figure 6.2, curves of different color (magenta, red, blue, green) correspond to different grid sizes $(48^3, 64^3, 80^3, 96^3)$. The top left plot in figure 6.2 shows that the shapes of the energy curves for different grids are qualitatively very similar and a zoomed in view shows that as the grid is refined, the energy curve will converge to the one with the finest grid. The top right plot and the bottom left plot in figure 6.2 shows convergence for the Lagrange multipliers. The bottom right plot demonstrates very good conservation of volume and area during simulation, where dashed lines are relative errors in volume and dotted dashed lines represent relative errors in area.

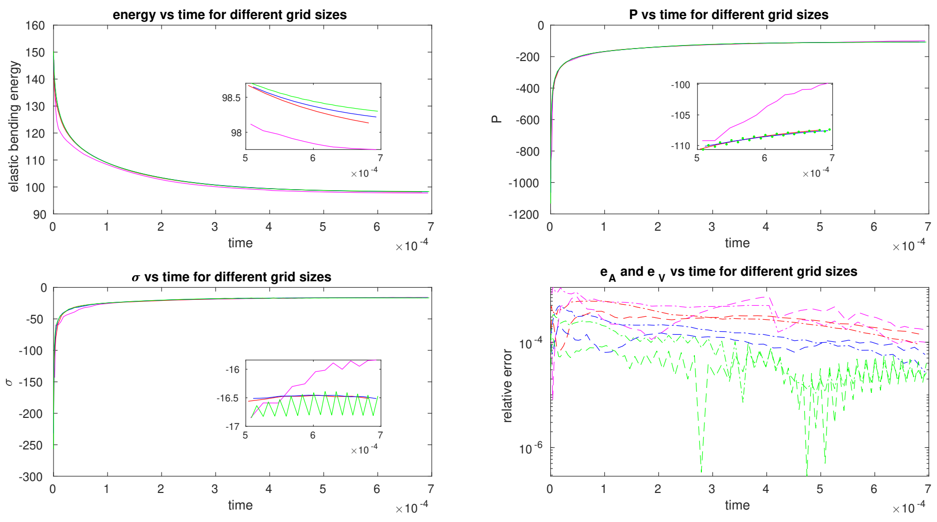

Figure 6.2. Top left: energy vs time for different grids. Grid sizes $(48^3, 64^3, 80^3, 96^3)$ are indicated by color of the curves (magenta, red, blue, green), which applies to other subplots as well. Top right and bottom left: convergence of Lagrange multipliers P, σ. Bottom left: relative errors in volume (dashed curves) and area (dotted dashed curves) vs time.

To verify convergence in time discretization, we relax an oblate with $v = 0.6$ on a 64^3 grid with different N_{CFL} since our adaptive Δt is computed from $\Delta t = N_{\mathrm{CFL}} \Delta s / v_{\max}$. In figure 6.3, curves of different color (magenta, red, blue) correspond to different values of N_{CFL} $(1, 0.5, 0.1)$. The top left plot in figure 6.3 shows that the shapes of the energy curves for different values of N_{CFL} are qualitatively very similar and a zoomed in view shows that as smaller values of N_{CFL} are used, oscillation will be suppressed and the energy curve will converge to the one with smallest N_{CFL}. The top right plot and the bottom left plot in figure 6.3 shows convergence for the Lagrange multipliers. The bottom right plot demonstrates very good conversation of volume and area during simulation, where dashed lines are relative errors in volume and dotted dashed lines represent relative errors in area.

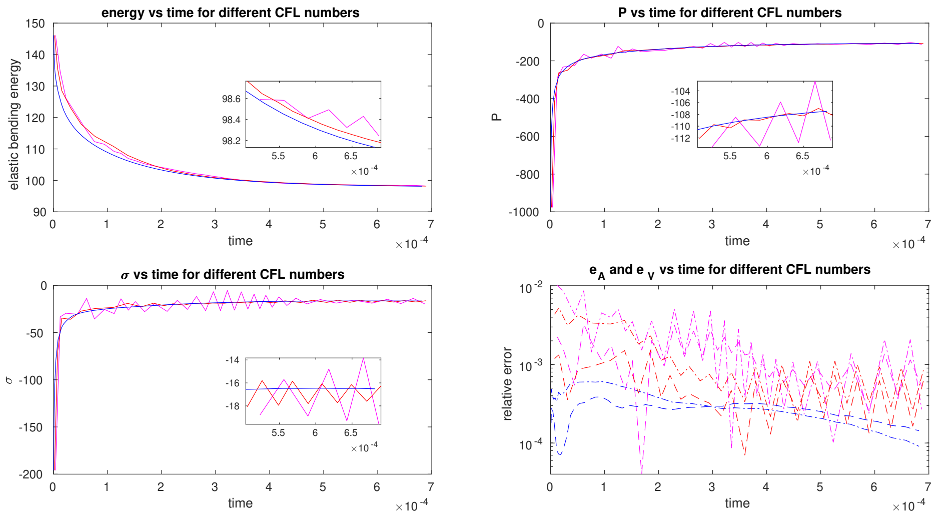

Figure 6.3. Top left: energy vs time for different grids. Different values of N_{CFL} $(1, 0.5, 0.1)$ are indicated by color of the curves (magenta, red, blue), which applies to other subplots as well. Top right and bottom left: convergence of Lagrange multipliers P, σ. Bottom left: relative errors in volume (dashed curves) and area (dotted dashed curves) vs time.

In summary, the semi-implicit scheme adopted from [smereka2003semi] still gives convergences when applied to more complicated scenarios, which ensures accuracy of the shape obtained in our simulation.

6.1.3 Effects of Constant Spontaneous Curvature

To explore effects of a constant spontaneous curvature, we repeat numerical experiments in section 6.1.1 with a non-zero C. Figure 6.4 shows the shape evolution of a vesicle whose initial shape is an oblate of with $v = 0.6$. The spontaneous curvature C is set to be 30 (note that in our convention the mean curvature of a unit sphere is 2). As can be seen from figure 6.4, a large negative spontaneous curvature will amplify small ripples (from numerical errors) and lead to growth of legs, which eventually results in pinching off of smaller vesicles. Figure 6.5 shows the shape dynamics of a vesicle whose initial shape is a prolate of reduced volume 0.6. Setting C to be 30 leads to the pinching of the prolate into three smaller vesicles. Figure 6.4-6.5 also demonstrates that curvature driven topological changes can be handled perfectly by our implementation of the level set method even in three dimensional space.

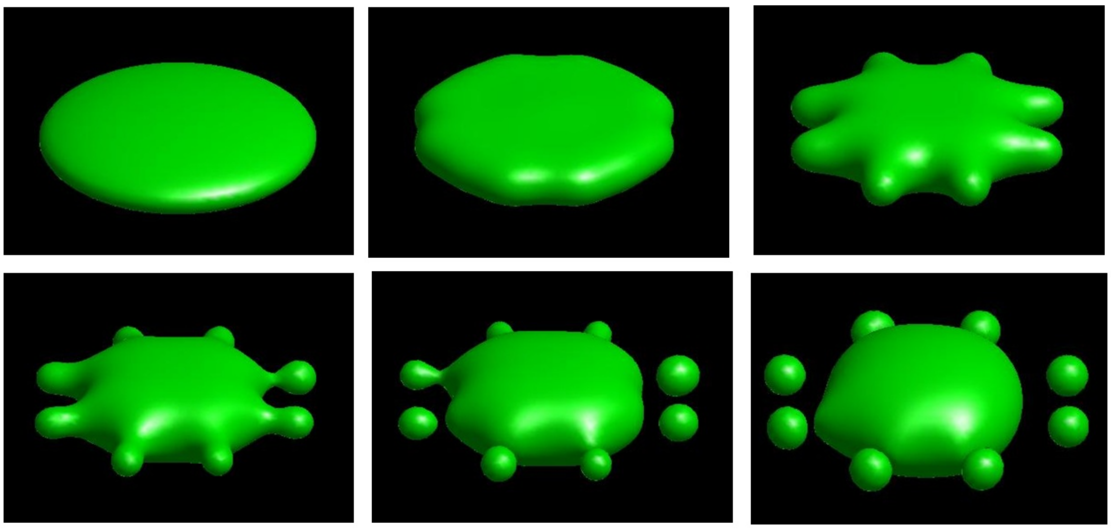

Figure 6.4. Shape evolution of an oblate with $v=0.6$ and spontaneous curvature $C=$ 30.

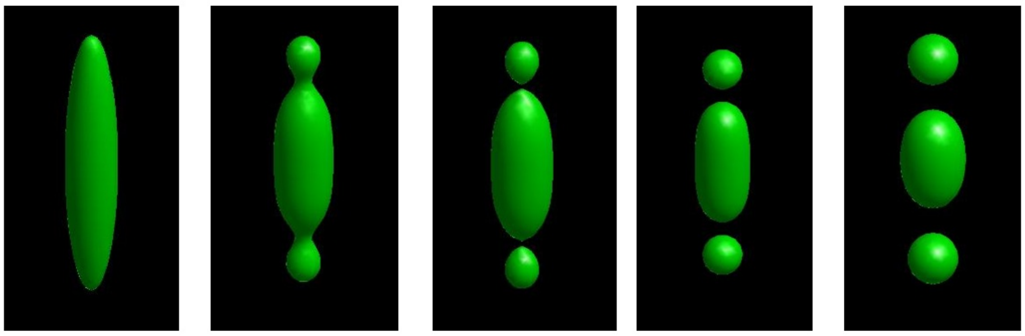

Figure 6.5. Shape evolution of a prolate with $v=0.6$ and spontaneous curvature $C=$ 30.

6.1.4 Effects of Spontaneous Curvature and Osmotic Pressure

In section 6.1.1-6.1.3, the reduced volume is kept constant by preserving volume and area of the vesicle. Experimentally, the volume of a vesicle can be adjusted by putting vesicles under a constant osmotic pressure [yanagisawa2008shape]. In this section, we explore effects of a nonzero spontaneous curvature and a constant osmotic pressure. This can be done by setting $P=P_0=$ constant and $\sigma_{\Delta A}=\sigma_{\Delta A_0}=$ constant. Note that setting $\sigma_{\Delta A}=\sigma_{\Delta A_0}$ is equivalent to setting $C=$ $\sigma_{\Delta A_0}h/\kappa$ and shifting σ to $\sigma+\frac{\kappa}{2}C^2$. Therefore we can explore effects of a constant spontaneous curvature by either specifying the value of C or the value of $\sigma_{\Delta A}$. In other words, both C and $\sigma_{\Delta A}$ can be interpreted as *generalized forces* with the area difference between the bilayer as *generalized displacements*. In this section we set at each time step

$$C=0, P=P_0, \frac{\mathrm{d}A}{\mathrm{d}t}=\frac{A_p \quad A(t)}{\Delta t}, \sigma_{\Delta A}=\sigma_{\Delta A_0}, \tag{6.13}$$

where A_p is set to be the initial area and $A(t)$ is the current area. Requirement Eq. (6.13) combined with Eqs. (6.5-6.7) will give σ that conserves area.

In figure 6.6, we start with a prolate with $v=0.5$ and set $P_0=$ 2000, $\sigma_{\Delta A_0}=0$. A large negative pressure will reduce volume of the vesicle and transform the prolate to a tube. In figure 6.7, we start with an oblate with $v=0.4$ and set $P_0=$ 1000, $\sigma_{\Delta A_0}=13$. The negative pressure term will deflate the oblate and the oblate eventually pinches into a torus. In figure 6.8, we show the evolution of an oblate with $v=0.75$ under a smaller pressure $P_0=$ 200 and different $\sigma_{\Delta A_0}$ ($\sigma_{\Delta A_0}=12$ for the top row and $\sigma_{\Delta A_0}=20$ for the bottom row). As can be seen, depending on the relative strength of the effects of spontaneous curvature and pressure, an oblate can either relax to a three leg starfish or a four leg starfish.

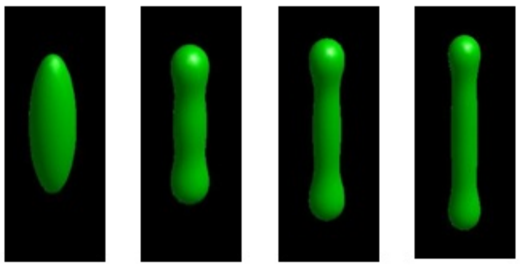

Figure 6.6. Relaxation of a prolate to a tube.

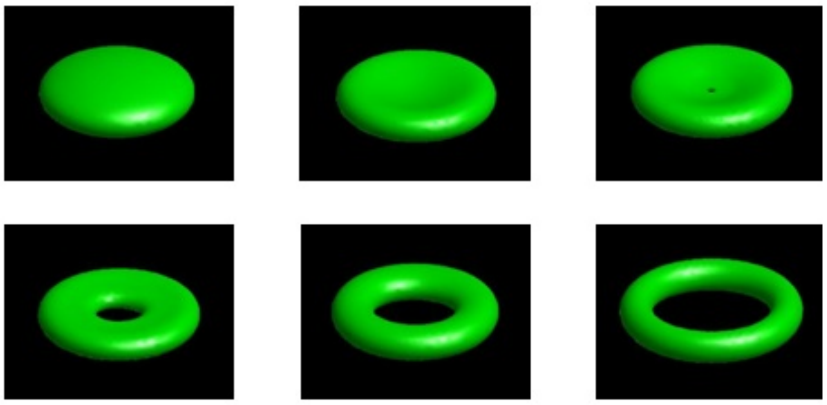

Figure 6.7. Pinching of an oblate to a torus.

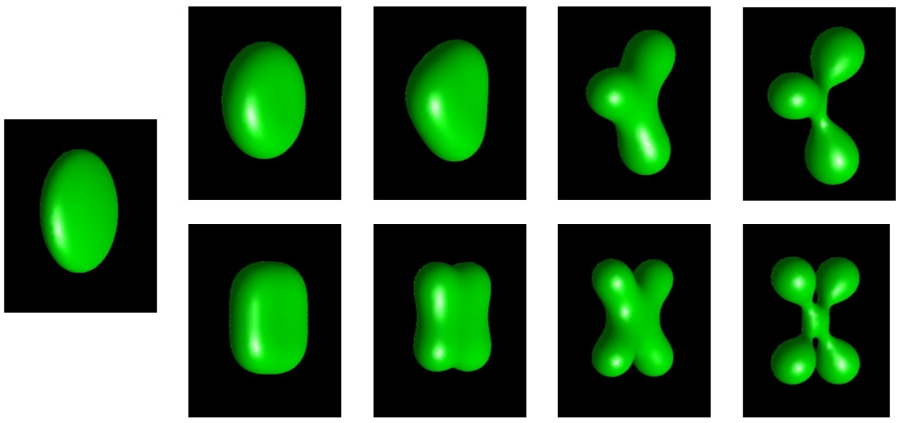

Figure 6.8. Deformation of an oblate into starfishes.

