Adaptive Spline-based Finite Element Method with Application to Phase-field Models of Biomembranes

by

Wen Jiang

Department of Mechanical Engineering and Materials Science
Duke University

Date: ____________________
Approved:

John Dolbow, Supervisor

Thomas Witelski

Wilkins Aquino

Guglielmo Scovazzi

Laurens Howle

Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mechanical Engineering and Materials Science in the Graduate School of Duke University 2015
Abstract

Adaptive Spline-based Finite Element Method with Application to Phase-field Models of Biomembranes

by

Wen Jiang

Department of Mechanical Engineering and Materials Science
Duke University

Date: ______________________

Approved:

___________________________

John Dolbow, Supervisor

___________________________

Thomas Witelski

___________________________

Wilkins Aquino

___________________________

Guglielmo Scovazzi

___________________________

Laurens Howle

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mechanical Engineering and Materials Science in the Graduate School of Duke University 2015
Abstract

Interfaces play a dominant role in governing the response of many biological systems and they pose many challenges to model and simulate with numerical methods. For sharp-interface models, for example, traditional finite element methods require the mesh to align with surfaces of discontinuities. Diffuse-interface models approximate the sharp interface with continuous variations of an order parameter, resulting in significant computational effort. To overcome these difficulties, we focus on developing a computationally efficient spline-based finite element method for interface problems.

A key challenge while employing B-spline basis functions in finite-element methods is the robust imposition of Dirichlet boundary conditions. We begin by examining weak enforcement of such conditions for B-spline basis functions, with application to both second- and fourth-order problems based on Nitsche’s approach. The use of spline-based finite elements is further examined along with a Nitsche technique for enforcing constraints on an embedded interface. We show how the choice of weights and stabilization parameters in the Nitsche consistency terms has a great influence on the accuracy and robustness of the method. For curved interfaces, to obtain optimal rates of convergence we employ a hierarchical local refinement approach to improve the geometrical representation of interface.

In multiple dimensions, a spline basis is obtained as a tensor product of the one-dimensional basis. This necessitates a rectangular grid that cannot be refined locally in regions of embedded interfaces. To address this issue, we develop an
adaptive spline-based finite element method that employs hierarchical refinement and coarsening techniques. The process of refinement and coarsening guarantees linear independence and retains the regularity of the basis functions. We further propose an efficient data transfer algorithm during both refinement and coarsening which yields accurate results.

The adaptive approach is applied to vesicle modeling which allows three-dimensional simulation to proceed efficiently. In this work, we employ a continuum approach to model the evolution of microdomains on the surface of Giant Unilamellar Vesicles. The chemical energy is described by a Cahn-Hilliard type density functional that characterizes the line energy between domains of different species. The generalized Canham-Helfrich-Evans model provides a description of the mechanical energy of the vesicle membrane. This coupled model is cast in a diffuse-interface form using the phase-field framework. The effect of coupling is seen through several numerical examples of domain formation coupled to vesicle shape changes.
Contents

Abstract iv
List of Tables x
List of Figures xi
List of Abbreviations and Symbols xvi
Acknowledgements xix

1 Introduction 1
  1.1 Cell membranes ......................................................... 2
  1.2 Continuum approach of vesicle modeling .......................... 3
    1.2.1 Canham-Helfrich-Evans model ................................. 3
    1.2.2 Phase separation in GUVs ...................................... 4
  1.3 Computational challenges .......................................... 5
    1.3.1 Local refinement in spline-based finite elements .......... 6
    1.3.2 Extended spline-based finite elements ........................ 7
  1.4 Overview of the thesis .............................................. 8

2 Spline-based Finite Elements 10
  2.1 Overview ............................................................... 10
  2.2 B-spline basis functions .......................................... 11
    2.2.1 Finite element basis with B-splines .......................... 12
    2.2.2 Dirichlet boundary conditions ............................... 14
2.3 Modeling second- and fourth-order problems with spline-based finite elements.............................................................................................................. 14
  2.3.1 Second-order boundary value problem ........................................................................................................ 15
  2.3.2 Fourth-order boundary value problem ..................................................................................................... 18
  2.3.3 Cahn–Hilliard equation ......................................................................................................................... 22

2.4 Summary .................................................................................................................................................. 24

3 A Robust Nitsche’s Formulation for Interface Problems with Spline-based Finite Elements 27
  3.1 Introduction .............................................................................................................................................. 27
  3.2 Model problem and variational formulation ............................................................................................. 30
    3.2.1 Weak Form .................................................................................................................................... 31
  3.3 Discretization with B-splines .................................................................................................................... 32
  3.4 Geometrical representation ...................................................................................................................... 34
  3.5 Analysis ................................................................................................................................................ 37
    3.5.1 Conditions for coercivity .................................................................................................................. 38
    3.5.2 Stability Parameter Evaluation ........................................................................................................ 39
  3.6 Numerical examples ................................................................................................................................. 44
    3.6.1 Fixed Circular Plate ......................................................................................................................... 45
    3.6.2 Kidney-shaped interface .................................................................................................................. 47
    3.6.3 Planar Interface ............................................................................................................................. 49
    3.6.4 Petal problem ............................................................................................................................... 51
    3.6.5 Bending of an elongated cantilevered beam .................................................................................... 54
  3.7 Summary ................................................................................................................................................ 57

4 Adaptive Refinement of Hierarchical Spline-based Finite Elements 59
  4.1 Introduction .............................................................................................................................................. 59
  4.2 Hierarchical B-splines ............................................................................................................................ 62
List of Tables

1.1 Refinement techniques and their properties. source: Vuong (2012) . . . 7
3.1 Refinement level for cubic B-spline basis function. . . . . . . . . . . . 44
4.1 The Poisson equation example: data transfer . . . . . . . . . . . . . . . 84
List of Figures

1.1 Giant Unilamellar Vesicles showing $L_o - L_d$ phase coexistence. The two lipid domains are fluoroscently labeled. Source: Baumgart et al. (2003a) ................................................................. 3

2.1 $C^2$ continuous cubic spline basis functions resulting from a uniform knot vector. ................................................................. 12

2.2 (a) Identification node for a bi-cubic spline basis function and (b) Computational domain (in grey) for the given geometry. .............. 13

2.3 Domain $\Omega$ bounded by Dirichlet ($\Gamma_d$) and Neumann ($\Gamma_h$) boundaries, with unit outward normal $\mathbf{n}$. ................................. 15

2.4 Solution to the 3D Poisson problem ........................................... 17

2.5 Convergence rates for the 3D Poisson problem in (a) $L_2$ norm and (b) energy norm. ................................................................. 18

2.6 Solution to the 3D biharmonic problem (first case). ..................... 20

2.7 Convergence rates for the 3D biharmonic (first case) in (a) $L_2$ and (b) energy norms. ................................................................. 21

2.8 Solution to the 3D biharmonic problem (second case). .................. 21

2.9 Convergence rates for the 3D biharmonic (second case) in (a) $L_2$ and (b) energy norms. ................................................................. 22

2.10 The Cahn-Hilliard equation example for $\bar{u} = 0.0$: evolution of the field with the $u(x, t) = 0.0$ iso-contours under coarsening. .............. 25

2.11 The Cahn-Hilliard equation example for $\bar{u} = 0.85$: evolution of the field with the $u(x, t) = 0.0$ iso-contours under coarsening. .............. 26

3.1 A domain $\Omega$ partitioned into regions $\Omega^1$ and $\Omega^2$ by the interface $\Gamma_s$. 30
3.2 The domain is divided in two sub-domains $\Omega^1$ and $\Omega^2$ by an embedded surface $\Gamma$. The black circles are the duplicated nodes whose support are cutted by the interface. 33

3.3 Overlapping element formulation for a quadrilateral element cut by an embedded interface. By construction, each element is associated with a $4 \times 4$ grid of bicubic B-splines. The black circles are the physical nodes corresponding to the background mesh and the hollow circles are the ghost nodes. 34

3.4 Geometry mismatch by using linear level set. 35

3.5 Mesh of the curved interface(red): elements cut by the interface are refined recursively, improving the geometrical representation. 35

3.6 (a) Discretization for the fixed circular plate with uniform loading. (b) Finite element approximation. 46

3.7 Convergence rates in $L_2$ norm. 46

3.8 Local geometric refinement and results for the kidney-shaped problem. 47

3.9 Comparison of convergence rates for the kidney-shaped problem in $L_2$ norm. 48

3.10 Interfacial location (red dashed line) for two versions of the planar interface problem. 49

3.11 Sensitivity in the bulk and flux error with degree of heterogeneity. The first row shows the comparison between the classical Nitsche method and the higher order $\gamma$-Nitsche method, and the second row shows the comparison between $\gamma$-Nitsche and higher order $\gamma$-Nitsche. 51

3.12 Sensitivity in the bulk and flux error with interfacial location ($\kappa^2/\kappa^1 = 10^{-6}$). The first row shows the comparison between classical Nitsche and higher order $\gamma$-Nitsche and the second row shows the comparison between $\gamma$-Nitsche and higher order $\gamma$-Nitsche. 52

3.13 Sensitivity in the bulk and flux error with interfacial location($\kappa^2/\kappa^1 = 10^{6}$). The first row shows the comparison between classical Nitsche and higher order $\gamma$-Nitsche and the second row shows the comparison between $\gamma$-Nitsche and higher order $\gamma$-Nitsche. 53

3.14 Local geometric refinement and results for the petal problem. 54
3.15 Convergence rates in $L_2$ norm for the petal problem. The first row shows the case of $\kappa^2/\kappa^1 = 10^{-6}$ and the second row shows the case of $\kappa^2/\kappa^1 = 10^6$. ................................................................. 55

3.16 Geometry and loading for the elongated cantilevered beam. ............ 56

3.17 Discretization and interface location ........................................... 57

3.18 Convergence rates in $L_2$ norm for the plane strain elasticity problem. 57

3.19 Comparison of surface tractions obtained using classical Nitsche with those obtained using higher order $\gamma$ Nitsche. ......................... 57

4.1 Subdivision of an original uniform B-spline into $p + 2$ contracted B-splines of half the knot span width, illustrated for polynomial degrees $p = 1$ through 3. ................................................................. 62

4.2 Hierarchical refinement of a set of B-spline basis functions (active basis functions are indicated with solid lines). ................................. 64

4.3 The weighted hierarchical basis: the sum of all the basis function values is 1 (the bottom image shows all active basis functions on both levels 0 and 1). ................................................................. 65

4.4 Truncated basis illustration: the truncated basis function is shown in red ................................................................. 67

4.5 The original refinement example .................................................... 75

4.6 Comparison between two different coarsening schemes: our scheme is shown on the left and the quasi-hierarchical unrefinement scheme (Grinspun et al., 2002) on the right. ............................................. 75

4.7 Sequence of dynamic refinement and coarsening. ......................... 76

4.8 The $L^2$ projection example: adaptive hierarchical mesh. ............... 82

4.9 $L^2$ norm convergence of cubic hierarchical B-spline basis for the 2D $L^2$ projection example. ................................................................. 83

4.10 The solution field $u$ corresponding to the adaptive hierarchical mesh shown in Figure 4.8(d). ................................................................. 83

4.11 The Poisson equation example: adaptive hierarchical mesh and field. 85

4.12 The advection example: cosine hill and adaptive mesh at $t = 0$. ....... 85
4.13 The advection example: a comparison between uniform mesh, adaptive mesh using global least square fitting and adaptive mesh using local least square fitting at time steps 0, 100, 200, 300, and 400.

4.14 CPU time consumed in solving stage and refinement/coarsening stage for the simulation shown in Figure 4.13.

4.15 Number of degrees of freedom for the adaptive mesh using local least square fitting data transfer scheme.

4.16 The Cahn-Hilliard equation example: evolution of the field and the mesh with the $u(x,t) = 0.0$ iso-contours under coarsening. The left column shows the results obtained with a uniform mesh and the middle and right columns correspond to the adaptive mesh. There are three levels of refinement and the effective fine grid resolution is $160 \times 160$.

4.17 Number of degrees of freedom for the simulation shown in Figure 4.16.

4.18 Total mass for the simulation shown in Figure 4.16 (the maximum error is $0.490\%$).

4.19 Surface of zero contour of (a) Initial shape, (b) equilibrium shape of case $\epsilon_4 = 0.05$, (c) equilibrium shape of case $\epsilon_4 = 0.2$.

4.20 Evolution of the $u = 0.0$ level curves by the strongly anisotropic Cahn-Hilliard equation with higher-order regularization at $t = 0, 0.0078, 0.0694$ and $1.1254$. The simulation was performed with a three-level refinement.

4.21 Adaptive hierarchical mesh at time $t = 0.0694$. The contours $u = 0.0$, $-0.3$ and $0.3$ are shown.

4.22 The anisotropic Cahn-Hilliard equation example for $\epsilon_4 = 0.05$: evolution of the field and the mesh with the $u(x,t) = 0.0$ iso-contours under coarsening. The left column shows the field and the right column corresponds to the adaptive mesh. There are three levels of refinement and the initial grid resolution is $100 \times 100$.

4.23 The anisotropic Cahn-Hilliard equation example for $\epsilon_4 = 0.2$: evolution of the field and the mesh with the $u(x,t) = 0.0$ iso-contours under coarsening. The left column shows the field and the right column corresponds to the adaptive mesh. There are three levels of refinement and the initial grid resolution is $100 \times 100$.

5.1 A smooth surface $S$ is represented as the zero level set of the phase field $\phi$. 

xiv
5.2 (a) Profile of an approximate delta function in the direction of the surface normal $\mathbf{n}$; (b) Approximate support of $\delta_S$ encompassing the surface ..................................................... 102

5.3 (a) 2D adaptively refined mesh. The contour $\phi = 0$ is shown in blue. (b) 3D adaptively refined mesh. Only one octant of the domain is shown and the contour $\phi = 0$ is shown in blue. .............................. 110

5.4 Flory–Huggins type double well potential ............................. 113

5.5 (a) Initial random state with a mean value $\bar{c} = 0.55$. (b) Final equilibrium state of a starfish vesicle. For this example, $\kappa(c_a) = 1.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 10$. (c) Final equilibrium state of a starfish vesicle. For this example, $\kappa(c_a) = 1.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = -10$. ............................... 115

5.6 (a) Final equilibrium state of a starfish vesicle. For this example, $\kappa(c_a) = 4.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 1.0 \times 10^{-16}$. (b) Experimentally observed curvature-dependent domain sorting (Baumgart et al., 2003b). ............................... 115

5.7 (a) Initial random state with a mean value $\bar{c} = 0.55$. For this example, $\kappa(c_a) = 4.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 1.0 \times 10^{-16}$. (b) and (c) Two perpendicular views of the final species distribution. The image opacity is set to be 50%. ............................... 116

5.8 (a) shows the initial phase-separated starfish vesicle. (b)-(f) shows the evolution of shape change. For these results, $\kappa(c_a) = 0.02$, $\kappa(c_b) = 0.02$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 10$. ............................... 116

5.9 Three perpendicular views of shape equilibrium. The image opacity is set to be 50%. ............................... 117

5.10 Evolution of shape changes and species transport (a)-(i). For these results, $\kappa(c_a) = 1.0$, $\kappa(c_b) = 4.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 8$. ............................... 118

5.11 A complete fission process for the example discussed in Section 5.7.3. ............................... 119

5.12 Time evolution of cross-section images for normal coarsening vesicle (a) and trapped coarsening vesicle (b). ............................... 120

5.13 Evolution of the energetic components for the example discussed in Section 5.7.3. The red line shows the trapped coarsening case and the blue line shows the normal coarsening case. The dimensionless time is plotted on a logarithmic scale. (a) Curvature energy. (b) Line energy. (c) Net energy. ............................... 121
List of Abbreviations and Symbols

Symbols

\( \kappa \) flexural rigidity associated with the mean surface curvature
\( \kappa_G \) flexural rigidity associated with the Gaussian curvature
\( H \) mean surface curvature
\( K \) Gaussian surface curvature
\( H_{sp} \) spontaneous curvature
\( \sigma \) line tension
\( f \) double-well potential associated with the chemical energy density
\( g \) double-well potential associated with the energy density of the phase field
\( \phi \) order parameter representing vesicle shape
\( c \) order parameter representing species concentration
\( \lambda \) line tension parameter associated with the gradient in \( c \)
\( \alpha \) line tension parameter associated with the gradient in \( \phi \)
\( \mu \) chemical potential
\( M \) species mobility
\( \delta_s \) approximate delta function
\( \Omega \) computational domain
\( \Omega_{\text{in}} \) internal domain of the vesicle
\( \Omega_{\text{out}} \) external domain of the vesicle
\( \epsilon_{\phi} \) dimensionless interface length associated with \( \phi \)
\( \epsilon_{c} \) dimensionless interface length associated with \( c \)
\( n \) normal pointing out of a surface
\( H_{\phi} \) diffuse-interface representation of the mean curvature
\( E_{c} \) chemical free-energy functional
\( E_{m} \) mechanical free-energy functional
\( E_{gc} \) penalty energy functional associated with geometric constraints
\( E \) total free-energy of the membrane
\( \alpha_{v} \) penalty parameter associated with volume constraint
\( \alpha_{a} \) penalty parameter associated with surface area constraint
\( \alpha_{m} \) penalty parameter associated with zero-flux constraint
\( 1 - n \otimes n \) projection operator
\( t \) time
\( \tau \) pseudo-time
\( \Gamma \) boundary of the domain
\( \Gamma_{d} \) Dirichlet boundary
\( \Gamma_{h} \) Neumann boundary
\( \Gamma_{s} \) Interface
\( \gamma_{1}, \gamma_{2} \) convexity-splitting constants
\( S \) vesicle surface
\( d \) distance function
\( h \) grid-size

Abbreviations

GUVs Giant Unilamellar Vesicles
X-FEM Extended Finite Element Method
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NURBS</td>
<td>Non-uniform Rational B-spline</td>
</tr>
<tr>
<td>IGA</td>
<td>Isogeometrical Analysis</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-aided Design</td>
</tr>
<tr>
<td>FEA</td>
<td>Finite Element Analysis</td>
</tr>
<tr>
<td>PDE</td>
<td>Partial Differential Equation</td>
</tr>
<tr>
<td>GMRES</td>
<td>Generalized Minimal Residual Method</td>
</tr>
</tbody>
</table>
Acknowledgements

Over the years at Duke I have been fortunate to have met so many extraordinary people who have supported my work and continually encouraged me through the writing of this dissertation. My sincere gratitude goes to all of them.

Above all, I would like to thank my advisor Professor John Dolbow for his inspirational advice and constant encouragement over the last several years, without which, this thesis would have been impossible. I have learned so much from his unique perspective on research and his sharp insight on almost all problems. His knowledgeable guidance helps to bring this thesis into being.

I am also very thankful to my excellent committee members. Professor Thomas Witelski’s excellent teaching on applied mathematics inspired me a lot in my subsequent studies. Professor Wilkins Aquino and Guglielmo Scovazzi helped me deepen my knowledge in finite element method through their wonderful courses and insightful discussions. My programming skills have been constantly improving thanks to Professor Laurens Howle’s helpful suggestions. Their generous help deserves my warm thanks.

I am always lucky to have been working with excellent collaborators during my research. I deeply appreciate Professor Isaac Hararis helpful comments on the embedded interface work. I am also very grateful to Professor Eliot Fried for those valuable discussions on the vesicles project.

My grateful thanks also go to many of my great colleagues. It is a great plea-
sure to work with Anand Embar on the vesicle project and his learning skills and problem-solving ability really impressed me. Chandrasekhar Annavarapu has always been ready with brilliant ideas and willing to offer his best suggestions. Temesgen Kindo has great passion for sharing his inspirational perspectives on many areas of computational mechanics. I would also want to express my thanks to Ziyu Zhang, Curtis Lee, Andrew Stershic and Yingjie Liu, for providing me a fun and stimulating environment in which I could learn and grow comfortably. We have shared many memorable times together.

Last but not least, I want to express my deep gratitude to my beloved families, for their understanding and invariable support throughout the duration of my PhD study.
Introductions arise in many areas of biological systems, such as membranes. Biomembranes are the fundamental separation structure for compartmentalizing biological systems. The membranes often exhibit lateral inhomogeneity due to a highly heterogeneous surface distribution of composition. The lateral heterogeneities of biomembranes play important roles in cell biological functions. The numerical modeling of phase separation is expected to improve the understanding of fundamental cell biological processes. The proposed research is aimed in this direction.

The mathematical model can be classified as falling into one of two categories, sharp-interface and diffuse-interface model. With a sharp-interface model, interfaces often represent infinitely thin surfaces of discontinuities. Diffuse-interface models replace the sharp interface with a continuous variation of an order parameter. Additionally, these models often demand higher order spatial derivatives and smoothness in the numerical approximation to solutions. In this regard, we develop spline-based finite elements that allow for both sharp- and diffuse-interface representation in an efficient and accurate manner.
1.1 Cell membranes

Biomembranes form the basic structural units of cellular structures. They are generally composed of mobile lipids, proteins and cholesterol. Several kinds of lipids are self-assembled in a lipid bilayer, within which transmembrane proteins are free to migrate and present a fluid-in-plane behavior (Seifert, 1997a). The proteins play a principle role in the morphology of a lipid bilayer by effectively breaking the bilayer symmetry (Kim et al., 1998). Vesicles are simple forms of closed biomembranes and serve as important but simplified models of more complex cell membranes. Bilayers that are a few nanometers thick form vesicles that range between 50nm to tens of micrometers (Deseri et al., 2008). Larger vesicles are commonly referred to as Giant Unilamellar Vesicles (GUVs), which provide a perfect tool for studying lipid phase separation and the appearance and behavior of lipid domains in membranes.

The surface distribution of lipids, proteins and cholesterol in cell membranes is known to be highly heterogeneous. Depending on the ambient temperature and surrounding conditions, each phospholipid molecule can admit a liquid ordered state ($L_0$) and liquid disordered state ($L_d$) (Baumgart et al., 2003a). Recent experiments on GUVs containing ternary mixtures of lipid components and cholesterol have shown the coexistence of liquid-ordered and liquid-disordered phases in the membrane. Specifically, membrane lipid phase coexistence is expected to modulate the shape transition process through the differing mechanical properties of the bulk domains and line tension at phase boundaries (Baumgart et al., 2005a). The long-range domain ordering in the form of locally parallel stripes and hexagonal arrays of circular domains is shown in Figure 1.1.

