

Topology Optimization using the Level Set and  
Extended Finite Element Methods:  
Theory and Applications

Hernan Villanueva

B.S. Mechanical Engineering, University of Kansas, 2011

M.S. Mechanical Engineering, University of Colorado Boulder, 2013

9 January 2015

## **Abstract**

Thesis directed by Professor Kurt Maute, Dr.-Ing.

The goal of this thesis proposal is to introduce a unified topology optimization framework that uses the **Level Set Method** (LSM) to describe the design geometry and the **eXtended Finite Element Method** (XFEM) to solve the governing equations and measure the performance of the design. The methodology is presented as an alternative to homogenization optimization approaches, and is able to accommodate a broad range of engineering design problems. The framework will be studied with applications in structural, incompressible Navier-Stokes flow, and scalar transport problems, to test the robustness and the characteristics of the method. A comparison of the framework against density approaches will be studied with regards to convergence, performance, and the capability to manufacture the designs.

# Contents

<b>List of Figures</b>	<b>5</b>
<b>List of Tables</b>	<b>12</b>
<b>1 Introduction</b>	<b>14</b>
<b>2 Background</b>	<b>23</b>
2.1 Optimization . . . . .	23
2.2 Topology optimization . . . . .	25
2.3 Density topology approaches . . . . .	27
2.3.1 Smoothing filter . . . . .	27
2.3.2 Density methods applied to fluid dynamics . . . . .	30
2.3.3 Discussion of the SIMP approach . . . . .	31
2.4 Level set method . . . . .	33
2.4.1 Topology optimization with the level set method . . . . .	36
2.5 Extended finite element method . . . . .	37
<b>3 A complete methodology for the implementation of XFEM inclusive models</b>	<b>41</b>
3.1 Introduction . . . . .	42
3.1.1 Finite element method . . . . .	42
3.1.2 Discontinuity . . . . .	42

3.1.3	eXtended Finite Element Method . . . . .	43
3.1.4	Level set method . . . . .	43
3.1.5	Delaunay triangulation . . . . .	44
3.2	Implementation . . . . .	46
3.2.1	Summary . . . . .	46
3.2.2	Glossary . . . . .	47
3.2.3	Procedure overview . . . . .	48
3.2.4	Implementation algorithms . . . . .	49
3.2.5	Solving the problem . . . . .	63
3.2.6	Preconditioner . . . . .	64
3.3	Corroboration and results . . . . .	66
3.3.1	Methodology . . . . .	66
3.3.2	Tests . . . . .	67
3.4	Conclusions . . . . .	70
3.5	Appendix A: Delaunay Triangulation code . . . . .	71
3.5.1	main.m . . . . .	71
3.5.2	xfem8isct.m . . . . .	73
3.5.3	xfem8tet.m . . . . .	74
3.5.4	number_configurations.m . . . . .	80
<b>4</b>	<b>Density and level set XFEM schemes for topology optimization of 3D structures</b>	<b>82</b>
4.1	Introduction . . . . .	83
4.2	Geometry Modeling . . . . .	88
4.3	Structural Analysis . . . . .	90
4.3.1	Structural Model . . . . .	90
4.3.2	Discretization . . . . .	93
4.3.3	Preconditioner . . . . .	94

4.4	Optimization Model . . . . .	96
4.4.1	XFEM . . . . .	97
4.4.2	SIMP . . . . .	97
4.5	Computational Considerations . . . . .	99
4.6	Numerical Examples . . . . .	101
4.6.1	Cube with center load . . . . .	103
4.6.2	Cuboid under torsion . . . . .	109
4.6.3	Two-phase Cantilevered Beam Design . . . . .	119
4.7	Conclusions . . . . .	122
<b>5</b>	<b>Draft: Level Set XFEM Topology Optimization of 3D Navier-Stokes and Scalar Transport Problems</b>	<b>123</b>
5.1	Methodology . . . . .	124
5.2	Numerical examples . . . . .	126
5.2.1	Optimization criteria . . . . .	127
5.2.2	Pipebend and Diffuser . . . . .	128
5.2.3	Micromixer . . . . .	132
5.3	Conclusions . . . . .	137
<b>6</b>	<b>Draft: Topology Optimization Approaches for the Curvature Minimization of Level Set Isocontours</b>	<b>138</b>
6.1	Curvature . . . . .	138
6.1.1	Comparison of curvature and mean curvature in $\mathbb{R}^2$ . . . . .	140
6.2	Curvature Squared . . . . .	142
6.3	Level set normal unit vector . . . . .	142
6.3.1	$\mathbf{n}_\phi$ . . . . .	143
6.3.2	$\mathbf{n}_u$ . . . . .	143
6.3.3	$\mathbf{n}_\psi$ . . . . .	145

6.3.4	$\mathbf{n}_g$	147
6.4	Methodology	152
6.4.1	Sweep setup	152
6.4.2	Preliminary sweep results	154
6.5	Preliminary optimization results	157
6.5.1	$\kappa_n^2 = \ \frac{d\mathbf{n}_u}{ds}\ ^2$	158
6.5.2	$\kappa_n^2 = \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}$ with $\mathbf{n}_g$	159
6.6	Conclusions	160
<b>7</b>	<b>Draft: Modeling of Multiple Scalar Transport Fields for an Atomic Layer Deposition Machine</b>	<b>161</b>
7.1	Conclusions	165
<b>8</b>	<b>Conclusions</b>	<b>166</b>
<b>9</b>	<b>Remaining Work</b>	<b>168</b>
9.1	Ghost penalty formulation	168
9.2	Curvature	168
9.3	Atomic layer deposition	169
9.4	Timeline	169
<b>Bibliography</b>		<b>170</b>

# List of Figures

1.1	The zero level set isolevel of the level set function, $\partial\Omega = \Gamma_{\phi=0}$ , in 1.1a divides the fixed mesh grid into different phase regions in 1.1b, where each phase may represent a different material or a different physics. . . . .	15
1.2	Setup of a structural topology optimization problem. A mesh of size $3L \times 2L$ , with $45 \times 30$ quadrilateral linear elements is anchored to the wall on its left side, and subject to a point load on its right side. . . . .	16
1.3	Comparison of the geometry representation of the density and level set methods for a “solid-void” structural topology optimization problem. . . . .	17
1.4	In an Ersatz material interpolation approach, the material properties of each finite element are interpolated proportional to the volumes of the solid and void phases. . . . .	18
1.5	Remeshing the design domain such that elements align with the zero level set isolevel is an alternative to the Ersatz material interpolation approach. . . .	18
1.6	In the XFEM, the elements cut by the zero level set isolevel are divided into subdomains and interfaces for integration. Circles represent the abscissae for the subdomains, while crosses represent the abscissae for the interface integration. . . . .	19
2.1	Progression of the optimization process. Objective is to minimize compliance subject to a maximum volume fraction of 0.5 for the solid phase. . . . .	26

2.2	The density Heaviside projection for various magnitudes of $\beta$ .	29
2.3	Influence of the interpolation penalty $\alpha_p$ .	32
2.4	Level set description of two circular inclusions with radii 0.667 and 0.333, respectively, moving towards each other. Level set functions can be used to describe complex topologies in a fixed mesh.	35
2.5	A two dimensional finite element has eight possible decompositions based on the nodal level set values.	37
2.6	Structural problem setup.	38
2.7	XFEM implementation example physical model	39
2.8	The center node, denoted by the color blue, uses different degrees-of-freedom to describe the disconnected phase regions. The subscripts denote the $m$ enrichment level. The maximum number of enrichment levels used for each phase is $M = 5$ . A value of 0 denotes the original finite element degrees-of-freedom, while other numbers indicate additional “enriched degrees-of-freedom”.	39
3.1	The zero level set isolevel of the level set function, $\partial\Omega = \Gamma_{\phi=0}$ , in 3.1a divides the fixed mesh grid into different phase regions in 3.1b, where each phase may represent a different material or a different physics.	44
3.2	Delaunay triangulation	45
3.3	Delaunay circumcircles	45
3.4	2D triangulation configurations	46
3.5	3D triangulation example.	47
3.6	Structural problem setup.	49
3.7	XFEM implementation example physical model	50
3.8	XFEM implementation example discrete model	50
3.9	Edge discretization of XFEM model example	53
3.10	Intersection point computation	54
3.11	Subphase computation algorithm.	55