6.1.5 Relaxation under Constrained Reduced Area Difference

By trying different values of P_0 and σ_{A_0} in section 6.1.4, we obtained shapes of different reduced volumes and reduced area differences. We can also constrain the reduced area difference by the

Lagrange multiplier method by setting

$$C=0, \frac{\mathrm{d}V}{\mathrm{d}t}=\frac{V_p \quad V(t)}{\Delta t}, \frac{\mathrm{d}A}{\mathrm{d}t}=\frac{A_p \quad A(t)}{\Delta t}, \frac{\mathrm{d}(\Delta a)}{\mathrm{d}t}=\eta_\varepsilon\left(\frac{\Delta a_p \quad \Delta a(t)}{\Delta t}\right),$$

where Δa_p is the prescribed reduced area difference, Δa is the current reduced area difference, $\eta_\varepsilon(\xi)=\begin{cases}\xi & \mathrm{abs}(\xi)\leqslant\varepsilon\\ \mathrm{sign}(\xi)\cdot\varepsilon & \mathrm{abs}(\xi)>\varepsilon\end{cases}$ where ε is a small positive constant limiting the time rate of change of the reduced area difference.

In the top row of figure 6.9, we start with an oblate with $v=0.6$ and $\Delta a=1.05$ (calculated from initial values) and relax it with $\Delta a_p=\Delta a$, i.e., Δa is conserved. The resulting shape loses cylindrical symmetry and becomes triangular. In the bottom row of figure 6.9, we start with a prolate with $v=0.9$ and $\Delta a=1.05$ and set $\Delta a_p=1.10$. The resulting shape also loses symmetry and deforms into a pear. In figure 6.10, we start with a triangular oblate resulting from the Δa preserving relaxation of an symmetrical oblate with $v=0.8, \Delta a=1.04$ and set $\Delta a_p=0.9$. The final shape is a stomatocyte that invaginates inward. In figure 6.11, we show the transformation from an oblate ($v=0.55$, $\Delta a=1.06$) to a two leg starfish ($\Delta a_p=1.22$) in the top row, and the transformation from a prolate ($v=0.55, \Delta a=1.47$) to a necklace ($\Delta a_p=1.60$).

It is interesting to see how those asymmetrical shapes (starfishes, triangular elliptocyte, pear, stomatocyte, etc) arise naturally from initially symmetrical shapes in the simulation.

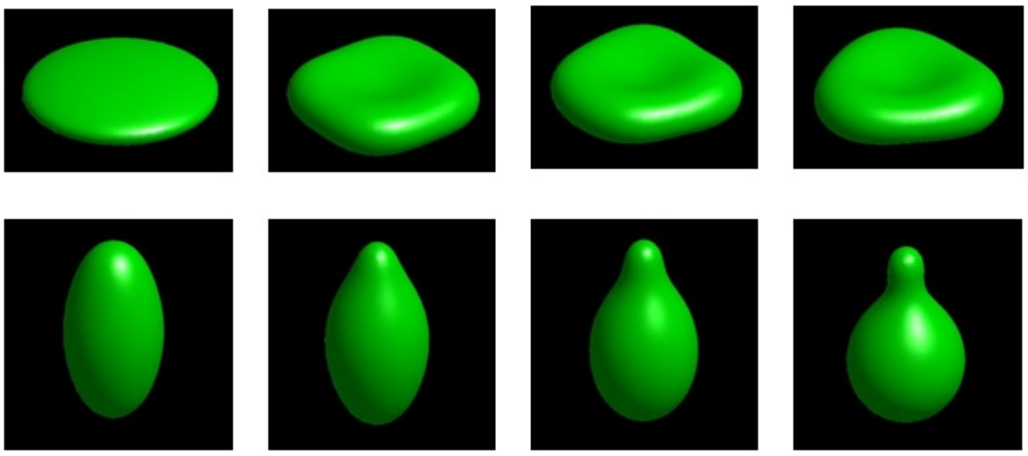

Figure 6.9. Top row: shape transformation of a symmetric oblate to a triangular elliptocyte. Bottom row: shape transformation for a prolate to a pear.

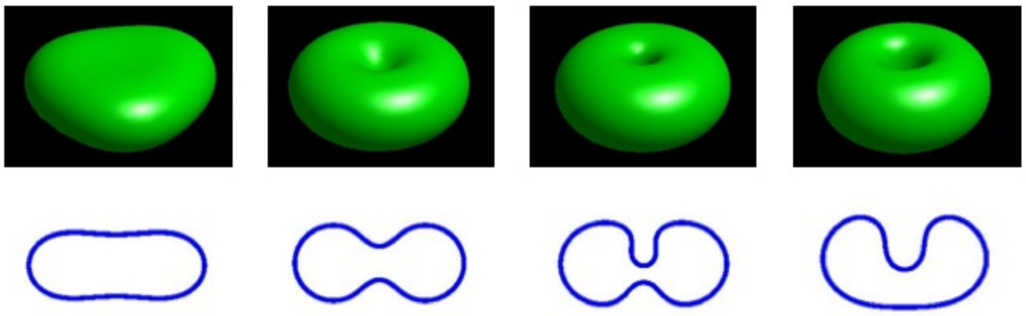

Figure 6.10. Top row: shape transformation of a triangular elliptocyte to a stomatocyte. Bottom row: cross section view of the top row. Note that the initial shape is not symmetric and the cross section view is not unique.

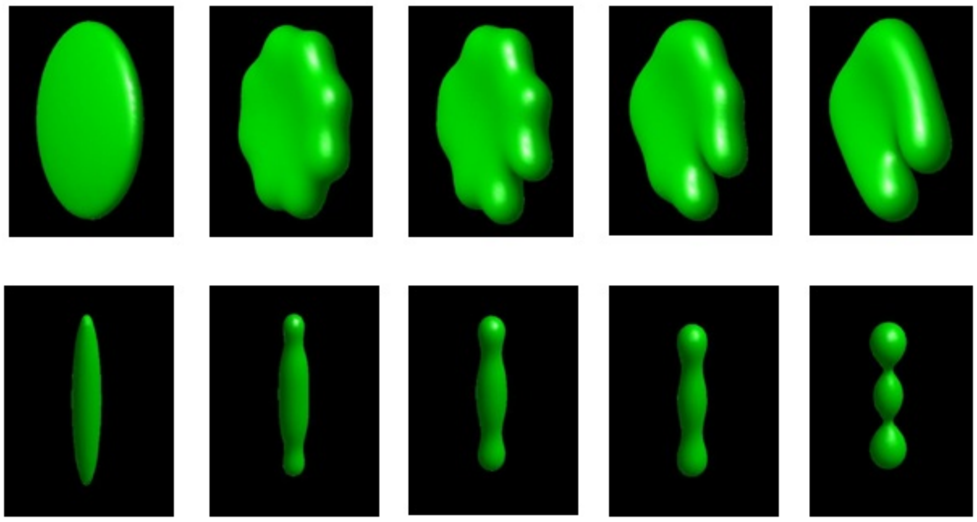

Figure 6.11. Top row: shape transformation of an oblate into a two leg starfish. Bottom row: shape transformation of a prolate to a necklace.

6.2 Shape of Multi-component Vesicles

In this section, we further develop numerical schemes introduced in section 6.1 to deal with multicomponent vesicles. Experimentally, lipid domains can be produced from mixtures of high melting temperature (saturated) lipids, low melting temperature (unsaturated) lipids and cholesterol [veatch2003separation]. Because of its hydrophobic steroid ring structure, cholesterol can fit in neither the solid-ordered phase of the saturated lipids(due to its peculiar size and shape) nor the Lo (lipid-disordered) phase of the unsaturated lipids (due to hydrophobicity), which then introduces the *Ld (lipid-ordered) phase* [mouritsen2005life]. The lipid-ordered phase, on the one hand, maintains fluidity and allows necessary rapid diffusion in the plane of the membrane, and on the other hand, features conformational order in the lipid chains that increases rigidity and stability of the membrane [mouritsen2005life]. Those lipid-ordered domains have many similarities with lipid-rafts found in vivo [harayama2018understanding] which are important in many membrane functions.

Theoretical and numerical studies of multi-domain vesicles begun as early as in 1990s. Jülicher and Lipowky seems to be the first to propose a Hamiltonian for multidomain vesicles which also takes into account effects of different Gaussian bending moduli for different domains [julicher1996shape]. Taniguchi performed numerical studies of phase separation dynamics and shape deformation of two component vesicles [taniguchi1996shape]. Jülicher and Lipowky's theory was then confirmed by Baumgart's experiments [baumgart2005membrane]. Recently, phase filed methods are adapted for numerical simulation of multicomponent vesicles [wang2008modelling, lowengrub2009phase, sohn2010dynamics]. Our level set approach is the first three dimensional simulation of multicomponent vesicles that takes into account differences in Gaussian bending moduli and can easily incorporate forces from the more general line Hamiltonian Eq. (3.2). In our simulation, vesicle shape evolves according to mechanical forces from variational calculations in chapter 3 while the driving forces calculated in the phase-field model is only thermodynamically consistent [wang2008modelling, lowengrub2009phase].

6.2.1 Hamiltonian of an Incompressible Biphasic Vesicle

Consider a vesicle with Ld and Lo phases and we use $\mathcal{P}$ and $\mathcal{P}^{+}$ to denote the surface patches corresponding to Ld and Lo phases. The plus/mius sign comes from our numerically convention

$$\mathcal{P} \quad = \{\boldsymbol{R}(Z) | \phi(Z) = 0, \psi(Z) < 0\}, \mathcal{P}^{+} = \{\boldsymbol{R}(Z) | \phi(Z) = 0, \psi(Z) > 0\}. \tag{6.14}$$

Thus $\boldsymbol{n}$ points into $\mathcal{P}^{+}$ from $\mathcal{P}\;\;$. The Hamiltonian for a biphasic vesicle can be written as

$$\begin{aligned} H_{\mathrm{bi}} = \quad & \int_{\mathcal{P}} \mathrm{d}A\left[\frac{\kappa^{\mathrm{Ld}}}{2}(K_M \quad C^{\mathrm{Ld}})^2 + \kappa_G^{\mathrm{Ld}} K\right] + \int_{\mathcal{P}^{+}} \mathrm{d}A\left[\frac{\kappa^{\mathrm{Lo}}}{2}(K_M \quad C^{\mathrm{Lo}})^2 + \kappa_G^{\mathrm{Ld}} K\right] \\ & + \sigma \;\left(\int_{\mathcal{P}} \mathrm{d}A \quad A \;\;\right) + \sigma^{+}\left(\int_{\mathcal{P}^{+}} \mathrm{d}A \quad A^{+}\right) \quad P\left(\int \frac{\boldsymbol{R}\cdot\boldsymbol{N}}{3}\mathrm{d}A \quad V_p\right) \\ & + \oint_{\partial\mathcal{P}} \mathrm{d}l(\kappa_l + \kappa_g k_g + \kappa_\tau \tau_g + \kappa_n k_n) + \int \mathrm{d}A \frac{\mu}{2}\left[(1 \quad c)^2 + \left(1 \quad \frac{1}{c}\right)^2\right], \end{aligned} \tag{6.15}$$

where $\kappa^{\mathrm{Ld}\;(\mathrm{Lo})}$, $\kappa_G^{\mathrm{Ld}(\mathrm{Lo})}$ is the mean curvature and Gaussian curvature bending moduli for the Ld(Lo) phase, $C^{\mathrm{Ld}(\mathrm{Lo})}$ the spontaneous curvature for the Ld(Lo) phase, $\sigma\;\;^{(+)}$ are Lagrange multipliers conserving area $A\;\;^{(+)}$ of the Ld(Lo) phase, P constrains volume of the vesicle, $\kappa_{l,g,\tau,n}$ can be interpreted as line tension constants associated with curve length, geodesic curvature, geodesic torsion and normal curvature. The c field in Eq. (6.15) measures local compression and stretching. Physically, c may be interpreted as density of lipid molecules of the vesicle. The hypothesis of being incompressible means density of lipids should remain constant during surface deformation. Dynamics of c is dictated by the conservation law

$$\frac{\partial c}{\partial t} + \nabla_{\|}\cdot(c\boldsymbol{v}) = 0, \tag{6.16}$$

where $\boldsymbol{v}$ is the velocity field of the vesicle. Initial value of c is set to be 1 everywhere. When $c>1(c<1)$, the vesicle is locally compressed(stretched). A large elastic moduli μ in Eq. (6.15) will help keep the vesicle incompressible by penalizing deviations of c from 1. This idea of defining a local field measuring local stretching is adopted from [aland2014diffuse]. The model defined in Eq. (6.15) is the spontaneous curvature model for biphasic vesicles from [julicher1996shape] with a generalized line tension term and an additional elastic energy term enforcing incompressibility.

In the practice of level set method, the level set function is periodically reinitialized to be a signed distance function to maintain numerical accuracy and stability [sussman1994level]. For the same reason, the auxiliary level set function ψ used to represent phase boundaries also needs periodic regularization. In [towers2009discretizing], it was shown that to obtain a convergent scheme for calculating length of phase boundary with Eq. (4.57), ϕ,ψ should be a signed distance function and their zero level set should be orthogonal to each other. In our numerical experiments, we found that ψ being a signed geodesic distance function gives better stability. The geodesics can be computed by iterating to equilibrium the following PDE [cheng2002motion] in virtual time τ

$$\frac{\partial\psi}{\partial\tau} + \mathrm{Sign}(\psi^0)(|\nabla_{\|}\psi| \quad 1) = 0. \tag{6.17}$$

We refer the reader to [zhang2020sixth] for the state of art numerical scheme for computing geodesics in the level set framework.