In the so-called raft model, the lateral heterogeneities of biomembranes arising from separation of lipids into two immiscible liquid phases, coupled with the protein distribution (Hess et al., 2005) play important roles in fundamental cell biological
processes, such as membrane signaling, trafficking (Pike, 2006; Simons and Ikonen, 1997) and are known to be central to the replication of viruses (Ono and Freed, 2001). The rafts may also be responsible for morphological changes such as budding and fission during endocytosis and exocytosis (Rauch and Farge, 2000), and membrane adhesion and fusion (Takeda et al., 2003). A comprehensive survey of theoretical models describing the physical principles governing raft formation and associated experimental approaches is given in Elson et al. (2010). The study of domain formation in GUVs containing ternary lipid-cholesterol mixtures is expected to improve the understanding of raft formation and raft behavior in cell membranes. The proposed research follows previous work in this direction (Embar et al., 2012).

1.2 Continuum approach of vesicle modeling

The continuum methods can be employed to describe the coupled process of shape change and species transport. The continuum approach is based on the curvature energy described in the generalized Helfrich model (Helfrich, 1973) supplemented by a line tension energy (Jülicher and Lipowsky, 1993, 1996).

1.2.1 Canham-Helfrich-Evans model
Canham (1970), Helfrich (1973) and Evans (1974) first proposed a purely mechanical model for studying the equilibrium shapes of single component vesicles such as red blood cells. For multi-component vesicles, line tension energy is introduced along the interface of the two phases. In its most general form, the energy functional reads,

$$E = \int_{\mathcal{S}} \frac{1}{2} \kappa(c) (H - H_{sp}(c))^2 \, da + \int_{\mathcal{S}} \kappa_G(c) K \, da + \int_{\mathcal{S}} \sigma \, da$$

(1.1)

where $c$ is the concentration corresponding to liquid-order and liquid-disordered phases, $S$ represents the membrane surface, $\kappa(c)$ and $\kappa_G(c)$ are the flexural rigidities corresponding to the mean curvature $H$ and Gaussian curvature $K$ of the membrane. $H_{sp}(c)$ is a phenomenological term referred to as the spontaneous curvature. The spontaneous curvature of a bilayer membrane is taken to represent the curvature of a bilayer in its natural state. $\sigma$ is the line tension, which is caused by the mismatch of different thicknesses of the lipid phases.

1.2.2 Phase separation in GUVs

Thermally and chemically induced phase separation have been observed to lead to domain formation on vesicles (Sackmann, 1995). GUVs of specified composition can separate into phases that resemble rafts (Figure 1.1). Such phase behavior depends on the composition of the lipid bilayer membrane as well as temperature and pressure. In the present work, we employ a Cahn-Hilliard model to study phase separation and domain formation on vesicles.

In the Cahn-Hilliard model, the total free energy of a mixture can be obtained by introducing an energy penalty for the interfaces and is given by

$$E_c = \int_{\Omega} f(c) + \frac{\lambda}{2} |\nabla c|^2 \, dv$$

(1.2)

where $f(c)$ is the free energy density of a homogeneous mixture and $\lambda$ is a parameter.
penalizing gradients in the volume fraction $c$. The continuity condition derived from (1.2) gives the Cahn-Hilliard equation for phase separation:

$$\frac{\partial c}{\partial t} + \nabla \cdot j = 0,$$

$$j = -M \nabla \mu, \quad (1.3)$$

$$\mu = \frac{\delta E_c}{\delta c}.$$  

where, $j$ is the flux, $\mu$ is the chemical potential and $M$ represents species mobility.

1.3 Computational challenges

In modeling vesicles (Biben et al., 2005; Campelo and Hernández-Machado, 2006; Lowengrub et al., 2009; Q and J, 2007; Embar et al., 2012), the phase-field method provides an implicit representation of the vesicle surface. Employing the phase-field approach, the governing equations are formulated in diffuse-interface form which leads to nonlinear fourth-order coupled PDEs. Solution to these higher order equations can be approximated by spline-based finite elements (Höllig, 2003; Embar et al., 2010a) which offer smooth B-spline basis functions to satisfy the requirement for $H^2$ regularity.

Since a diffuse interface model for biomembranes eliminates the need to explicitly track the membrane surface, the initial code development is relatively straightforward, in comparison to using a finite element discretization of the equations on a triangulated evolving surface. However, the computational effort of this model is always very large mainly due to the following two reasons. First, the diffuse interface model increases the dimension of the physical problem. The two-dimensional vesicle surface is embedded in a three-dimensional computational grid. Second, small grid spacing is needed to resolve the narrow interfacial regions. Therefore, a challenging issue is the development of efficient, adaptive, three-dimensional spline-based finite
elements for phase field simulations.

1.3.1 Local refinement in spline-based finite elements

B-splines and non-uniform rational B-splines (NURBS) were originally used for the exact representation of geometry in computer graphics, and were later employed as basis functions in finite element approximations. Combining these two techniques leads to the concept of isogeometric analysis (IGA) as proposed by Hughes et al. (2005a) to bridge the gap between computer-aided design (CAD) and finite element analysis (FEA). The splined-based finite element method can be considered a special case of IGA, in which the geometry does not necessarily adopt the spline bases. Thus the local refinement approach in IGA can be applied to spline-based finite elements without any modification. A comprehensive comparison of the existing refinement techniques is listed in Table 1.1. The hierarchical refinement approach is adopted in our research to model vesicle membranes adaptively due to two considerations. First, it should work in any dimension, especially three dimensions because the vesicle is implicitly represented in a three-dimensional space. Second, it should not change the regularity of the basis functions since approximating the solution to fourth-order partial differential equations requires $H^2$ regularity.

Recent studies (Vuong et al., 2011; Schillinger et al., 2012; Bornemann and Cirak, 2013) have focused on using hierarchical B-splines in IGA to achieve adaptivity. However, there are some issues which have not been addressed, such as the coarsening algorithm and the data transfer techniques. These issues are very important for dynamic refinement in which the refinement region will change during the computation. These issues are addressed in this work.
Table 1.1: Refinement techniques and their properties. source: Vuong (2012)

<table>
<thead>
<tr>
<th>Refinement Technique</th>
<th>Local</th>
<th>Linear Independence</th>
<th>Nested</th>
<th>Partition of Unity</th>
<th>Special Case Uniform Geometry</th>
<th>Representability</th>
<th>Arbitrary Degree/Continuity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform B-spline Refinement</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Multipatch</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>T-splines</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>HTP-splines</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Hierarchical B-splines</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

1.3.2 Extended spline-based finite elements

While the spline-based finite elements perform efficiently diffuse-interface models, its capability for sharp-interface problems needs to be further explored. For many sharp-interface problems, the extended finite element method (X-FEM) facilitates the treatment of both stationary and evolving interfaces with complex geometry. This work investigates the use of B-spline basis functions in conjunction with the X-FEM.

One of our aims is to develop a finite element method that alleviates meshing constraints and allows the investigation of interfacial effects by embedding interfaces within a background mesh. We facilitate this embedding by enriching the standard finite element basis functions in the vicinity of the interface with Heaviside functions. In other words, the interface is represented independently of the finite element mesh. We then develop stabilized variational methods based on Nitsche’s technique that efficiently enforce these constraints while preserving the accuracy of interfacial fluxes.
1.4 Overview of the thesis

The broader goal of this work is to develop a computationally efficient spline-based finite element method for both sharp- and diffuse-interface problems. The proposed methods are then applied to a phase field model of multicomponent vesicles.

B-splines are most easily constructed over rectilinear domains and mapped into arbitrarily shaped domains. The standard IGA further requires a boundary and interface fitted discretization. This usually entails a tremendous meshing effort for complex geometry. To overcome these limitations, we develop a spline-based finite element method that allows the boundary and interface to be embedded within the background mesh. The boundary and interfacial constraints are then weakly enforced by Nitsche’s method.

The phase-field method provides an implicit representation of the vesicle surface and naturally allows for topological changes. However the computational effort of this model is always very intensive on a uniform mesh. To make three dimensional simulations proceed efficiently, we further develop an adaptive spline-based finite element method based on hierarchical refinement.

The thesis is organized as follows. Chapter 2 presents the details of building a finite element basis with uniform splines. The performance of the B-spline based finite element method is presented with several benchmark numerical examples of second and fourth-order boundary value problems. In Chapter 3, the use of spline-based finite elements is examined along with a Nitsche technique for enforcing constraints on an embedded interface. Chapter 4 presents adaptive spline-based finite elements that employ hierarchical refinement and coarsening techniques with an efficient data transfer algorithm. Chapter 5 describes the numerical modeling of vesicles. Numerical examples of domain evolution coupled to vesicle shape deformation are presented.
sion are also considered. Chapter 6 provides concluding remarks and comments on some future directions for this work.
2

Spline-based Finite Elements

2.1 Overview

Smooth functions such as B-splines are becoming increasingly popular for use in finite element methods. The construction of conforming finite elements for fourth-order problems requires solution and weighting spaces to be in $H^2$. This corresponds to the use of shape functions with $C^1$ continuity. Classical finite elements employ Hermite basis functions that possess global $C^1$ continuity. However, the construction of Hermite polynomials on unstructured meshes with complex geometry is difficult. Another approach is to rewrite the governing equations as two coupled, second-order systems. This mixed formulation not only increases the degrees of freedom for the problem, but also requires the discrete space to satisfy the $inf-sup$ condition. Alternatively, B-spline basis functions provide higher-order continuity and do not introduce additional degrees of freedom to the problem. In this Chapter, a conforming finite element basis for second- and fourth-order systems is constructed with B-splines.
2.2 B-spline basis functions

A brief account of the construction of B-Spline basis functions in one dimension is given below. The extension to multiple dimensions is straightforward with the use of tensor product splines. A more detailed treatment of the subject can be found in Piegl and Tiller (1997).

Consider a knot vector \( \Xi \), where,
\[
\Xi = \{ \xi_1, \xi_2, \ldots, \xi_{n+p+1} \}
\] (2.1)
with \( n \) denoting the number of control points and \( p \), the degree of the spline. B-spline basis functions are defined recursively starting with piecewise constants (degree \( p=0 \)):
\[
N_{I,0} (\xi) = \begin{cases} 
1 & \text{if } \xi_I \leq \xi < \xi_{I+1} \\
0 & \text{otherwise}
\end{cases}
\] (2.2)

For \( p \geq 1 \):
\[
N_{I,p} (\xi) = \frac{\xi - \xi_I}{\xi_{I+p} - \xi_{I}} N_{I,p-1} (\xi) + \frac{\xi_{I+p+1} - \xi_I}{\xi_{I+p+1} - \xi_{I+1}} N_{I+1,p-1} (\xi)
\] (2.3)

Multivariate B-splines are constructed using a tensor product generalization of univariate B-splines. Generated from univariate knot vectors: \( \Xi^l = \{ \xi_{l1}, \xi_{l2}, \ldots, \xi_{ln_l+p_l+1} \} \), where \( l = 1, \ldots, \text{dim} \), the multivariate basis functions \( B_{i,p} \) can be constructed as:
\[
B_{i,p} (\xi) = \prod_{l=1}^{\text{dim}} N_{i_l,p_l} (\xi_l)
\] (2.4)

Figure 2.1(a) illustrates five cubic B-spline basis functions over a uniform knot vector in one dimension. An example of a bicubic spline is shown in Figure 2.1(b). It is generated using a tensor product of the 1-D spline highlighted in Figure 2.1(a).

The following well-known properties of uniform B-spline basis functions are relevant to B-spline finite element methods:
Figure 2.1: \( C^2 \) continuous cubic spline basis functions resulting from a uniform knot vector.

- **Local support**:
  \[
  \{ (\xi) : N_{i,p}(\xi) \neq 0 \} \subset [\xi_i, \xi_{i+p+1}] \tag{2.5}
  \]

- **Local linear independence**:
  \[
  \sum_i c_i N_{i,p}(\xi) = 0 \iff c_i = 0 \tag{2.6}
  \]

- **Nonnegativity**:
  \[
  N_{i,p}(\xi) \geq 0 \tag{2.7}
  \]

- **Partition of unity (Marsden’s Identity (de Boor, 2001))**:
  \[
  \sum_i N_{i,p}(\xi) = 1, \xi \in [\xi_1, \xi_{n+p+1}] \tag{2.8}
  \]

2.2.1 Finite element basis with B-splines

Cubic B-spline basis functions constructed on a uniform grid are used as a finite element basis in this study. With a Galerkin approximation, spatial discretization of
the weak form is carried out such that $u^h \in \mathcal{U}^h \subset \mathcal{U}$ and $w^h \in \mathcal{V}^h \subset \mathcal{V}$. We write

\[ w^h(x) = \sum_{I} N_I (x(\xi)) \ c_I \tag{2.9a} \]

\[ u^h(x) = \sum_{I} N_I (x(\xi)) \ d_I \tag{2.9b} \]

for the approximations to the test and trial functions, respectively. Here, $\xi$ denotes the parametric coordinates used to construct the B-spline basis. Bicubic B-splines are globally $C^2$ continuous and they satisfy the requirement of the $H^2$ conforming discrete space for the 4th order problem. Each B-spline function is identified by the point having the lowest knot coordinate in its support. In two dimensions, this translates to being the lower left corner of the spline support, as shown in Figure 2.2(a). Knot spans subdivide the computational domain into ‘elements’ with characteristic side-lengths $h$. With this construction, relevant bicubic B-splines for a general physical domain are shown in Figure 2.2(b). The circular markers represent the lower left corner of each bicubic spline whose support extends into the physical domain.

**Figure 2.2:** (a) Identification node for a bi-cubic spline basis function and (b) Computational domain (in grey) for the given geometry.
2.2.2 Dirichlet boundary conditions

The B-spline basis is non-interpolatory, and this poses a challenge for the imposition of Dirichlet-type boundary conditions with the finite element method. One straightforward approach is to reduce the continuity of the basis functions by knot-repetition. The basis can be rendered interpolatory at any knot when the knot multiplicity $k = p$ (Hughes et al., 2005b). However, this technique is limited to the case that the discretization grid conforms to the boundary.

In the present study, we use Nitsche’s method to weakly enforce Dirichlet boundary conditions. With a sufficiently large stabilization parameter, Nitsche’s formulation leads to a variationally consistent, symmetric and positive-definite system. The minimum value of the stabilization parameter that guarantees coercivity of the bilinear form can be determined by the solution of an eigenvalue problem. This basic idea is then applied to fourth-order problems in which the Dirichlet boundary conditions involve both the primary function and its derivative.

It bears emphasis that this approach is also applicable to the case that the grid does not conform to the boundary. However, for curved boundaries the level of accuracy in the geometric representation of the boundary becomes important when higher-order rates of convergence are sought with these methods. The approach to improve the geometrical representation of the curved embedded boundary is proposed in Chapter 3.

2.3 Modeling second- and fourth-order problems with spline-based finite elements

In this Section, second- and fourth-order boundary value problems and their Nitsche’s variational formulations are described. Numerical examples are provided to demonstrate the performance of splines as finite element basis functions and the efficacy of
Nitsche’s method in weakly imposing Dirichlet boundary conditions. Additionally, we present the numerical simulation of the Cahn-Hilliard equation in three dimensions.

Figure 2.3 illustrates the setting for a standard boundary value problem. The domain \( \Omega \) is bounded by the surface \( \Gamma \) such that, \( \Gamma = \Gamma_d \cup \Gamma_h \), where \( \Gamma_d \) and \( \Gamma_h \) are the Dirichlet and Neumann boundaries, respectively. The vector \( \mathbf{n} \) represents the outward normal to the boundary at any point along \( \Gamma \).

![Figure 2.3: Domain \( \Omega \) bounded by Dirichlet (\( \Gamma_d \)) and Neumann (\( \Gamma_h \)) boundaries, with unit outward normal \( \mathbf{n} \).](image)

### 2.3.1 Second-order boundary value problem

A Poisson equation is considered as a representative second-order problem. The Dirichlet or Neumann conditions at any point along the boundary are given by the value of the function or the normal derivative at that point.
The boundary value problem is defined by

\[-\Delta u = f \quad \text{in } \Omega, \quad (2.10a)\]
\[u = g \quad \text{on } \Gamma_d, \quad (2.10b)\]
\[\nabla u \cdot \mathbf{n} = h \quad \text{on } \Gamma_h, \quad (2.10c)\]

where \(\Delta\) denotes the Laplacian, and \(\nabla\) the gradient operator.

Using Nitsche’s method, the variational form of the Poisson equation can be stated as:

Find \(u \in \mathcal{U}\) such that

\[a(w, u) = l(w) \quad (2.11)\]

where

\[a(w, u) = \int_{\Omega} \nabla w \cdot \nabla u \, dv - \int_{\Gamma_d} w (\nabla u \cdot \mathbf{n}) \, da - \int_{\Gamma_d} u (\nabla w \cdot \mathbf{n}) \, da + \alpha \int_{\Gamma_d} w u \, da, \quad (2.12a)\]

\[l(w) = \int_{\Omega} w f \, dv + \int_{\Gamma_h} w h \, da - \int_{\Gamma_d} g (\nabla w \cdot \mathbf{n}) \, da + \alpha \int_{\Gamma_d} w g \, da, \quad (2.12b)\]

for all \(w \in \mathcal{V}\). Here, the solution and weighting spaces are such that both \(\mathcal{U} = H^1\) and \(\mathcal{V} = H^1\), where \(H^1\) is the Sobolev space of square integrable functions with square integrable first derivatives.

Nitsche’s terms along the Dirichlet boundary give the method variational consistency. The stabilization term is added to ensure the coercivity of the bilinear form and the stabilization parameter \(\alpha\) can be determined by an eigenvalue problem at the element level.
Solution of Poisson equation in a unit cube

We consider a 3D Poisson equation in a unit cube with homogeneous boundary conditions. The governing equation and boundary conditions are

\[ \Delta u = \sin(\pi x) \sin(\pi y) \sin(\pi z), \quad (2.13a) \]
\[ u(x, y, z) = 0 \text{ along the boundaries}, \quad (2.13b) \]

for which the analytical solution is

\[ u(x, y, z) = \frac{1}{3\pi^2} \sin(\pi x) \sin(\pi y) \sin(\pi z). \quad (2.14) \]

The Nitsche’s variational form with \( f = -\sin(\pi x) \sin(\pi y) \sin(\pi z) \) is discretized with uniform cubic B-splines. The stability parameters are evaluated for each element sharing the Dirichlet boundary, by solving local eigenvalue problems as explained in Embar et al. (2010a). Figure 2.4 shows the approximation to the solution for

\[
\begin{array}{c}
\text{Figure 2.4: Solution to the 3D Poisson problem}
\end{array}
\]

a uniform grid with \( h = 1/30 \). A quartic convergence rate in the \( L_2 \) norm of the error and a cubic rate of convergence in the energy norm is obtained as shown in Figure 2.5.
2.3.2 Fourth-order boundary value problem

A non-homogeneous biharmonic equation is considered for a representative fourth order problem. Here, the Dirichlet boundary conditions are specified by values of the function and its normal derivative along the boundary. Neumann boundary conditions are specified in terms of second and third-order derivatives.

The fourth-order boundary value problem can be stated as,

\[ \Delta^2 u = f \quad \text{in } \Omega, \]  
\[ u = g \ ; \ \nabla u \cdot \mathbf{n} = h \quad \text{on } \Gamma_d, \]  
\[ \Delta u = p \ ; \ \nabla(\Delta u) \cdot \mathbf{n} = q \quad \text{on } \Gamma_h \]

where \( \Delta^2 \) denotes the biharmonic operator.

It is worth noting here that other combinations of boundary conditions can also exist. Numerical examples having different combination of boundary conditions are presented.

Nitsche’s method leads to the following variational formulation that can be stated as:
Find $u \in \mathcal{U}$ such that

$$a(w, u) = l(w)$$

for all $w \in \mathcal{V}$, where

$$a(w, u) = \int_{\Omega} \Delta w \Delta u \, dv + \int_{\Gamma_d} w (\nabla (\Delta u) \cdot \mathbf{n}) \, da + \int_{\Gamma_d} u (\nabla (\Delta w) \cdot \mathbf{n}) \, da + \alpha_1 \int_{\Gamma_d} w u \, da$$

$$- \int_{\Gamma_d} (\nabla w \cdot \mathbf{n}) \Delta u \, da - \int_{\Gamma_d} (\nabla u \cdot \mathbf{n}) \Delta w \, da + \alpha_2 \int_{\Gamma_d} (\nabla w \cdot \mathbf{n}) (\nabla u \cdot \mathbf{n}) \, da$$

(2.17a)

$$l(w) = \int_{\Omega} w f \, dv - \int_{\Gamma_h} w q \, da + \int_{\Gamma_d} g (\nabla (\Delta w) \cdot \mathbf{n}) \, da + \alpha_1 \int_{\Gamma_d} w g \, da$$

$$+ \int_{\Gamma_h} (\nabla w \cdot \mathbf{n}) p \, da - \int_{\Gamma_d} h \Delta w \, da + \alpha_2 \int_{\Gamma_d} (\nabla w \cdot \mathbf{n}) h \, da$$

(2.17b)

Here, the solution and weighting spaces are such that both $\mathcal{U} = H^2$ and $\mathcal{V} = H^2$; where $H^2$ is the Sobolev space of square integrable functions with square integrable first and second derivatives. The stabilization parameters $\alpha_1$ and $\alpha_2$ are chosen to guarantee coercivity of the Nitsche form (2.17a), as described in Embar et al. (2010a).

3D biharmonic equation

We now examine a 3D biharmonic problem in a unit cube. In the first case, Dirichlet boundary conditions are specified only on the primary function along the boundaries, thus only the stability parameter $\alpha_1$ is active. The governing equation and boundary conditions are given as follows:

$$\Delta^2 u = \sin(\pi x) \sin(\pi y) \sin(\pi z)$$

(2.18a)

$$u = 0 \text{ and } \Delta u = 0 \text{ along the boundaries,}$$

(2.18b)
The analytical solution for this problem is given by

\[ u(x, y, z) = \frac{1}{9\pi^4} \sin(\pi x) \sin(\pi y) \sin(\pi z). \]  

(2.19)

\textbf{Figure 2.6:} Solution to the 3D biharmonic problem (first case).