3.12 XFEM implementation example 2 . . . . .	60
3.13 XFEM implementation example 3 . . . . .	62
3.14 Diffusion problem setup . . . . .	67
3.15 Mesh refinement sweep interface error . . . . .	68
3.16 Mesh refinement sweep L2 error . . . . .	68
3.17 Conductivity refinement sweep interface error . . . . .	69
3.18 Conductivity refinement sweep L2 error . . . . .	69
3.19 Condition number comparison - no preconditioner . . . . .	70
3.20 Condition number comparison - with preconditioner . . . . .	71
 4.1 Initial design with array of spherical inclusions for the cantilever beam example of Section 4.6.3; blue represents phase “A” material and red represents phase “B”. . . . .	90
4.2 Two-phase problem. . . . .	91
4.3 Intersection patterns for a two-dimensional QUAD4 element. . . . .	99
4.4 Initial assignment of enrichment levels. . . . .	99
4.5 Initial and final assignment of enrichment levels considering a cluster of elements.	100
4.6 Cube with center load. . . . .	103
4.7 SIMP results for cube with center load problem; clockwise: bottom, side, top, and clipped views. . . . .	104
4.8 IDL post-processing of SIMP results for cube with center load problem. . . .	105
4.9 Initial level set configurations for cube with center load problem. . . . .	106
4.10 LSM-XFEM results for cube with center load problem; clockwise: bottom, side, top, and clipped views. . . . .	107
4.11 Evolution of strain energies in the optimization process for SIMP and LSM-XFEM approaches. . . . .	107
4.12 Cuboid under torsion model. . . . .	110

4.13	SIMP results of cuboid under torsion problem for different levels of mesh refinement; SIMP parameters: $p = 3$ , $r_\rho = 1.6$ of the element edge length, $\beta = 0$ .	111
4.14	LSM-XFEM results of cuboid under torsion problem for two levels for mesh refinement; LSM-XFEM parameters: $r_\phi = 1.6$ of the element edge length, no perimeter constraint.	113
4.15	SIMP results for different projection parameters $\beta = [0.0, 4.0, 8.0]$ ; mesh size: $120 \times 30 \times 30$ .	114
4.16	IDL post-processing of SIMP results for cuboid under torsion problem; the vertical lines mark the threshold values at which the volume constraint is satisfied.	115
4.17	SIMP and LSM-XFEM results for larger smoothing radius; mesh size: $120 \times 30 \times 30$ .	117
4.18	LSM-XFEM results for cuboid under torsion problem using a perimeter constraint: (a) LSM-XFEM restarted from SIMP and (b) LSM-XFEM restarted from LSM-XFEM result; mesh size: $120 \times 30 \times 30$ .	118
4.19	Initial setup for two-phase cantilever beam problem.	119
4.20	SIMP and LSM-XFEM results for different stiffness ratios: (from left to right) $E_B = 0.5E_A$ , $E_B = 0.1E_A$ , $E_B = 0.01E_A$ , $E_B$ is void; SIMP results (top row); LSM-XFEM results (bottom row).	120
4.21	LSM-XFEM optimized two-phase design for $E_B = 0.1E_A$ realized by 3D printing.	121
5.1	Setup for the pipebend problem. A parabolic inlet flow in the $x$ -direction, with a maximum value of 1.0 enters the design domain through a circular inclusion of diameter $L$ in the minimum YZ facet. The outlet is modeled as a circle with diameter $L$ and is located in the minimum $XZ$ facet with a constant pressure of 1.0.	128

5.2	Setup for the diffuser problem. A parabolic inlet flow in the $x$ -direction, with a maximum value of 1.0 enters the design domain through the entire minimum YZ facet. The outlet, modeled as a circle of diameter $\frac{5}{3}L$ is located in the maximum YZ facet with a constant pressure of 1.0. . . . .	129
5.3	Initial designs for the pipe bend problem. . . . .	131
5.4	Initial designs for the diffuser problem. . . . .	131
5.5	Results for the Stokes and Navier-Stokes pipebend optimization problems with a Reynolds number of 1. . . . .	132
5.6	Results for the Stokes and Navier-Stokes diffuser optimization problems with a Reynolds number of 1. . . . .	133
5.7	Setup for the micromixer problem. A parabolic inlet flow in the $x$ -direction, with a maximum value of 1.0 enters the design domain through a circular inclusion of diameter $L$ through the minimum YZ facet. The upper area of the inlet has a scalar value of 1.0, while the lower area has a value of 0.0. The outlet is located in the maximum YZ facet with a constant pressure of 1.0. .	134
5.8	Results for the Stokes and Navier-Stokes micromixer optimization problems with a Reynolds number of 1 and a thermal conductivity of $k_{xx}^f = 0.001$ . Legend shows the magnitude of the scalar field. . . . .	135
5.9	Streamlines for the Stokes and Navier-Stokes micromixer optimization problems with a Reynolds number of 1 and a thermal conductivity of $k_{xx}^f = 0.001$ . .	136
6.1	Discontinuities in the $\mathbf{n}_\phi$ field at a node. . . . .	144
6.2	A level set field for a circular inclusion of radius 0.5, and their respective hyperbolic tangent projections for different values of $\epsilon$ . . . . .	146
6.3	Sinusoidal wave with an amplitude of 0.50. Mesh discretization is $45 \times 30$ elements. The red dot indicates one of the points at which the level set function intersects an element. The red dotted line indicates all cut elements that lie on the same interface of the red dot within a search radius $r_\kappa = 0.4$ . .	148

6.4	Zoomed image for the cut elements within the search radius $r_\kappa = 0.4$ . Blue elements represent the cut element that are on the same interface. . . . .	149
6.5	Discretization of the structural linear beam elements. Inner nodes have two separate rotational degrees-of-freedom, left and right, one for each adjacent element. Nodes at the end of the interface only have a single rotational degree-of-freedom. All nodes rest on a field of artificial springs. . . . .	150
6.6	Computation of $\alpha_{\text{CR}}$ . $\alpha_{\text{CR}}$ is a function of the geometrical normal unit vectors of the adjacent elements. . . . .	151
6.7	Setup for sweeping over the radius of a circle ranging from 0.25 to 0.95. . . .	153
6.8	Setup for sweeping over the amplitude of a sinusoidal wave ranging from 0.25 to 0.75. . . . .	154
6.9	Squared curvature and absolute error for the circular sweep using the $\mathbf{n}_\phi$ formulation, with mesh $45 \times 30$ . . . . .	155
6.10	Squared curvature and absolute error for the circular sweep using the $\mathbf{n}_\phi$ formulation, with mesh $180 \times 120$ . . . . .	155
6.11	Squared curvature and absolute error for the circular sweep using the $\mathbf{n}_u$ formulation, with mesh $45 \times 30$ . . . . .	156
6.12	Squared curvature and absolute error for the sinusoidal wave sweep using the $\mathbf{n}_u$ formulation, with mesh $45 \times 30$ . . . . .	156
6.13	Setup for the curvature optimization study. A mesh of size $3L \times 2L$ , with $45 \times 30$ quadrilateral linear elements is anchored to the wall on its left side, and subject to a point load on its right side. . . . .	158
6.14	Optimized geometry for computing strain energy over the interface 6.2 with formulation $\mathbf{n}_u$ . $w_\Pi = 1$ and $w_\kappa = 0.1$ . . . . .	158
6.15	Optimized geometry for computing strain energy over the interface 6.2 with formulation $\mathbf{n}_u$ . $w_\Pi = 1$ and $w_\kappa = 1$ . . . . .	159

6.16 Optimized geometry for computing strain energy over the interface 6.26 with formulation $\mathbf{n}_g$ . $w_{\Pi} = 1$ and $w_{\kappa} = 0.1$ . $k_{\kappa} = 0.01$ and $r_{\kappa} = 1.2$ .	159
6.17 Optimized geometry for computing strain energy over the interface 6.26 with formulation $\mathbf{n}_g$ . $w_{\Pi} = 1$ and $w_{\kappa} = 1$ . $k_{\kappa} = 0.01$ and $r_{\kappa} = 1.2$ .	160
7.1 CAD model of the atomic layer deposition machine. The holes at the top represent the inlets through which the chemical reactants will enter. The conveyor belt at the bottom moves the web in the direction of the arrows as the chemicals are deposited. The machine has open boundaries which may cause reactions with air, and lead to imperfections in the deposition.	162
7.2 Cross-sectional view of the ALD machine. The black dotted line on the right represents a symmetry boundary condition. The coating head and sled are made of solid aluminum. The web on which the reactants are deposited is a polymer. The blue dotted line represents the gas inlets, and the red dotted line represents the open boundary conditions.	163
7.3 View of the coating sled after a deposition was performed. The picture shows that large particle clusters are formed at the open boundaries. Our hypothesis is that air reacts at the edges with the TMA reactant.	163