The above regularization makes sense only if the phase boundary is located on the surface of the vesicle. However, in simulation of biphasic vesicles, the phase boundary will shrink into a very small circle and eventually disappear after domains of a different phase pinches off. In this case of biphasic vesicles pinching off at phase boundaries, we add an additional energy term $H_{\mathrm{reg}}(\psi,\nabla\psi)$ to regularize ψ

$$H_{\mathrm{reg}}(\psi,\nabla\psi) = \int \frac{D_N}{2}(\boldsymbol{N}\cdot\nabla\psi)^2 \mathrm{d}V + \int p(|\nabla\psi|)\mathrm{d}V. \tag{6.18}$$

The first integral in Eq. (6.18) diffuses ψ in the normal direction $\boldsymbol{N}$ of the surface with a diffusion coefficient D_N, which is also employed in the phase-field method [wang2008modelling] to keep two tanh profile orthogonal. The second integral in Eq. (6.18) is adopted from the distance regularized level set method developed by Li in a series of papers [li2005level, li2010distance], where $p(|\nabla\psi|)$ has a minimum value of zero at $|\nabla\psi|=1$. Minimization of $\int p(|\nabla\psi|)\mathrm{d}V$ will help keep $|\nabla\psi|$ close to 1. This functional plays a similar role to the Lyapunov functional giving rise to the Cahn-Hilliard equation for phase separation. While the latter keeps ψ a tanh profile, the former maintains ψ close to a signed distance map. We set $p(|\nabla\psi|)$ to be a quadratic function following [li2005level]

$$p(|\nabla\psi|) = \frac{D_n}{2}(1 \quad |\nabla\psi|)^2.$$

More choices of $p(|\nabla\psi|)$ are discussed in [li2010distance]. In practice, when there is no pinching of phase boundaries, we solve Eq. (6.17) to regularize the auxiliary level set function ψ. When a narrow neck is detected, we switch to minimize Eq. (6.18). We set $D_N = 1$ and $D_n = 1$ in our numerical experiments.

To derive dynamical equations for ϕ and ψ, we calculate variation of Eq. (6.15) under a variation $\boldsymbol{R} \to \boldsymbol{R} + \delta\boldsymbol{R}$. Under this variation, the lipid density variation is given by Eq. (3.31). If the surface energy density $f(c)$ is only a function of c, then $\delta f(c) = (\partial f / \partial c)\delta c$ and the variation of $\int f(c)\mathrm{d}A$ is given by Eq. (3.33). Comparison between Eq. (3.33) and Eq. (3.26) implies that a density dependent surface energy density $f(c)$ is equivalent to a surface tension $f(c) \quad c(\partial f / \partial c)$. Let us denote by σ_c this equivalent tension for the incompressibility penalty term $f_{\text{in}}(c) = \frac{\mu}{2}[(1 \quad c)^2 + (1 \quad 1/c)^2]$. Then,

$$\sigma_c \equiv f_{\text{in}} \quad c\frac{\partial f_{\text{in}}}{\partial c} = \frac{\mu}{2}\left[\left(\frac{1}{c} \quad 1\right)\left(\frac{3}{c} \quad 1\right) + (1 \quad c^2)\right]. \tag{6.19}$$

Now, from Eqs. (3.33,6.19) and results in chapter 3, we have

$$\frac{\delta H_{\text{bi}}}{\delta \boldsymbol{R}} = \int \boldsymbol{f}_{\text{su}}\mathrm{d}A + \int_{\partial\mathcal{P}} \boldsymbol{g}_{\text{bo}}\mathrm{d}l \tag{6.20}$$

where the surface bending force $\boldsymbol{f}_{\text{su}}$ and line forces $\boldsymbol{g}_{\text{bo}}$ located at phase boundary are

$$\begin{aligned}
\boldsymbol{f}_{\text{su}} &= \nabla_{\|}\sigma_c \quad \left[\kappa\left(\Delta_{\|}K_M + \frac{1}{2}K_M^3 \quad 2K_MK + 2CK \quad \frac{1}{2}C^2K_M\right) \quad (\sigma + \sigma_c)K_M \quad P\right]\boldsymbol{N} && (6.21)\\
\boldsymbol{g}_{\text{bo}} &= (\kappa^{\text{Ld}} \quad \kappa^{\text{Lo}})\nabla_{\perp}K_M\boldsymbol{N} \quad \left[\frac{\kappa^{\text{Ld}}}{2}(K_M \quad C^{\text{Ld}})^2 \quad \frac{\kappa^{\text{Lo}}}{2}(K_M \quad C^{\text{Lo}})^2\right]\boldsymbol{n}\\
&\quad \left\{\left(\frac{\kappa_G^{\text{Ld}} \quad \kappa_G^{\text{Lo}}}{2} + \kappa_g\right)(K\boldsymbol{n} + \nabla_s\tau_g\boldsymbol{N}) + (\sigma \quad \sigma^+)\boldsymbol{n}\right\} + \kappa_l(k_g\boldsymbol{n} + k_n\boldsymbol{N})\\
&\quad + \kappa_\tau[(\nabla_s k_g \quad k_n\tau_g)\boldsymbol{N} \quad (\nabla_s k_n + k_g\tau_g)\boldsymbol{n}] + k_n[(\nabla_s\tau_g + k_gB_\perp)\boldsymbol{n} + \tau_g^2\boldsymbol{n}] && (6.22)
\end{aligned}$$

where

$$\begin{aligned}
\kappa &\equiv \kappa^{\text{Lo}}H(\psi) + \kappa^{\text{Ld}}(1 \quad H(\psi))\\
C &\equiv C^{\text{Lo}}H(\psi) + C^{\text{Ld}}(1 \quad H(\psi))\\
\sigma &\equiv \sigma^+H(\psi) + \sigma \quad (1 \quad H(\psi))
\end{aligned}$$

with $H(\psi)$ as the Heaviside function of ψ. Note that presence of κ_g is equivalent to a difference of Gaussian bending moduli κ_G^{Ld} and κ_G^{Lo}, which is a direction result from the Gauss-Bonnet theorem.

The normal speed v_{su} of the surface is determined from the balance of bending force and viscous force in the $\boldsymbol{N}$ direction

$$(\boldsymbol{f}_{\text{su}} + \delta(\psi)|\nabla\psi|\boldsymbol{g}_{\text{bo}}) \cdot \boldsymbol{N} = \zeta_{\text{bi}}v_{\text{su}}, \tag{6.23}$$

where the coefficient $\delta(\psi)|\nabla\psi|$ diffuses line force to surface and we will take the viscosity coefficient ζ_{bi} to be one for simplicity. The normal speed v_{bo} of the phase boundary is determined from balance of forces in the $\boldsymbol{n}$ direction

$$\boldsymbol{g}_{\text{bo}} \cdot \boldsymbol{n} = \xi_{\text{bi}}(v_{\text{bo}} \quad v_{\text{sn}}) \tag{6.24}$$

where the viscosity coefficient ξ_{bi} is set to be one and v_{sn} is the $\boldsymbol{n}$ component of surface velocity set by

$$\boldsymbol{f}_{\text{su}} \cdot \boldsymbol{n} = \zeta_{\text{bi}}v_{\text{sn}}, \tag{6.25}$$

with the visocisity coefficent ζ_{bi} being one. The Lagrange multipliers $\sigma \quad, \sigma^+, P$ are determined from constrains on volume, total area and distribution of areas among different phases. The time rates of change of total volume V and total area A are still given by Eqs. (6.5,6.6) with the replacement

$$\boldsymbol{v} \cdot \boldsymbol{N} = v_{\text{su}}. \tag{6.26}$$

The third constraint is now the one imposed on the area $A \quad$ of the Ld phase, whose time rate of change is now given by

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \int_{\mathcal{P}} (\quad K_M)v_{\text{su}}\mathrm{d}A + \int_{\partial\mathcal{P}} v_{\text{bo}}\mathrm{d}l. \tag{6.27}$$

In all of our simulations, we will always keep the ratio of different phases a constant. We will therefore set

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \frac{A\ \ (0)\quad A\ \ (t)}{\Delta t} \tag{6.28}$$

most of the time, where $A\ \ (0)$ is the initial area of the Ld phase.

Now we have three fields ϕ, ψ, c to evolve. The level set function ϕ still follows the level set equation

$$\frac{\partial\phi}{\partial t} + v_{\mathrm{su}}|\nabla\phi| = 0 \tag{6.29}$$

and is handled in the same manner as in the previous section. The evolution of the auxiliary level set function ψ is dictated by

$$\frac{\partial\psi}{\partial t} + v_{\mathrm{bo}}|\nabla\psi| = \quad \frac{\delta H_{\mathrm{reg}}}{\delta\psi}, \tag{6.30}$$

where

$$\begin{aligned}\frac{\delta H_{\mathrm{reg}}}{\delta\psi} &= \nabla\cdot\left\{(D_N\boldsymbol{N}\cdot\nabla\psi)\boldsymbol{N} + \frac{p'(|\nabla\psi|)}{|\nabla\psi|}\nabla\psi\right\}\\ &= D_N\left(K_M\frac{\partial\psi}{\partial N} + N^i\nabla_i N^j\nabla_j\psi + N^iN^j\nabla_i\nabla_j\psi\right)\\ &\quad + \frac{p'(|\nabla\psi|)}{|\nabla\psi|}\Delta\psi + \frac{p''(|\nabla\psi|)|\nabla\psi|\quad p'(|\nabla\psi|)}{|\nabla\psi|^3}\nabla\psi\otimes\mathrm{Hessian}(\psi)\otimes(\nabla\psi)^{\mathrm{T}}.\end{aligned}$$

The advection velocity $\boldsymbol{v_c}$ for c is determined from

$$\boldsymbol{f}_{\mathrm{su}} = \zeta_{\mathrm{bi}}\boldsymbol{v}_c \tag{6.31}$$

and the dynamical equation for c is

$$\frac{\partial c}{\partial t} + \nabla_{\|}\cdot(c\boldsymbol{v}_c) = 0. \tag{6.32}$$

It is worth discussing how we relate velocity fields $v_{\mathrm{su}}, v_{\mathrm{bo}}, \boldsymbol{v}_c$ to force densities $\boldsymbol{f}_{\mathrm{su}}, \boldsymbol{g}_{\mathrm{bo}}$ in our simplified hydrodynamic model. The surface force density $\boldsymbol{f}_{\mathrm{su}}$ drives advection of ϕ, ψ and c, so $\boldsymbol{f}_{\mathbf{su}}$ will contribute to all of them. We assume that the line force $\boldsymbol{g}_{\mathrm{bo}}$ will introduce a singular surface force and therefore contribute to v_{su} via $\delta(\psi)|\nabla\psi|\boldsymbol{g}_{\mathrm{bo}}\cdot\boldsymbol{N}$. This force will also drive minimization of boundary energy even if the vesicle is held static by some external force. Thus v_{bo} should be directly linked to $\boldsymbol{g}_{\mathrm{bo}}$ and we assumed a simple linear relation. We did not include $\boldsymbol{g}_{\mathrm{bo}}$ directly in $\boldsymbol{v}_c$. We assumed that $\boldsymbol{g}_{\mathrm{bo}}$ will induce a tension jump of $(\sigma\quad\sigma^+)$ between the Ld and Lo phase and it is this jump in tension that leads to changes in c via compression or stretching of the vesicle (see Eq. (3.31)), whose effects are included by the dependence of $\boldsymbol{f}_{\mathbf{su}}$ on $\sigma^{\pm}$. In a more realistic hydrodynamical simulation, the line force density $\boldsymbol{g}_{\mathrm{bo}}$ will give rise to a locally varying tension field with jumps at phase boundaries instead of two constant tensions $\sigma^{\pm}$ with different phases. This also makes it very difficult for us to conserve areas of small domains in the presence of more than one domain. Still, we can make many interesting simulations within this very simple framework. More physically consistent models will be the topic of future research.

Eqs. (6.29,6.30,6.32) are handled with the same semi-implicit scheme as is for Eq. (6.9). For the lipid density field c, we use diffusion operator Δ to smooth its dynamics. For ψ, we still use the BiLaplacian operator. Thus, ϕ, c and ψ will be updated by

$$\phi^{n+1} = \phi^n\quad\Delta t(\mathbb{1}+\alpha\Delta t\Delta^2)\ ^1\{v_{\mathrm{su}}|\nabla\phi|\}^n \tag{6.33}$$

$$c^{n+1} = c^n\quad\Delta t(\mathbb{1}\quad\alpha\Delta t\Delta)\ ^1\{\nabla_{\|}\cdot(c\boldsymbol{v}_c)\}^n \tag{6.34}$$

$$\psi^{n+1} = \psi^n\quad\Delta t(\mathbb{1}+\alpha\Delta t\Delta^2)\ ^1\left\{v_{\mathrm{bo}}|\nabla\psi| + \frac{\delta H_{\mathrm{reg}}}{\delta\psi}\right\}^n. \tag{6.35}$$

The algorithm for biphasic vesicle dynamics is then as follows:

Initialize the level set functions ϕ, ψ on a three dimensional cartesian grid. Usually we initialize ϕ to be a sphere. Field ψ will be iterated to be the Geodesics on the surface with the numerical method from [zhang2020sixth]. Field c is set to be one everywhere. Set $D_N = 0$ and $D_n = 0$.

Repeat until the maximum time or the maximum number of iterations is reached:

i. Calculate geometry fields with the discretization from Chapter 4.

ii. Calculate surface force density $\boldsymbol{f}_{\mathrm{su}}$ and $\boldsymbol{g}_{\mathrm{bo}}$ without terms from Lagrange multipliers $\sigma^{\pm}, P$.

iii. Find the maximum $v_{\max}$ value of $|\boldsymbol{f}_{\mathrm{su}}|$ near the interface and use $v_{\max}\Delta t = N_{\mathrm{CFL}}\Delta s$ to determine time step Δt. We take the CFL number $N_{\mathrm{CFL}} = 1$ for simulations in this section.

iv. Solve for the Lagrange multipliers $\sigma^{\pm}, P$ by specifying $\mathrm{d}V/\mathrm{d}t, \mathrm{d}A/\mathrm{d}t, \mathrm{d}A\quad/\mathrm{d}t$ in Eqs. (6.5,6.6,6.27).

v. Calculate the complete $\boldsymbol{f}_{\mathrm{su}}$ and $\boldsymbol{g}_{\mathrm{bo}}$. Then find $v_{\mathrm{su}}, \boldsymbol{v}_c, v_{\mathrm{bo}}$.

vi. Calculate $(\phi^{n+1}, c^{n+1}, \psi^{n+1})$ according to Eqs. (6.33,6.34,6.35)

vii. Reinitialize ϕ if necessary.

viii. Calculate maximum $\left(\Delta s\sqrt{(k_n)^2 + (k_g)^2}\right)$. If this number is larger than 30, we stop using Eq. (6.17) to regularize ψ and set $D_N = 1, D_n = 1$ thereafter. Those numbers are found by try and err with numerical experiments.