Plots of the solution for a uniform grid size \( h = 1/30 \) are given in Figure 2.6. Figure 2.7 shows optimal convergence rates in both the \( L_2 \) and energy norms. While a quartic rate is obtained for the \( L_2 \) norm of the error, the energy norm has a quadratic rate of convergence owing to the occurrence of higher order derivatives in the bilinear form.

In the second case, we consider a 3D biharmonic equation with the exact solution

\[ u(x, y, z) = (1 - \cos(2\pi x))(1 - \cos(2\pi y))(1 - \cos(2\pi z)). \]  

(2.20)

The forcing term \( f \) and Dirichlet condition \( u \) and \( \nabla u \cdot n \) are obtained from \( u \). For this numerical example, the stability parameters are associated with both the function and its derivative. Separate eigenvalue problems are set up and solved for each parameter in every boundary element.

Plots of the solution for a uniform grid size \( h = 1/30 \) are given in Figure 2.8. Figure 2.9 shows optimal convergence rates in both the \( L_2 \) and energy norms. While
Figure 2.7: Convergence rates for the 3D biharmonic (first case) in (a) $L_2$ and (b) energy norms.

Figure 2.8: Solution to the 3D biharmonic problem (second case).

A quartic rate is obtained for the $L_2$ norm of the error, the energy norm has a quadratic rate of convergence.
2.3.3 Cahn–Hilliard equation

The governing equation is given as

\[
\frac{\partial u}{\partial t} = \nabla \cdot \left( M(u) \nabla \left( \frac{\partial f}{\partial u} - 2\gamma \Delta u \right) \right) \tag{2.21}
\]

subject to zero mass flux boundary conditions,

\[
\nabla u \cdot n = 0 \quad \text{and} \quad \nabla \left( \frac{\partial f}{\partial u} - 2\gamma \Delta u \right) \cdot n = 0 \quad \text{on } \Gamma \times (0, T) \tag{2.22}
\]

and initial conditions,

\[
u(x, 0) = u_0(x) \quad \text{in } \Omega \tag{2.23}
\]

For this example, we consider a free energy function of the form

\[
f(u) = \frac{1}{12} u^4 - \frac{1}{2} u^2 \tag{2.24}
\]

The chemical potential is thus given by

\[
\mu(u) = f'(u) = \frac{1}{3} u^3 - u \tag{2.25}
\]
The binodal points are evaluated as the two global minima of the chemical free energy function and are given by, $u_a = -\sqrt{3}$ and $u_b = \sqrt{3}$.

Accounting for the zero mass flux boundary conditions, the Nitsche weak form can be derived from the Cahn–Hilliard equation (2.21). Assuming a constant mobility $M$, the weak form can be stated as,

$$\int_{\Omega} w \frac{\partial u}{\partial t} \, dv + M \int_{\Omega} f''(u)(\nabla w \cdot \nabla u) \, dv + \gamma M \int_{\Omega} \Delta w \cdot \Delta u \, dv - \gamma M \int_{\Gamma} \Delta u (\nabla w \cdot n) \, da$$

$$- \gamma M \int_{\Gamma} \Delta w (\nabla c \cdot n) \, da + \alpha \int_{\Gamma} (\nabla w \cdot n)(\nabla c \cdot n) \, da = 0 \quad (2.26)$$

for all $w \in \mathcal{V} \subset H^2(\Omega)$. Here, $H^2(\Omega)$ is the Sobolev space of order 2. $\mathcal{U}$ is the admissible space of concentration fields such that $\mathcal{U} \subset H^2$.

A cubic B-spline basis is employed to obtain a conforming discretization of (2.26). Backward-Euler method is used for time integration. The discretized system of nonlinear equations is then solved using a full Newton-Raphson scheme.

Three-dimensional numerical examples are considered here. The constant $\gamma$ is taken to be $0.025^2$ and the problem is solved on a $60^3$ mesh. The mean value of the initial spatial distribution of concentration $\bar{u}$ is set to be in the spinodal region. Being inherently unstable, the mixture is expected to evolve into two distinct phases, each phase being represented by a global minimum of the free energy function $f$.

(a) $\bar{u}=0.0$

We present snapshots of isosurfaces of the concentration in Figure 2.10. We observe that this randomly perturbed constant concentration evolves to a complex interconnected pattern.

(b) $\bar{u}=0.85$

We present snapshots of isosurfaces of the concentration in Figure 2.11. In this
case the masses of the two phases were significantly different, which leads to the nucleation.

2.4 Summary

The spline-based finite element method is an ideal technology for the study of equations involving higher-order differential operators. This Chapter presents a robust Nitsche's formulation to weakly impose Dirichlet boundary conditions. The accuracy of the approach is verified for three-dimensional second- and fourth-order linear problems and optimal rates of convergence are obtained. The performance of the approach for a fourth-order nonlinear evolution equation is further examined through a three-dimensional Cahn-Hilliard phase-field model.
Figure 2.10: The Cahn-Hilliard equation example for $\bar{u}=0.0$: evolution of the field with the $u(x, t) = 0.0$ iso-contours under coarsening.
Figure 2.11: The Cahn-Hilliard equation example for \( \bar{u} = 0.85 \): evolution of the field with the \( u(x, t) = 0.0 \) iso-contours under coarsening.
A Robust Nitsche’s Formulation for Interface Problems with Spline-based Finite Elements

3.1 Introduction

The extended finite element method (X-FEM) facilitates the treatment of both stationary and evolving interfaces with complex geometry. Pertinent applications range from the modeling of dynamic crack propagation (Moes et al., 1999) to investigating phase transformation/solidification (Ji et al., 2002). Optimal convergence rates have routinely been demonstrated for elements using linear shape functions. In this Chapter, our purpose is to investigate the use of B-spline basis functions in conjunction with the X-FEM.

In the X-FEM, it is not required that the finite element mesh aligns with the interface geometry, circumventing the need to generate conforming meshes as the interface evolves. Frequently, with such embedded approaches, the interface is implicitly represented by a level set function. The level of accuracy in the geometric representation of the interface becomes important when higher-order rates of convergence are sought with these methods. To improve the geometrical representation of
the curved interface, the approximation can be classified as falling into one of two categories, p-refinement and h-refinement. The p-refinement approach is to use higher order interpolation for the interface. Cheng and Fries (2010) used a higher order representation of the level set to describe the interface with higher-order accuracy. Kästner et al. (2013) employed a similar idea and further improved the compatibility between the level set representation and the integration subdomains by computing the level set values at local elements. Higher-order splines were used by Benowitz and Waisman (2013) to interpolate the arbitrary shape of inclusions. The development of NURBS-based subelements for the representation of curved material interfaces can be found in Haasemann et al. (2011). These issues have also been examined recently in the context of discontinuous Galerkin method for interface problems. In particular, Huynh et al. (2013) proposed the use of superparametric elements at the interface to recover optimal rates of convergence.

The h-refinement approach is to locally refine the mesh in the vicinity of the interface. The level set function is interpolated by means of finite element shape functions, which need not necessarily be the same ones used for the finite element approximation. This allows one to introduce an adapted mesh to describe the geometry while keeping a higher order approximation of field quantities on a uniform coarse mesh. Thanks to this approach, Dréeau et al. (2010) and Legrain et al. (2012) obtained optimal rates of convergence with higher order X-FEM for the case of free surfaces and material interfaces with complex geometry. Moumnassi et al. (2011) also proposed a similar method to represent arbitrary parametric surfaces implicitly with level sets. In this study, we employ similar techniques to represent the level set on a refined sub-mesh. Our contention is that, in contrast to p-refinement, such an approach is straightforward to implement and readily extends to multiple dimensions.

Nitsche’s method has been used extensively in conjunction with the X-FEM for
weakly enforcing constraints on embedded interfaces (Sanders et al., 2009, 2012; Hautefeuille et al., 2012). The method employs a stabilization parameter that should be chosen to be sufficiently large as to maintain coercivity. Both global (Griebel and Schweitzer, 2003) and element level (Embar et al., 2010b; Dolbow and Harari, 2009) calculations have been employed to obtain suitable parameters. Recently, a novel weighting for the interfacial consistency terms arising in a Nitsche variational form was proposed by Annavarapu et al. (2012a,b, 2013, 2014) to address the lack of robustness exhibited by the classical Nitsche formulation for embedded interface problems. They derived closed-form analytical expressions for the interfacial weights and stabilization parameters for the case of constant strain triangular and tetrahedral elements.

With higher-order elements or basis functions, closed-form expressions for these terms have yet to be derived. In this work, we propose a new formulation to find the weights for higher order elements. By using the generalized inverse estimate given in Barbosa and Hughes (1991), we establish a clear relationship between the weights and stabilization parameters. This allows us to make a choice for the weights which least affects the stability of the method in the presence of both small cut elements and large heterogeneities in the data. The proposed approach is compared with the one proposed in Annavarapu et al. (2012a) for higher order elements and demonstrates better robustness. We also describe how the singularity of the matrices in the generalized eigenvalue problem can result in unstable numerical solutions. To overcome this difficulty, we deflate the matrices and remove the singularities, resulting in a robust eigenvalue evaluation.

This Chapter is organized as follows. In Section 3.2, we define the model problems and the associated variational formulations with emphasis on the terms stemming from Nitsche’s method. The discretizations with B-Splines and extended B-Splines enriched with Heaviside functions are then described in Section 3.3. The geomet-
rical error and hierarchical local refinement approach is discussed in Section 3.4. Section 3.5 then outlines a method to identify suitable weighting and stabilization terms. Numerical examples that demonstrate the accuracy of the proposed method against the classical Nitsche approach are presented in Section 3.6. Finally, in Section 3.7, we provide a summary and concluding remarks.

3.2 Model problem and variational formulation

Figure 3.1 illustrates a domain $\Omega$, partitioned into domains $\Omega^1$ and $\Omega^2$ by an interface $\Gamma_*$. The normals $n^m$ are considered as outward pointing from their corresponding domains and the unit normal $n$ to $\Gamma_*$ is chosen to point outward from $\Omega^1$ as shown.

![Figure 3.1: A domain $\Omega$ partitioned into regions $\Omega^1$ and $\Omega^2$ by the interface $\Gamma_*$.](image)

We begin by considering a relatively simple Poisson problem on the domains $\Omega^1$ and $\Omega^2$: Find $u$ such that

\begin{align}
-\nabla \cdot \kappa^m \nabla u^m &= f^m \quad \text{in } \Omega^m, \\
u^m &= g^m \quad \text{on } \Gamma^m_d, \\
\kappa^m \nabla u^m \cdot n^m &= h^m \quad \text{on } \Gamma^m_h,
\end{align}

(3.1a) (3.1b) (3.1c)
coupled together with the interface conditions:

\[
\begin{align*}
\llbracket \kappa \nabla u \rrbracket \cdot \mathbf{n} &= \tilde{j} & \text{on } \Gamma_s, \\
\llbracket u \rrbracket &= \tilde{i} & \text{on } \Gamma_s,
\end{align*}
\]

(3.2a) \hspace{1cm} (3.2b)

where, \( f^m, g^m \) and \( h^m \) are known and \( \tilde{i} \) and \( \tilde{j} \) are sufficiently smooth known functions on the interface. The shorthand notation \( \llbracket u \rrbracket \) corresponds to the jump, \( u^2 - u^1 \).

### 3.2.1 Weak Form

In our method, both the boundary and interfacial conditions are enforced weakly with Nitsche’s method. Nitsche’s method involves additional integrals along the Dirichlet boundary and interface which give the method variational consistency. The variational form consists of parts associated with bulk and interfacial terms. We write this in a general form as: Find \( u \in \mathcal{U} \) such that

\[
a_b (w, u) + a_i (w, u) = l_b (w) + l_i (w)
\]

(3.3)

for all \( w \in \mathcal{V} \). The solution and weighting spaces are such that both \( \mathcal{U} = H^1(\Omega^1 \cup \Omega^2) \) and \( \mathcal{V} = H^1(\Omega^1 \cup \Omega^2) \) where \( H^1 \) is the Sobolev space of square integrable functions with square integrable first derivatives. Here, the bulk contributions are

\[
a_b (w, u) = \sum_m \int_{\Omega^m} \nabla w^m \cdot \kappa^m \nabla u^m \, d\Omega - \sum_m \int_{\Gamma_d^m} w^m \kappa^m (\nabla u^m \cdot \mathbf{n}^m) \, d\Gamma \\
- \sum_m \int_{\Gamma_d^m} \kappa^m (\nabla w^m \cdot \mathbf{n}^m) u^m \, d\Gamma + \alpha_b \sum_m \int_{\Gamma_d^m} w^m u^m \, d\Gamma
\]

(3.4a)

\[
l_b (w) = \sum_m \int_{\Omega^m} w^m f^m \, d\Omega + \sum_m \int_{\Gamma_h^m} w^m h^m \, d\Gamma \\
- \sum_m \int_{\Gamma_d^m} \kappa^m (\nabla w^m \cdot \mathbf{n}^m) g^m \, d\Gamma + \alpha_b \sum_m \int_{\Gamma_d^m} w^m g^m \, d\Gamma
\]

(3.4b)
On the interface, we have

$$a_i(w, u) = -\int_{\Gamma_*} [w] \langle \kappa \nabla w \rangle_\gamma \cdot n \, d\Gamma - \int_{\Gamma_*} [u] \langle \kappa \nabla w \rangle_\gamma \cdot n \, d\Gamma + \alpha_i \int_{\Gamma_*} [w] [u] \, d\Gamma$$

(3.5a)

$$l_i(w) = -\int_{\Gamma_*} \tilde{i} \langle \kappa \nabla w \rangle_\gamma \cdot n \, d\Gamma + \alpha_i \int_{\Gamma_*} [w] \tilde{i} \, d\Gamma + \int_{\Gamma_*} \tilde{j} \langle w \rangle_{1-\gamma} \, d\Gamma$$

(3.5b)

where the shorthand notation $\langle \cdot \rangle_\gamma = (\gamma)(\cdot)^1 + (1 - \gamma)(\cdot)^2$ refers to the weighted average of a quantity $(\cdot)$ across the interface. Here $\gamma$ and stability parameters $\alpha_i$ and $\alpha_b$ are real numbers. A possible strategy for identifying these parameters will be discussed in 3.5.

### 3.3 Discretization with B-splines

We cover the domain $\Omega = \Omega^1 \cup \Omega^2$ with a set of non-overlapping elements. The interface $\Gamma_*$ and the boundary $\Gamma_d = \Gamma^1_d \cup \Gamma^2_d$ are allowed to be embedded in the mesh in the sense that they are allowed to cut through the elements. We construct a piecewise linear approximation to the interface and locate it through a zero isosurface of a level set function in the domain. For every B-spline element that is intersected by an interface, we have twice as many basis functions as an “uncut” element to facilitate the construction of the spline interpolant on either side of the interface, see Figure 3.2. We represent the overlapping element formulation for a cut quadrilateral with a line element in Figure 3.3.

The finite element discretization is now constructed on this overlapping domain.
Figure 3.2: The domain is divided in two sub-domains $\Omega^1$ and $\Omega^2$ by an embedded surface $\Gamma_*$. The black circles are the duplicated nodes whose support are cutt ed by the interface.

The test and trial functions are given by:

$$w^h(x) = \sum_{m=1}^{2} \sum_{i \in I^m} H^m N_i(x) \ c_i$$  \hspace{1cm} (3.6a) \hspace{1cm}

$$u^h(x) = \sum_{m=1}^{2} \sum_{i \in I^m} H^m N_i(x) \ d_i$$  \hspace{1cm} (3.6b) \hspace{1cm}
Figure 3.3: Overlapping element formulation for a quadrilateral element cut by an embedded interface. By construction, each element is associated with a $4 \times 4$ grid of bicubic B-splines. The black circles are the physical nodes corresponding to the background mesh and the hollow circles are the ghost nodes.

where $I^m$ is the set of all nodes whose corresponding shape functions have support overlapping the discretized domain $\Omega^m$ and $H^m$ is the characteristic function given by

$$H^m(x) = \begin{cases} 
1 & \text{if } x \in \Omega^m_h \\
0 & \text{otherwise}
\end{cases} \quad (3.7)$$

3.4 Geometrical representation

We employ a piecewise-linear representation for the interface using a level-set approximation over a set of elements that are in general distinct from those used for the B-spline approximation. The piecewise-linear approximation to the geometry introduces some error as shown in Figure 3.4, but it greatly simplifies integration over the material regions.

The approach is to improve the geometrical representation of the interface while keeping the same “coarse” mesh for the B-splines, thus keeping the same number of degrees of freedom. Accordingly, the level set is constructed over a hierarchically refined sub-mesh. Each cell cut by the zero iso-surface is split and forms a hierarchical
structure. For example, a quadrilateral in 2D is divided into 4 quadrilaterals with half element size. Level set values are then computed on the new vertices. The procedure is done recursively up to a user defined maximum depth. The associated improvement in the geometrical representation for a curved interface is illustrated in Figure 3.5.

In our study, we measure the numerical bulk error using the $L_2$ norm as described in Huynh et al. (2013):
\[ \| u - u^h \|_{\Omega} := \left( \int_{\Omega} (u - u^h)^2 \right)^{1/2} = \left( \int_{\Omega \setminus A_h} (u - u^h)^2 + \int_{A_h} (u - u^h)^2 \right)^{1/2} \] (3.8)

The \( A_h \), shown in Figure 3.4, is the region between the exact interface and the computed level set. The complement of \( A_h \) in \( \Omega \) is denoted by \( \Omega \setminus A_h \). In this region, for a sufficiently refined discretization, \( u^h \sim u^1 \) while the true solution \( u = u^2 \). Consequently, we assume that the absolute value \( |u - u^h| \) is on the order of the jump \( \bar{i} \). For simplicity, we assume that the jump \( \bar{i} \) is constant in the region \( A_h \) and then the \( L_2 \) error norm is given by

\[ \| u - u^h \|_{\Omega} \approx \left( \int_{\Omega \setminus A_h} (u - u^h)^2 + \text{volume}(A_h) \bar{i}^2 \right)^{1/2} \] (3.9)

The approximation error represented by the first term in the above equation decreases as \( (h_a)^{p+1} \), \( h_a \) being the relative approximation element size, and \( p \) the degree of precision in the basis functions. Numerical experiments (Dréeau et al., 2010) show that with a piecewise-linear level set the volumetric error in the second term decreases as \( h_g \), \( h_g \) being the relative geometrical element size. We wish to identify the appropriate size for \( h_g \) such that the geometrical error converges at the same rate as the approximation error. This implies that the geometrical element size is:

\[ h_g = (h_a)^{p+1} \] (3.10)

With cubic B-splines, this relation becomes:

\[ h_g = h_a^4 \] (3.11)

This relationship gives the correct refinement level for the geometry in order to maintain optimal rates of convergence.
3.5 Analysis

In this section, conditions for the coercivity of the Nitsche forms are derived for embedded interface problems. In the discrete setting, the variational form 3.3 can be written as:

\[ a_b (w^h, u^h) + a_i (w^h, u^h) = l_b (w^h) + l_i (u^h) \quad \forall w^h \in V^h \]  

(3.12)

We analyze the bilinear forms related to the boundary and interfacial constraints respectively, thus the bulk quantity over the domain is divided equally into each bilinear form. Furthermore, we make use of the following definitions of the \( L_2 \) norm of a quantity over a domain (\( \Omega \)) or along a Dirichlet boundary (\( \Gamma_d \)) or along an interface (\( \Gamma_* \)).

\[
\| \nabla w \| = \left( \sum_m \int_{\Omega_m} \nabla w^m \cdot \kappa^m \nabla w^m \, d\Omega \right)^{1/2} 
\]

(3.13a)

\[
\| w \|_{\Gamma_d} = \left( \sum_m \int_{\Gamma_d^m} w^m w^m \, d\Gamma \right)^{1/2} 
\]

(3.13b)

\[
\| w \|_{\Gamma_*} = \left( \int_{\Gamma_*} w w \, d\Gamma \right)^{1/2} 
\]

(3.13c)
3.5.1 Conditions for coercivity

To examine coercivity of the discrete version of the Nitsche bilinear form \( a_b(w^h, u^h) \), we start by using \( w^h \in V^h \) in place of \( u^h \). The bilinear form can be written as

\[
a_b(w^h, w^h) + a_i(w^h, w^h) = \frac{1}{2} \sum_{m} \int_{\Omega_m} \nabla w^m \cdot \kappa^m \nabla w^m \, d\Omega \\
- 2 \sum_{m} \int_{\Gamma_d} w^m \kappa^m \left( \nabla w^m \cdot n^m \right) \, d\Gamma \\
+ \alpha_b \sum_{m} \int_{\Gamma_d} w^m w^m \, d\Gamma + \frac{1}{2} \sum_{m} \int_{\Omega_m} \nabla w^m \cdot \kappa^m \nabla w^m \, d\Omega \\
- \int_{\Gamma_\ast} \left[ [w^h] \langle \kappa \nabla w^h \rangle_\gamma \cdot n \right] \, d\Gamma - \int_{\Gamma_\ast} \left[ [w^h] \langle \kappa \nabla w^h \rangle_\gamma \cdot n \right] \, d\Gamma \\
+ \alpha_i \int_{\Gamma_\ast} \left[ [w^h] \right] \left[ [w^h] \right] \, d\Gamma \tag{3.14}
\]

Applying definitions (3.13), the above equation can be rewritten as

\[
a_b(w^h, w^h) + a_i(w^h, w^h) = \frac{1}{2} \| \nabla w^h \|^2 - 2 \sum_{m} \int_{\Gamma_d} w^m \kappa^m \left( \nabla w^m \cdot n^m \right) \, d\Gamma \\
+ \alpha_b \| w^h \|^2_{1, \Gamma_d} + \frac{1}{2} \| \nabla w^h \|^2 - 2 \int_{\Gamma_\ast} \left[ [w^h] \langle \kappa \nabla w^h \rangle_\gamma \cdot n \right] \, d\Gamma \\
+ \alpha_i \| [w^h] \|_{1, \Gamma_\ast}^2 \tag{3.15}
\]

Employing Young’s inequality with \( \epsilon_b \) and \( \epsilon_i \) in (3.15), we get

\[
a_b(w^h, w^h) + a_i(w^h, w^h) \geq \frac{1}{2} \| \nabla w^h \|^2 - \frac{1}{\epsilon_b} \| w^h \|^2_{1, \Gamma_d} - \epsilon_b \| \left( \kappa \nabla w^h \cdot n \right) \|^2_{1, \Gamma_d} \\
+ \alpha_b \| w^h \|^2_{1, \Gamma_d} + \frac{1}{2} \| \nabla w^h \|^2 - \frac{1}{\epsilon_i} \| [w^h] \|_{1, \Gamma_\ast}^2 \\
- \epsilon_i \left[ \langle \kappa \nabla w^h \rangle_\gamma \cdot n \right]_{1, \Gamma_\ast}^2 + \alpha_i \| [w^h] \|_{1, \Gamma_\ast}^2 \tag{3.16}
\]
where $\epsilon_b > 0$ and $\epsilon_i > 0$.