# List of Tables

3.1	Point ID to cell IDs connectivity table . . . . .	51
3.2	Edge to points and cells IDs connectivity table . . . . .	52
3.3	Main-phase and sub-phase table for pseudo-elements . . . . .	55
3.4	Initial node-element table . . . . .	57
3.5	Flipping the enrichment levels . . . . .	58
3.6	Final node-element table . . . . .	59
3.7	Element to enrichment table . . . . .	59
3.8	Initial node-element table configuration 2 . . . . .	61
3.9	Final node-element table configuration 2 . . . . .	61
3.10	Element to enrichment table configuration 2 . . . . .	61
3.11	Initial node-element table configuration 3 . . . . .	63
3.12	Final node-element table for configuration 3 . . . . .	63
3.13	Element to enrichment table configuration 3 . . . . .	63
4.1	Comparison of strain energies of SIMP and LSM-XFEM results for cube with center load problem; the corresponding designs are shown in Figs. 4.7 and 4.10.106	
4.2	Strain energies of SIMP results for cuboid under torsion problem; SIMP parameters: $p = 3$ , $r_\rho = 1.6$ of the element edge length, and $\beta = 0$ ; the corresponding designs are shown in Fig. 4.13. . . . .	110

4.3	Strain energies of LSM-XFEM results for cuboid under torsion problem; LSM-XFEM parameters: $r_\phi = 1.6$ of the element edge length, no perimeter constraint; the corresponding designs are shown in Fig. 4.14. . . . .	112
4.4	Strain energies of SIMP results for different projection parameters $\beta = [0.0, 4.0, 8.0]$ ; the corresponding designs are shown in Fig. 4.15. . . . .	114
4.5	Strain energies of the SIMP-IDL designs of cuboid under torsion problem; the corresponding designs are shown in Fig. 4.13. . . . .	116
4.6	Strain energies of SIMP and LSM-XFEM results for cuboid under torsion problem using a mesh independent filter; the corresponding designs are shown in Fig. 4.17. . . . .	117
4.7	Strain energies of LSM-XFEM results for cuboid under torsion problem using a perimeter constraint and different initial designs; the corresponding final designs are shown in Fig. 4.18. . . . .	119
4.8	Strain energies of SIMP and LSM-XFEM results for different stiffness ratios; the corresponding designs are shown in Fig. 4.20. . . . .	121
5.1	Parameters for bend problem. . . . .	130
5.2	Parameters for diffuser problem. . . . .	130
5.3	Parameters for diffuser problem. . . . .	134
6.1	Parameters for the curvature optimization study. . . . .	157

# Chapter 1

## Introduction

The goal of this thesis proposal is to introduce a unified topology optimization framework that uses the **L**evel **S**et **M**ethod (LSM) to describe the design geometry and the e**X**tended **F**inite **E**lement **M**ethod (XFEM) to solve the governing equations and measure the performance of the design. The framework will be referred to as the LSM-XFEM optimization method.

Topology optimization approaches (see Sections 2.1 and 2.2) seek the optimal geometry and/or material layout of a body within a given design domain. Popular schemes, such as density methods (Section 2.3), have been widely applied to topology optimization of different physics, such as structures, fluid dynamics, etc. In structural topology optimization, the density method works by transforming the structural optimization problem into a standard nonlinear program where the design variables are coefficients of the governing equation. These design variables are introduced as the material density of the finite elements. Density approaches cannot describe the physics at the phase interface accurately without extensive mesh refinement. This may lead to unphysical responses (Section 2.3.3). Therefore, problems that require an accurate description of the boundary conditions at the interface may not be suited for density approaches. The use of level set methods in topology optimization emerged as an alternative to the density approach. Level set methods (Section 2.4) are able to provide

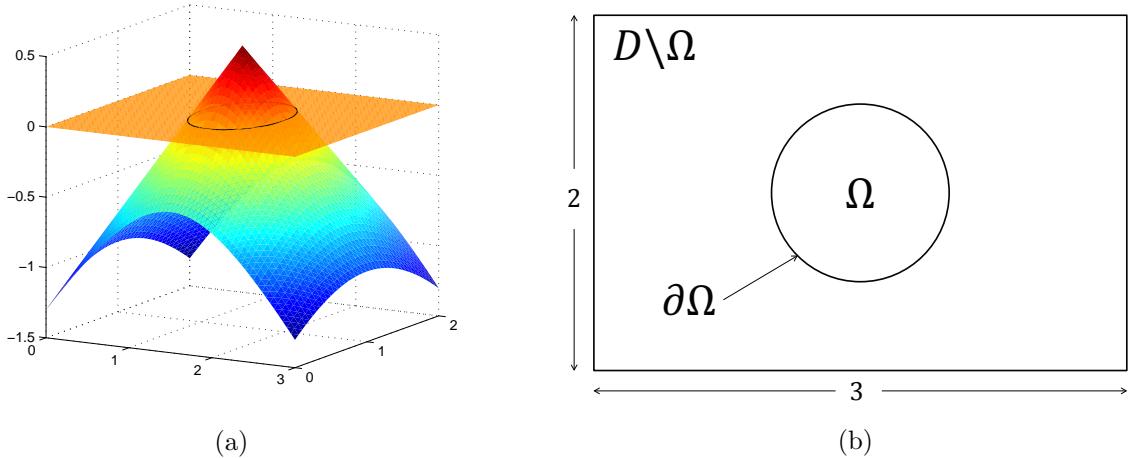


Figure 1.1: The zero level set isolevel of the level set function,  $\partial\Omega = \Gamma_{\phi=0}$ , in 1.1a divides the fixed mesh grid into different phase regions in 1.1b, where each phase may represent a different material or a different physics.

an accurate representation of the layout by dividing the design domain into phase regions, where each region may represent a different physics or a different material, as shown in Figure 1.1.

To illustrate the differences in the geometry representation between the density and level set methods, consider the “solid-void” structural optimization problem presented in Figure 1.2, where the objective is to minimize the structural compliance by changing the material layout, subject to a maximum volume fraction of 0.5 for the solid phase. The results for both the density and level set approaches are shown in Figure 1.3. Notice both designs generate similar truss-like geometries. Figure 1.3a shows how the density method represents the interface between the solid and void materials as intermediate “grey” densities, while the level set method in Figure 1.3b describes the geometry by cutting the elements.