6.2.2 Deformation of a Two Domain Biphasic Vesicle

In this section, we give an illustration of a typical shape evolution driven by line tension. The vesicle is initialized to be a sphere and the Ld phase (colored in red in Figure 6.12) covers 20% of the whole area. The only nonzero parameters in this simulation are $\kappa^{\mathrm{Ld}} = \kappa^{\mathrm{Lo}} = 1, \kappa_l = 30$. The final reduced volume is set to be 0.90. The Lagrange multipliers are determined from

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \eta_\varepsilon\left(\frac{V_p \quad V(t)}{\Delta t}\right), \frac{\mathrm{d}A}{\mathrm{d}t} = \frac{A_p \quad A(t)}{\Delta t}, \frac{\mathrm{d}A}{\mathrm{d}t} = \frac{A_p \quad A\quad(t)}{\Delta t}$$

at each time step, where $V(t), A(t), A\quad(t)$ is the current volume, current total area, current area of Ld phase, the prescribed volume $V_p = 0.9V(0)$ and $A_p = A(0), A\quad(t) = A\quad(0)$, ε is a small number limiting the time rate of change of the volume. We are thus conserving total area and area of different phases in our simulation while allowing the volume to decrease until the prescribed volume is reached. As can be seen from Figure 6.12, the Ld phase bulge out under effects of the line tension and a kink is introduced at the phase boundary. By starting with different phase distributions and changing simulation parameters, we are able to explore effects of various physical origins.

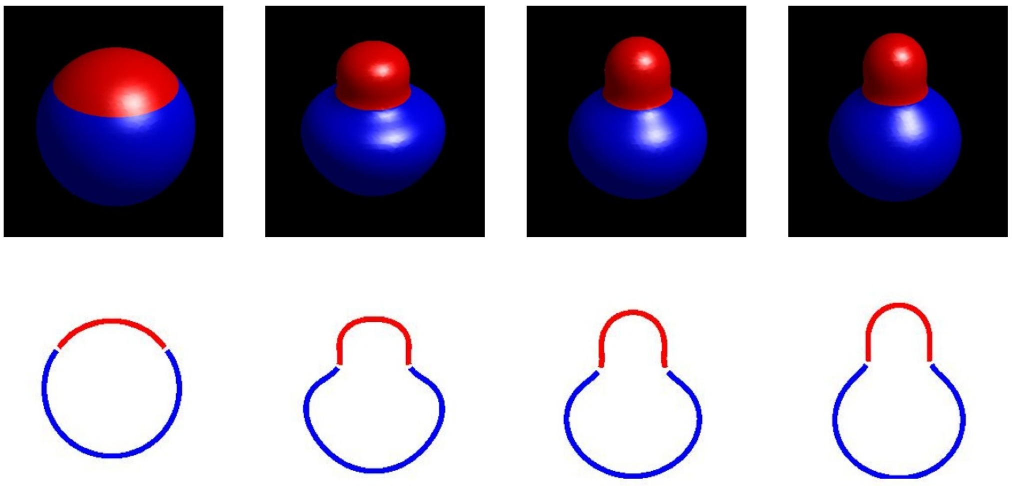

Figure 6.12. Deformation of a two domain biphasic vesicle.

6.2.3 Convergence Verification

To make sure that our simulation will yield better results when finer grids are used, we run our simulation described in the previous section on grids of different sizes and compare the final shape and energy. The final shapes at numerical time 0.021 are shown in Figure 6.13, where grid sizes goes from 48^3 in the left to 64^3 in the middle and 80^3 in the right. As can be seen from Figure 6.13, the final shapes are almost the same. A more quantitative comparison is given in Table 6.1, where the final total energy, line energy and bending energies for Ld phase and Lo phase are given. Those close values indicates numerical convergence of our scheme.

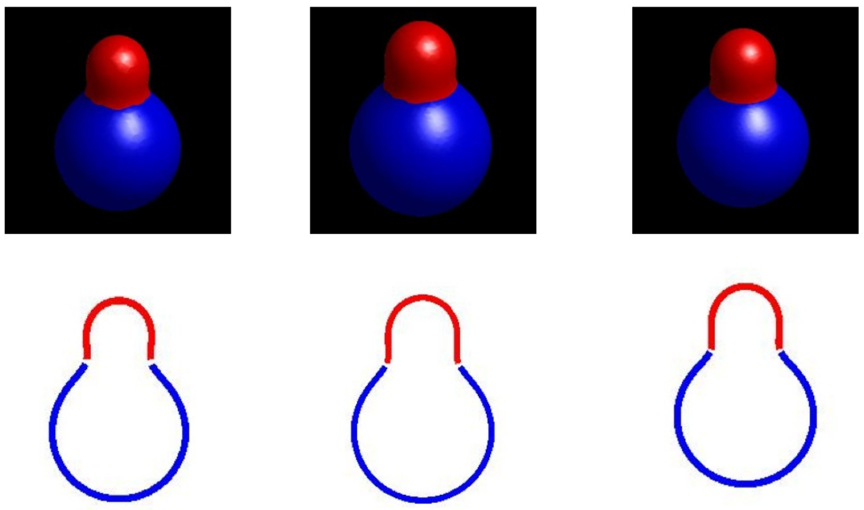

Figure 6.13. Final shapes in 3D view and cross section view at the same numerical time on different grids. Grid sizes from left to right: $48^3, 64^3, 80^3$.

grid sizes	48^3	64^4	80^3
total energy	122.138	123.786	124.095
line energy	86.282	88.190	88.021
bending energy for Ld phase	14.637	14.366	14.508
bending energy for Lo phase	21.219	21.230	21.566

Table 6.1. Final energies from simulation on different grids.

6.2.4 Pinching of Biphasic Bidomain Vesicles

In this section, we explore conditions under which a two domain biphasic vesicles will pinch. In particular, we will look at effects of the prescribed reduced volume, a constant spontaneous curvature and line tension constant κ_l.

For a biphasic bidomain vesicle with the ratio of surface area for the Ld phase being r, it is straightforward to find the maximum reduced volume $\mathrm{rd}_{\max}(r)$ that allows pinching of vesicle into two parts to be $(1 \quad r)^{3/2}+r^{3/2}$. When $r=0.5$, $\mathrm{rd}_{\max}\ \frac{1}{2})=\frac{\sqrt{2}}{2}$. Thus if the prescribed final reduced volume is larger than $\frac{\sqrt{2}}{2}$, no matter how large the line tension is, the vesicle will not pinch. This is illustrated in Figure 6.14, where we show **final** shape of vesicles with different prescribed reduced volume v_p and line tension constant κ_l. The only nonzero parameters are $\kappa^{\mathrm{Ld}}=\kappa^{\mathrm{Lo}}=1$,

$\kappa_l \neq 0$. For the top row, $v_p = 0.8 > \frac{2}{\sqrt{2}}$. Going from left to right on the top row, the line tension κ_l are set to be $5, 10, 30$ respectively. When κ_l is small, the vesicle looks like a prolate from the previous section. When κ_l is large, the two parts of the vesicle becomes more like spheres under constraints of volume and area. v_p is set to be 0.7 for the bottom row and κ_l is $4, 5, 10$ going from left to right. As expected, when the prescribed reduced volume allows pinching, very narrow neck can be obtained and as κ_l is further increased, vesicle will pinch at phase boundary. Even though Figure 6.14 shows **final** shapes of vesicles with different v_p and κ_l, from left to right we may also regard them as **dynamical** shape transformations with increasing line tension.

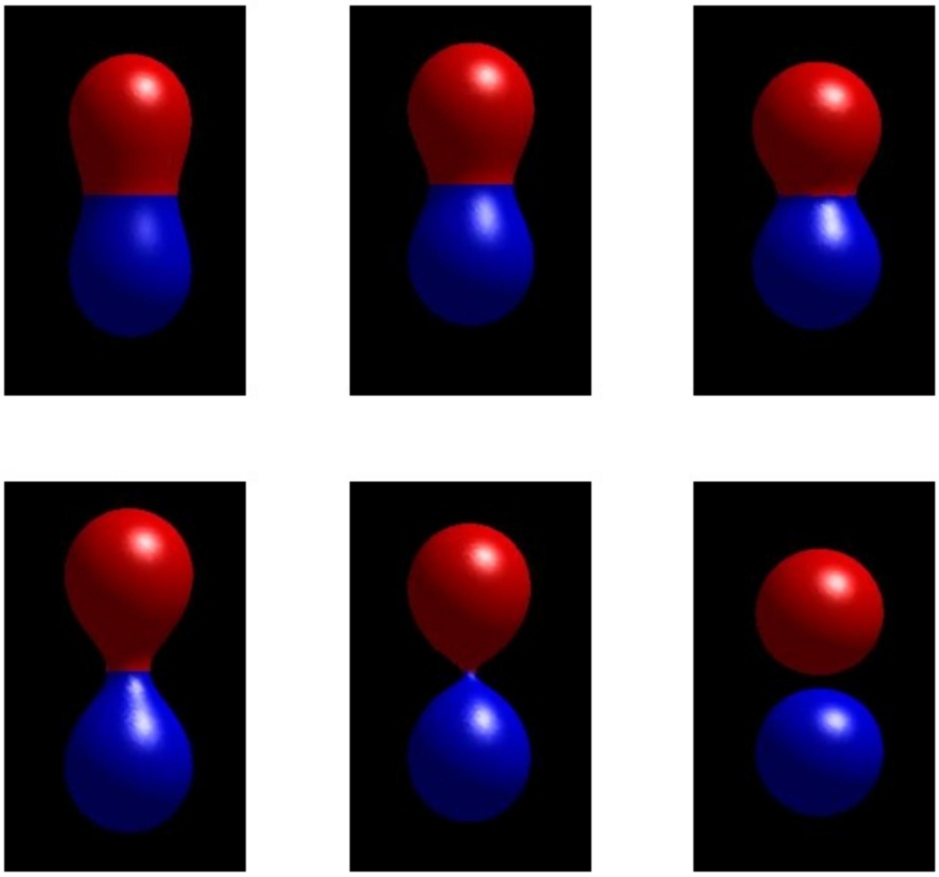

Figure 6.14. Final shapes of vesicles with different prescribed reduced volume v_p and κ_l. For top row, $v_p = 0.8$, κ_l is 5,10,30 from left to right. For bottom row, $v_p = 0.7$, κ_l is 4,5,10 from left to right.

Now we study effects of a constant spontaneous curvature C. We first run our simulation with different v_p, κ_l and $C^{\mathrm{Ld}} = C^{\mathrm{Lo}} = 0$ for a number of steps. Then we set $C^{\mathrm{Ld}} = C^{\mathrm{Lo}} = C$ and observe effects of different C. This may be regarded as a preliminary investigation of the effects of curvature inducing proteins added to systems of biphasic vesicles. The results are shown in Figure 6.15. For the first row, we set $v_p = 0.7, \kappa_l = 5$ and the first figure shows shapes of the vesicle just before C is set to 5. The next five pictures in the first row shows evolution of the same vesicle. Note that $v_p = 0.7, \kappa_l = 5, C = 0$ leads to a very narrow neck but no pinching (bottom middle figure in Figure 6.14 shows the final shape). Thus adding curvature inducing proteins may have an effect of destabilizing a very narrow neck and inducing pinching of vesicles. When the same nonzero $C(= 5)$ is set for a vesicle with $v_p = 0.7, \kappa_l = 4$ (bottom left figure in Figure 6.14), the neck radius will shrink but the vesicle is unable to pinch (middle row in Figure 6.15). If we increase C to 10, then the vesicle will readily pinch into two (bottom row in Figure 6.15).

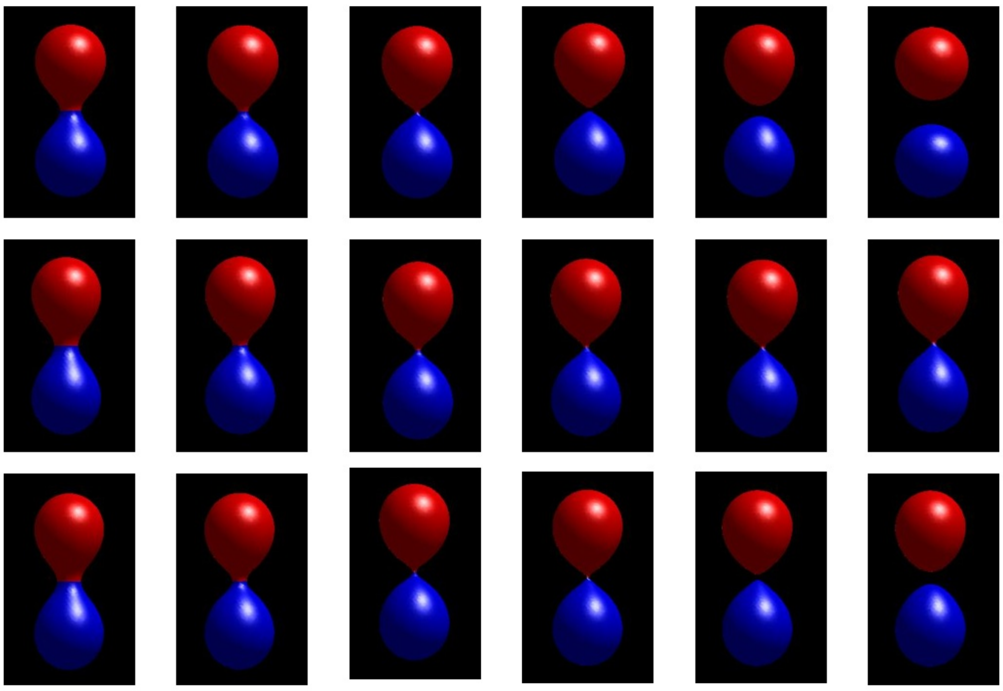

Figure 6.15. Effects of a nonzero spontaneous curvature. The first column from the left are starting shapes resulting from simulations with $C^{\mathrm{Ld}}=C^{\mathrm{Lo}}=0, r=0.5$ (area ratio of Ld phase), $v_p=0.7$ and different κ_l(top row $\kappa_l=5$, middle row and bottom row $\kappa_l=4$). Then we set $C^{\mathrm{Ld}}=C^{\mathrm{Lo}}=C$ (top row and middle row $C=$ 5, bottom row $C=$ 10) and show the effects of different spontaneous for each row.