There exist mesh dependent constants $C_b$ and $C_i$ such that

$$\| (\kappa \nabla w^h \cdot n) \|_{\Gamma_d}^2 \leq C_b \| \nabla w^h \|^2$$  \hspace{1cm} (3.17)

$$\| \langle \kappa \nabla w^h \rangle_{\gamma} \cdot n \|^2_{\Gamma_*} \leq C_i \| \nabla w^h \|^2$$  \hspace{1cm} (3.18)

The bilinear form can therefore be rewritten as

$$a_b (w^h, w^h) + a_i (w^h, w^h) \geq \left( \frac{1}{2} - \epsilon_b C_b \right) \| \nabla w^h \|^2 + \left( \alpha_b - \frac{1}{\epsilon_b} \right) \| w^h \|^2_{\Gamma_d}$$ \hspace{1cm} (3.19)

$$+ \left( \frac{1}{2} - \epsilon_i C_i \right) \| \nabla w^h \|^2 + \left( \alpha_i - \frac{1}{\epsilon_i} \right) \| [w^h] \|^2_{\Gamma_*}$$

Taking $\epsilon_b < \frac{1}{2C_b}$ and $\epsilon_i < \frac{1}{2C_i}$, coercivity of the bilinear form is ensured when $\alpha_b > \frac{1}{\epsilon_b}$ and $\alpha_i > \frac{1}{\epsilon_i}$. In the present study, values of $\alpha_b = 4C_b$ and $\alpha_i = 4C_i$ are used.

### 3.5.2 Stability Parameter Evaluation

The mesh dependent constants $C_b$ in equation (3.17) and $C_i$ in equation (3.18) can be determined globally by solving a generalized eigenvalue problem at the Dirichlet boundary and interface. The eigenvalue problem can be posed as:

$$Ax = \lambda Bx$$  \hspace{1cm} (3.20)

where the elements of $A$ and $B$ follow directly from equations (3.17) and (3.18) as proposed in Embar et al. (2010b). Instead of using this global approach, the present work employs a local element-wise approach to determine the stabilization parameters where we define $\gamma_e^m$, $\Omega_e^m$ and $\Gamma_{\gamma e}$ are the element-wise weights, domain and interface respectively.

The weights $\gamma$ in the weak form can be chosen as

$$\gamma_e^m = \frac{\text{meas}(\Omega_e^m) / \kappa^m}{\text{meas}(\Omega_e^m) / \kappa^1 + \text{meas}(\Omega_e^m) / \kappa^2} \quad \text{for } m = 1, 2$$  \hspace{1cm} (3.21)
which was proposed by Annavarapu et al. (2012a). This choice of $\gamma$ effectively alleviates the numerical issues associated with pathological cases such as elements with very small partial volume fractions and/or large material heterogeneities. It was proposed for constant strain triangular and tetrahedral elements, but its performance with higher order elements has not been investigated.

Here, we propose a general method to find the weights that is particularly suited to higher order elements. Since the flux within an element no longer remains constant, the qualitative dependence between the weights and the stabilization parameters has to be established via additional element level eigenvalue calculations.

Rather than make use of (3.18) directly, we consider two one-sided inequalities. There exist mesh dependent constants $C^1_e$ and $C^2_e$ such that

$$
\int_{\Gamma^{*e}} (\nabla w^h \cdot \mathbf{n}) (\nabla w^h \cdot \mathbf{n}) d\Gamma \leq C^1_e \int_{\Omega^1_e} \nabla w^h \cdot \nabla w^h d\Omega \quad (3.22a)
$$

$$
\int_{\Gamma^{*e}} (\nabla w^{2h} \cdot \mathbf{n}) (\nabla w^{2h} \cdot \mathbf{n}) d\Gamma \leq C^2_e \int_{\Omega^2_e} \nabla w^{2h} \cdot \nabla w^{2h} d\Omega \quad (3.22b)
$$

A good estimate of $C^m_e$ can be determined by solving a generalized eigenvalue problem as mentioned above.

For the weighted-average flux at the interface we have:

$$
\| \langle \kappa \nabla w^h \rangle \cdot \mathbf{n} \|_{\Gamma^{*e}}^2 = \int_{\Gamma^{*e}} (\gamma^1_e \kappa^1 \nabla w^h \cdot \mathbf{n} + \gamma^2_e \kappa^2 \nabla w^{2h} \cdot \mathbf{n})^2 d\Gamma,
$$

$$
\leq \int_{\Gamma^{*e}} \left( (\gamma^1_e \kappa^1 \nabla w^h \cdot \mathbf{n})^2 (1 + \delta) + (\gamma^2_e \kappa^2 \nabla w^{2h} \cdot \mathbf{n})^2 (1 + 1/\delta) \right) d\Gamma,
$$

$$
\leq \int_{\Omega^1_e} \left( (\gamma^1_e)^2 (\kappa^1)^2 C^1_e \nabla w^h \cdot \nabla w^h (1 + \delta) \right) d\Omega
$$

$$
+ \int_{\Omega^2_e} \left( (\gamma^2_e)^2 (\kappa^2)^2 C^2_e \nabla w^{2h} \cdot \nabla w^{2h} (1 + 1/\delta) \right) d\Omega \quad (3.23)
$$
where the first line follows from the definition of the weighted average flux at the interface, and the second on using Young’s inequality for any \( \delta > 0 \). The third line follows on using the generalized inverse estimate (3.22a) and (3.22b), assuming \( \kappa^1 \) and \( \kappa^2 \) are constant over the element. If we choose

\[
\delta = \frac{C^2_e \kappa^2 (\gamma^2_e)^2}{C^1_e \kappa^1 (\gamma^1_e)^2}
\]

then

\[
\| \langle \kappa \nabla w^h \rangle \gamma \cdot n \|_{\Gamma_{se}}^2 \leq \sum_m \int_{\Omega^m_e} \nabla w^{m^h} \cdot \kappa^m \nabla w^{m^h} d\Omega \times \left( C^1_e (\gamma^1_e)^2 \kappa^1 + C^2_e (\gamma^2_e)^2 \kappa^2 \right)
\]

(3.24)

Then from (3.24), (3.22a), (3.22b) and (3.18), the mesh dependent parameter \( C_i \) obeys the following relation:

\[
C_i \geq \left( C^1_e (\gamma^1_e)^2 \kappa^1 + C^2_e (\gamma^2_e)^2 \kappa^2 \right)
\]  

(3.25)

We notice that as \( (C^m_e) \to \infty \) which might be caused by elements with small volume fractions (see (3.22a) and (3.22b)) or as \( \kappa^m \) tends to a large value, the above would result in an unusually large estimate for the element level stabilization parameter.

Here, we propose a choice of the weights

\[
\gamma^m_e = \frac{1/((C^m_e \kappa^m))}{1/(C^1_e \kappa^1) + 1/(C^2_e \kappa^2)}
\]

(3.26)

which yields

\[
C_i \geq \frac{1}{1/(C^1_e \kappa^1) + 1/(C^2_e \kappa^2)}
\]

(3.27)

Clearly, this choice of weights alleviates the numerical issues for the aforementioned pathological cases and provide us with a robust form of the Nitsche’s method for higher order basis functions. As an additional remark, we note that for constant strain elements, the inverse estimate constant \( C^m_e \) represents the ratio of \( \text{meas}(\Gamma_{se}) \)
to \( \text{meas}(\Omega^m_e) \), and therefore the weights and the stabilization parameters become identical to (3.21) proposed by Annavarapu et al. (2012a).

Now we revisit the generalized eigenvalue problems for determining the mesh dependent constant for our present method and the one proposed by in Annavarapu et al. (2012a). For robustness, it is important that the solutions to those generalized eigenvalue problems are obtained accurately.

We begin by considering the generalized eigenvalue problem associated with the weighting proposed by Annavarapu et al. (2012a). The elements of the matrices \( \mathbf{A} \) and \( \mathbf{B} \) in this method are given by

\[
[A]_{ij} = \int_{\Gamma_{\neq e}} \langle \kappa \nabla N_i^\gamma \cdot \mathbf{n} \rangle \langle \kappa \nabla N_j^\gamma \cdot \mathbf{n} \rangle d\Gamma \quad (3.28a)
\]

\[
[B]_{ij} = \sum_m \int_{\Omega^m_e} \nabla N_i^m \cdot \kappa^m \nabla N_j^m \ d\Omega \quad (3.28b)
\]

Matrix \( \mathbf{A} \) and \( \mathbf{B} \) have two blocks and the size of each block is \( 16 \times 16 \) (for bicubic B-splines in two dimensions) representing the contribution from each side of the interface. We observe that \( \mathbf{B} \) is rank deficient by construction. The conditioning of \( \mathbf{B} \) would become even worse in the presence of small cut elements and large material heterogeneities. The reason is that, for small cut elements, the entries of the block associated with small volume are close to zero; for large material contrast, the ratio of the large entries associated with large \( \kappa \) in \( \mathbf{B} \) to the small entries associated with small \( \kappa \) becomes large.

By contrast, with our approach the elements of matrices \( \mathbf{A} \) and \( \mathbf{B} \) are given by

\[
[A]^m_{ij} = \int_{\Gamma_{\neq e}} (\nabla N_i^m \cdot \mathbf{n}) (\nabla N_j^m \cdot \mathbf{n}) d\Gamma \quad (3.29a)
\]

\[
[B]^m_{ij} = \int_{\Omega^m_e} \nabla N_i^m \cdot \nabla N_j^m \ d\Omega \quad (3.29b)
\]
In this case, the generalized eigenvalue problem is performed on each side of the interface, and matrix $A^m$ and $B^m$ have only one $16 \times 16$ block. As a result, the small cut elements and large material heterogeneities do not have a great influence on the conditioning of $B^m$.

The matrix $B$ is singular, with a null space spanned by a single vector corresponding to a constant field. This vector is also in the null space of $A$. Consequently, this vector is an eigenvector of the generalized eigenvalue problem associated with an indeterminate eigenvalue, which can be a source of ill conditioning in practical calculations.

To overcome this difficulty, we deflate both matrices to remove the singularity of $B$. Since the null space of $B$ is known, this is a straightforward procedure. Here we consider the $16 \times 16$ matrices (3.29a) and (3.29b). The common nullspace basis for these two matrices is a vector $[1 \ 1 \ 1 \cdots]_{1\times16}$, denoted by $x$. Suppose that we want to obtain the deflated matrices with the first row being zeros in each. Denote the first entry of $x$ by $x_1$ and the first row of $A$ and $B$ by $a_1$ and $b_1$ respectively. The deflated matrices $\tilde{A}$ and $\tilde{B}$ are given by

$$\tilde{A} = A - \frac{1}{x_1} x^T a_1 \quad (3.30a)$$

$$\tilde{B} = B - \frac{1}{x_1} x^T b_1 \quad (3.30b)$$

Now the first rows of $\tilde{A}$ and $\tilde{B}$ become zeros and then we can delete the first row and the first column and obtain two order-reduced $15 \times 15$ matrices denoted by $\hat{A}$ and $\hat{B}$. Consequently the deflated eigenvalue problem is given by:

$$\hat{A}x = \hat{\lambda} \hat{B}x \quad (3.31)$$

The deflated eigenvalue problem retains the determinate eigenvalues of the original problem. The deflated matrix $\hat{A}$ remains singular, but this poses no difficulty as we are interested in the largest eigenvalue of the deflated eigenvalue problem.
Table 3.1: Refinement level for cubic B-spline basis function.

<table>
<thead>
<tr>
<th>elements by side</th>
<th>$h_a$</th>
<th>$h_g$</th>
<th>refinement level</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>$\frac{1}{10}$</td>
<td>$\frac{1}{10} \times \left(\frac{1}{2}\right)^0$</td>
<td>0</td>
</tr>
<tr>
<td>20</td>
<td>$\frac{1}{20}$</td>
<td>$\frac{1}{20} \times \left(\frac{1}{2}\right)^3$</td>
<td>3</td>
</tr>
<tr>
<td>40</td>
<td>$\frac{1}{40}$</td>
<td>$\frac{1}{40} \times \left(\frac{1}{2}\right)^6$</td>
<td>6</td>
</tr>
</tbody>
</table>

It is worth remarking that in plane elasticity the null space of $B$ has dimension three, and again is a subspace of the null space of $A$. Since the basis of the null space of $B$ is known, the same deflation procedure may be employed.

3.6 Numerical examples

We consider a series of two-dimensional benchmark problems to evaluate the accuracy and robustness of the proposed methods. Throughout this section and in the figure legends, we denote our method by higher order $\gamma$-Nitsche, the weighted form of Nitsche’s method proposed by Annavarapu et al. (2012a) by $\gamma$-Nitsche, and the standard form by simply Nitsche. The cubic B-spline basis functions are employed in all the following examples except for the problem of bending of a cantilevered elongated rectangle where quadratic splines are used. We determine the stability parameters by solving the deflated eigenvalue problem in all the following examples and further highlight its importance in the planar interface problem. Table 3.1 shows the chosen refinement level for each mesh which is used in our convergence studies. Relative error norms are used for convergence studies in all numerical examples that follow. The accuracy in the bulk field is evaluated using a normalized $L_2$ norm of the error,

$$\varepsilon_{L_2}(u)_{\Omega} = \frac{\sum_m \| u^h - u \|_{L_2,\Omega^m}}{\sum_m \| u \|_{L_2,\Omega^m}} \quad (3.32)$$

The accuracy in the flux is examined using a normalized $L_2$ norm of the error on
the interface,
\[ \varepsilon_{L^2} |_{\Gamma_*} = \frac{\| \phi^h - \phi \|_{L^2, \Gamma_*}}{\| \phi \|_{L^2, \Gamma_*}} \]  \quad (3.33)

where \( \phi \) denotes the flux. In the following examples, we consider the flux from both sides of the interface as well as the average. In the last example, the energy norm is given by

\[ \varepsilon_{\text{energy}} = \left( \frac{1}{2} \sum_{m} \int_{\Omega^m} (\epsilon^h - \epsilon) : (\sigma^h - \sigma) \text{d}\Omega \right)^{\frac{1}{2}} \]  \quad (3.34)

where \( \epsilon \) and \( \sigma \) are strain and stress respectively.

### 3.6.1 Fixed Circular Plate

As a first example, we consider a one-sided fourth order problem with a curved boundary embedded in the background mesh. The boundary is represented by the linear level sets with the proposed refinement approach. The boundary value problem is given by

\[ \Delta^2 u = q \]  \quad (3.35a)

\[ u (r = 0.5) = 0 \]  \quad (3.35b)

\[ \frac{\partial u}{\partial r} (r = 0.5) = 0 \]  \quad (3.35c)

where, \( r \) is the radial distance from the plate center. A uniform distribution of transverse loading is assumed such that,

\[ q = \frac{q_0}{D} \]  \quad (3.36)

The field \( u \) represents the transverse plate displacement while \( D \) is the flexural stiffness of the plate. The ratio \( (q_0/D) \) is taken to be unity in this example. The analytical solution for this problem is given by:

\[ u = \frac{q_0}{64D} \left(1 - r^2\right)^2 \]  \quad (3.37)
Figure 3.6: (a) Discretization for the fixed circular plate with uniform loading. (b) Finite element approximation.

Figure 3.7: Convergence rates in $L_2$ norm.

Figure 3.6(a) shows a sample uniform discretization where the grid does not conform to the Dirichlet boundary. The boundary is represented by a linear level set function with local refinement. In order to avoid a singular global stiffness matrix, the nodes whose support do not overlap the physical domain are eliminated from numerical computation. The corresponding finite element approximation is shown in Figures 3.6(b). Convergence studies give spatial convergence at a near quartic rate as shown in Figure 3.7. Thus, dealing with curved embedded boundaries, the
proposed method allows one to use higher order approximation shape functions while keeping the optimal convergence rate.

### 3.6.2 Kidney-shaped interface

We next consider a problem described in Huynh et al. (2013). The problem is defined on the domain $\Omega = [-1, 1] \times [-1, 1]$. The shape of the interface is governed by the following level set function

$$
\psi = (2 ((x + 0.5)^2 + y^2) - x - 0.5)^2 - ((x + 0.5)^2 + y^2) + 0.1
$$

The jump problem is given by (3.1) where $\kappa^1 = 1$ and $\kappa^2 = 10$. The interfacial condition and source term are derived from the following exact solution

$$
u = \begin{cases} 
\sin(2x^2 + y^2 + 2) + x & \text{in } \Omega^1 = \{x : \psi(x) < 0\} \\
0.1 \cos(1 - x^2 - y^2) & \text{in } \Omega^2 = \{x : \psi(x) > 0\}
\end{cases}
$$

We also employ (3.39) on the outermost boundary of $\Omega$ as a Dirichlet condition, weakly imposed with Nitsche’s method.

![Diagram](image)

(a) local refined geometric cells with level 3  
(b) finite element solution

**Figure 3.8:** Local geometric refinement and results for the kidney-shaped problem.
In this curved interface problem, the accuracy highly depends on the geometric representation of the interface. In order to obtain an optimal convergence rate, we choose the correct refinement level from Table 3.1. The finite element mesh with embedded interface is shown in Figure 3.8(a), where the geometrical refinement level of 3 is shown. The corresponding solution is shown in Figure 3.8(b). We report optimal rates of convergence in the $L_2$ norm of the bulk error with an optimal geometrical representation of the curved interface, as shown in Figure 3.9(a). Our results appear to contradict the claims of Huynh et al. (2013) regarding the accuracy of embedded methods.

Without geometrical refinement, the bulk error converges with only linear order as shown in Figure 3.9(b). Since the flux error is evaluated on the discretized interface, the geometric representation does not have much influence on the normalized error. We report the optimal convergence rate in flux for both cases. However, if we approximate the exact integral using our geometric refinement technique for the most refined case, the error in the weighted interfacial flux for approximation mesh size $h = 0.05$ is 4.17% with geometrical refinement and 14.23% without geometrical refinement, respectively.

![Graphs showing convergence rates](image)

(a) convergence rate with geometrical refinement (b) convergence rate without geometrical refinement

**Figure 3.9:** Comparison of convergence rates for the kidney-shaped problem in $L_2$ norm.
3.6.3 Planar Interface

We next consider a simple two-sided model problem on the domain $\Omega = [0, 1] \times [0, 1]$ with a planar interface. The problem has homogeneous Dirichlet boundary condition on $\Gamma_d = \{x : x = 0, 1\}$ and Neumann boundary conditions on $\Gamma_n = \{x : y = 0, 1\}$. Choosing $f^1 = 1 + x^2$ and $f^2 = 1 + (x - 1)^2$, the problem has a simple analytical solution given by

$$u = \begin{cases} \frac{(3\kappa^1 + \kappa^2)x}{4(\kappa^1)^2 + 4\kappa^1\kappa^2} - \frac{x^2}{2\kappa^1} - \frac{x^4}{12\kappa^1} & \text{in } \Omega^1 = \{x : \psi(x) < 0\} \\ \frac{\kappa^2 - \kappa^1 + (3\kappa^1 + \kappa^2)x}{4(\kappa^2)^2 + 4\kappa^1\kappa^2} - \frac{x^2}{2\kappa^2} - \frac{(1-x)^4}{12\kappa^2} & \text{in } \Omega^2 = \{x : \psi(x) > 0\} \end{cases} \quad (3.40)$$

In this example, to highlight the robustness of the proposed formulation by contrasting its performance with the $\gamma$-Nitsche and classical Nitsche methods, the stability parameters are determined by solving the original generalized eigenvalue problems, i.e. without deflation.

We first study the sensitivity of the bulk and interfacial flux with respect to contrast in material properties across the interface. At the interfacial location in Figure 3.10(a), $\psi(x) = 5x - y - 89/36$, we vary the material parameters such that the
ratio between $\kappa^2$ and $\kappa^1$ ranges from $10^{-20}$ to $10^{20}$. From Figure 3.11(a) and 3.11(b), it is clear that with the classical Nitsche method, the bulk field and interfacial flux is erroneous when the contrast in material properties is large. On the other hand, the $\gamma$-Nitsche and higher order $\gamma$ Nitsche remain well behaved. The error plots for these two weighted Nitsche methods are shown in Figure 3.11(c) and 3.11(d). We report some spikes in the error for the $\gamma$-Nitsche method.

We next examine the bulk and interfacial flux error by varying the interface location (see Figure 3.10(b)) for a high contrast in material properties. The results are shown in Figure 3.12 and Figure 3.13. It is clear from the plots that the classical Nitsche method results in wildly oscillating behavior in the bulk and flux error fields. By contrast, the weighted Nitsche methods yield better performance. We once again report some spikes in the error for the $\gamma$-Nitsche method.

It is worth remarking that the spikes in the error for the $\gamma$-Nitsche method is related to the generalized eigenvalue problem. Without deflation, the conditioning of the matrix in the original generalized eigenvalue problem for $\gamma$-Nitsche is much worse for the pathological cases where those spikes occur. The result is a poor choice for the stability parameter with $\gamma$-Nitsche. By employing the deflated formulation discussed in 3.5.2, the $\gamma$-Nitsche and higher order $\gamma$-Nitsche methods yield identical results. Thus we conclude by remarking that the higher order $\gamma$-Nitsche’s method presented in this work leads to more robust performance and the proposed deflated approach greatly improves the fidelity of the generalized eigenvalue problem. For the purpose of robustness and accuracy, the higher order $\gamma$-Nitsche with deflated eigenvalue formulation is strongly recommended.