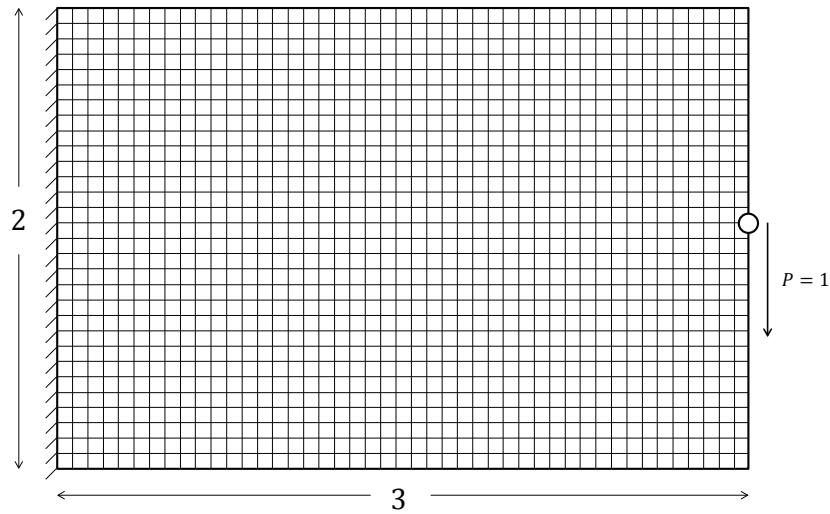
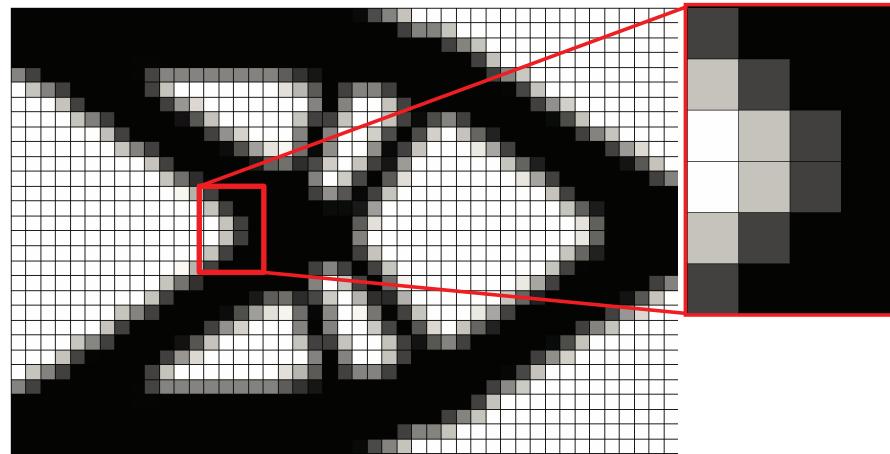
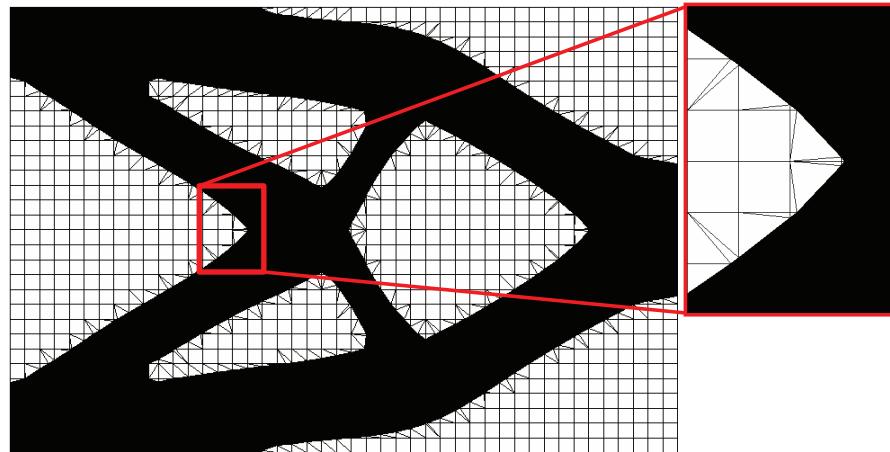


Figure 1.2: Setup of a structural topology optimization problem. A mesh of size  $3L \times 2L$ , with  $45 \times 30$  quadrilateral linear elements is anchored to the wall on its left side, and subject to a point load on its right side.



(a) Density method.



(b) Level set method.

Figure 1.3: Comparison of the geometry representation of the density and level set methods for a “solid-void” structural topology optimization problem.

In the level set method, the material properties of the structural finite elements are interpolated between the void and solid phases, proportional to the volumes of the individual phases. This approach is called Ersatz material interpolation and it is very similar to the interpolation used in density methods. Therefore, it leads to similar issues with regard to enforcing boundary conditions and predicting the physical response along the boundary. The process is illustrated in Figure 1.4.

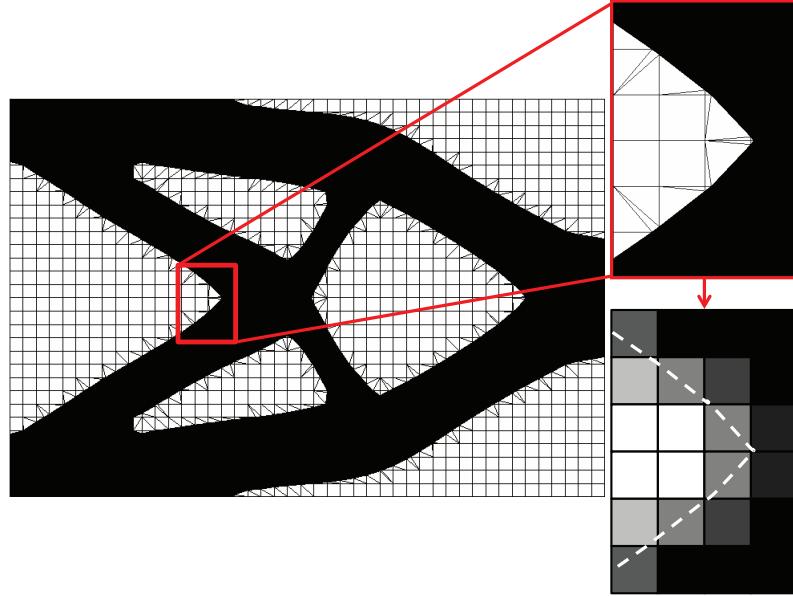


Figure 1.4: In an Ersatz material interpolation approach, the material properties of each finite element are interpolated proportional to the volumes of the solid and void phases.

An alternative to the Ersatz interpolation is to repeatedly generate new meshes that align with the geometry of the zero level set isolevel, as shown in Figure 1.5. However, generating an entirely new body fitted mesh typically suffers from robustness and efficiency, particularly for three dimensional problems. It was also shown by [92] and [121] that this method affects the convergence of the optimization process.

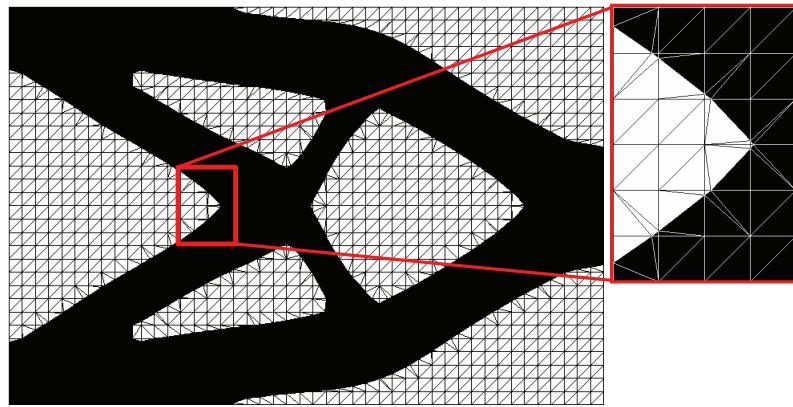


Figure 1.5: Remeshing the design domain such that elements align with the zero level set isolevel is an alternative to the Ersatz material interpolation approach.

Another approach is used by the XFEM (Section 2.5). The XFEM is an immersed bound-

ary technique that works on fixed grids, and decomposes the cut elements into subdomains and interfaces that it uses for integration, as shown in Figure 1.6.

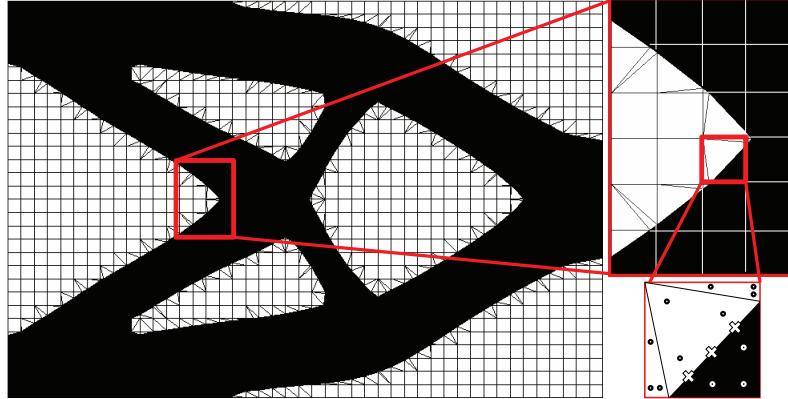


Figure 1.6: In the XFEM, the elements cut by the zero level set isolevel are divided into subdomains and interfaces for integration. Circles represent the abscissae for the subdomains, while crosses represent the abscissae for the interface integration.

The XFEM enriches the finite element solution space allowing for discontinuities (see Sections 4.3.2 and 4.5 for details). In this thesis proposal, we will use a Heaviside enrichment strategy, which allows for solution fields with discontinuities  $C^{-1}$ .