From the above numerical experiments, we see that reduced volume v_p, ratio (r) of different phases (or area of lipid rafts), line tension constant κ_l, spontaneous curvature C can all play a role in the pinching of biphasic vesicles. A complete exploration of effects of all parameters in Eqs. (3.1,3.2) is beyond the scope of the current thesis.

6.2.5 Pinching of Multidomain Biphasic Vesicles

For simulation of bidomain vesicles, it is not necessary to impose the incompressibility condition as long as we keep ratio of each phase a constant. For multidomain vesicles, incompressibility becomes necessary if we want prevent nonphysical changes of domain sizes, as is shown in Figure 6.16. Note that in [wang2008modelling], the author termed this nonphysical domain coarsening the Oswald ripening, but this is simply due to a lack of imposed incompressibility. For the vesicle in Figure 6.16, we set $V_p=0.9V(0)$ and we are conserving total area and area of each phase. Other parameters are $\kappa^{\mathrm{Ld}}=1, \kappa^{\mathrm{Lo}}=10, \kappa_l=20$.

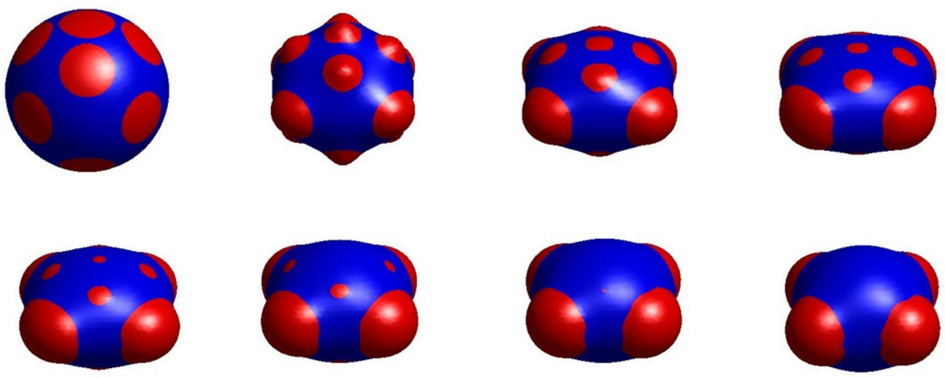

Figure 6.16. Domain coarsens if we set $\mu=0$.

To impose incompressibility, we set $\mu = 1000$. The resulting shape evolution at the same numerical time is illustrated in Figure 6.17. Domain shrinking and coarsening is not eliminated in Figure 6.17 but greatly reduced. A better way to impose incompressibility is to solve for a local tension so that the surface divergence of the velocity field is zero. This is currently under investigation.

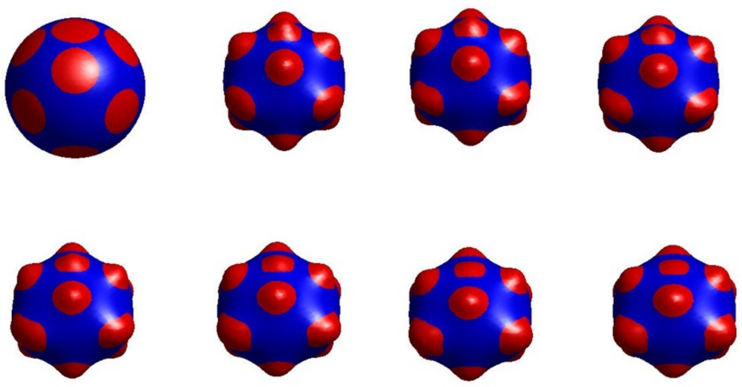

Figure 6.17. Domain becomes more stable if we impose the incompressibility condition by setting $\mu = 1000$.

For the vesicle evolution in Figure 6.18, we changed the relative bending moduli of Ld (red) and Lo (blue) phases. Now we have $\kappa^{\mathrm{Ld}} = 10, \kappa^{\mathrm{Lo}} = 1, \kappa_l = 20$. We also allows a smaller volume by setting $V_p = 0.8V(0)$. Since domains in red are more rigid now, the tendency for them to flatten out dominates the tendency to bulge out. This flattening transformation unexpectedly creates narrow channels among nearby domains and then under the effects of line tension, smaller domains begin to fuse into large circular ones. This fusion of domains and the large bending moduli for the red phase together makes the shape of the vesicle more and more planar. Note that we observed some nonphysical shrinking of domains on the sides even though we set $\mu = 1000$ for this case. This is yet to be addressed. Still, Figure 6.18 is an interesting example showing the effects of different bending moduli and the interplay of domain fusion and shape dynamics of the vesicles.

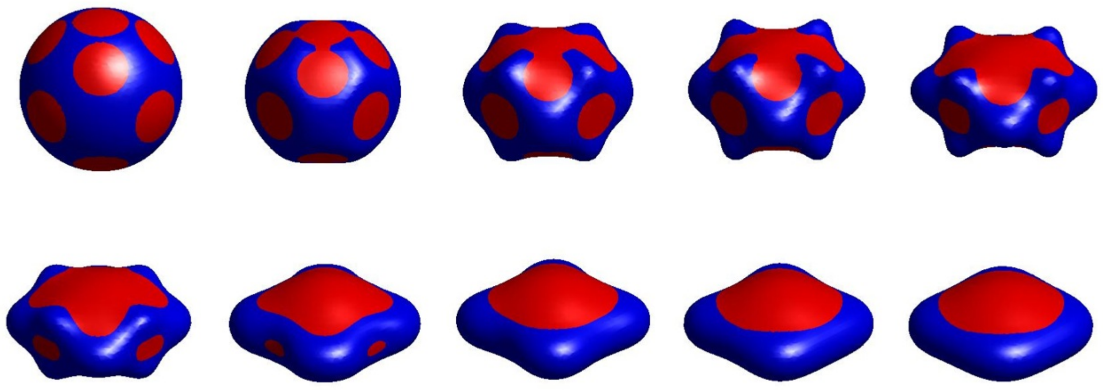

Figure 6.18. Vesicle shape dynamics under the effects of different bending moduli and domain fusion.

The next two numerical examples are inspired by experiments from [yanagisawa2008shape], where the author subjected phase separated vesicles to osmotic pressure differences and observed outside and inside budding. In Figure 6.19, we set $V_p = 0.8V(0), \kappa^{\mathrm{Ld}} = \kappa^{\mathrm{Lo}} = 1, \kappa_l = 100, \mu = 1000$. To maintain stability of the simulation, we use the maximum line curvature of the phase boundary to detect imminent pinching of domains. In practice, we calculate max $\sqrt{(k_n)^2 + (k_g)^2}$ near the phase boundary and when this number is larger than $30/\Delta x$, where Δx is the grid spacing, we stop using Eq. (6.17) to regularize ψ and set $D_N = 1, D_n = 1$ thereafter. This partial pinching of vesicles is very similar to the experiments illustrated Figure 2a from [yanagisawa2008shape].

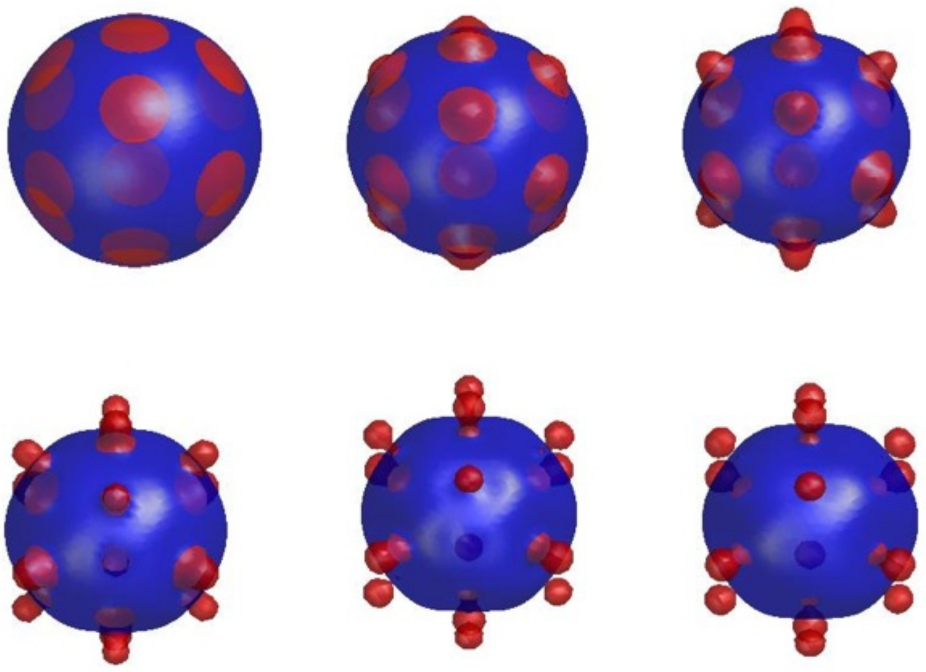

Figure 6.19. Outward partial pinching of multidomain biphasic vesicles under osmotic pressure.

In Figure 6.20, we let the phase separated domains invaginate inwardly at the start of the simulation. This initial invagination could be induced experimentally by subjecting ternary vesicles to osmotic pressure before phase separation begins [yanagisawa2008shape]. Also, we directly set the pressure P to be 600 instead of calculating it as a Lagrange multiplier. In this case, under a negative pressure, phase separated domains bud inside. This is very similar to experimental observation of Figure 2b in [yanagisawa2008shape].

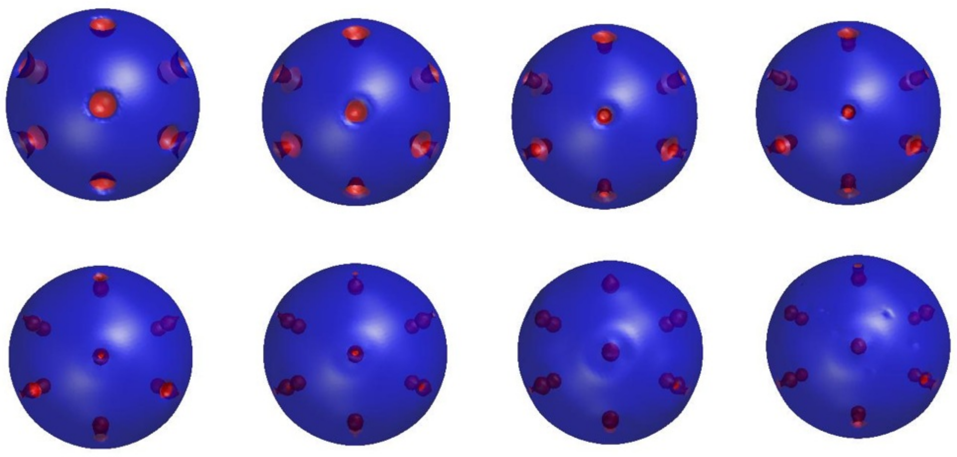

Figure 6.20. Inward partial pinching of multidomain biphasic vesicles under osmotic pressure.

By adjusting values of different parameters and starting with different initial conditions, we are able to explore effects of many parameters such as line tension constant κ_l, bending moduli, prescribed reduced volume, osmotic pressure etc. We may also explore effects of different Gaussian moduli by using a nonzero κ_g and other line tension parameters such as κ_τ, κ_n. A thorough analysis of complete line energy (Eq. (3.2)) is beyond the scope of the current paper and will be presented elsewhere. Still, our numerical scheme integrates different physical forces more naturally than that of [wang2008modelling] and allows more parameters to be explored.

6.3 Coupling Between Shape Dynamics of Vesicles and Protein Kinetics

In previous sections, bending moduli and spontaneous curvatures are assumed to be constant either through the whole vesicle, or through different phases. In reality, those parameters depend not only on lipid composition, but also on the oligomerization of curvature scaffolding proteins and the reversible insertion of protein regions that act like wedges in membranes [mcmahon2005membrane]. On the other hand, molecular dynamics simulation and *in vivo* budding assays show that proteins with an amphipathic helix can sense membrane curvature and preferably localize to regions of high curvature where they can then mediate membrane scission [martyna2016curvature, rossman2010influenza]. Research investigating the interplay of membrane curvature generation and sensing is still a rapidly developing field [baumgart2011thermodynamics]. In this section, we develop a simple model for curvature sensing and curvature generation by protein molecules in the level set framework.