50
Figure 3.11: Sensitivity in the bulk and flux error with degree of heterogeneity. The first row shows the comparison between the classical Nitsche method and the higher order $\gamma$-Nitsche method, and the second row shows the comparison between $\gamma$-Nitsche and higher order $\gamma$-Nitsche.

3.6.4 Petal problem

We consider the jump problem where the interface geometry is defined by the collection of points

\[
X = 0.02\sqrt{5} + (0.5 + 0.2\sin(5\theta))\cos(\theta)
\]

\[
Y = 0.02\sqrt{5} + (0.5 + 0.2\sin(5\theta))\sin(\theta) \quad -\pi \leq \theta < \pi
\]
the data is given by:

\[ f_1 = -4(1 + x^2 + y^2)e^{x^2+y^2} \quad \text{in } \Omega^1 \quad (3.41a) \]

\[ f_2 = -16(x^2 + y^2) \quad \text{in } \Omega^2 \quad (3.41b) \]

\[ \bar{i} = \frac{1}{\kappa^2} \left( (x^2 + y^2)^2 - 0.1 \log(2\sqrt{x^2 + y^2}) \right) - \frac{1}{\kappa^2} e^{x^2+y^2} \quad \text{on } \Gamma_* \quad (3.41c) \]

\[ \bar{j} = (4(x^2 + y^2) - 0.1(x^2 + y^2)^{-1})(xn_1 + yn_2) - 4(x^3n_1 + y^3n_2) \quad \text{on } \Gamma_* \quad (3.41d) \]

Figure 3.12: Sensitivity in the bulk and flux error with interfacial location ($\kappa^2/\kappa^1 = 10^{-6}$). The first row shows the comparison between classical Nitsche and higher order $\gamma$-Nitsche and the second row shows the comparison between $\gamma$-Nitsche and higher order $\gamma$-Nitsche. 

on the domain $\Omega = [-1.0, 1.0] \times [-1.0, 1.0]$. The interior region is taken as $\Omega^1$, and
To obtain an optimal convergence rate, we choose the recommended refinement level of the interface and a correct weighting for the interfacial consistency terms. To material contrast, the accuracy depends on both a good geometric representation - Nitsche and the second row shows the comparison between classical Nitsche and higher order Nitsche (higher order).

The analytical solution is

\[ u(x, y) = \begin{cases} 
\frac{1}{\kappa} e^{x^2+y^2} & \text{in } \Omega^1 \\
\frac{1}{\kappa} \left( (x^2 + y^2)^2 - 0.1 \log(2\sqrt{x^2 + y^2}) \right) & \text{in } \Omega^2 
\end{cases} \quad (3.42) \]

In this petal problem, due to the complicated interfacial geometry and large material contrast, the accuracy depends on both a good geometric representation of the interface and a correct weighting for the interfacial consistency terms. To obtain an optimal convergence rate, we choose the recommended refinement level.
from Table 3.1 and employ the higher order $\gamma$-Nitsche method. The finite element mesh with embedded interface is shown in Figure 3.14(a), where the geometrical refinement level of 3 is shown. The corresponding solution is shown in Figure 3.14(b). We report optimal rates of convergence in the $L_2$ norm of the bulk and flux error with the higher order $\gamma$-Nitsche method, as shown in Figure 3.15. The results also indicate that the classical Nitsche method fails to converge in the $L_2$ norm of the interfacial flux error.

3.6.5 **Bending of an elongated cantilevered beam**

As a final example, we consider a problem in plane strain elasticity. The general formulation is well known and not repeated here. The extension of our Nitsche approach is relatively straightforward. Let the domain be a a rectangle of length $L$ and height $D$ as shown in Figure 3.16. The rectangle is clamped at the end $x = 0$ and subjected to a parabolic traction at the free end. The parabolic traction is given by:

$$t(y) = -\frac{P}{2t} \left( \frac{D^2}{4} - y^2 \right)$$

(3.43)
where $I = D^3/12$ is the moment of inertia and $P$ is the resultant.

This example is also considered in Nguyen et al. (2013); Ruess et al. (2014) where the classical Nitsche's form with a simple average weighting in the context of higher order B-splines and NURBS is used. Here we demonstrate the robustness of our approach particularly for the accuracy of the interfacial tractions. The analytical expressions for the displacement field are given by

$$u_x(x, y) = \frac{Py}{6EI} \left[ (6L - 3x)x + (2 + \nu) \left( y^2 - \frac{D^2}{4} \right) \right]$$

$$u_y(x, y) = -\frac{P}{6EI} \left[ 3\nu y^2 (L - x) + (4 + 5\nu) \frac{D^2x}{4} + (3L - x)x^2 \right]$$

(3.44)

Figure 3.15: Convergence rates in $L_2$ norm for the petal problem. The first row shows the case of $\kappa^2/\kappa_1^3 = 10^{-6}$ and the second row shows the case of $\kappa^2/\kappa_1^3 = 10^6$. 
The exact stress components are

\[
\sigma_{xx}(x, y) = \frac{P(L - x)y}{I}, \quad \sigma_{yy}(x, y) = 0, \quad \sigma_{xy}(x, y) = -\frac{P}{2I} \left( \frac{D^2}{4} - y^2 \right) \tag{3.45}
\]

In this example, the exact solution is cubic and so the problem represents a kind of patch test with cubic B-splines. To make the study more interesting, we employ quadratic B-spline basis functions. The Poisson ratio is \( \nu = 0.3 \) and the rectangle dimensions are \( D = 6 \) and \( L = 48 \). The rectangle is divided into two domains by an interface shown in Figure 3.16. The interface is placed at an incline such that there are elements with very small partial volume fractions. The finite element mesh with embedded interface is shown in Figure 3.17. Figure 3.18 demonstrates that the optimal convergence rates are obtained in both displacement and energy for this weakly coupled two-domain jump problem. This result is also compared with the single domain problem without interface. We plot the equivalent traction, \( i.e. \alpha [u] - \langle \kappa \nabla u \rangle_\gamma \cdot n \) at each Gauss point on the interface in Figure 3.19. It is clear that even with an homogeneous material, results obtained with the classical Nitsche’s method are not accurate and barely usable. On the other hand, the results of the higher order \( \gamma \)-Nitsche method remain well behaved.

![Figure 3.16: Geometry and loading for the elongated cantilevered beam.](image-url)
Figure 3.17: Discretization and interface location

Figure 3.18: Convergence rates in $L_2$ norm for the plane strain elasticity problem.

(a) displacement norm
(b) energy norm

Figure 3.19: Comparison of surface tractions obtained using classical Nitsche with those obtained using higher order $\gamma$ Nitsche.

(a) Normalized tangential traction along the interface
(b) Normalized normal traction along the interface

3.7 Summary

This Chapter presents a robust spline-based finite element method for embedded interface problems. In the vicinity of the interface, discretization is based on the use of overlapping B-spline basis functions, akin to enrichment with the Heaviside func-
tion in the X-FEM. The constraints along the boundary and interface are enforced weakly by employing a variation of Nitsche’s method. The accuracy of the method stems from a number of considerations. Firstly, in order to obtain optimal rates of convergence, the geometrical representation and the finite element approximation are balanced. In our framework, a level set function is used to implicitly represent the interface. The linear discretization of the level set is defined on a hierarchical sub-mesh in the vicinity of the interface, such that the accuracy of the geometrical representation is sufficient. Secondly, we have shown how the choice of weights in the Nitsche consistency terms has a great influence on the accuracy and robustness of the method. In our approach, the weights for the higher order basis functions are obtained by first determining two mesh dependent constants. These two constants are bounded from below by the maximum eigenvalue problem. The solution to the generalized eigenvalue problem in the finite element formulation is further stabilized by matrix deflation.
4

Adaptive Refinement of Hierarchical Spline-based Finite Elements

4.1 Introduction

Adaptivity has been an issue that the IGA community has examined in several different contexts. This spans early work on multipatch refinement, to work with T-splines, to more recent work on hierarchical B-splines. These works can all be viewed with the perspective that while adaptivity is often desirable, the development of elegant and efficient algorithms can be challenging. This Chapter concerns adaptive refinement with B-splines. These functions are becoming increasingly popular for use in both computer aided design (CAD) and finite element approximations to solutions of partial differential equations.

Isogeometric analysis (Hughes et al., 2005a) has been largely based on NURBS due to their preeminence in engineering design. Although smooth NURBS basis functions offer important computational advantages over standard finite elements (Cottrell et al., 2006; Evans et al., 2009), local refinement is severely limited by its tensor product construction. The refinement of NURBS has to propagate throughout the
global domain. Instead of implementing local refinement on a single domain, one approach is to subdivide the parametric domain into several patches and refine patches uniformly. This is followed by the weak imposition of continuity constraints across patch boundaries. This multipatch approach was discussed in Cottrell et al. (2009) and Kagan et al. (1998, 2003) for isogeometric analysis and spline-based finite elements. The drawback of this approach is that one only obtains $C^0$ continuity between patches and it is difficult to make the refinement process truly adaptive and dynamic.

T-splines were introduced as a superior alternative to NURBS. This is due in part not only to their ability to model complicated designs as a single, watertight geometry (Sederberg et al., 2003) but also to facilitate local refinement (Sederberg et al., 2004). These properties make T-splines a promising technology for isogeometric discretization. The basic idea of T-splines relies on the viewpoint of a basis and each basis function (also called blending function) is defined on a local knot vector. T-splines can be seen as a mesh-based technique in which all of the basis functions are constructed from mesh structures. T-splines were initially defined for degree three and the generalization to arbitrary polynomial degree is complicated. In some pathological cases the refinement of T-splines is not as local as one would expect (Bazilevs et al., 2010; Dörfel et al., 2010) and might introduce a linear dependence (Buffa et al., 2010). Recently, an effective T-spline local refinement algorithm has been developed to ensure linear independence and reduce the complexity of the allowable mesh topologies during refinement (Scott et al., 2012; Li et al., 2012), but this algorithm is still under investigation in three dimensions.

Hierarchical tensor product splines (HTP-splines) were also introduced in Deng et al. (2008) as a CAD technique. Although the required local behavior of the refinement algorithm and linear independence can be guaranteed, the regularity of the basis is restricted to be $C^1$. The details about the application of this method in isogeometric analysis can be found in Nguyen-Thanh et al. (2011).
The hierarchical B-splines first introduced by Forsey and Bartels (1988) allow local refinement of a surface by adding patches representing finer details. The resulting set of basis functions is not always linearly independent. The construction for linearly independent hierarchical B-spline basis functions was developed by Kraft (1997), and this approach is better suited to the finite element method. The hierarchical approach was further investigated by Höllig and co-workers for local mesh refinement in spline-based finite elements (Höllig et al., 2001; Höllig, 2003). The basic idea of hierarchical B-spline refinement is to replace coarser B-spline functions by finer B-spline functions. In this sense, the hierarchical B-spline approach can be included in the basis refinement framework (Grinspun, 2003) in which other popular methods, such as finite element, wavelets and subdivision, can also be applied. The basis refinement method adopts a basis viewpoint. With this approach one does not need to fix problematic T-vertices arising from bisecting an element. Moreover, the continuity of the discrete space does not change. In the context of isogeometric analysis, it should be noted that the geometric description does not change under refinement. In this Chapter we focus on developing hierarchical spline-based finite elements, with particular emphasis on refinement and coarsening algorithm and data transfer techniques.

This Chapter is organized as follows. In Section 4.2, we introduce hierarchical spline-based finite elements. In Section 4.3, we discuss the object-oriented implementation of the proposed approach. The refinement and coarsening algorithms and data transfer techniques are provided in Section 4.4. Numerical examples that demonstrate the efficacy of the proposed strategy are presented in Section 4.5. Finally, we provide a summary and concluding remarks in the last section.
4.2 Hierarchical B-splines

4.2.1 Hierarchical B-spline space

With standard adaptivity, elements are refined and coarsened. This gives rise to T-junctions and the need to impose multi-point constraints. By contrast, the hierarchical approach replaces individual basis functions on coarse grids with a set of basis functions constructed over a finer grid. As such, it may be viewed as a node-based approach to adaptivity rather than an element-based approach. Thus, this approach requires only that the basis functions used be refinable, and for B-splines this requirement is fulfilled by the following subdivision formula.

The univariate B-spline basis function with grid size $h$ can be expressed as a linear combination of B-splines with grid size $h/2$:

$$N_p(\xi) = 2^{-p} \sum_{j=0}^{p+1} \binom{p+1}{j} N_p(2\xi - j)$$

The subdivision of an original uniform B-spline for polynomial degrees $p = 1$ through 3 is illustrated in Figure 4.1.

![Subdivision of an original uniform B-spline into $p + 2$ contracted B-splines of half the knot span width, illustrated for polynomial degrees $p = 1$ through 3.](image)

**Figure 4.1**: Subdivision of an original uniform B-spline into $p + 2$ contracted B-splines of half the knot span width, illustrated for polynomial degrees $p = 1$ through 3.

The extension to multivariate B-splines is straightforward with the use of tensor
product structures. The subdivision equation then becomes:

$$B_p(\xi) = \sum_{j} \left( \prod_{l=1}^{\dim} 2^{-p_l} \left( \frac{p_l + 1}{J_l} \right) N_{p_l}(2\xi^l - j_l) \right)$$  \hspace{1cm} (4.2)$$

Following the subdivision relationships derived above, one can define a hierarchical B-spline space (Kraft, 1997; Höllig, 2003). The hierarchical B-spline space $$S^K$$, corresponding to a nested sequence of domains

$$\Omega = \Omega_0 \supset \Omega_1 \supset \Omega_2 \supset \cdots \supset \Omega_K = \emptyset$$  \hspace{1cm} (4.3)$$
is spanned by the B-splines

$$B_{h_k}^{i}, \quad i \in I_k, h_k = 2^{-k}h, 0 \leq k < K,$$  \hspace{1cm} (4.4)$$

where $$I_k$$ denotes the indices $$i$$ for which $$\bar{\Omega} \cap \text{supp} \ B_{h_k}^{i}$$ is a subset of $$\bar{\Omega}_k$$ (with nonzero measure) but not contained in $$\bar{\Omega}_{k+1}$$. In other words, we replace the coarse basis functions $$B_{h_k}^{i}$$ with the fine basis functions $$B_{h_{k+1}}^{i}$$ in $$\Omega_{k+1}$$. Hence we arrive at the hierarchical nestedness relation

$$S^0 \subset S^1 \subset \cdots \subset S^K,$$  \hspace{1cm} (4.5)$$

where the basis of the $$S^0$$ is initialized as $$\{ \phi \in B_{1,p}^{h_0} \mid \text{supp} \ \phi \subseteq \Omega_0 \}$$. Figure 4.2 provides a simple illustration of how two coarse-scale basis functions on level 0 can be replaced with finer-scale basis functions on level 1. The collection of functions indicated with solid lines (at both levels 0 and 1) then form a hierarchical B-spline space.

This approach yields a linearly independent set of functions, as proved by Kraft (1997). An alternative definition of nested spaces and domains provided by Vuong et al. (2011) leads to a more flexible refinement region. More specifically, in Vuong’s definition the boundaries of adjacent domains $$\Omega_k$$ and $$\Omega_{k+1}$$ need not be disjoint.
The hierarchical nestedness relation (4.5) provides an elementary refinement strategy: since \( S_k \subset S_{k+1} \), based on subdivision formulas, any B-spline function \( B_{i,p}^{h_k} \) on level \( k \) can be exactly represented by the finer basis \( B_{i,p}^{h_{k+1}} \). Therefore, the subdivision equation (4.2) is also called the refinement equation (Krysl et al., 2003), i.e.

\[
\phi_i^{(k)}(x) = \sum_j a_{ij}^{(k+1)} \phi_j^{(k+1)}(x)
\]

where we simplify the hierarchical B-spline function \( B_{i,p}^{h_k} \) as \( \phi_i^{(k)} \) and write the binomial coefficient as \( a_{ij} \). The refinement equation results in a statement that the continuity of the basis space is preserved since any finite linear combination of \( C^m \) functions remains \( C^m \) continuous.

![Figure 4.2: Hierarchical refinement of a set of B-spline basis functions (active basis functions are indicated with solid lines).](image)

We now introduce some terminology. The refinement set of the coarser-scale function \( \phi_i^{(k)} \) consists of all the basis functions that are shown on the right side of the refinement equation. If \( \phi_j^{(k+1)} \) belongs to the refinement set, we refer to \( \phi_i^{(k)} \) as a parent of \( \phi_j^{(k+1)} \) and \( \phi_j^{(k+1)} \) as a child of \( \phi_i^{(k)} \). Generally a function may have multiple parents as well as multiple children. For non-interpolatory basis functions, the fact that none of the child basis functions is owned by a single parent makes
the refinement strategies mentioned in Krysl et al. (2003) unsuitable for B-splines of polynomial degree $p > 1$.

4.2.2 The weighted hierarchical basis

The partition of unity property of uniform B-splines, combined with the non-negativity and compact support of the functions, leads to the fact that a B-spline curve is completely contained within the convex hull defined by its control points. This is important for the representation of geometry in computer graphics applications. By satisfying a partition of unity, we regain the affine covariance property (Cottrell et al., 2009; Hughes et al., 2005a) which is essential for isogeometric analysis. The work done by Melenk and Babuška (1996) further highlights the importance of attaining a partition of unity in the standard finite element method. However, the hierarchical basis does not constitute a partition of unity on its own. Thus, some modifications are needed to recover this property within the framework of the hierarchical approach.

![Figure 4.3](image)

**Figure 4.3**: The weighted hierarchical basis: the sum of all the basis function values is 1 (the bottom image shows all active basis functions on both levels 0 and 1).

To recover a partition of unity, Krysl et al. (2003, 2004) adds a weight to each
basis function and modifies the weights by applying a similar algorithm to the one used for data transfer. This algorithm is effective but only works for basis functions that satisfy the Kronecker delta property. Based on the same idea, a weighted hierarchical basis for B-splines is proposed. As proved by Vuong et al. (2011), any weight associated to a function is zero or strictly positive, ensuring non-negativity. These weights can be easily obtained by recursively applying the refinement relation (4.6). Figure 4.3 shows what the collection of splines looks like for a weighted hierarchical basis that also represents a partition of unity.

4.2.3 The truncated hierarchical basis

The basic idea of the hierarchical approach is to locally enrich the approximation space by adding basis functions on a higher level. As a result, the basis functions associated to different hierarchical levels are overlapping. This situation is even worse for B-spline functions because a B-spline function is supported by $(p + 1)^l$ elements and some of them will extend to the region on which the finer B-spline functions are activated. As mentioned by Schillinger et al. (2012), some parent basis functions can be implicitly represented by a linear combination of some of their inactive children when the other children have been already added into the basis function set. This can be easily understood by rewriting the refinement relation (4.6) as

$$
\phi_i^{(k)}(x) - \sum_{j}^{\text{active}} a_{ij}^{(k+1)} \phi_j^{(k+1)}(x) = \sum_{j}^{\text{inactive}} a_{ij}^{(k+1)} \phi_j^{(k+1)}(x)
$$

(4.7)

The parent basis function now can be replaced by the linear combination of inactive children basis functions on the right side of (4.7). This technique is illustrated in Figure 4.4. The process of truncation not only decreases the overlapping basis functions on different hierarchical levels, but also improves the numerical properties such as conditioning and sparsity of the stiffness matrix. The details of the truncated basis
for hierarchical splines is explained in Giannelli et al. (2012), in which a truncation mechanism is proposed to preserve all the nice properties of hierarchical B-splines, such as linear independence, non-negativity and partition of unity.

Although the truncated hierarchical B-spline basis has some remarkable properties, it leads to a complex algorithm (Schillinger et al., 2012). More specifically, the truncated basis loses its parent-child relation and thus cannot directly apply the refinement relation (4.6). To the best of our knowledge, there is no investigation on the dynamic adaptive refinement method as well as the data transfer scheme for such basis. Furthermore, Giannelli et al. (2012); Schillinger et al. (2012) shows that the condition number of the matrix is improved by applying truncation mechanism but still remains the same order of magnitude as the one of the matrix obtained without truncation. Hence its advantage seems to be moderate and we omit the truncation procedure in our current implementation.

![Figure 4.4: Truncated basis illustration: the truncated basis function is shown in red](image)

4.3 Object-oriented implementation of the hierarchical spline-based FEM

The basis viewpoint adopted by the hierarchical approach greatly reduces the difficulty of implementation since the algorithmic issues of compatibility are entirely
circumvented (Grinspun et al., 2002). Some efficient implementations can be found in Krysl et al. (2004); Vuong et al. (2011); Schillinger et al. (2012); Bornemann and Cirak (2013). More specifically, a tree data structure is employed by Schillinger et al. (2012) and each basis function is defined on the node or leaf. Another interesting subdivision-based technique is presented by Bornemann and Cirak (2013) to facilitate the integration of hierarchical B-splines within a standard finite element framework. Instead of directly integrating the basis functions from different levels, all basis functions are projected onto the finest level based on subdivision formulas and the integration only applies on those finest basis functions. A slightly more general and versatile object-oriented implementation is introduced by Krysl et al. (2003, 2004). Their data structures and algorithms apply to a variety of practically important standard finite elements: triangular and quadrilateral elements in two dimensions, or hexahedral elements in three dimensions. Our implementation of the hierarchical spline-based finite element method is based on this object-oriented framework. Some of the most important data structures related to the hierarchical B-splines are described below.

1. **GEOMETRIC CELL (GCELL)**

   The geometric cell (GCELL) is one of the basic classes in our implementation and it can be understood as an element in the B-spline based finite element method (Embar et al., 2010a) with additional functionalities. GCELL is an abstract class and its specialization depends on the dimension, such as GCELL\_2D for the two dimensional case. The B-spline basis function is defined on GCELLs. When a B-spline basis function is going to be refined, all the GCELLs on which this basis function is defined will be divided into finer, nested GCELLs. The nested GCELLs form a tree data structure.