As the topology of the level set function changes, it will occur that a phase subdomain is small enough to cause numerical instabilities in the system, and not lead to convergence. Several approaches have been studied in the literature to try and ameliorate this issue. [56] introduced a preconditioner that scales the solution space proportional to the areas of the subdomains. [17, 93, 94] used a ghost penalty formulation to smooth the gradients of the solution at the facets of the cut elements. Both approaches presented favorable results for two dimensional diffusion and incompressible Navier-Stokes flow problems.

Perimeter or curvature measures can be computed at the level set interface using the XFEM. These measures can be introduced into the optimization problem in order to provide a global shape control feature, and prevent the appearance of small floating particles [64].

This thesis proposal will analyze the characteristics of the proposed LSM-XFEM method as follows:

1. [64] studied the XFEM decomposition of the cut elements into subdomains, and the application of multiple enrichment levels in two dimensional problems. Our first hypothesis is that the same approach used by [64] can be extended to three dimensions. Section 3 studies the algorithmic details of the approach, and the robustness of the method to describe complex three dimensional geometries using the level set method.
2. The next question is whether the method can accurately describe the physics at the phase interface without extensive mesh refinement. If so, does it provide an advantage over the interface description provided by the density methods? There are several approaches to approximate the solution at the interface of the level set field, as described in Section 2.5. We will study the application of stabilized Lagrange multipliers in Section 4, and of the Nitsche method in Section 5.
3. If the framework can describe complex geometries generated by a level set field, as well as accurately represent the physics at the phase interface, how does it fare overall against density approaches? Section 4 applies the framework to three dimensional structural problems, and provides a comprehensive comparison against density methods. The objective is to study convergence rates of the optimization problems, shape control of the optimized geometries, postprocessing capabilities of the methods to generate manufacturable designs, and the effects of providing different initial designs to the optimization process. The hypothesis is that the framework provides a better description of the geometry and material distribution than density methods, and therefore requires coarser meshes which leads to faster computations. Furthermore, it provides the capability to extract surface meshes directly from the level set function and manufacture the designs using three dimensional printing. This feature would be promising for rapid-prototyping.
4. In section 4.3.3, we will expand the scaling preconditioning scheme to structural problems. The preconditioner was previously studied for two dimensional heat conduction

problems. The scaling should ameliorate the effects that small intersection regions cause to the solution. In section 5, we will study an alternate approach, the face-oriented ghost-penalty, and its effects on incompressible Navier-Stokes flows. This penalty formulation has been studied in the literature for diffusion and incompressible Navier-Stokes problems, and our hypothesis is that it can be applied in our optimization framework to ensure convergence for three dimensional problems.

5. Once we can ensure convergence of the solution, the framework should be able to solve a broad range of design engineering problems. In this thesis proposal, we will study structural, incompressible Navier-Stokes flow, and scalar transport problems. These physics will help understand the characteristics, and the capability of the method to be applied to real-world engineering problems. Structural problems are studied in Section 4. Incompressible Navier-Stokes and scalar transport problems are studied in 5.
6. Controlling the shape of the optimized geometry is important in manufacturing. The effects of using a smoothing filter for the level set field (see Section 2.4) will be studied in Section 4.6.2. This filter will also be compared against the density filter (see Section 2.3.1) as a way of controlling the minimum feature size. Regularization techniques applied to the level set interface, such as measuring the curvature and perimeter, may cause different optimized geometries to emerge. The use of these measures and its effects on the finalized design will be studied in Section 6.

The studies proposed above will help us understand the characteristics and capabilities of the framework. The final step is to use this knowledge to study the method in the context of a real-world problem that requires the modeling of a three dimensional incompressible Navier-Stokes flow with a scalar transport field in Section 7.

The specific objectives of this thesis proposal are then: (i) To generate a robust LSM-XFEM topology optimization scheme; (ii) To compare the LSM-XFEM optimization scheme with traditional homogenization methods, such as SIMP, and study the advantages and

disadvantages of our formulation; (iii) and to explore the characteristics of the methodology through cases studies in selected applications.

The layout of this thesis proposal is as follows: Section 1 presented the motivation and goals of the study, Section 2 presents an overview of the concepts required to understand this work, Section 3 presents the algorithm to implement an XFEM framework in software, Section 4 presents the application of the methodology in structural problems, and compares the results to density approaches, Section 5 presents the application of the method to incompressible Navier-Stokes flows and scalar transport problems, Section 6 studies regularization techniques to improve the manufacturing of the design, Section 7 uses the methodology for a real-world design engineering problem. Finally, Sections 9 and 8 show the remaining work and conclusions of this thesis proposal.

# Chapter 2

## Background

This chapter presents a brief overview on topology optimization, the level set method, and finite element methods. The information presented in this chapter is sufficient such that the reader can understand the framework in which the thesis is developed, but it is not comprehensive. References are provided for the reader who wishes to see more details on the topics.

### 2.1 Optimization

An optimization problem is a problem in which you seek to find the best solution from the set of all feasible solutions. There are two categories of optimization problems, and their classification depends on whether the problem variables are continuous or discrete. In this work, we will focus on optimization problems with continuous variables. An optimization problem with discrete variables is known as a combinatorial optimization problem, and the reader is directed to [75] for a review.

The standard form for an optimization problem with continuous variables is:

$$\begin{aligned} & \min_{\mathbf{s}} \mathcal{F}(\mathbf{s}, \mathbf{u}(\mathbf{s})), \\ \text{s.t. } & \begin{cases} \mathbf{s}, & \text{subject to design constraints } \mathcal{G}_j \leq 0, \\ \mathbf{u}, & \text{solves } W = 0 \text{ for a given } \mathbf{s}, \end{cases} \end{aligned} \quad (2.1)$$

where the goal is to minimize some objective functional  $\mathcal{F}$  with respect to the vector of design variables  $\mathbf{s}$ . The design variables are subject to the  $j$ -th design constraint  $\mathcal{G}_j$ , which are enforced as an inequality problem. In general, the objective and constraints depend on the optimization variables,  $\mathbf{s}$ , and the state variables,  $\mathbf{u}$ . These state variables,  $\mathbf{u}$ , satisfy the weak form of the equilibrium equation  $W$ , which is enforced as an equality constraint. By convention, Equation 2.1 defines a minimization problem. Maximization problems are handled by placing a negative sign on the objective functional  $\mathcal{F}$ .

There are several optimization algorithms available to update the design variables. For structural optimization problems with non-trivial and multiple constraints, the Method of Moving Asymptotes (MMA) [107], and its globally convergent counterpart, the Globally Convergent Method of Moving Asymptotes (GCMMA) [109] have become the algorithms of choice. The work of this document will focus on the GCMMA algorithm. For a reference on more advanced mathematical tools, such as the Sequential Quadratic Programming (SQP), the Sparse Nonlinear OPTimizer (SNOPT) [39], or the Interior Point OPTimizer [119], the reader is referred to the work of [101].

Gradients of the objective functional and design constraints with respect to the design variables are required by the mathematical algorithms specified above. The sensitivities for the optimization problems presented in this document will be computed using the adjoint method [18, 38].

In the following sections, we will delve into the approaches available to solve the optimization problem.