6.3.1 Diffusion and Advection of Curvature Sensing and Generation Proteins on a Static Surface

As a first step, let us investigate how protein molecules should move tangentially on a static curved surface. The Hamiltonian of the system now depends only on the protein concentration field c_p and can be written as

$$H(c_p) = \int \frac{\kappa(c_p)}{2}[K_M \quad C(c_p)]^2 \mathrm{d}A, \tag{6.36}$$

where the bending moduli $\kappa(c_p)$ and the spontaneous curvature $C(c_p)$ now depends on protein concentration c_p. The Hamiltonian in Eq. (6.36) measures the mismatch energy between membrane bending and protein packing. Protein molecules will be advected so as to to minimize this mismatch energy. Suppose that we subject protein molecules to a position variation $\delta\boldsymbol{R}_p = W^\alpha \boldsymbol{S}_\alpha$, constrained to the surface of the membrane. Then the variation of concentration c_p will be given by

$$\delta_p(c_p) = \quad \nabla_\parallel \cdot (c_p \delta\boldsymbol{R}_p), \tag{6.37}$$

which simply says that concentration change is minus the outgoing material flux induced by $\delta\boldsymbol{R}_p$ and we used subscript p to stress that this variation is related to position changes of protein molecules (in contrast to the one without this subscript related to changes in surface position). Note the difference between Eq. (3.31) and Eq. (6.37), where the former variation arises from stretching and compression of lipids while the latter one results from directional advection of protein molecules. Now if the surface energy density $f(c_p)$ is only a function of c_p, the variation of $\int f(c_p)\mathrm{d}A$ is given by Eq. (3.47) and we can interpret the integrand $\left(\quad c_p \nabla_\parallel \frac{\partial f}{\partial c_p}\right)$ of the surface integral in Eq. (3.47) as molecular forces from lipids to protein molecules. The advection velocity

$\boldsymbol{v}_p$ of proteins molecules can then be determined from balance of this molecular force and viscosity

$$c_p \nabla_{\|} \frac{\partial f}{\partial c_p} = \zeta_p \boldsymbol{v}_p, \tag{6.38}$$

where ζ_p is the viscosity coefficient for lipids and protein molecules. The dynamics of c_p is given by the conservation law

$$\frac{\partial c_p}{\partial t} + \nabla_{\|} \cdot (c_p \boldsymbol{v}_p) = 0. \tag{6.39}$$

Let us take the pear shaped vesicle from Figure 6.9 as the curved surface and observe the advection of protein molecules under different models for $\kappa(c_p)$ and $C(c_p)$. In this thesis, we restrict our models to the linear ones adopted by [lowengrub2016numerical] based on experimental measurements from [jin2006measuring]. In particular, we set

$$\kappa(c_p) = \kappa_0 + \kappa_1 c_p, C(c_p) = C_0 + C_1 c_p. \tag{6.40}$$

Firstly, let us assume that the bending moduli κ is independent of c_p and the spontaneous curvature $C(c_p)$ is linearly dependent on c_p. Numerically, we set $\kappa_0 = 1, \kappa_1 = 0, C_0 = 0, C_1 = \ 1$ and $c_p(S, t = 0) = 4.75$. Then we solve Eq. (6.39) on this curved surface. The numerical scheme is the same as that for c in Eq. (6.34). The resulting density dynamics of c_p is shown in the top row of Figure 6.21, where we see that protein molecules tend to aggregate near the pointy pole of the pear and flow away from the saddle shaped neck region where the mean curvature is approximately zero. The total mass of protein molecules is set such that the equilibrium distribution of c_p is locally equal to the mean curvature. As a second example, we set $\kappa_0 = 1, \kappa_1 = 10, C_0 = 0, C_1 = 0, c_p(S, t = 0) = 1$. This time, we observe how protein molecules redistribute by dependence of bending moduli on c_p. The result is illustrated in the bottom row of Figure 6.21. As expected from minimization of Eq. (6.36), protein molecules now will congregate at the neck region and stay away from the most curved part of the surface. When both κ_1 and C_1 are nonzero, the equilibrium distribution of c_p will depend on local mean curvature and the relative importance of κ_1 and C_1. With minimal modification, we can also explore effects of non-linear models for $\kappa(c_p), C(c_p)$ and dependence of spontaneous Gaussian curvatures on c_p [schmidt2013influenza], which will be the topic for future research.

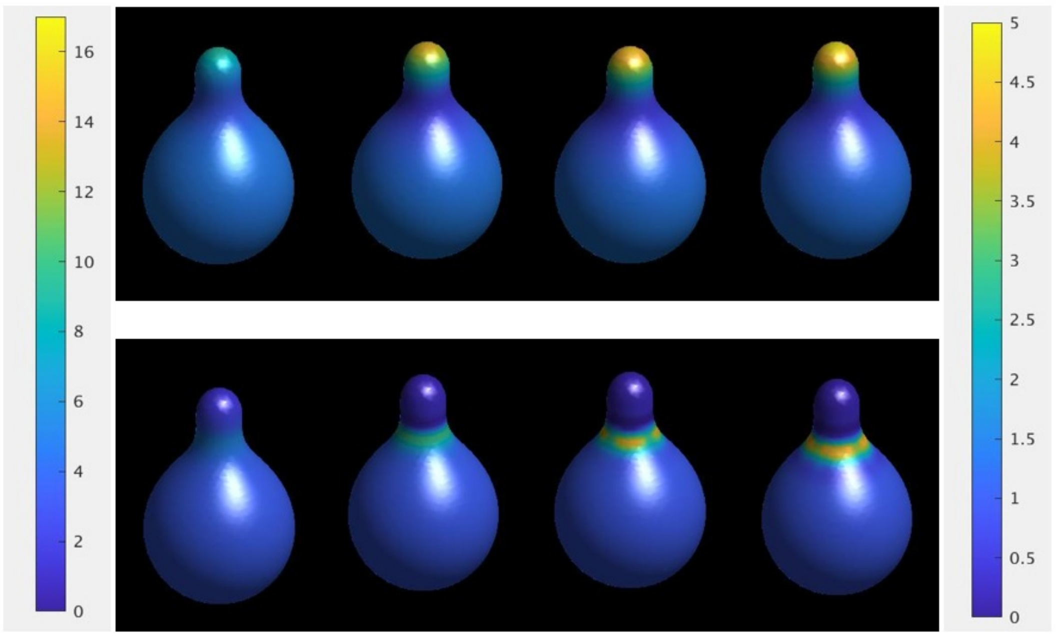

Figure 6.21. Protein redistribution due to curvature generation and sensing under different models. Left colorbar and top row: protein molecules localize to regions of high curvature. Right colorbar and bottom row: protein molecules localize to regions of small curvature.

6.3.2 Coupling Between Shape Dynamics of Single Phase Vesicle and Protein Kinetics

Having verified the numerical model for protein kinetics on a static surface, we move to combine vesicle shape dynamics with protein kinetics in this section. The Hamiltonian for this system is very similar to Eq. (6.1):

$$\begin{aligned} H_{\mathrm{ps}} \;=\; & \int \frac{\kappa(c_p)}{2}(K_M \quad C(c_p))^2 \mathrm{d}A + \sigma\Big(\int \mathrm{d}A \quad A_p\Big) \quad P\Big(\int \frac{\boldsymbol{R}\cdot\boldsymbol{N}}{3}\mathrm{d}A \quad V_p\Big) \\ & + \int \mathrm{d}A \frac{\mu}{2}\Big[(1 \quad c)^2 + \Big(1 \quad \frac{1}{c}\Big)^2\Big] \end{aligned} \tag{6.41}$$

where $\kappa(c_p)$ and $C(c_p)$ are now functions of density of curvature generation and sensing proteins, and we add the last term from Eq. (6.15) to impose incompressibility. The subscript *ps* in Eq. (6.41) indicates that this Hamiltonian accounts for protein kinetics on a single phase vesicle. The elastic bending force density $\boldsymbol{f}_{\mathrm{ps}}$ results from

$$\frac{\delta H_{\mathrm{ps}}}{\delta \boldsymbol{R}} = \int \boldsymbol{f}_{\mathrm{ps}} \mathrm{d}A \tag{6.42}$$

and the velocity $\boldsymbol{v}_{\mathrm{ps}}$ of the vesicle is determined from $\boldsymbol{f}_{\mathrm{ps}} = \zeta_{\mathrm{ps}} \boldsymbol{v}_{\mathrm{ps}}$ as is the case for Eq. (6.4), where ζ_{ps} is taken to be one. Note that in the calculation of Eq. (6.42), δc_p is handled in the same way as δc in Eq. (3.31). Since the variation of all of the terms in Eq. (6.41) has been calculated before, we ignore the explicit form here. The resulting flow velocity $\boldsymbol{v}_{\mathrm{sp}}$ will be the one driving deformation of the vesicle and density dynamics of c. In presence of a large μ in Eq. (6.41), $\nabla_{\|} \cdot \boldsymbol{v}_{\mathrm{sp}} \cong 0$ and the flow is approximately incompressible, as is verified in Figure 6.17. If we set $\boldsymbol{v}_p$ in Eq. (6.39) to be $\boldsymbol{v}_{\mathrm{sp}}$, c_p will remain a constant if initially set to be a constant. We therefore postulate that in addition to moving passively with lipids with velocity $\boldsymbol{v}_{\mathrm{sp}}$, additional tangential forces $\boldsymbol{f}_p$ will generate an extra velocity $\boldsymbol{v}_{\mathrm{xp}}$ for protein molecules via

$$\boldsymbol{f}_p = \zeta_p \boldsymbol{v}_{\mathrm{xp}} \tag{6.43}$$

and

$$\frac{\delta_p H_{\mathrm{ps}}}{\delta \boldsymbol{R}_p} = \int \boldsymbol{f}_p \mathrm{d}A, \tag{6.44}$$

where we set $\zeta_p = \frac{1}{3}$ to speed up tangential localization of protein molecules. The dynamics of the system then consists of the level set equation Eq. (6.9) for ϕ with $\boldsymbol{v} = \boldsymbol{v}_{\mathrm{ps}}$, the conservation law Eq. (6.32) for c with $\boldsymbol{v}_c = \boldsymbol{v}_{\mathrm{sp}}$ and the mass conservation Eq. (6.39) for protein molecules with $\boldsymbol{v}_p = \boldsymbol{v}_{\mathrm{ps}} + \boldsymbol{v}_{\mathrm{xp}}$. The entire algorithm is similar to that presented in Section 6.1 with additional computation to evolve c and c_p in each step.

As an example, we now allow the pear shaped vesicle in Figure 6.21 to bend along with protein diffusion and advection. We set $\kappa_0 = 1/12, \kappa_1 = 11/12, C_0 = 0, C_1 = \quad 20$ in Eq. (6.40) as our physical model for the $\kappa(c_p), C(c_p)$ and $c_p(S, t = 0) = 1$ as the initial condition. The Lagrange multipliers σ and P are determined from conservation of volume and area, i.e. $\mathrm{d}V/\mathrm{d}t = (V_p \quad V(t))/\Delta t$, $\mathrm{d}A/\mathrm{d}t = (A_p \quad A(t))/\Delta t$. The resulting shape dynamics of the vesicle and the density dynamics of protein molecules is illustrated in Figure 6.22. At first, protein molecules flow into the pointy pole of the vesicle and become relatively depleted in the neck region. This will reduce the bending stiffness of the neck region and help narrowing the radius of the neck, which is due the dependence of bending moduli on c_p. After the radius of the neck becomes sufficiently small, the effect of curvature sensing of protein molecules dominates and protein begin to accumulate near the neck region (see the second graph in the bottom of Figure 6.22), facilitating scission of the pear shaped vesicle.

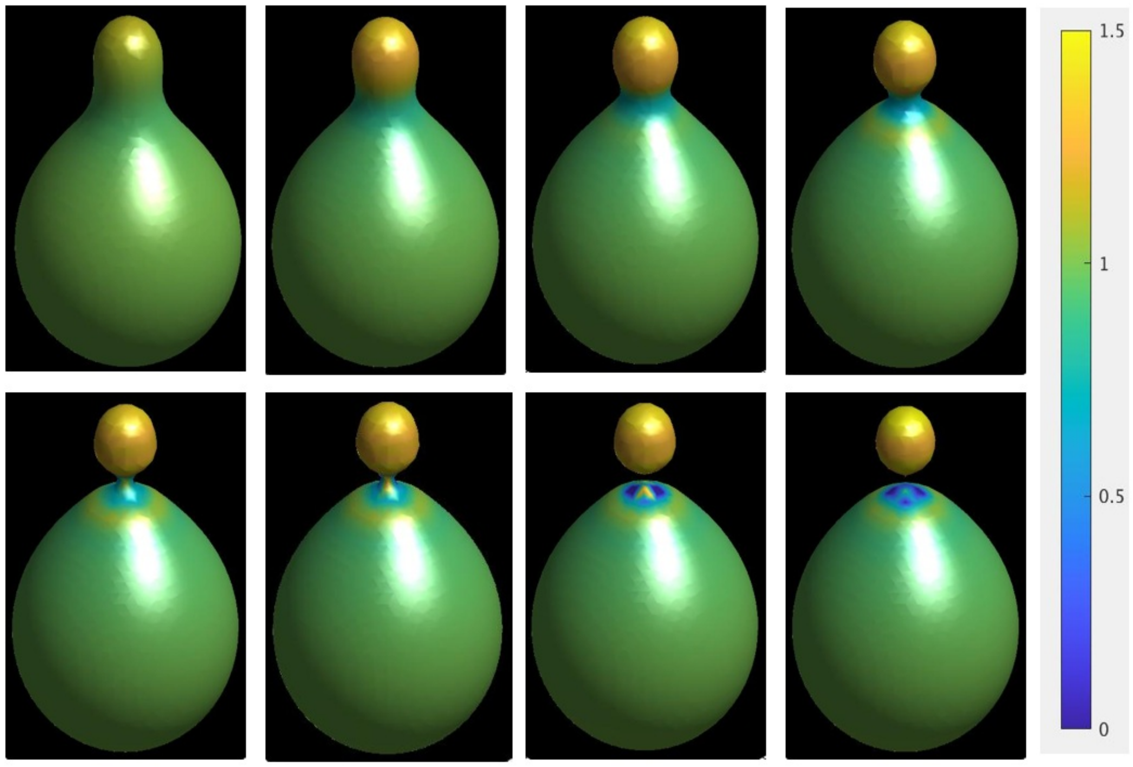

Figure 6.22. Protein diffusion and advection due to curvature generation and sensing on a pear shaped vesicle help scission of the vesicle. Colorbar on the right indicates values of c_p on the surface. Numerical time for each subgraph (from left to right, from top to bottom): $3.95\times 10^{\ 6}, 2.41\times 10^{\ 5}, 3.94\times 10^{\ 5}, 5.51\times 10^{\ 5}$, $6.19\times 10^{\ 5}, 6.25\times 10^{\ 5}, 6.34\times 10^{\ 5}, 6.47\times 10^{\ 5}$.

6.3.3 Coupling Between Shape Dynamics of Biphasic Vesicle and Protein Kinetics

In this section, we take a look at how protein dependent bending moduli and spontaneous curvature will influence shape dynamics of biphasic vesicles. In particular, we will look at how protein advection affects scission of the biphasic vesicles. The Hamiltonian $H_{\rm pb}$ is almost the same with Eq. (6.15) for the biphasic vesicles with the additional assumption that bending moduli $\kappa^{\rm Ld(Lo)}$ and the spontaneous curvatures $C^{\rm Ld(Lo)}$ are dependent on density of protein molecules. A linear model is explored in this section

$$\kappa^{\rm Ld}(c_p) = \kappa_0^{\rm Ld} + \kappa_1^{\rm Ld} c_p, \kappa^{\rm Lo}(c_p) = \kappa_0^{\rm Lo} + k_1^{\rm Lo} c_p, C^{\rm Ld} = C^{\rm Lo} = C_0 + C_1 c_p. \tag{6.45}$$

which will add extra terms to the surface bending force $\boldsymbol{f}_{\rm su}$ and line forces $\boldsymbol{g}_{\rm bo}$ in Eqs. (6.21,6.22). The normal speed $v_{\rm su}$ of the surface, the normal speed $v_{\rm bo}$ of the phase boundary and the advection velocity $\boldsymbol{v}_c$ for c are still computed with Eqs. (6.23,6.24,6.31). The extra tangential velocity $\boldsymbol{v}_{\rm xp}$ for protein molecules results from Eqs. (6.43,6.44) with $H_{\rm ps}$ replaced by $H_{\rm pb}$. The tangential advection velocity for protein molecules is $\boldsymbol{v}_p = \boldsymbol{v}_c + \boldsymbol{v}_{\rm xp}$. The dynamics of the whole system includes the level set equation Eq. (6.29) for ϕ, the evolution equation Eq. (6.30) of the auxiliary level set function ψ, mass conservation Eq. (6.32) for c and Eq. (6.39) for c_p. Regularization of the level set functions ϕ and ψ is handled in the same manner. The entire scheme is the same as the one from section 6.2.1, with additional steps for protein advection.