2. **FIELD**
The class FIELD is a fundamental concept, and any field can be represented by the sum

\[ u^h(x) = \sum_j \phi_j(x)u_j \]  

(4.8)

where \( u^h(x) \) is an approximation in finite element space and \( \phi_j(x) \) is the B-spline basis function and \( u_j \) is the associated degree-of-freedom. Each term in the sum (4.8) is expressed through an object which binds the degree-of-freedom to its basis function. This object is a class parameterized with the number of components, thus the approximated function is able to represent a scalar, vector, or tensor function.

3. BASIS FUNCTION (BFUN)

BFUN provides access to the standard B-spline basis functions. BFUN maintains a list of all the GCELLs over which the B-spline basis function is defined. A member data is introduced in BFUN as the weight associated with this basis function to recover the partition of unity. The value of the weight can be set to 1 if the partition of unity is not needed, otherwise it will be determined by applying a subdivision formula.

To evaluate the spatial integrals we perform numerical quadrature on the GCELL that are the leaves of the refinement hierarchy. The evaluation needs to traverse from the leaf towards the root based on the parent-child relation. The basis function evaluation is expressed on the 0 (coarsest) level and the evaluation on other levels can be performed as follows:

\[ N^k(\xi^l) = N^0(2^k \xi^l) \]  

(4.9a)

\[ \frac{\partial N^k(\xi^l)}{\partial \xi^l} = \frac{1}{2^k} \frac{\partial N^0(2^k \xi^l)}{\partial \xi^l} \]  

(4.9b)

4. BASIS FUNCTION SET(BFUN_SET)

69
The basis function set (BFUN_SET) collects basis functions defined on a particular geometric mesh. The BFUN_SET generates an opaque identifier that is used to access a basis function. Usually this identifier is carried by a finite element node, but due to the hierarchical structures the nodes associated with B-spline functions on different levels may refer to the same node point in space. To resolve this issue, each node is also equipped with its hierarchical level so that we obtain a one-to-one link between the unique identifier (node) and the basis function. Each basis function in the BFUN_SET is also equipped with a flag indicating whether it is active or not and only the active basis functions compose the linearly independent hierarchical B-spline space.

4.4 Adaptive Refinement/Coarsening and Data Transfer

The versatility of hierarchical B-spline refinement has been demonstrated for a series of fluid and solid mechanics problems in two and three dimensions (Vuong et al., 2011; Schillinger et al., 2012; Bornemann and Cirak, 2013). In these numerical examples, only the refinement process is involved. However, for many applications coarsening is very important, in particular to maintain efficiency. Furthermore, to satisfy the needs of an adaptive procedure, algorithms capable of transferring data between hierarchical structures are required.

4.4.1 Adaptive Refinement/Coarsening Procedure

We first define some notations used in the algorithms:

- \( \mathcal{P}(\phi) \) is the parent set of \( \phi \)
- \( \mathcal{C}(\phi) \) is the child set of \( \phi \)
- \( \mathcal{B} \) is the set of active basis functions
- \( \mathcal{B}^k \) is the basis function set consisting of active \( k \)-level basis functions
1. **BASIC RULES**

As proposed by Krysl et al. (2003, 2004), we simplify our algorithmic design by adhering to three rules:

**Rule 1** The refining/unrefining of a function on level $k$ may activate or deactivate that function or any of its children on level $k + 1$; no other function may be affected.

**Rule 2** A function on level $k + 1$ ($k + 1 > 1$) may be refined only when all of its parents on level $k$ have been refined.

**Rule 3** A function on level $k$ may be coarsened only if (i) it was previously refined and (ii) all of its children on level $k + 1$ are not refined.

Rules 2 and 3 enforce the disjoint condition of a nested domain $\partial \Omega_k \cap \partial \Omega_{k+1} = \emptyset$ (see equation (4.3)). This also implies the common rule of one-level-difference refinement of neighbors in finite element meshes.

2. **REFINEMENT**

Algorithm 3 describes the complete refinement procedure. Kraft’s algorithm is used to ensure linear independence. On the one hand, the parent basis function should be removed from the basis function set if it can be completely represented by its children. The automatic detection of this situation is achieved by counting the number of unrepresented children of the parent basis function (see Algorithm 1). On the other hand, we should always build a nested domain as described in Algorithm 2.

3. **COARSENING**

Algorithm 7 describes the complete coarsening procedure. Although coarsening can be simply considered as the opposite operation of refinement, special care
Algorithm 1 Propagate New Representation ($\phi$)

// When a basis function $\phi$ becomes active, its parents will add one represented child
for all $\phi_p \in \mathcal{P}(\phi)$ do
    $\phi_p$ decreases the number of unrepresented children by one
    if the number of unrepresented children of $\phi_p$ is zero then
        if $\phi_p$ is active then
            call Deactivate ($\phi_p$)
        else
            call Propagate New Representation ($\phi_p$)
        end if
    end if
end for

Algorithm 2 Form Nested Domain (refined set $S_{ref}$, level $k$)

// collect current refined domain
for all $\phi \in S_{ref}$ do
    $\mathcal{D} \cup = \text{supp } \phi$
end for
// collect basis functions covered by the domain $\mathcal{D}$
for all $\phi \in \mathcal{B}^k$ do
    if $\text{supp } \phi \subset \mathcal{D}$ then
        if $\phi$ is not refined then
            call Refine($\phi$)
        end if
    end if
end for

Algorithm 3 Refinement

for $k = 0, 1, \ldots$ do
    $S_{ref} = \emptyset$
    for all $\phi \in \mathcal{B}^k$ do
        if $\text{error}(\phi) > \text{tolerance}$ then
            call Refine($\phi$)
            $S_{ref} \cup = \phi$ // collect the refined basis functions
        end if
    end for
    // ensure refinement forms a nested domain
    call Form Nested Domain($S_{ref}$, $k$)
end for

Algorithm 4 Activate ($\phi$)

Set $\phi$ to be active and add it to the $\mathcal{B}$
call Propagate New Representation ($\phi$)
should be taken to maintain the hierarchical structure. Firstly, to ensure linear independence, we provide an automatic detection mechanism to reactivate a basis function when it has no longer been completely represented by its children (see Algorithm 6). Secondly, by looping over all the basis functions on level $k$, we activate the targeted basis functions which need to be unrefined and collect the children of the remaining basis functions into the $S_{set\_keep}$ set. Thirdly, we loop over all the basis functions on level $k+1$ and deactivate those basis functions which are not in the $S_{set\_keep}$ set. This coarsening procedure maintains the nested domain structure and linear independence.

A simple illustration of our coarsening technique is shown in Figure 4.5 and 4.6. As a comparison, another coarsening approach (Grinspun et al., 2002) is also demonstrated. In Figure 4.5, the original two level refinement of a cubic B-spline function is shown. Two basis functions in dashed lines on level 0 are replaced with their children on level 1. The basis function in red is targeted to be unrefined. In our algorithm (see Figure 4.6(a)), the targeted basis function is activated and only two of its children are deactivated because the rest of the three children still have a refined parent (dashed lines) and they should be kept in the basis function set. Figure 4.6(b) shows another coarsening scheme where all the children of the targeted basis function are deactivated. Although both of these two approaches lead to a linearly independent set of functions, only our approach forms the hierarchical space.
Algorithm 6 Propagate Lost Representation ($\phi$)

// When a basis function $\phi$ becomes inactive, its parents will decrease one represented child
for all $\phi_p \in P(\phi)$ do
  $\phi_p$ increases the number of unrepresented children by one
  if ($\phi_p$ is inactive) && (the number of unrepresented children of $\phi_p$ is one) then
    call Activate ($\phi_p$)
  end if
end for

Algorithm 7 Coarsening

for $k = K - 1, K - 2, \ldots$ do
  $S_{set\to\text{keep}} = \emptyset$
  for all $\phi \in B^k$ do
    if error($\phi$) < tolerance then
      if check rule #3 returns true then
        call Activate($\phi$)
      end if
    end if
    // collect all child functions that would be needed
    if $\phi$ is active then
      for all $\phi_c \in C(\phi)$ do
        $S_{set\to\text{keep}} \cup = \phi_c$
      end for
    end if
  end for
  // deactivate abundant basis function on level $k + 1$
  for all $\phi \in B^{k+1}$ do
    if ($\phi$ is inactive) && ($\phi \notin S_{set\to\text{keep}}$) then
      call Deactivate ($\phi$)
    end if
  end for
end for

For many unsteady problems, a dynamic adaptive method is needed to track the features of the result as the computation progresses. By the processes of refinement and coarsening, an adaptive refinement method should keep the problem size in check. In our hierarchical refinement framework, the increasing number of basis functions by refining a parent basis function is not generally the same, because some of its children might have already existed in the current basis function set. The

Algorithm 8 Deactivate ($\phi$)

Set $\phi$ to be inactive and remove it from the $B$
call Propagate Lost Representation ($\phi$)
coarsening process also has the same issue. In this regard it is worthwhile to check whether our hierarchical B-spline refinement approach can keep the total number of basis functions roughly constant for a simple advection problem. In Figure 4.7, the refinement region (marked in gray) is moving to the right with a fixed length. Suppose that we wish to refine basis functions that are fully within this region. According to this criterion, we have two basis functions refined at the initial step. In the next step, the refinement region translates to a new position, leading to the refinement of a

Figure 4.5: The original refinement example

Figure 4.6: Comparison between two different coarsening schemes: our scheme is shown on the left and the quasi-hierarchical unrefinement scheme (Grinspun et al., 2002) on the right.
new basis function as well as the coarsening of the previously refined basis function. As can be seen, our refinement and coarsening scheme keeps the number of total active basis functions constant (12), and also forms a nested hierarchical structure. Although simple, this example suggests that our approach is capable of efficiently handling the situation where dynamic refinement is needed to capture time-evolving critical areas.

4.4.2 Data Transfer

By the process of refinement and coarsening, we produce a new basis function set and need to transfer the previous field (the source) to the current field (the destination). Due to the precision of the refinement algorithm field transfers from coarse meshes to finer meshes are exact. However, if the destination field is defined on a mesh coarser than the source field, the transfer will generally result in some information being lost. The exception concerns the special case where the source field can be spanned

\[ \text{Figure 4.7: Sequence of dynamic refinement and coarsening.} \]
by the basis functions in the destination mesh.

**Algorithm 9 Propagate Coefficients (φ)**

```plaintext
for all φ_c ∈ C(φ) do
    u_{φ_c} += a_{φ,φ_c} u_{φ_c}
    // u_{φ_c} is the coefficient associated with φ_c
    // a_{φ,φ_c} is the weight of φ_c in the refinement relation of φ (4.6)
    if φ_c is inactive then
        // recursively propagate the parent coefficient to its children
        call Propagate Coefficients (φ_c)
    end if
end for
```

**Algorithm 10 Unrefine Coefficients**

```plaintext
// local least squares fitting
solve min_{φ_i ∈ S, φ ∈ S} ∫_{Ω} \frac{1}{2}(∑_{φ_i ∈ S} \tilde{φ}_i(x) \bar{u}_i - ∑_{φ_i ∈ S} φ_i(x) u_i)^2 dΩ
```

**Algorithm 11 Field Transfer (F_{source}, F_{destination})**

```plaintext
for all φ ∈ B_{destination} do
    u_φ := 0 // initialization
end for
for all φ ∈ B_{source} do
    if φ is refined then
        // refinement data transfer
        call Propagate Coefficient (φ)
    else if φ is unrefined then
        // collect coarsening region
        Ω_{coarsen} ∪ = supp φ
    else
        // this basis function does not change
        u_φ += \tilde{u}_φ
        // u_φ and \tilde{u}_φ are the coefficients associated with φ respectively
    end if
end for
// get the unrefined coefficients
call Unrefine Coefficients
```

The source field, written as

\[ \tilde{u}(x) = \sum_i \tilde{φ}_i(x) \bar{u}_i \] (4.10)
where $\tilde{\phi}_i \in \tilde{\mathcal{B}}$ and $\tilde{u}_i$ are the B-spline functions and dofs (degrees-of-freedom), is to be transferred to the destination field, written as

$$u(x) = \sum_i \phi_i(x) u_i$$  \hspace{1cm} (4.11)

where $\phi_i \in \mathcal{B}$ and $u_i$ are the corresponding B-spline functions and dofs of the destination field. The parameters $\tilde{u}_i$ are all known. Thus we have a classical function approximation problem and for a non-interpolating basis a least squares fitting can be used to solve for the unknown parameters (Bornemann and Cirak, 2013). Standard procedures rely on the use of global least-squares algorithms, but these introduce significant computational expense.

Alternatively, we propose an efficient field transfer algorithm by taking advantage of the hierarchical B-spline properties. Our data transfer algorithm proceeds as follows. For refinement, the data transfer technique follows naturally from the refinement relation (4.6). By multiplying both sides of equation (4.6) by $u_i^k$ and summing over index $i$, we obtain

$$\sum_i u_i^{(k)} \phi_i^{(k)}(x) = \sum_i u_i^k \sum_j a_{ij}^{(k+1)} \phi_j^{(k+1)}(x) \quad \forall x$$

$$= \sum_j \sum_i \left( a_{ij}^{(k+1)} u_i^{(k)} \right) \phi_j^{(k+1)}(x) \quad \forall x$$

$$= \sum_j \left[ \sum_i \left( a_{ij}^{(k+1)} u_i^{(k)} \right) \right] \phi_j^{(k+1)}(x) \quad \forall x \hspace{1cm} (4.12)$$

and the degree of freedom corresponding to the new basis function can be determined (see Algorithm 9).

For coarsening, we limit the least-squares fit to those basis functions whose support fully cover the coarsening region. The coarsening region is defined as the union
of the support of unrefined basis functions

\[ \mathcal{D}_{\text{coar}} = \cup \text{supp } \phi_{\text{unref}} \]  

(4.13)

In practical terms, the value of the destination field at a position outside the coarsening region will remain unchanged. To better illustrate the idea of data transfer, we introduce the concept of a subfield. A subfield is defined as a field consisting of basis functions from a subset of the original basis function set. The source subfield and destination subfield related to coarsening are defined as

\[ \tilde{u}_{\text{sub}} = \sum_{\phi_{i} \in \tilde{S}} \tilde{\phi}_{i}(x) \tilde{u}_{i}, \]  

(4.14)

where \( \tilde{S} = \{ \tilde{\phi}_{i} \mid \text{supp } \tilde{\phi}_{i} \subseteq \mathcal{D}_{\text{coar}} \} \), and

\[ u_{\text{sub}} = \sum_{\phi_{i} \in S} \phi_{i}(x) u_{i}, \]  

(4.15)

where \( S = \{ \phi_{i} \mid \text{supp } \phi_{i} \subseteq \mathcal{D}_{\text{coar}} \} \), respectively. The dofs for basis functions in \( \tilde{S} \) are transferred to the basis functions in \( S \) by a local least squares method (see Algorithm 10), which only requires the solution of a small linear system.

As an additional remark, we note that the data transfer technique can be used to determine the weights for the weighted hierarchical B-spline basis. The initialized B-spline is a partition of unity and so their sum yields a field that is constant, with a value of one. By transferring this constant one field to the field consisting of the new basis functions, the dofs of these basis functions are used as the weights. With these weights, the weighted hierarchical basis satisfies partition of unity.

Finally, we conclude by remarking that since the concept of subdivision can also be applied to NURBS bases (Schillinger et al., 2012), a data transfer scheme for NURBS can be established in a similar way.
4.5 Numerical examples

For the examples in this section, we employ an error indicator that lumps the elemental error to each basis function. In particular, to capture steep gradients in the solution we use a simple gradient based error indicator $\varepsilon_\phi$ associated with basis function $\phi$,

$$\varepsilon_\phi = \frac{\sum_{e=1}^{n} \left( \int_{\Omega_e} |\nabla u|^2 d\Omega_e \right)^{1/2}}{\sum_{e=1}^{n} V_e}$$  \hspace{1cm} (4.16)

where $n$ is the number of elements associated with the basis function and $V_e$ is the element volume. If $\varepsilon_\phi$ is larger than the average error:

$$\varepsilon_\phi > \frac{C_{ref}}{n_{basis}} \sum_{i=1}^{n_{basis}} \varepsilon_{\phi_i}$$  \hspace{1cm} (4.17)

with $n_{basis}$ being the number of all basis functions, the basis function is refined. The additional constant $C_{ref}$ is introduced to empirically fine-tune the threshold for the specific problem. Similarly, if the error is smaller than an empirically scaled value of the average error

$$\varepsilon_\phi < \frac{C_{unref}}{n_{basis}} \sum_{i=1}^{n_{basis}} \varepsilon_{\phi_i}$$  \hspace{1cm} (4.18)

the basis function is tagged for coarsening.

4.5.1 $L^2$ Projection of Phase Field Function

In phase field methods, interfaces can be approximated as the zero level set of a field $\tilde{u}$ given by

$$\tilde{u} = \tanh \left( \frac{d(\mathbf{x})}{\sqrt{2}\epsilon} \right)$$  \hspace{1cm} (4.19)

where $d(\mathbf{x})$ is the signed distance of any point in the domain to the interface and $\sqrt{2}\epsilon$ represents the width of the interface. Since the B-spline functions are not
interpolants, we use an $L^2$ projection to obtain the coefficient of each basis function. This can be written as

$$
\min_{\tilde{u}} \int_{\Omega} \frac{1}{2} (u^h - \tilde{u})^2 \, d\Omega \quad (4.20)
$$

where $u^h = \sum_i \phi_i(x) u_i$ is the finite element approximation and $\phi_i$ is the B-spline basis function.

We solve this problem in $\Omega = [0, 1]^2$ and define the zero level set of $\tilde{u}$ as a circle with radius $r = 0.3$. To resolve the narrow interface, Sun and Beckermann (2007) suggests the grid spacing $h$ be smaller than the width of the interface $\epsilon$. To test our refinement scheme, we choose a small $\epsilon = 0.005$ which requires very fine grids in the vicinity of the interface.

We begin with a $64 \times 64$ grid of cubic B-splines and subsequently refine up to level $k = 3$ with smallest mesh size $h = 0.002$. In each refinement step, we generate an additional overlay level by adaptively refining the interface region, see Figure 4.8. The approximation $u^h$ corresponding to $k = 3$ is shown in Figure 4.10. Figure 4.9 shows the convergence in the $L^2$ norm, obtained with uniform h-refinement and adaptive hierarchical refinement. It is worth mentioning that in this example the optimal rate of convergence is 2 with respect to the number of degrees of freedom. With uniform refinement, we observe an optimal rate of convergence. With adaptivity, by contrast, we observe a higher rate of convergence. To obtain an error of approximately $10^{-6}$, the hierarchical basis requires fewer than 10% of the total number of degrees of freedom for a uniform grid.
Figure 4.8: The $L^2$ projection example: adaptive hierarchical mesh.

(a) Mesh of level $k = 0$, $n_{dofs} = 4,489$.

(b) Mesh of level $k = 1$, $n_{dofs} = 7,267$.

(c) Mesh of level $k = 2$, $n_{dofs} = 13,369$.

(d) Mesh of level $k = 3$, $n_{dofs} = 28,597$. 
4.5.2 Poisson Equation

We consider a Poisson equation on a unit square with Dirichlet boundary conditions on all four edges. The governing equation and boundary conditions are

\[
\Delta u = 4 \\
u(x, 0) = x^2 \\
u(x, 1) = x^2 + 1 \\
u(y, 0) = y^2 \\
u(y, 1) = y^2 + 1
\]

for which the analytical solution is

\[
u(x, y) = x^2 + y^2
\]

As the solution is a quadratic field, it can be exactly represented using cubic splines. As such, this example can be viewed as a kind of ”patch test”. In this case, we use this problem to verify our data transfer scheme as follows. By the process
of refinement, we expect that the $L^2$ norm of the error should not change. Since coarsening is precise when the approximated solution is contained in the unrefined trial space, it is likewise expected that coarsening in this problem will not change the $L^2$ norm of the error.

The sequence of adaptive hierarchical meshes involving both refinement and coarsening can be seen in Figure 4.11. Specifically, the mesh is refined from (a) to (c) and subsequently unrefined from (c) to (e). The field is only calculated once for the initial mesh and then the field is transferred between different meshes. From (d) to (c), we coarsen only some of the basis functions on level $k = 1$ for the purpose of testing the local least squares fitting scheme. The $L^2$ norm of the error for each step can be found in Table 4.1 and remains in the order of $10^{-14}$. The slight difference that occurs during coarsening is likely due to the truncation error introduced in solving the local least squares system.

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
<th>(f)</th>
</tr>
</thead>
<tbody>
<tr>
<td>#dofs</td>
<td>361</td>
<td>853</td>
<td>2,296</td>
<td>1,666</td>
<td>853</td>
<td>361</td>
</tr>
<tr>
<td>#eqns solved in data transfer</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>727</td>
<td>175</td>
<td>180</td>
</tr>
</tbody>
</table>

### 4.5.3 Advection Problem

We consider a pure advection equation to test our dynamic adaptive algorithm on an unsteady problem, given as

\[
\frac{\partial u(x, y, t)}{\partial t} + a \frac{\partial u(x, y, t)}{\partial x} = 0 \quad \text{on } \Omega \quad [0, 1]^2 \quad \text{(4.23)}
\]

where $a = 0.1$ is the velocity in the $x$ direction. As shown in Figure 4.12, the initial condition is given as $u(x, y, 0) = 0.1 \times \{1 - \cos[20\pi(x - 0.2)]\}$ on $[0.2, 0.3] \times [0, 1]$ and zero elsewhere and the inflow boundary condition is given as $u(0, y, t) = 0$. We
Figure 4.11: The Poisson equation example: adaptive hierarchical mesh and field.

apply standard SUPG stabilization and choose \( \tau = 1.0/\sqrt{(2.0/dt)^2 + (2a/h)^2} \) where \( dt = 0.0125 \) is the time step size and \( h \) is the element size.

Figure 4.12: The advection example: cosine hill and adaptive mesh at \( t = 0 \).
The initial mesh has $20 \times 20$ elements and the maximum allowable hierarchical level is 3, as shown in Figure 4.12. The adaptive hierarchical mesh is automatically generated to capture the moving cosine hill by the refinement and coarsening scheme based on the gradient error indicator. We compare the finite element approximations along the line $y = 0.5$ between a uniform mesh and an adaptive mesh with the same smallest element size. For the adaptive case, we report results using data transfer schemes employing both global and local least squares fitting. As can be seen in Figure 4.13, these results agree very well.