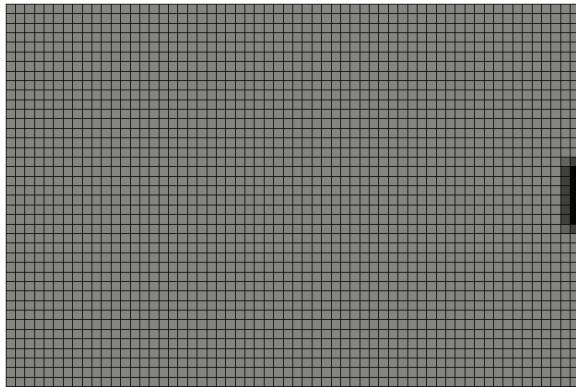
## 2.2 Topology optimization

Topology optimization is a type of optimization problem in which you seek the optimal geometry and/or material layout of a body within a given design domain  $D$ . For example, mathematically we can formulate the following generic topology optimization problem:

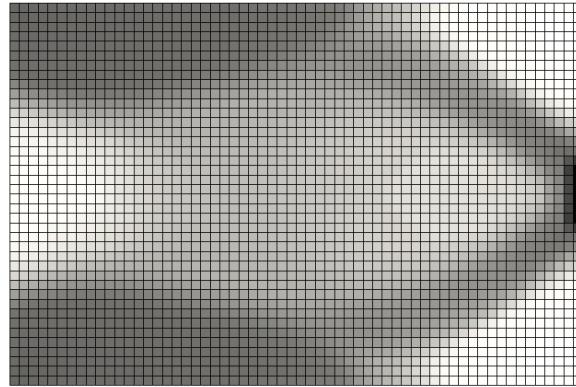
$$\min_{\mathbf{s}} \int_{\Omega} f(\mathbf{s}(\mathbf{x})) d\Omega \quad (2.2)$$

which, analogous to Equation 2.1 is subject to design constraints and equilibrium equations. The goal of the optimization problem is to minimize some objective functional  $\int_{\Omega} f(\mathbf{s}(\mathbf{x})) d\Omega$  over the design domain with respect to the design variables  $\mathbf{s}(\mathbf{x})$ , where  $\Omega$  represents the shape of the design. The optimization problem will set the design variables as “on” or “off” in order to minimize the design objective, while satisfying the design constraints. We can represent a design variable at a point  $i$  as “on” by setting  $s_i = 1.0$ , and “off” by  $s_i = 0.0$ . A quick example is described in Figures 1.2 and 2.1. Figure 1.2 shows the setup for a topology optimization problem with an objective of minimal compliance and subject to a maximum volume fraction of 0.5 for the solid phase. Figure 2.1 shows the changes in the design, where the design variables at the elements are represented as “on” (black) or “off” (white) or in-between (grey).

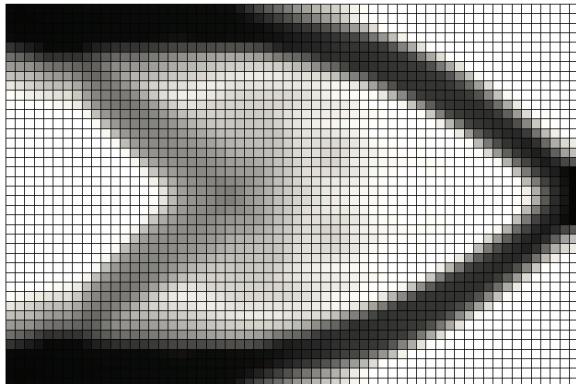
Topology optimization provides the ability to create designs that are not always intuitive, or to improve on existing designs. Topology optimization methods were initially developed to create conceptual designs in the early stage of the design process [11, 90]. It is interesting to note that the earliest recorded paper on topology optimization dates back to 1904, with the work of Australian inventor Michell in the derivation of optimality criteria for least-weight layout of trusses [72]. And while topology optimization focused for decades on structural design, it has recently found its application in a wide range of physical disciplines [10] including acoustics [30], wave propagation [31], electromagnetics [55, 96] and optics [12, 49]. For a more detailed review on topology optimization, please refer to [11, 28].



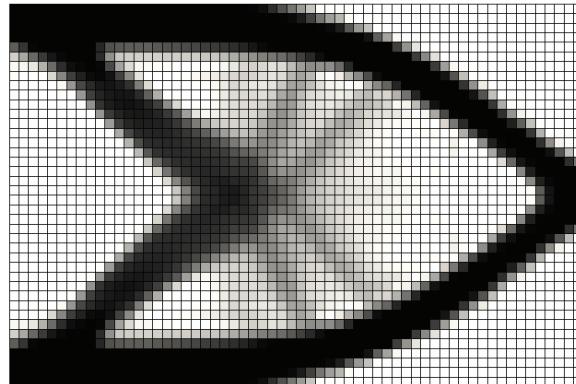
(a) Step 0.



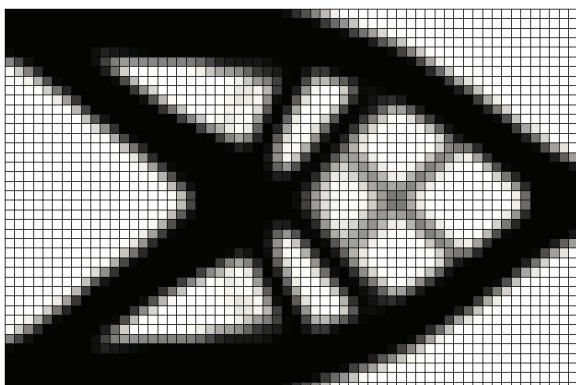
(b) Step 5.



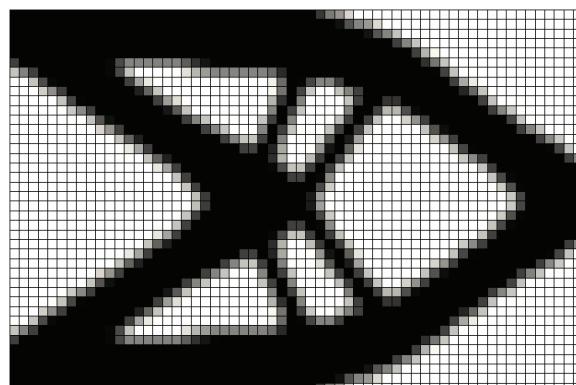
(c) Step 10.



(d) Step 15.



(e) Step 25.



(f) Step 50.

Figure 2.1: Progression of the optimization process. Objective is to minimize compliance subject to a maximum volume fraction of 0.5 for the solid phase.

Structural topology optimization, specifically topology optimization of continuum structures, is in its mathematical nature one of the most challenging optimization problems [11]. However, in 1988, [8] introduced their seminal paper on the homogenization method. In this method, the design domain is assumed to be formed by a material with micro-scale voids, and the topology optimization problem seeks the optimal porosity of the porous medium in order to minimize the objective functional. Due to its effectiveness and simplicity, homogenization-based methods found a lot of applications in structural design, and quickly became the main approach in structural topology optimization [7]. The homogenization method works by transforming the structural optimization problem into a standard nonlinear program where the design variables are coefficients of the underlying governing equation, and therefore is capable of producing internal holes in the design domain without *a priori* knowledge of them.

## 2.3 Density topology approaches

In 1989, Bendsøe [7], Zhou and Rozvany [129] introduced the most popular of homogenization methods, the “solid isotropic material with penalization” (SIMP). In SIMP, the design variables represent the artificial densities,  $\rho(\mathbf{x})$ , of a group of elements in a fixed finite element grid (our design domain  $D$ ) and their material properties are parameterized in terms of a set of material interpolation functions such that intermediate properties are penalized. The typical approach is to assume a value of  $\rho = 1.0$  represents solid material, while  $\rho = 0.0$  represents void.

### 2.3.1 Smoothing filter

An additional numerical scheme is necessary to smear out numerical instabilities. This is referred to as the filtering method [97, 99]. For example, in structural topology optimization, the density, rather than being equal to a single variable, can be computed from a linear smoothing filter as follows:

$$\tilde{\rho}(\mathbf{s}) = \frac{\sum_{i=1}^E w_i s_i}{\sum_{i=1}^E w_i} \quad (2.3)$$

with

$$w_i = \max(0, r_\rho - \|\mathbf{x}_i - \mathbf{x}\|) \quad (2.4)$$

where  $s_i$  is equivalent to an artificial density  $\rho$  at a point  $\mathbf{x}_i$ ,  $\mathbf{x}_i$  is the location of the node at which the design variable  $i$  is defined,  $\tilde{\rho}(\mathbf{s})$  is the density at a point  $\mathbf{x}$ ,  $w_i$  is the factor of point  $\mathbf{x}$  with respect to the design variable  $i$ ,  $r_\rho$  is the filter radius, and  $E$  is the number of elements in the design domain.