Inspired by the experiments of [rossman2010influenza], we let a phase separated vesicle bend under the effects of line tension for a certain steps with constant $\kappa^{\rm Ld}, \kappa^{\rm Lo}, C^{\rm Ld}, C^{\rm Lo}$, then we turn on dynamics for c and c_p by setting setting C_1 to be nonzero and adding a constant initial profile for c and c_p. The purpose is to simulate how phase separated vesicle will respond to curvature generation and sensing of protein molecules. In the first example, at the start, we set $\kappa^{\rm Ld} = 1$, $\kappa^{\rm Lo} = 5, \kappa_l = 50, \kappa_G^{\rm Ld} \quad \kappa_G^{\rm Lo} = 3.6$, $C^{\rm Ld} = C^{\rm Lo} = 0$, ratio of surface area for the Ld phase be 0.1, pressure

$P = 100$. The Lagrange multipliers $\sigma^{\pm}$ are determined from conservation of area for each phase. Note that here we used experimental data from [baumgart2005membrane] to set values for $\kappa^{\mathrm{Ld/Lo}}$, $\kappa_l, \kappa_G^{\mathrm{Ld}} \quad \kappa^{\mathrm{Lo}}$. We did not prescribe a reduced volume here. Instead, a constant osmotic pressure P is used, which is due to experimental observation of changes of volume during deformation of biphasic vesicles [rossman2010influenza]. We let the vesicle deform with those parameters for 200 steps. The resulting shape is the shown in Figures 6.23 and 6.24 as the starting point for our simulation. The dynamics in the first 200 steps is very similar to Figure 6.12 and is omitted here. Then we set $c_p(S, t = 0) = 1, c(S, t = 0) = 1, \mu = 1000, C^{\mathrm{Ld}} = C^{\mathrm{Lo}} = C_0 + C_1 c_p$ and start solving dynamics for c and c_p in addition to that of ϕ and ψ. For Figure 6.23, we used $C_0 = 0, C_1 = \quad 5$ while for Figure 6.24 $C_0 = 0, C_1 = \quad 1$. In Figure 6.23, protein molecules are advected to the region of high curvature and depleted near the neck region. Then as the curvature of the neck region continue to increase as it is getting narrower, density of protein increases here. Finally, part of the Ld phase (colored in red) buds off from the mother vesicle. This result is very similar to the two dimensional simulation of [lowengrub2009phase] but at odds with experimental findings of [rossman2010influenza] where the Ld phase buds off exactly at the phase boundary. Shape dynamics in Figure 6.24 is very similar to that in Figure 6.23, except that scission happens at the phase boundary which can be qualitatively compared with experiments in [rossman2010influenza]. In Figure 6.23, scission of vesicle is dominated by curvature generation molecules which resembles scission in Figures 6.4 and 6.5. In Figure 6.24, addition of curvature generation molecules help mediate scission through line tension and the physics is more related to the simulation presented in Figures 6.14,6.15,6.19,6.20. We also noticed in Figure 6.24, it takes a much longer time before scission happens. This can be a result of slower advection due to smaller C_1.

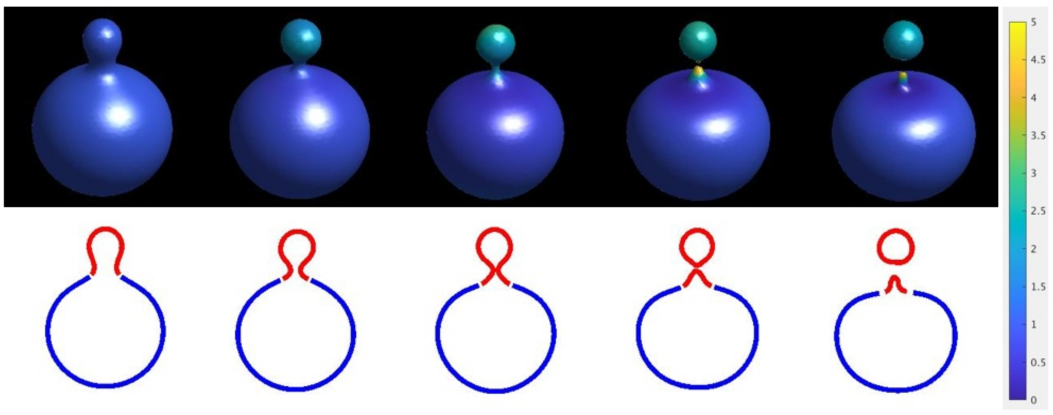

Figure 6.23. The Ld phase of a biphasic vesicle buds off due to curvature generation molecules. Numerical time for each frame from left to right: $7.45 \times 10^{\ 3}, 8.67 \times 10^{\ 3}, 1.01 \times 10^{\ 2}, 1.11 \times 10^{\ 2}, 1.14 \times 10^{\ 2}$.

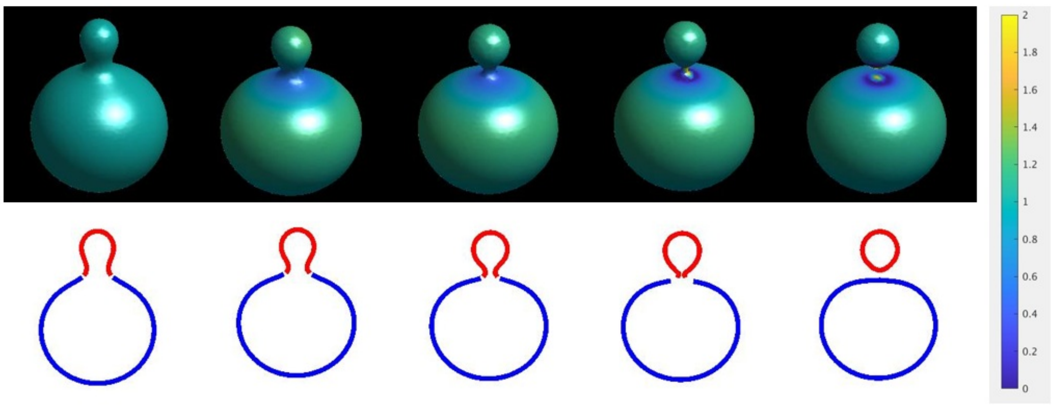

Figure 6.24. Curvature generation molecules help facilitate budding of biphasic vesicles. Numerical time for each frame from left to right: $7.45 \times 10^{\ 3}, 5.65 \times 10^{\ 2}, 5.71 \times 10^{\ 2}, 5.74 \times 10^{\ 2}, 5.74 \times 10^{\ 2}$.

The above numerical experiments shows the capability of our numerical scheme. A more thorough exploration of different models for protein membrane interaction and more detailed comparison with experimental data will be the topic of future research.

Chapter 7
Conclusion

In this thesis, we made some efforts towards a comphrehensive understanding of vesicle shape dynamics. In particular, we calculated variations for a Hamiltonian for lipid bilayer vesicles with higher order curvature energy from [deserno2015fluid], spatially varying bending coefficients, and a more general line energy term. The resulting elastic forces can be used in descriptions of more general elastic surfaces. We then showed how those invariant forces can be calculated in the cartesian coordinate within the framework of level set methods. We developed a sixth-order accurate scheme for Hamilton-Jacobi equations with level set-defined boundary conditions, which allows accurate simulation of elastic surfaces and can find many applications in other sciecne and engineering problems. We developed a semi-implicit numerical scheme [smereka2003semi] for the dynamics of single phase vesicles and explored effects of spontaneous curvature, osmotic pressure and constrained reduced area difference on the equilibrium shape of single phase vesicles. We found that large spontaneous curvatures lead to scission of vesicles and under an osmotic pressure, cylindrically symmetric vesicles will lose this symmetry and develop complex morphologies. Many of those intriguing shapes have been observed experimentally. We emphasize here the importance of volume constrains in the dynamics of closed vesicles.

We then borrowed ideas from [cheng2002motion] to develop a numerical scheme for biphasic vesicles by representing the phase boundary as a codimension-2 level set in three dimensional space. This idea is also used in the phase-filed simulation of biphasic vesicles. The level set formulation has the advantage of representing arbitrary complex models and evolve the system with physical forces rather than simply time-step the system in a thermodynamically consistent way. For instance, phase-filed formulation is unable to account for effects of different Gaussian bending moduli of different phases, while our level set formulation can easily take this difference into account and include more general forces from Hamiltonians in Eqs. (3.1,3.2). As far as we know, no other fully three dimensional numerical schemes can take into accounts of all of those effects beyond the canonical Helfrich model. We implemented two methods to regularize the auxiliary level set function. When there is no pinching along the phase boundary, we calculate the geodesics to the phase boundary periodically with the numerical scheme from [zhang2020sixth]. In the presence of singularities, we borrowed techniques from the distance regularized level set method [li2005level, li2010distance] to maintain stability of our scheme. We then investigated how prescribed reduced volume, line tension, spontaneous curvature, ratio of different phases of a biphasic vesicle affects pinching of bidomain biphasic vesicles. A penalty term is used to impose incompressibility of the vesicle. Interplay of domain fusion and shape dynamics of multidomain vesicles is presented. Outward and inward budding of multidomain vesicles under constant osmotic pressure is observed numerically, which is inspired by experiments from [yanagisawa2008shape].

Finally, a simple numerical model for curvature generation and curvature sensing of molecules on the membrane is proposed. Numerical experiments are designed and implemented to show how dependence of bending moduli and spontaneous curvature on protein density redistribute proteins on regions of different curvatures. Then we combined protein advection and shape dynamics of single phase vesicles and biphasic vesicle and investigated pinching of pear shaped vesicles and biphasic vesicles in the presence of protein molecules. It was found that curvature generation and sensing of protein molecules can facilitate scission of budded regions of the membrane. Two types of pinching for biphasic vesicles are found. When large curvature are induced by protein molecules, scission of lipid rafts happens away from the phase boundary. While protein molecules can generate a relatively small curvature, less sensitive curvature sensing lead to slow advection and longer time are needed before complete budding of phase separated region. This is a starting point for more quantitative researches on scission of biphasic vesicles.

Overall, we developed necessary theoretical tools and numerical schemes to simulate various shape dynamics of vesicles. Our numerical experiments presented here shows the capability of those tools, which can be used to answer many other interesting question, for instance, effects of the difference of Gaussian bending moduli, higher order curvature energy, geodesic torsion and normal curvature of phase boundaries. Still, more can be done to enhance our methods. A better scheme to impose incompressibility and to solve conservation law on a curved surface that may undergo large deformation is needed to have a more accurate dynamics for c and c_p. More physical insights may be gained if the full Navier-Stokes equations are used to compute advection speed. These will be left for future research.

Bibliography

[adalsteinsson1995fast] David Adalsteinsson and James A Sethian. A fast level set method for propagating interfaces. *Journal of computational physics*, 118(2):269–277, 1995.

[adalsteinsson1999fast] David Adalsteinsson and James A Sethian. The fast construction of extension velocities in level set methods. *Journal of Computational Physics*, 148(1):2–22, 1999.

[adalsteinsson2003transport] David Adalsteinsson and James A Sethian. Transport and diffusion of material quantities on propagating interfaces via level set methods. *Journal of Computational Physics*, 185(1):271–288, 2003.

[aland2014diffuse] Sebastian Aland, Sabine Egerer, John Lowengrub, and Axel Voigt. Diffuse interface models of locally inextensible vesicles in a viscous fluid. *Journal of computational physics*, 277:32–47, 2014.

[alberts2014molecular] B. Alberts, A.D. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell: Sixth International Student Edition*. 500 Tips. Garland Science, Taylor and Francis Group, 2014.

[bardi1991nonconvex] Martino Bardi and Stanley Osher. The nonconvex multidimensional riemann problem for hamilton–jacobi equations. *SIAM Journal on Mathematical Analysis*, 22(2):344–351, 1991.

[baumgart2011thermodynamics] Tobias Baumgart, Benjamin R Capraro, Chen Zhu, and Sovan L Das. Thermodynamics and mechanics of membrane curvature generation and sensing by proteins and lipids. *Annual review of physical chemistry*, 62:483–506, 2011.

[baumgart2005membrane] Tobias Baumgart, S Das, Watt Wetmore Webb, and James Thomas Jenkins. Membrane elasticity in giant vesicles with fluid phase coexistence. *Biophysical journal*, 89(2):1067–1080, 2005.

[bertalmio2001variational] Marcelo Bertalmıo, Li-Tien Cheng, Stanley Osher, and Guillermo Sapiro. Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, 174(2):759–780, 2001.

[bertalmio1999region] Marcelo Bertalmio, Guillermo Sapiro, and Gregory Randall. Region tracking on level-sets methods. *IEEE Transactions on Medical Imaging*, 18(5):448–451, 1999.

[biben2005phase] Thierry Biben, Klaus Kassner, and Chaouqi Misbah. Phase-field approach to three-dimensional vesicle dynamics. *Physical Review E*, 72(4):41921, 2005.

[burchard2001motion] Paul Burchard, Li-Tien Cheng, Barry Merriman, and Stanley Osher. Motion of curves in three spatial dimensions using a level set approach. *Journal of Computational Physics*, 170(2):720–741, 2001.

[capovilla2002lipid] R Capovilla, J Guven, and JA Santiago. Lipid membranes with an edge. *Physical Review E*, 66(2):21607, 2002.

[capovilla2002stresses] Riccardo Capovilla and Jemal Guven. Stresses in lipid membranes. *Journal of Physics A: Mathematical and General*, 35(30):6233, 2002.