All calculations for this example were carried out on a workstation with Xeon 2.93 GHz CPU and 24GB of memory. Both a direct and a Jacobi preconditioned iterative solver GMRES were employed for the linear algebraic system of equations. The accumulated CPU time for 400 time steps consumed in the solving stage (includes assembling matrix and solving linear system) and the refinement/coarsening stage (includes activating/deactivating basis functions and data transfer) is shown in Figure 4.14. We note that the solution time is greatly reduced by locally refining the mesh. Compared to the global least square fitting approach, the use of a local scheme results in significant computational savings in the refinement/coarsening stage. Although in this 2D example the computation time does not vary significantly between the direct and iterative solvers, for large problems (particularly in parallel) iterative solvers are desirable. In Figure 4.15, it is interesting to find that by using the adaptive refinement method the number of basis functions is almost kept constant. This is consistent with our analysis in the previous section.

4.5.4 Cahn-Hilliard Equation

To show the robustness of the algorithm, we present simulations of spinodal decomposition in a 2D domain using the adaptive refinement method. The governing equation is given in Chapter 2 with constant $\gamma$ taken to be $0.01^2$.  

86
In this simulation, we consider the evolution of an initially dispersed phase-separated state. As the system coarsens to reduce the total surface energy, the number of second-phase regions decreases. As seen in Figure 4.16, the results of the uniform mesh and the adaptive mesh are nearly identical. Moreover, the adaptive mesh adapts around the interfaces; away from interfaces the gradient of $u$ is small.
Figure 4.14: CPU time consumed in solving stage and refinement/coarsening stage for the simulation shown in Figure 4.13.

Figure 4.15: Number of degrees of freedom for the adaptive mesh using local least square fitting data transfer scheme.

Figure 4.17 shows that the number of degrees of freedom rapidly decreases during the early stages of the problem and then remains relatively stable. Although the solution to the Cahn-Hilliard equation conserves mass, mesh coarsening introduces some mass loss. The evolution of the total mass is shown in Figure 4.18 and the
maximum percentage error in the mass is below 0.5%. The results once again support the contention that using an adaptive mesh algorithm results in accurate results and significant computational savings compared to uniform mesh approaches.

4.5.5 Anisotropic Cahn-Hilliard Equation

In the last example, we solve the anisotropic Cahn-Hilliard equation using the adaptive refinement method. In the phase-field model, the free energy functional is given by

$$\mathcal{F} = \int_\Omega \left( f(u) + \frac{\epsilon^2}{2} |\nabla u|^2 \right) \, d\Omega$$  \hspace{1cm} (4.24)

Requiring the free energy functional to decease monotonically the conservation
of mass results in the Cahn-Hilliard equation,

\[ \frac{\partial u}{\partial t} = M \Delta \mu \] \hspace{1cm} (4.25a)

\[ \mu = f'(u) - \nabla \cdot (\epsilon(n)^2 \nabla u) - \partial_x (|\nabla u|^2 \epsilon(n) \partial_{u_x} \epsilon(n)) \] \hspace{1cm} (4.25b)

\[ - \partial_y (|\nabla u|^2 \epsilon(n) \partial_{u_y} \epsilon(n)) - \partial_z (|\nabla u|^2 \epsilon(n) \partial_{u_z} \epsilon(n)) \]

subject to zero mass flux boundary conditions,

\[ \nabla u \cdot n = 0 \quad \text{and} \quad \nabla \mu \cdot n = 0 \quad \text{on} \: \Gamma \times (0, T) \] \hspace{1cm} (4.26)

and initial conditions,

\[ u(x, 0) = u_0(x) \quad \text{in} \: \Omega \] \hspace{1cm} (4.27)
Figure 4.17: Number of degrees of freedom for the simulation shown in Figure 4.16.

Figure 4.18: Total mass for the simulation shown in Figure 4.16 (the maximum error is 0.490%).

M is the mobility and it is assumed constant for simplicity. \( \hat{\partial}_p \) is partial derivative with respect to the variable \((\ast)\). The anisotropic interface energy is included in this model by writing \( \epsilon \) as a function of the local unit normal vector \( \mathbf{n} = \nabla u/|\nabla u| \) in the interfacial region. Thus,

\[
\epsilon(\mathbf{n}) = \epsilon_0(1 - 3\epsilon_1) \left( 1 + \frac{4\epsilon_1}{1 - 3\epsilon_1} \frac{u_x^4 + u_y^4 + u_z^4}{|\nabla u|^4} \right)
\]

(4.28)
The constant parameter $\epsilon_0$ and $\epsilon_4$ are related to the strength and the magnitude of the anisotropy in the interface energy, respectively. The subscripts denote partial differentiation with respect to $x$, $y$ and $z$.

For this example, we consider a free energy function of the form

$$f(u) = \frac{1}{4}(1-u^2)^2 \quad (4.29)$$

The two global minima is at $u = -1$ and $u = 1$.

When the interfacial energy anisotropy is sufficiently strong, there are missing orientations in the equilibrium level curves of the diffuse interface solutions and the anisotropic Cahn-Hilliard equation becomes ill-posed. To overcome this difficulty, a higher-order derivative term is added to the energy functional. The regularized free energy functional is given by

$$\mathcal{F} = \int_{\Omega} \left( f(u) + \frac{\epsilon^2}{2} |\nabla u|^2 + \frac{\delta^2}{2} (\Delta u)^2 \right) d\Omega \quad (4.30)$$

where $\delta$ is the corner energy regularization parameter and taken to be $1.0 \times 10^{-3}$ in the present study. The regularized anisotropic Cahn-Hilliard equation becomes

$$\frac{\partial u}{\partial \ell} = M \Delta \mu \quad (4.31a)$$

$$\mu = f'(u) - \nabla \cdot (\epsilon(n)^2 \nabla u) - \partial_x \left(|\nabla u|^2 \epsilon(n) \epsilon_x \right) - \partial_y \left(|\nabla u|^2 \epsilon(n) \epsilon_y \epsilon(n) \right) - \partial_z \left(|\nabla u|^2 \epsilon(n) \epsilon_z \right) + \delta^2 (\Delta^2 u) \quad (4.31b)$$

The Nitsche weak form can be derived from the governing equation (4.31), accounting for zero mass flux boundary conditions. The governing equation is a sixth order PDE and the cubic spline basis functions which are globally $C^2$ continuous are used for discretization. The semi-implicit scheme is employed for time integration and the
Cahn-Hilliard equation is then rewritten as

\[
\frac{\partial u}{\partial t} = \left( \frac{u^{n+1} - u^n}{\Delta t} \right) = M \Delta \left( f'(u)|^{n+1} - \nabla \cdot (\epsilon_0 \nabla u) |^{n+1} \right) \\
- \nabla \cdot \left( (\epsilon(n) \nabla u) \right)^n - \partial_x \left( |\nabla u|^2 \epsilon(n) \partial_u n \right) |^n \\
- \partial_x \left( |\nabla u|^2 \epsilon(n) \partial_u n \right) |^n - \partial_x \left( |\nabla u|^2 \epsilon(n) \partial_u n \right) |^n + \delta^2 (\Delta^2 u) |^{n+1} \right)
\]

(4.32)

We first consider the evolution of shapes toward equilibrium with two different anisotropic coefficients. The evolution shapes were computed using the circular shape as the initial condition,

\[
u(x, y) = \tanh \left( \frac{0.25 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2}}{\sqrt{2} \epsilon_0} \right)
\]

(4.33)

where \(\epsilon_0 = 0.01\).

The anisotropic coefficient \(\epsilon_4\) is taken to be 0.05 and 0.2 respectively. The analytical solution shows that the missing orientations occur for \(\epsilon_4 > \frac{1}{15}\). Thus for the strong anisotropy case \(\epsilon_4 = 0.2\), the regularization is required for well-posedness. Figure 4.19 shows the initial shape and equilibrium shape for \(\epsilon_4 = 0.05\) and \(\epsilon_4 = 0.2\) respectively.

We present snapshots of the evolving \(u = 0.0\) level sets for case \(\epsilon_4 = 0.2\) in Figure 4.20. It is clear that three hills and two valleys form at the compass points of the circle. A single hill structure merges at long times. A snapshot of the adaptive mesh at time \(t = 0.0694\) is shown in Figure 4.21. The mesh has three levels of refinement with an initial mesh size of 100 \(\times\) 100.

To illustrate the robustness of the proposed method, we examine the evolution of a random distribution of concentration. We take the same anisotropic coefficients in the previous example and the initial distribution is random with a mean value \(\bar{u} = -0.3\). We present snapshots of the evolving \(u = 0.0\) level sets for case \(\epsilon_4 = 0.05\) in Figure 4.22 and for case \(\epsilon_4 = 0.2\) in Figure 4.23 respectively.
Figure 4.19: Surface of zero contour of (a) Initial shape. (b) equilibrium shape of case $\epsilon_4 = 0.05$. (c) equilibrium shape of case $\epsilon_4 = 0.2$.

4.6 Summary

In this Chapter, we introduced an adaptive refinement approach using hierarchical spline-based finite elements. Compared to other local refinement approaches for B-splines, our approach guarantees linear independence and easily extends to arbitrary dimensions, while maintaining the regularity of the basis functions. We outlined the refinement and coarsening algorithms and described an efficient data transfer technique. Since our refinement algorithm is precise, we transfer the field exactly. For coarsening, we perform a local least square fitting which turns out to be fast and accurate. Although we placed emphasis on B-spline basis functions in this work, the approach readily extends to NURBS and refinable splines in general.
Figure 4.20: Evolution of the $u = 0.0$ level curves by the strongly anisotropic Cahn-Hilliard equation with higher-order regularization at $t = 0, 0.0078, 0.0694$ and $1.1254$. The simulation was performed with a three-level refinement.

Figure 4.21: Adaptive hierarchical mesh at time $t = 0.0694$. The contours $u = 0.0, -0.3$ and $0.3$ are shown.
(a) $t=0.0346$

(b) $t=0.1534$
Figure 4.22: The anisotropic Cahn-Hilliard equation example for $\epsilon_4 = 0.05$: evolution of the field and the mesh with the $u(x, t) = 0.0$ iso-contours under coarsening. The left column shows the field and the right column corresponds to the adaptive mesh. There are three levels of refinement and the initial grid resolution is $100 \times 100$. 

(c) $t=1.2622$

(d) steady state
(a) $t=0.0346$

(b) $t=0.1534$
Figure 4.23: The anisotropic Cahn-Hilliard equation example for $\epsilon_4 = 0.2$: evolution of the field and the mesh with the $u(x,t) = 0.0$ iso-contours under coarsening. The left column shows the field and the right column corresponds to the adaptive mesh. There are three levels of refinement and the initial grid resolution is $100 \times 100$. 

(c) $t=1.2622$

(d) steady state
Modeling the Evolution of Microdomains on Giant Unilamellar Vesicles with a Phase-Field Approach

5.1 Introduction

We employ a continuum approach to describe the coupled process of shape change and species transport. The continuum approach is based on the generalized Helfrich curvature energy (Helfrich, 1973) supplemented by a line tension energy (Jülicher and Lipowsky, 1993, 1996). In this work, the line tension energy is characterized by a Cahn-Hilliard-type energy. Following our previous work (Embar et al., 2012), the shape change and phase separation are characterized by distinctly different time scales. This is largely motivated by the experimental observation that the time scale for shape change is much shorter than that associated with domain growth (Yanagisawa et al., 2007).

In modeling vesicles (Biben et al., 2005; Campelo and Hernández-Machado, 2006; Lowengrub et al., 2009; Q and J, 2007; Embar et al., 2012), the phase-field method provides an implicit representation of the vesicle surface. Employing the phase-field approach, the governing equations are formulated in diffuse-interface form leading to
nonlinear fourth-order coupled PDEs. An adaptive finite element method based on a mixed-form of the fourth-order system was used to simulate a phase-field model of vesicles (Du and Zhang, 2008; Lowengrub et al., 2009). In this work, we approximate the solution to the fourth-order equations using spline-based finite elements (Höllig, 2003; Embar et al., 2010a) which offer smooth B-spline basis functions to satisfy the requirement $H^2$ regularity. The hierarchical refinement approach is further adopted to model the vesicles adaptively.

This Chapter is organized as follows. Section 5.2 provides a brief description of the phase-field framework. Energetic contributions including mechanics and mass transport are described in Section 5.4. Diffuse-interface forms of the governing equations for shape equilibrium and species transport are provided in Section 5.5. The details of the numerical modeling using the adaptive refinement of hierarchical spline-based finite elements is given in Section 5.6. Numerical examples are discussed in Section 5.7, followed by a summary.

5.2 Phase-field framework

In the present work, a phase-field approach is used to cast all governing equations in a diffuse-interface form. On employing a phase field $\phi$, the vesicle mid-surface $S$ is implicitly represented by a three-dimensional layer of controllable width $\epsilon_\phi$. This varies smoothly yet steeply across the layer. The phase-field approach not only eliminates the need to explicitly track sharp surfaces, but also naturally allows for topological changes. As depicted in Figure 5.1, $\phi$ varies between $\phi = -1$ (vesicle interior) and $\phi = 1$ (vesicle exterior) and its zero level-set ($\phi = 0$) represents $S$. Moreover within the transition layer of any implicitly defined surface,

$$n = \frac{\nabla \phi}{|\nabla \phi|} \quad (5.1)$$

represents the unit normal (oriented in the direction of increasing $\phi$).
Figure 5.1: A smooth surface $S$ is represented as the zero level set of the phase field $\phi$.

5.3 Approximate delta function

Figure 5.2: (a) Profile of an approximate delta function in the direction of the surface normal $\mathbf{n}$; (b) Approximate support of $\delta_s$ encompassing the surface

To obtain a diffuse-interface form, we represent the surface integral of a function
$f$ defined on $S$ as an equivalent volume integral over $\Omega$ via (Smereka, 2006)

$$
\int_{S} f_s da = \int_{\tilde{\Omega}} \tilde{f} \delta_S dv, \quad (5.2)
$$

where $\tilde{f}$ is defined in a neighborhood of $S$ and the restriction of $\tilde{f}$ to $S$ coincides with $f_s$. Here, $\delta_S$ is an approximate delta function with the property

$$
\int_{-\infty}^{\infty} \delta_S dr = 1, \quad (5.3)
$$

where $r$ represents the distance along the surface normal $n$. Figure 5.2(a) illustrates the profile of an approximate delta function with the horizontal axis representing the signed distance to the surface $S$. Essentially, $\delta_S$ restricts any function defined over $\Omega$ to a neighborhood of $S$. In the present work, the expression for this approximate delta function is given by

$$
\delta_S \approx \frac{3\sqrt{2}}{4} \left( \frac{(1 - \phi^2)^2}{\epsilon_{\phi}} + \frac{\epsilon_{\phi}}{2} |\nabla \phi|^2 \right). \quad (5.4)
$$

5.4 Energetics

Experimental evidence (Baumgart et al., 2005b) suggests that phase separation is strongly coupled to the shape change of the vesicle. This effect of coupling is seen through a mechanical energy that drives curvature-dependent domain sorting and a line tension that favors the formation of buds. The total energy of the vesicle is taken to be a sum of chemical energy, mechanical energy and geometrical constraint penalty energy, all of which are consistently non-dimensionlized.

$$
E = E_c + E_m + E_{gc} \quad (5.5)
$$
5.4.1 Chemical energy

The present work focuses on the evolution of microdomains in GUVs composed of cholesterol and two lipid species. It is assumed that both lipid species dimerize with cholesterol, mainly resulting in a binary mixture of liquid-ordered and liquid-disordered phases.

The GUV is represented by an oriented surface $S$ bounding a region $R$. The dimensionless purely chemical free-energy of the GUV is taken to be of Cahn–Hilliard type (Cahn and Hilliard, 1958)

$$E_c = \int_S \left( f(c_S) + \frac{\epsilon_c^2}{2} |\nabla_S c_S|^2 \right) da, \quad (5.6)$$

where $c_S$ represents the concentration, $f$ is a double-well potential, $\epsilon_c$ can be considered as the dimensionless thickness of the interface separating the phases and $\nabla_S$ is the gradient operator on $S$. The Euclidean counterparts of the surface gradient is given as (Greer et al., 2006)

$$\nabla_S() = (I - n \otimes n)\nabla() \quad (5.7)$$

where $n$ represents the unit normal to the level surfaces of $\phi$ and is given by (5.1). The operator $(I - n \otimes n)$ projects the Euclidean gradient $\nabla()$ onto the level surfaces of the phase field $\phi$.

To obtain a diffuse-interface form of the functional (5.8), we represent the surface integral as an equivalent volume integral thanks to an approximate delta function $\delta_S$. The dimensionless diffuse-interface form of the chemical energy can then be written as

$$E_c \approx \int_\Omega \left( f(c) + \frac{\epsilon_c^2}{2} |(I - n \otimes n)\nabla c|^2 \right) \delta_S dv, \quad (5.8)$$

where $c$ is defined in a neighborhood of $S$ by extending $c_S$ off the surface.
In the present work, the double-well potential based on the Flory-Huggins theory of binary polymer mixtures (Flory, 1942; Huggins, 1942) is given by

\[ f(c) = c \ln c + (1 - c) \ln(1 - c) + \chi c(1 - c), \quad (5.9) \]

where \( \chi > 2 \) is the interaction parameter.

5.4.2 Mechanical energy

The dimensionless mechanical free-energy of the GUV is taken to be characterized by the generalized Canham–Helfrich–Evans curvature energy (Canham, 1970; Helfrich, 1973; Evans, 1974) given by

\[ E_m = \frac{\epsilon_c^2}{2} \int_S \kappa(c_S) (H - H_{sp}(c_S))^2 \, da, \quad (5.10) \]

where \( \kappa \) and \( H_{sp} \) are the concentration-dependent flexural rigidity and spontaneous curvature. For simplicity, Gaussian curvature energy is not considered in this study. Notice that the curvature energy (5.10) is scaled by the factor \( \epsilon_c^2 \) that arises as a consequence of rendering the energy per unit area dimensionless.

The mean curvature can be expressed in terms of the phase field \( \phi \) as

\[ H_\phi = \frac{1}{2} \nabla \cdot \left( \frac{\nabla \phi}{|\nabla \phi|} \right) = \frac{1}{2|\nabla \phi|} \left( \Delta \phi - \frac{1}{2|\nabla \phi|^2} (\nabla \phi \cdot \nabla |\nabla \phi|^2) \right). \quad (5.11) \]

Proceeding as in the derivation of (5.8) the dimensionless diffuse-interface form of the curvature energy (5.10) is obtained as

\[ E_m \approx \frac{\epsilon_c^2}{2} \int_\Omega \kappa(c) (H_\phi - H_{sp}(c))^2 \delta_S \, dv, \quad (5.12) \]

where \( c \) is now defined in a neighborhood of \( S \).
5.4.3 Geometric constraints

The lipid bi-layer (Evans and Skalak, 1980; Seifert, 1997b) is characterized by local lateral incompressibility. In the present study, a global area conservation condition is used as a constraint to model shape equilibrium of isolated vesicles. In addition, due to the fact that the bending resistance of vesicles is too weak to sustain osmotic pressure, the volume enclosed by a vesicle can be considered to be fixed. Here the constraints of surface area and volume conservation are implemented using a penalty method.

The global geometric constraints of fixed enclosed volume and surface area can be represented in equivalent diffuse-interface forms. The dimensionless versions of the enclosed volume and surface area are given by

\[
V_{in} \approx \frac{1}{2} \int_{\Omega} (1 - \phi) \, dv. \tag{5.13}
\]

and

\[
A \approx \int_{\Omega} \frac{3 \sqrt{2}}{4} \left( \frac{(1 - \phi^2)^2}{4 \epsilon_\phi} + \frac{\epsilon_\phi}{2} |\nabla \phi|^2 \right) \, dv. \tag{5.14}
\]

respectively.

Denoting the initial enclosed volume and surface area by \( V_0 \) and \( A_0 \) respectively, the penalty energy \( E_{gc} \) arising from the constraints on internal volume and surface area is given by

\[
E_{gc} = \frac{\alpha_v}{2} \left( \frac{V_{in}}{V_0} - 1 \right)^2 + \frac{\alpha_a}{2} \left( \frac{A}{A_0} - 1 \right)^2, \tag{5.15}
\]

where \( \alpha_v \) and \( \alpha_a \) are penalty parameters associated, respectively, with the constraints on volume and surface area.
5.5 Governing equations

The governing equations in the diffuse-interface form are presented here. The equation governing species transport on the vesicle resembles the Cahn–Hilliard equation for binary mixtures (Cahn and Hilliard, 1958). Shape equilibrium is reached by requiring the first variation of the functional $E$ to vanish.

5.5.1 Species transport

To describe the process of phase separation on the GUV surface, a Cahn-Hilliard type equation is derived for a binary mixture of lipid species. The chemical potential is represented by the first variation of both mechanical and chemical energy with respect to the concentration $c_s$. The dimensionless equation of the species conservation on $S$, and the corresponding chemical potential $\mu_s$ are

$$\frac{\partial c_s}{\partial t} = \nabla_s \cdot (M(c_s) \nabla_s \mu_s),$$  \hspace{1cm} (5.16a)

$$\mu_s = \frac{\delta E_c}{\delta c_s} + \frac{\delta E_m}{\delta c_s},$$  \hspace{1cm} (5.16b)

where $M(c_s)$ is the concentration dependent species mobility. In the present work, the mobility $M$ is assumed constant for the sake of simplicity.

In this chemical potential, whereas the first term $\delta E_c/\delta c_s$ contributes to phase separation (i.e., spinodal decomposition and coarsening), the second term $\delta E_m/\delta c_s$ leads to curvature-dependent domain sorting.

The dimensionless diffuse-interface form of the species transport equation can be approximated as

$$\frac{\partial c}{\partial t} = \text{div}(M(1 - n \otimes n) \nabla \mu),$$  \hspace{1cm} (5.17a)

$$\mu = \epsilon_c^2 \left( \frac{1}{2} \kappa'(c)(H_\phi - H_{sp}(c))^2 - \kappa(c)(H_\phi - H_{sp}(c))H'_{sp}(c) \right)$$

$$+ f'(c) - \epsilon_c^2 \text{div}((1 - n \otimes n) \nabla c).$$  \hspace{1cm} (5.17b)
where all quantities of interest are defined in a neighborhood of $S$. To prevent the concentration from diffusing between the level sets, it is desirable to ensure that the concentration is constant normal to $S$. In the present work, the zero flux constraint is weakly imposed using the penalty method,

$$E_p = \alpha_m \int_\Omega (\nabla c \cdot \mathbf{n})^2 \delta_s \, dv,$$

where $\alpha_m$ is the penalty parameter.