The filter in Equation 2.3 prevents the formation of features smaller than  $r_\rho$ , and serves as a minimum feature size control. However, this comes at the cost of forming intermediate densities along the material interface. Methods for smearing out intermediate densities have been proposed by [34, 100, 106]. [40] proposed a density projection method to reduce the volume occupied by material with intermediate densities. This projection is based on a smoothed Heaviside function and applied to the densities as follows:

$$\hat{\rho}(\tilde{\rho}) = 1 - e^{-\beta\tilde{\rho}} + \tilde{\rho}e^{-\beta} \quad (2.5)$$

where  $\hat{\rho}$  is the projected density, and the parameter  $\beta \geq 0$  controls the crispness of the projection. Notice that for  $\beta = 0$  we recover the original density  $\hat{\rho} = \tilde{\rho}$ . As we increase  $\beta$ , more and more intermediate densities are penalized towards the value of 1.0, as shown in Figure 2.2. Note, however, that if the objective and/or constraints of the optimization problem find the intermediate densities to be beneficial, the optimization algorithm will ignore the effects of this projection scheme. The reader is referred to [40, 41, 100, 114, 122] for more details on projection schemes. In structural topology optimization, it is important to model the relation between density and stiffness. SIMP models the stiffness proportional to the density in the power  $p$ , where  $p > 1$  [9], in order to guarantee a well-posed optimization

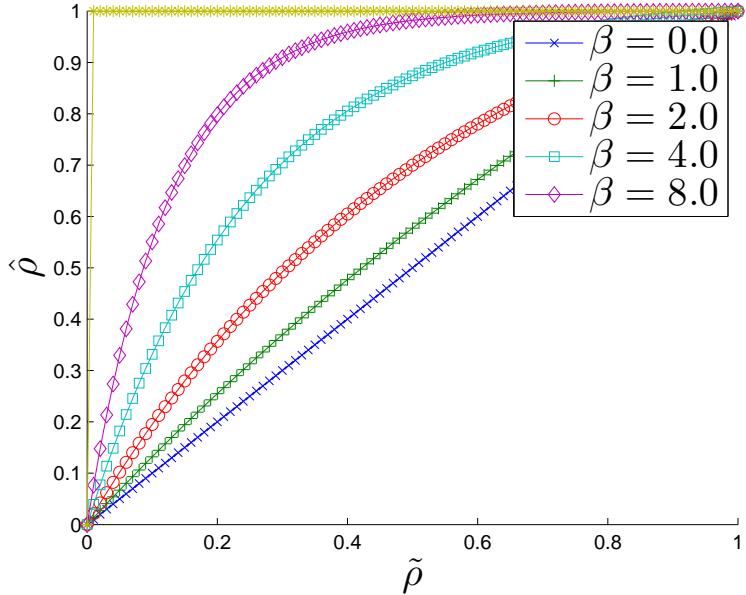


Figure 2.2: The density Heaviside projection for various magnitudes of  $\beta$ .

problem [9]. The structural stiffness for a “solid-void” problem can then be formulated as:

$$E(\mathbf{x}) = \hat{\rho}^p E_0 \quad (2.6)$$

where  $E_0$  is the initial structural stiffness of the material. Typically, the parameter  $p$  is set to 1, and then increased as the optimization progresses [91]; this is the so-called continuation method [102]. It was shown that if one uses  $p > 3$ , we recover a black-and-white binary-like material distribution [11]. That is why the density approach has been referred to as a *pixelated* geometric model.

The SIMP method can be expanded to model a multi-material optimization problem. For example, applying a “rule of mixture”, we can model a two-phase material in structural topology optimization by modifying Equation 2.6 as:

$$E(x) = \hat{\rho}^p E_A + (1 - \hat{\rho}^p) E_B \quad (2.7)$$

where  $E_A$  represents the stiffness of material “A”, and  $E_B$  represents the stiffness of

material “B”. Notice that if we model the material phase “B” as void, and set  $E_B = 0$  we recover the original equation from 2.6. This method uses a single design variable field to model up to two different materials.

For a three-phase or more material optimization problem, we require an extended power law interpolation with multiple design variable fields (i.e.  $\mathbf{s}_1$ ,  $\mathbf{s}_2$ , etc.), as shown in [89, 115]. In general, the SIMP method requires  $(n - 1)$  design variable fields for  $n$  distinct material phases [115].

### 2.3.2 Density methods applied to fluid dynamics

Several applications require finding the optimal geometries of systems to improve the performance of internal and external flows [66]. Adopting the concept of density methods, [13] extended the methodology to fluid related problems. They modeled the influence of a wall or body in the fluid flow by representing it as a body force in the incompressible Navier-Stokes equations as:

$$\rho^f \left( \frac{\partial v_i}{\partial t} + v_j \frac{\partial v_i}{\partial x_j} \right) = \frac{\partial}{\partial x_j} \left( -p \delta_{ij} + \mu^f \left( \frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j} \right) \right) + f_i \quad (2.8)$$

$$\frac{\partial v_i}{\partial x_i} = 0 \quad (2.9)$$

where  $\rho^f$  is the fluid density,  $v_i$  is the velocity along the spatial dimension  $i$ ,  $p$  is the fluid mechanical pressure,  $\mu^f$  is the fluid dynamic viscosity, and  $f_i$  is the force exerted by the porous media:

$$f_i = -\alpha v_i \quad (2.10)$$

This methodology is referred to as the Brinkman penalization. Similar to structural topology optimization problems, we set the design variables,  $\mathbf{s}$ , to represent the fluid fraction at a point of the design domain, and set  $\mathbf{s} = \gamma = \gamma(\mathbf{x})$ , where  $(0 \leq \gamma \leq 1)$ . Typically,  $\gamma = 1$  represents the fluid domain, and  $\gamma = 0$  represents the solid domain.

The coefficient  $\alpha$  can be interpolated from the design variables as:

$$\alpha = \alpha_{max} \gamma \quad (2.11)$$

The parameter  $\alpha_{max}$  should be large enough such that the term  $\mathbf{f}$  in Equation 2.8 sufficiently penalizes the flow to  $\mathbf{u} = 0$ . [50] set  $\alpha_{max}$  to:

$$\alpha_{max} = \left(1 + \frac{1}{Re}\right) \chi \quad (2.12)$$

where  $Re$  is the Reynolds number, and  $\chi$  is set to a very large value, i.e.  $10^4$ .

However, this linear interpolation causes large gradients in the fluid flow, which cause numerical issues and may lead to the optimization problem converging to a local minimum. [13] introduced a convex interpolation:

$$\alpha(\mathbf{x}) = \alpha(\gamma(\mathbf{x})) = \alpha_{max} + \gamma(\alpha_{min} - \alpha_{max}) \frac{1 + \alpha_p}{\gamma + \alpha_p} \quad (2.13)$$

Figure 2.3 shows the influence of the perimeter  $\alpha_p$ . In general, we want to choose  $\alpha_p$  to be as low as possible, but high enough to prevent intermediate porosities from showing up in the optimization process. In the work of [52],  $\alpha_p$  was chosen to be 0.01 with favorable results.  $\alpha_{min}$  is set to zero, such that at its minimum,  $\alpha = 0$  recovers the original Navier-Stokes equations. For more details on topology optimization of Stokes and Navier-Stokes flows, the reader is referred to [66].

### 2.3.3 Discussion of the SIMP approach

The concept of relating some artificial densities to the stiffness in structural problems can be expanded to other physics disciplines. The SIMP method can be used to describe material properties in thermal conductivity, magnetic permeability, porosity, etc.; and therefore, it found its way to a wide range of applications. The method requires a relatively small

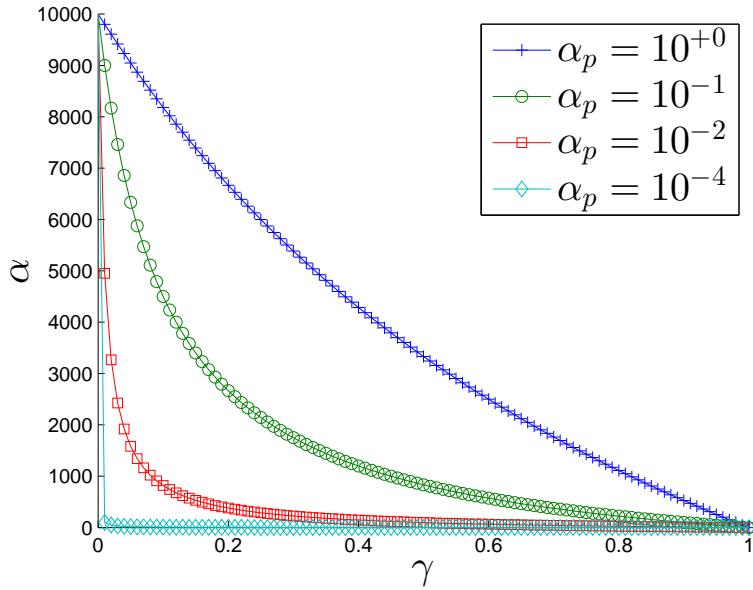


Figure 2.3: Influence of the interpolation penalty  $\alpha_p$ .

amount of iterations in order for the optimization problem to converge to an optimal design (at least for “solid-void” problems). SIMP has this capability because it operates on the entire design domain and not only on the boundaries; therefore, it does not suffer from localization effects. The approach is also suitable for a wide combination of design constraints, multiple load conditions, and extremely large (often 3D) systems. The educational article by Sigmund [97] detailing a 99-line SIMP-code implemented in Matlab, as well as his web-based topology optimization program [110]<sup>1</sup> played an important role in the acceptance of the SIMP method in both the academic and industry communities. Virtually all industrial optimization software uses the SIMP method as their optimization method of choice due to its ease of implementation.