[cheng2002motion] Li-Tien Cheng, Paul Burchard, Barry Merriman, and Stanley Osher. Motion of curves constrained on surfaces using a level-set approach. *Journal of Computational Physics*, 175(2):604–644, 2002.

[chopp2001some] David L Chopp. Some improvements of the fast marching method. *SIAM Journal on Scientific Computing*, 23(1):230–244, 2001.

[chopp2009another] David L Chopp. Another look at velocity extensions in the level set method. *SIAM Journal on Scientific Computing*, 31(5):3255–3273, 2009.

[chopp1991computing] David Layne Chopp. Computing minimal surfaces via level set curvature flow. Technical Report, Elsevier, 1991.

[chopp1999motion] David Chopp and James A Sethian. Motion by intrinsic laplacian of curvature. *Interfaces and Free boundaries*, 1(1):107–123, 1999.

[coquerelle2016fourth] Mathieu Coquerelle and Stéphane Glockner. A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces. *Journal of Computational Physics*, 305:838–876, 2016.

[crane2013geodesics] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. Geodesics in heat: a new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)*, 32(5):152, 2013.

[deserno2015fluid] Markus Deserno. Fluid lipid membranes: from differential geometry to curvature stresses. *Chemistry and physics of lipids*, 185:11–45, 2015.

[du2008second] Antoine du Chéné, Chohong Min, and Frédéric Gibou. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *Journal of Scientific Computing*, 35(2-3):114–131, 2008.

[du2006simulating] Qiang Du, Chun Liu, and Xiaoqiang Wang. Simulating the deformation of vesicle membranes under elastic bending energy in three dimensions. *Journal of computational physics*, 212(2):757–777, 2006.

[duchemin2014explicit] Laurent Duchemin and Jens Eggers. The explicit–implicit–null method: removing the numerical instability of pdes. *Journal of Computational Physics*, 263:37–52, 2014.

[engquist2005discretization] Björn Engquist, Anna-Karin Tornberg, and Richard Tsai. Discretization of dirac delta functions in level set methods. *Journal of Computational Physics*, 207(1):28–51, 2005.

[fasshauer2007meshfree] Gregory E Fasshauer. *Meshfree approximation methods with MATLAB*, volume 6. World Scientific, 2007.

[goldman2005curvature] Ron Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7):632–658, 2005.

[greer2006fourth] John B Greer, Andrea L Bertozzi, and Guillermo Sapiro. Fourth order partial differential equations on general geometries. *Journal of Computational Physics*, 216(1):216–246, 2006.

[grinfeld2013introduction] Pavel Grinfeld. *Introduction to tensor analysis and the calculus of moving surfaces*. Springer, 2013.

[guckenberger2016bending] Achim Guckenberger, Marcel P Schraml, Paul G Chen, Marc Leonetti, and Stephan Gekle. On the bending algorithms for soft objects in flows. *Computer Physics Communications*, 207:1–23, 2016.

[harayama2018understanding] Takeshi Harayama and Howard Riezman. Understanding the diversity of membrane lipid composition. *Nature Reviews Molecular Cell Biology*, 19(5):281, 2018.

[harten1987uniformly] Ami Harten and Stanley Osher. Uniformly high-order accurate nonoscillatory schemes. i. *SIAM Journal on Numerical Analysis*, 24(2):279–309, 1987.

[helfrich1973elastic] Wolfgang Helfrich. Elastic properties of lipid bilayers: theory and possible experiments. *Zeitschrift für Naturforschung C*, 28(11-12):693–703, 1973.

[hw2002stomatocyte] Gerald Lim HW, Michael Wortis, and Ranjan Mukhopadhyay. Stomatocyte–discocyte–echinocyte sequence of the human red blood cell: evidence for the bilayer–couple hypothesis from membrane mechanics. *Proceedings of the National Academy of Sciences*, 99(26):16766–16769, 2002.

[jiang2000weighted] Guang-Shan Jiang and Danping Peng. Weighted eno schemes for hamilton–jacobi equations. *SIAM Journal on Scientific computing*, 21(6):2126–2143, 2000.

[jiang1996efficient] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1):202–228, 1996.

[jin2006measuring] Albert J Jin, Kondury Prasad, Paul D Smith, Eileen M Lafer, and Ralph Nossal. Measuring the elasticity of clathrin-coated vesicles via atomic force microscopy. *Biophysical journal*, 90(9):3333–3344, 2006.

[julicher1996shape] Frank Jülicher and Reinhard Lipowsky. Shape transformations of vesicles with intramembrane domains. *Physical Review E*, 53(3):2670, 1996.

[laadhari2017fully] Aymen Laadhari, Pierre Saramito, Chaouqi Misbah, and Gábor Székely. Fully implicit methodology for the dynamics of biomembranes and capillary interfaces by combining the level set and newton methods. *Journal of Computational Physics*, 343:271–299, 2017.

[stanfordbunny] Stanford Computer Graphics Laboratory. The Stanford 3D Scanning Repository. `http://graphics.stanford.edu/data/3Dscanrep/`, 2014. Accessed: 2020-02-18.

[li2005level] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Level set evolution without re-initialization: a new variational formulation. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 430–436. IEEE, 2005.

[li2010distance] Chunming Li, Chenyang Xu, Changfeng Gui, and Martin D Fox. Distance regularized level set evolution and its application to image segmentation. *IEEE transactions on image processing*, 19(12):3243–3254, 2010.

[lingwood2010lipid] Daniel Lingwood and Kai Simons. Lipid rafts as a membrane-organizing principle. *Science*, 327(5961):46–50, 2010.

[lipnikov2019high] Konstantin Lipnikov and Nathaniel Morgan. A high-order discontinuous galerkin method for level set problems on polygonal meshes. *Journal of Computational Physics*, 397:108834, 2019.

[liu1994weighted] Xu-Dong Liu, Stanley Osher, Tony Chan et al. Weighted essentially non-oscillatory schemes. *Journal of computational physics*, 115(1):200–212, 1994.

[lowengrub2016numerical] John Lowengrub, Jun Allard, and Sebastian Aland. Numerical simulation of endocytosis: viscous flow driven by membranes with non-uniformly distributed curvature-inducing molecules. *Journal of computational physics*, 309:112–128, 2016.

[lowengrub2009phase] John S Lowengrub, Andreas Rätz, and Axel Voigt. Phase-field modeling of the dynamics of multicomponent vesicles: spinodal decomposition, coarsening, budding, and fission. *Physical Review E*, 79(3):31926, 2009.

[macdonald2008level] Colin B Macdonald and Steven J Ruuth. Level set equations on surfaces via the closest point method. *Journal of Scientific Computing*, 35(2-3):219–240, 2008.

[macdonald2009implicit] Colin B Macdonald and Steven J Ruuth. The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM Journal on Scientific Computing*, 31(6):4330–4350, 2009.

[martyna2016curvature] Agnieszka Martyna, Jordi Gomez-Llobregat, Martin Linde´n, and Jeremy S Rossman. Curvature sensing by a viral scission protein. *Biochemistry*, 55(25):3493–3496, 2016.

[mcmahon2005membrane] Harvey T McMahon and Jennifer L Gallop. Membrane curvature and mechanisms of dynamic cell membrane remodelling. *Nature*, 438(7068):590–596, 2005.

[miao1994budding] Ling Miao, Udo Seifert, Michael Wortis, and Hans-Günther Döbereiner. Budding transitions of fluid-bilayer vesicles: the effect of area-difference elasticity. *Physical Review E*, 49(6):5389, 1994.

[min2010reinitializing] Chohong Min. On reinitializing level set functions. *Journal of computational physics*, 229(8):2764–2772, 2010.

[min2007second] Chohong Min and Frédéric Gibou. A second order accurate level set method on non-graded adaptive cartesian grids. *Journal of Computational Physics*, 225(1):300–321, 2007.

[morris2019solvated] RG Morris, TR Dafforn, and MS Turner. Solvated membrane nanodiscoids: a probe for the effects of gaussian curvature. *ArXiv preprint arXiv:1904.00710*, 2019.

[mouritsen2005life] Ole G Mouritsen. *Life-as a matter of fat*. Springer, 2005.

[noyhouzer2016ferrocene] Tomer Noyhouzer, Chloe´ L'Homme, Isabelle Beaulieu, Stephanie Mazurkiewicz, Sabine Kuss, Heinz-Bernhard Kraatz, Sylvain Canesi, and Janine Mauzeroll. Ferrocene-modified phospholipid: an innovative precursor for redox-triggered drug delivery vesicles selective to cancer cells. *Langmuir*, 32(17):4169–4178, 2016.

[osher2006level] Stanley Osher and Ronald Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006.

[osher1988fronts] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[peng1999pde] Danping Peng, Barry Merriman, Stanley Osher, Hongkai Zhao, and Myungjoo Kang. A pde-based fast local level set method. *Journal of computational physics*, 155(2):410–438, 1999.

[rossman2010influenza] Jeremy S Rossman, Xianghong Jing, George P Leser, and Robert A Lamb. Influenza virus m2 protein mediates escrt-independent membrane scission. *Cell*, 142(6):902–913, 2010.

[russo2000remark] Giovanni Russo and Peter Smereka. A remark on computing distance functions. *Journal of Computational Physics*, 163(1):51–67, 2000.

[ruuth2008simple] Steven J Ruuth and Barry Merriman. A simple embedding method for solving partial differential equations on surfaces. *Journal of Computational Physics*, 227(3):1943–1961, 2008.

[salac2011level] David Salac and M Miksis. A level set projection model of lipid vesicles in general flows. *Journal of Computational Physics*, 230(22):8192–8215, 2011.

[schmidt2013influenza] Nathan W Schmidt, Abhijit Mishra, Jun Wang, William F DeGrado, and Gerard CL Wong. Influenza virus a m2 protein generates negative gaussian membrane curvature necessary for budding and scission. *Journal of the American Chemical Society*, 135(37):13710–13719, 2013.

[seifert1997configurations] Udo Seifert. Configurations of fluid membranes and vesicles. *Advances in physics*, 46(1):13–137, 1997.

[sethian1996fast] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[sezgin2017] Erdinc Sezgin, Ilya Levental, Satyajit Mayor, and Christian Eggeling. The mystery of membrane organization: composition, regulation and roles of lipid rafts. *Nat. Rev. Cell Biol.*, 18(1):361–374, 2017.

[shu1988efficient] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, 77(2):439–471, 1988.

[siegel2010fourth] David P Siegel. Fourth-order curvature energy model for the stability of bicontinuous inverted cubic phases in amphiphile- water systems. *Langmuir*, 26(11):8673–8683, 2010.

[smereka2003semi] Peter Smereka. Semi-implicit level set methods for curvature and surface diffusion motion. *Journal of Scientific Computing*, 19(1):439–456, 2003.

[smereka2006numerical] Peter Smereka. The numerical approximation of a delta function with application to level set methods. *Journal of Computational Physics*, 211(1):77–90, 2006.

[smit2005grid] J Smit, M van Sint Annaland, and JAM Kuipers. Grid adaptation with weno schemes for non-uniform grids to solve convection-dominated partial differential equations. *Chemical engineering science*, 60(10):2609–2619, 2005.

[sohn2010dynamics] Jin Sun Sohn, Yu-Hau Tseng, Shuwang Li, Axel Voigt, and John S Lowengrub. Dynamics of multicomponent vesicles in a viscous fluid. *Journal of Computational Physics*, 229(1):119–144, 2010.

[strain1999fast] John Strain. Fast tree-based redistancing for level set computations. *Journal of Computational Physics*, 152(2):664–686, 1999.

[sussman1994level] Mark Sussman, Peter Smereka, and Stanley Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational physics*, 114(1):146–159, 1994.

[taniguchi1996shape] Takashi Taniguchi. Shape deformation and phase separation dynamics of two-component vesicles. *Physical review letters*, 76(23):4444, 1996.

[tornberg2004numerical] Anna-Karin Tornberg and Björn Engquist. Numerical approximations of singular source terms in differential equations. *Journal of Computational Physics*, 200(2):462–488, 2004.

[towers2007two] John D Towers. Two methods for discretizing a delta function supported on a level set. *Journal of Computational Physics*, 220(2):915–931, 2007.

[towers2008convergence] John D Towers. A convergence rate theorem for finite difference approximations to delta functions. *Journal of Computational Physics*, 227(13):6591–6597, 2008.

[towers2009discretizing] John D Towers. Discretizing delta functions via finite differences and gradient normalization. *Journal of Computational Physics*, 228(10):3816–3836, 2009.

[towers2009finite] John D Towers. Finite difference methods for approximating heaviside functions. *Journal of Computational Physics*, 228(9):3478–3489, 2009.

[tu2003lipid] ZC Tu and ZC Ou-Yang. Lipid membranes with free edges. *Physical Review E*, 68(6):61915, 2003.

[veatch2003separation] Sarah L Veatch and Sarah L Keller. Separation of liquid phases in giant vesicles of ternary mixtures of phospholipids and cholesterol. *Biophysical journal*, 85(5):3074–3083, 2003.

[wang2008modelling] Xiaoqiang Wang and Qiang Du. Modelling and simulations of multi-component lipid membranes and open membranes via diffuse interface approaches. *Journal of mathematical biology*, 56(3):347–371, 2008.

[xu2003eulerian] Jian-Jun Xu and Hong-Kai Zhao. An eulerian formulation for solving partial differential equations along a moving interface. *Journal of Scientific Computing*, 19(1-3):573–594, 2003.

[yanagisawa2008shape] Miho Yanagisawa, Masayuki Imai, and Takashi Taniguchi. Shape deformation of ternary vesicles coupled with phase separation. *Physical review letters*, 100(14):148102, 2008.

[zhang2020sixth] Tiankui Zhang and Charles W Wolgemuth. Sixth-order accurate schemes for reinitialization and extrapolation in the level set framework. *Journal of Scientific Computing*, 83(2), 2020.

[zhao1996variational] Hong-Kai Zhao, Tony Chan, Barry Merriman, and Stanley Osher. A variational level set approach to multiphase motion. *Journal of computational physics*, 127(1):179–195, 1996.

[zhong1989bending] Ou-Yang Zhong-Can and Wolfgang Helfrich. Bending energy of vesicle membranes: general expressions for the first, second, and third variation of the shape energy and applications to spheres and cylinders. *Physical Review A*, 39(10):5280, 1989.