5.5.2 Shape equilibrium

The dimensionless equation of shape equilibrium is obtained by requiring the first variation of the functional $E$ to vanish.

$$\frac{\delta E}{\delta \phi} = \frac{\delta E_c}{\delta \phi} + \frac{\delta E_m}{\delta \phi} + \frac{\delta E_{gc}}{\delta \phi} = 0 \quad (5.19)$$

The variations $\delta E_c/\delta \phi$ and $\delta E_m/\delta \phi$ govern shape changes in response to energetic penalties associated with line tension and curvature, respectively.

On employing (5.4) for the approximate delta function $\delta_s$, each of the terms in the shape equation (5.19) can be evaluated as

$$\frac{\delta E_c}{\delta \phi} = \frac{3\sqrt{2}}{4} \left( \frac{1}{\epsilon} \psi(c) f'(\phi) - \epsilon \left( \psi(c) \Delta \phi + \nabla \phi \cdot \nabla \psi(c) \right) \right), \quad (5.20a)$$

$$\frac{\delta E_m}{\delta \phi} = \frac{3\sqrt{2}\epsilon^2}{16} \left( \Delta (\kappa(c) F(\phi)) - \frac{1}{\epsilon^2} \kappa(c) F(\phi) f''(\phi) \right)$$

$$+ \frac{2\sqrt{2}}{\epsilon} \kappa(c) H_{sp}(c) F(\phi) \phi, \quad (5.20b)$$

$$\frac{\delta E_{gc}}{\delta \phi} = - \frac{\alpha_v}{2V_0} (V_R - 1) + \frac{3\sqrt{2}}{4\epsilon \alpha} \kappa(A_R - 1) \left( f'(\phi) - \epsilon^2 \Delta \phi \right), \quad (5.20c)$$

108
wherein

\[ f(\phi) = \frac{1}{4}(1 - \phi^2)^2, \]

\[ F(\phi) = \epsilon_\phi \Delta \phi - \frac{1}{\epsilon_\phi} f'(\phi) - \sqrt{2} (1 - \phi^2) H_{sp}(c), \]

\[ \psi(c) = f_{f-H}(c) + \frac{\epsilon_c^2}{2} |(I - n \otimes n) \nabla c|^2. \]

In the present work, shape equilibrium is achieved by relaxing (5.19) to

\[ \frac{\partial \phi}{\partial \tau} = - \frac{\delta E}{\delta \phi} \quad (5.21) \]

and approximating the solution to the steady state of (5.21), i.e. \( \partial \phi/\partial \tau \approx 0 \). In (5.21), \( \tau \) represents pseudo (i.e., non-physical) time.

### 5.6 Numerical modeling

#### 5.6.1 Spatial discretization

The diffuse-interface forms of the shape change and species transport equations are nonlinear fourth-order partial-differential equations and are discretized using the spline-based finite element method. The computational effort of this model is always very intensive on a uniform mesh. The adaptive hierarchical refinement approach allows the simulation to proceed efficiently. In the present work, 2D examples have 3 levels of refinement whereas 3D examples have 2 levels of refinement, see Figure 5.3 for a typical example of adaptively refined 2D and 3D meshes, respectively. The grid size is chosen to be the smaller of \( \epsilon_\phi \) and \( \epsilon_c \). Such a level of discretization is found to provide a sufficiently accurate representation of the phase field \( \phi \) and the approximate delta function \( \delta_S \).
Figure 5.3: (a) 2D adaptively refined mesh. The contour $\phi = 0$ is shown in blue. (b) 3D adaptively refined mesh. Only one octant of the domain is shown and the contour $\phi = 0$ is shown in blue.

5.6.2 Time discretization

To avoid the computation of directional derivatives of the nonlocal terms ($V_R$ and $A_R$ in (5.20c)), a semi-implicit time stepping scheme is used. While the concave terms are treated explicitly, convex terms are treated implicitly. The resulting computational framework ensures that the functional $E$ does not increase along solution paths (Eyre, 1997). The semi-implicit scheme for the gradient-flow augmentation (5.21) of the shape equation is then rewritten as

$$\frac{\partial \phi}{\partial \tau} = \left( \frac{\phi^{n+1} - \phi^n}{\Delta \tau} \right) = -(1 + \gamma_1) \left. \frac{\delta E_m}{\delta \phi} \right|_{n}^{n+1} - \left. \frac{\delta E_{cv}^{\text{cvx}}}{\delta \phi} \right|^{n+1}$$

$$- \left( \frac{\delta E_{gc}}{\delta \phi} - \gamma_1 \frac{\delta E_m}{\delta \phi} \right) \left|^{n} - \left. \frac{\delta E_{cv}^{\text{cvx}}}{\delta \phi} \right|^{n} ,$$

(5.22)
with

\[
E_{cv}^{\Omega} \approx \int_{\Omega} \psi(c) \left( f(\phi) - \gamma_2 \phi^2 + \left( \frac{\epsilon_1^2}{2} - \gamma_2 \right) |\nabla \phi|^2 \right) \, dv, 
\]

\[
E_{cvx} \approx \gamma_2 \int_{\Omega} \psi(c) \left( \phi^2 + |\nabla \phi|^2 \right) \, dv,
\]

in which \( \psi \) and \( f \) are defined immediately after (5.20). The scaling factors \( \gamma_1 \) and \( \gamma_2 \) are taken to be 15 and 2 to ensure concavity of \( E_{cv}^{\Omega} \).

5.6.3 Computational scheme

The model is intended to study a regime where phase separation occurs on a time scale significantly longer than that associated with shape changes. While a natural time scale can be associated with species transport, the vesicle is assumed to instantaneously reach shape equilibrium at all times during the evolution process. This phenomenon is numerically simulated using a scheme in which the vesicle is driven to shape equilibrium at each time step involving coupled species transport and shape evolution. The initial spatial distribution of the concentration \( c \) of lipid species is prescribed on the whole computational domain such that \( c(S, t = 0) = c_0^S \). The computational framework for the coupled process is outlined below:

**Main solve (Coupling process)**

Do while \( \partial c / \partial t > 0 \):

1. Time march one step by solving equations (5.17) for \( c^{n+1} \) implicitly with Newton-Raphson.

2. Using a semi-implicit scheme, drive the system to shape equilibrium at \( t^{n+1} \) by solving equation (5.21) to steady state. During this process, the concentration must be coupled to the deforming level sets of \( \phi \) using data extension.
Numerically, steady state is considered to be achieved when

\[ \left( \int_{\Omega} \left( \frac{\partial \phi}{\partial \tau} \right)^2 \, dv \right)^{\frac{1}{2}} < \text{tol}_\phi, \quad \text{for shape equilibrium,} \quad (5.24a) \]

\[ \left( \int_{\Omega} \left( \frac{\partial c}{\partial t} \right)^2 \, dv \right)^{\frac{1}{2}} < \text{tol}_c, \quad \text{for species transport,} \quad (5.24b) \]

For the numerical examples considered in the present work, values of 0.1 for \( \text{tol}_\phi \) and 0.01 for \( \text{tol}_c \) were found to provide an adequate representation of steady state.

5.7 Numerical results

In the present work, the effect of the material parameters on the coupled phenomenon of species transport and shape equilibrium is studied qualitatively. Accordingly, \( \kappa \) and \( H_{sp} \) of each phase are chosen relative to their values in the other phase. The dimensionless interface thickness \( \epsilon_c \) is chosen to be small relative to the vesicle dimensions.

Numerical examples simulating the coupled model use concentration dependent dimensionless flexural rigidity \( \kappa \) and spontaneous curvature \( H_{sp} \). To exaggerate the difference in the flexural rigidities and spontaneous curvatures of the two phases, an exponential variation with concentration is assumed:

\[ \kappa(c) = \exp(a_\kappa c + b_\kappa), \quad (5.25a) \]

\[ H_{sp}(c) = \exp(a_H c + b_H). \quad (5.25b) \]

The parameters \((a_\kappa, b_\kappa)\) and \((a_H, b_H)\) are estimated based on the values of \( \kappa \) and \( H_{sp} \) in the phases corresponding to \( c_a \) and \( c_b \). A small value of order \( 1.0 \times 10^{-16} \) is used to approximate \( H_{sp} = 0 \).

Unless mentioned otherwise, all examples employ a Flory–Huggins type double-well potential with the interaction parameter \( \chi = 2.25 \) and, thus, with minima
$c_a \approx 0.2244$ and $c_b \approx 0.7756$ distinguishing the phases. The potential used in the present study is shown in Figure 5.4.

![Flory-Huggins type double well potential](image)

**Figure 5.4**: Flory–Huggins type double well potential

Unless mentioned otherwise, all examples use $\epsilon_\phi = 0.02$ and $\epsilon_c = 0.02$ while the vesicle dimensions are of $O(1)$. The following values for penalty parameters are found to enforce the geometric constraints quite accurately: $\alpha_v = 10^4$ and $\alpha_a = 10^3$. The zero-flux constraint (5.18) is imposed with $\alpha_m = 10^4$.

While solving the gradient-flow augmentation (5.21) of the shape equilibrium equation, pseudo-time increments are taken to be in the interval $[10^{-7}, 10^{-3}]$ for two- and three-dimensional numerical examples with multiple phases. Data extension is performed with a uniform pseudo-time step of $10^{-2}$. Dimensionless time increments for species transport are taken to be in the range $[10^{-6}, 10^{-1}]$.

5.7.1 *curvature-dependent domain sorting*

Two- and three-dimensional examples are considered here to observe the evolution of the curvature-dependent domain sorting. The starfish vesicle starts with a random initial distribution of concentration about a mean value $\bar{c} = 0.55$, as shown
in Figure 5.5(a) and Figure 5.7(a). The shape is not allowed to change during evolution.

We first demonstrate how the spontaneous curvature affects the domain sorting. In Figure 5.5, while the dimensionless flexural rigidity is assumed constant and equal to unity, the dimensionless spontaneous curvatures are taken to be $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 10$ for the example in Figure 5.5(b) and $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = -10$ for the example in Figure 5.5(c). Figure 5.5(b) shows that phase $b$ moves toward the positive curvature regions and Figure 5.5(c) shows that phase $b$ moves toward the negative curvature regions. The phase $a$ accumulates in the rest regions of the geometry.

We next examine how the flexural rigidity affects the domain sorting. In Figure 5.6(a) and Figure 5.7, the dimensionless spontaneous curvatures are taken to be $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 1.0 \times 10^{-16}$ and the dimensionless flexural rigidity is taken to be $\kappa(c_a) = 4.0$ and $\kappa(c_b) = 1.0$. Figure 5.6 shows that species reorganize on the surface in such a way that phase $b$ moves toward the higher curvature regions (both of top and bottom of the arms of vesicle) of the geometry while phase $a$ accumulates in regions with relatively low curvature. This numerical result represents good qualitative agreement with the experimental observations shown in Figure 5.6(b). In contrast to the two-dimensional case, the saddle regions of the vesicle in Figure 5.7 have small mean curvature. Thus the phase $a$ accumulates in both the saddle shapes and the lower curvature region of the arms of the starfish vesicle.

5.7.2 Shape deformation dominated by line tension and spontaneous curvature

The initial shape of the vesicle is taken to be a seven-armed starfish (Wintz et al., 1996). A phase-separated state is used for the initial concentration distribution. The dimensionless flexural rigidity is taken to be $\kappa(c_a) = 0.02$ and $\kappa(c_b) = 0.02$. 

114
Figure 5.5: (a) Initial random state with a mean value $\bar{c} = 0.55$. (b) Final equilibrium state of a starfish vesicle. For this example, $\kappa(c_a) = 1.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 10$. (c) Final equilibrium state of a starfish vesicle. For this example, $\kappa(c_a) = 1.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = -10$.

Figure 5.6: (a) Final equilibrium state of a starfish vesicle. For this example, $\kappa(c_a) = 4.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 1.0 \times 10^{-16}$. (b) Experimentally observed curvature-dependent domain sorting (Baumgart et al., 2003b).

The dimensionless spontaneous curvatures are taken to be $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 10$. In this case, shape change is driven by both line tension and spontaneous curvature. Figure 5.8 shows that at an early stage spherical caps are formed mainly by driving phase $b$ towards its natural curvature. As the evolution proceeds, the line tension mainly drives the shape change and finally leads to vesicle
Figure 5.7: (a) Initial random state with a mean value $\bar{c} = 0.55$. For this example, $\kappa(c_a) = 4.0$, $\kappa(c_b) = 1.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 1.0 \times 10^{-16}$. (b) and (c) Two perpendicular views of the final species distribution. The image opacity is set to be 50%.

Figure 5.8: (a) shows the initial phase-separated starfish vesicle. (b)-(f) shows the evolution of shape change. For these results, $\kappa(c_a) = 0.02$, $\kappa(c_b) = 0.02$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 10$. 
fission. Three perpendicular views of shape equilibrium are shown in Figure 5.9.

5.7.3 Spinodal decomposition and coarsening in 3D

Both the examples in this section use an interaction parameter in the double-well potential of $\chi = 2.75$ with the corresponding double-well minima at $c_a \approx 0.0995$ and $c_b \approx 0.9005$. The vesicle starts with a random initial distribution of concentration about a mean value $\bar{c} = 0.30$. The initial condition with random distribution is commonly observed in experiments with GUVs (Baumgart et al., 2003b; Yanagisawa et al., 2007). In experiments, the phase separation and the associated shape changes of the vesicles containing a mixture of lipids and cholesterol is observed by reducing the temperature below the transition temperature. The coupling process of shape changes and phase separation are numerically simulated in these examples.

Ellipsoidal vesicle with random distribution

The initial shape of the vesicle is taken to be a prolate ellipsoid. The flexural rigidities of phase $a$ and phase $b$ are taken to be in the ratio $1 : 4$. The spontaneous curvature of phase $a$ is taken to be $H_{sp}(c_a) = 10^{-16}$, while phase $b$ is assumed to have $H_{sp}(c_b) = 8$. The progress of shape change and species transport is shown in Figure 5.10. The
Figure 5.10: Evolution of shape changes and species transport (a)-(i). For these results, $\kappa(c_a) = 1.0$, $\kappa(c_b) = 4.0$, $H_{sp}(c_a) = 1.0 \times 10^{-16}$ and $H_{sp}(c_b) = 8$.

The complete fission process shown in Figure 5.11 suggests that minimizing line tension energy leads to the formation of buds of phase $b$ and finally leads to the separation of buds. The separated buds are eventually absorbed into the main body of the vesicle through a process like Oswald ripening.

Spherical vesicle with random distribution

The initial shape of the vesicle is taken to be a sphere of radius $0.35$ units. The flexural rigidity of phase $a$ and phase $b$ are taken to be in the ratio $1 : 4$. The spontaneous curvature of phase $a$ is taken to match the curvature of the sphere $H_{sp}(c_a) = 1.0/0.35$, while phase $b$ is assumed to have $H_{sp}(c_b) = 8$.

A sphere represents a minimum in terms of the surface area enclosing a prescribed
volume. Thus, a perfectly spherical vesicle with constraints on volume and surface area cannot undergo any shape change. In this example, we consider two cases. In the first case, no geometric constraints are imposed, so as to allow for shape changes. This approximates experiments where the shape of the vesicle is a slight deviation from a sphere (for a given volume, the vesicle has excess surface area in comparison to a perfect sphere of the same volume). In the other case, geometric constraints are imposed, so the shape changes are not allowed.

The present two cases are intended to illustrate the difference between the normal coarsening and the trapped coarsening effect as described in (Yanagisawa et al., 2007). In experiments, for normal coarsening, the vesicle has a nearly spherical shape and its domain size grows as a power law. In contrast to normal coarsening, the trapped coarsening of a vesicle having a flaccid spherical shape is characterized by a significant slowing down of the domain growth. For the numerical simulations,
Figure 5.12: Time evolution of cross-section images for normal coarsening vesicle (a) and trapped coarsening vesicle (b).

Figure 5.12(a) and Figure 5.12(b) are the time evolution of the normal coarsening and the trapped coarsening domains, respectively. For normal coarsening, the domains do not bud and the coarsening proceeds seamlessly and reaches a steady state at $t = 0.0252$. On the other hand, in the case of trapped coarsening, domain coarsening is suppressed at a certain domain size and proceeds much slower than the normal
Figure 5.13: Evolution of the energetic components for the example discussed in Section 5.7.3. The red line shows the trapped coarsening case and the blue line shows the normal coarsening case. The dimensionless time is plotted on a logarithmic scale. (a) Curvature energy. (b) Line energy. (c) Net energy.

coarsening. The energy evolution plot in Figure 5.13 also indicates that for the trapped coarsening case, the domain coarsening was trapped at a certain size roughly
from $t = 10^{-2}$ to $t = 10^1$.

5.8 Summary

In this Chapter, a continuum chemo-mechanical model is used to numerically simulate the formation and evolution of microdomains on the deforming surface of GUVs. This model is cast in a phase-field framework. The phase-field variables $\phi$ and $c$ represent the vesicle surface and the phase boundary respectively. The coupling between mechanics and chemistry is captured by considering concentration-dependent mechanical properties such as flexural rigidity and spontaneous curvature. The variational form of the governing equations are discretized using a spline-based finite element method. Simulations in three dimensions are required to illustrate the effects of line tension, and the adaptive hierarchical refinement approach allows the three-dimensional simulation to proceed in an efficient manner.

Several numerical examples are used to illustrate the coupling of species transport to vesicle shape changes. The curvature-dependent domain sorting suggests that the vesicle shape and its associated mechanical properties strongly affect phase separation. The shape deformation of a phase-separated starfish vesicle indicates that line tension favors the formation of buds and can lead to fission. The simulations with initial random distribution provide a good qualitative agreement with experimental observations on vesicles. Particularly, the growth dynamics of domains is investigated to exhibit the different growth rates between normal and trapped coarsening. The further quantitative study of the rate of increase of the average size of domains could be performed by using statistical measures, such as structure factors.
In this dissertation, we developed spline-based finite element methods that can treat both sharp and diffuse interface problems in a computationally efficient manner. The proposed approach allows dynamic local refinement in the vicinity of the interface geometry, which leads to significant computational savings compared to fixed mesh approaches. The interface geometry is embedded in the finite element mesh. This flexibility translates to significant savings in computational time for problems where the interface exists on a complex topology or evolves in time.

We examined the use of spline-based finite elements along with a Nitsche technique for enforcing constraints on the boundary and interface. We showed that the level of accuracy in the geometric representation of the interface becomes important when high-order rates of convergence are sought with these methods. To obtain optimal rates of convergence, we employed a hierarchical local refinement approach to improve the geometrical representation of curved interfaces. We further proposed a novel weighting for the interfacial consistency terms arising in the Nitsche variational form with B-splines. A qualitative dependence between the weights and the stabilization parameters was established with additional element level eigenvalue
calculations. An important consequence of this weighting is that the bulk and the interfacial fields remains well behaved in the presence of large heterogeneities as well as elements with arbitrarily small volume fractions.

To proceed simulation in an optimal fashion, we present adaptive spline-based finite elements that employ hierarchical refinement and coarsening techniques. Hierarchical B-spline refinement augments the approximation space through the use of finer, more compactly supported B-splines. With this approach, compatibility is automatic because problematic T-vertices do not arise. The process of refinement does not change the regularity of the basis functions, and can be easily generalized to arbitrary dimensions. An efficient data transfer technique is further proposed to transfer the field in an accurate manner.

The adaptivity technique is then applied to the phase-field model of multicomponent vesicles. A continuum chemo-mechanical model is employed to model the evolution of microdomains on the surface of Giant Unilamellar Vesicles. We provide numerical examples to demonstrate the coupling process of vesicle shape change and species transport. In particular, the curvature-dependent domain sorting and line tension dominated shape changes are presented. We further studied the effect of line tension and curvature elasticity on the kinetics of microdomain growth.

There are several avenues that could be explored in the future. From a method development perspective, the extended spline-based finite elements could be extended to model nonlinearities at the interface and handle multiple interfaces with a single element. While there appear to be additional challenges in extending the proposed approach to three-dimensional problems when it comes to computational geometry, we contend that our approach is nonetheless simpler to implement than a higher-order approximation of the interface.

The parallel implementation of the proposed refinement approach is a nontrivial endeavor, particularly with regard to maintaining good load balancing. We also in-
tend to examine better error estimates for guiding refinement and coarsening. The Bezier extraction operator can be incorporated into the hierarchical spline-based finite elements to achieve better performance. In addition, the current implementation employed a direct solver for the linear algebraic system of equations. For large problems in parallel, an efficient iterative solver is desirable. A multigrid solver with similar properties of hierarchical refinement approach shows some promise in this direction.

For the vesicle modeling, the inclusion of Gaussian curvature energy can be important to model vesicle budding and fission accurately. The growth dynamics of domains on ternary fluid vesicles has been studied experimentally using fluorescence microscopy. In numerical simulation, a quantitative comparison would be made possible by extracting the structure factors on the vesicle’s surface. It would also be interesting to introduce anisotropic surface energy to study the phase separation and its associated shape change.
Bibliography


128


Biography

Wen Jiang was born in Sichuan, China on August 25th, 1984. He graduated from the Beijing University of Aeronautics and Astronautics in Beijing, China with a B. S. in Aircraft Design in 2007 and a M. S. in Solid Mechanics in 2009. In the fall of 2009, he joined the Department of Mechanical Engineering and Materials Science as a graduate fellow and received his doctorate in February, 2015.

Wen has co-authored several peer-reviewed journal articles, including “Adaptive refinement of the hierarchical b-spline finite element method and data transfer scheme” (Jiang and Dolbow, 2014) and “A robust Nitsche’s formulation for interface problems with spline-based finite elements” (Jiang et al., 2015). In addition, the journal articles “Modeling the Evolution of Microdomains in Giant Unilamellar Vesicles with a Phase-Field Approach” is currently in revision.