The SIMP method typically describes the interface between the different material domains either by using intermediate densities or by discrete material distributions, which may lead to jagged boundaries. In both cases, the representation of the interface is not precise, and therefore, the enforcement of boundary conditions at the interface is hindered. This may result in non-physical responses, such as premature yielding [69] in structural me-

---

<sup>1</sup>The software is now available on iOS phones and tablets.

chanics, fluid flow penetrating solid material in low Reynolds number flow [54], and scalar fields diffusing through solid material at low Péclet number flow [65]. This issue can be mitigated by representing the material interface more accurately either by mesh refinement or adaptive re-meshing [67, 68]. However, for problems that require an accurate geometrical description of the interface, such as stresses in elasticity, boundary layer problems in fluids and skin-depth issues in electromagnetics, SIMP (and other material interpolation methods) will fail due to the jagged edges obscuring the physics [27, 126].

Using material interpolation to address multi-material optimization problems is not a physics-based technique, and has been shown to violate the Hashin-Shtrikman bounds for low values of  $\rho$  and large values of  $p$  [45]. Furthermore, modeling multiple material phases can become complicated [128] and lead to slow convergence due to the larger number of iterations, as shown in [113].

## 2.4 Level set method

The level set method applied to topology optimization arose as a technique capable of overcoming some of the shortcomings of the density approach. The main advantage of the method is that it allows for the description of complex geometries and the variation of the shape and topology of our design without introducing intermediate materials. A level set approach is a *region-based* model with explicit boundaries, in contrast to the *pixelated* model of the density method [115].

The level set method was first used to implicitly represent a structural boundary by [88]. Since then, it has been applied extensively in the field of imaging and computer visions [86]. The level set method eventually found its way to topology optimization, as the technique is well suited for the task: level set functions can form holes, split into multiple pieces, or merge with other level set functions [2, 88, 115].

The basic idea behind using the level set method to represent a shape boundary  $\Gamma = \partial\Omega$

is to express a curve or surface as the zero level set of a higher dimensional implicit function  $\phi$ , as shown in Figures 2.4b, 2.4d, and 2.4f, such that:

$$\Gamma = \{\mathbf{x} : \phi(\mathbf{x}) = 0\} \quad (2.14)$$

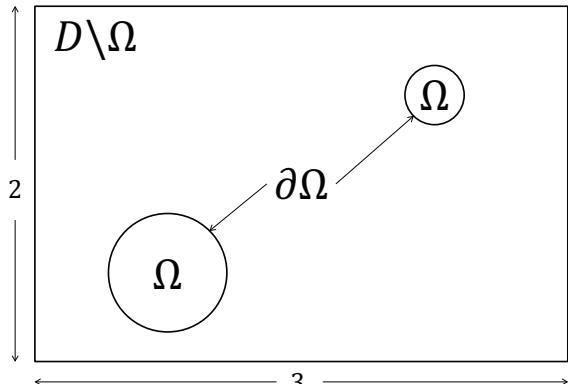
Then we can divide the design domain into phase regions as:

$$\begin{aligned} \phi(\mathbf{x}) &> 0 \quad \forall \mathbf{x} \in \Omega \setminus \partial\Omega \text{ (inside the region)} \\ \phi(\mathbf{x}) &= 0 \quad \forall \mathbf{x} \in \partial\Omega \text{ (on the boundary)} \\ \phi(\mathbf{x}) &< 0 \quad \forall \mathbf{x} \in D \setminus \Omega \text{ (outside the region)} \end{aligned} \quad (2.15)$$

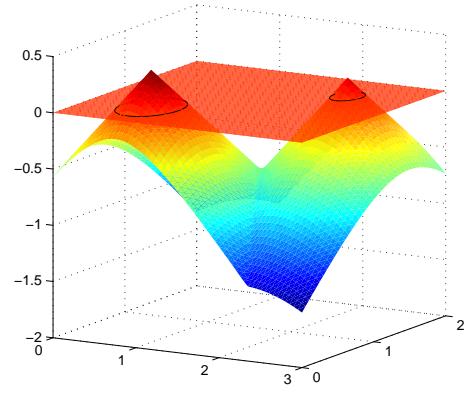
where  $D$  represents the design domain, either bounded or unbounded, and contains all possible admissible shapes of  $\Omega$ , as shown in Figures 2.4a, 2.4c, and 2.4e. Each phase region may represent a different material [2, 87, 95, 115] or a different physics [37, 59]. In the work of this thesis, the level set method is discretized with a finite element mesh.

Multiple level set functions can be used to model more than two phase regions 2.15. As with the level set method itself, the use of multiple level set functions originated in image processing [112]. This so-called “color” level set method requires  $m$  level set functions to model  $n = 2^m$  different phase regions. For a reference on the method, the reader is referred to [115, 116].

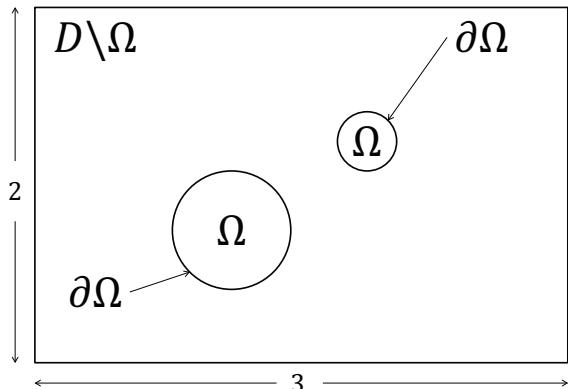
The material distribution in the design domain can be determined from the phase region of the level set field. Several methods exists to describe this distribution. Density based level set methods describe the phase regions by either using element-wise constant material fractions or by mapping the level set field directly to a point [123]. These techniques are denoted as Ersatz material approaches (see Figure 1.4) [4, 117], and while they ease the computational complexity, they lead to modeling errors. In this work, we will triangulate the element cut by the zero level set isolevel, and then perform our operations on the subdomains and on the interface (see Figure 1.6) [64, 113]. This methodology will avoid the need to



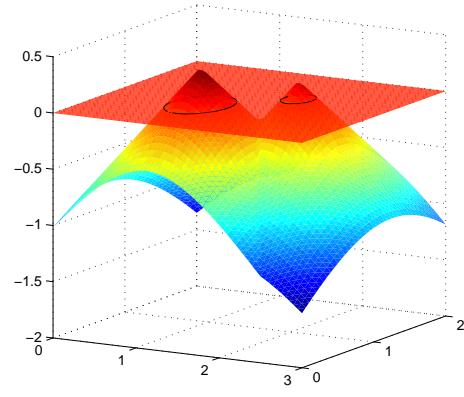
(a)



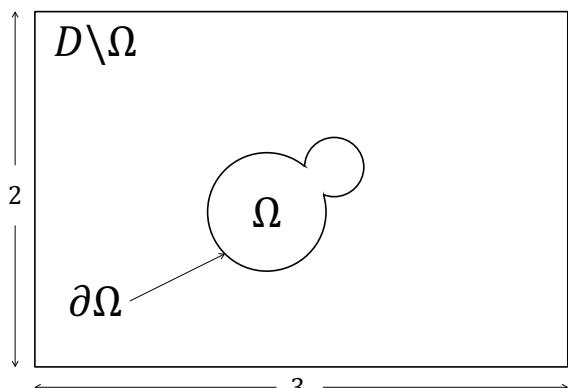
(b)



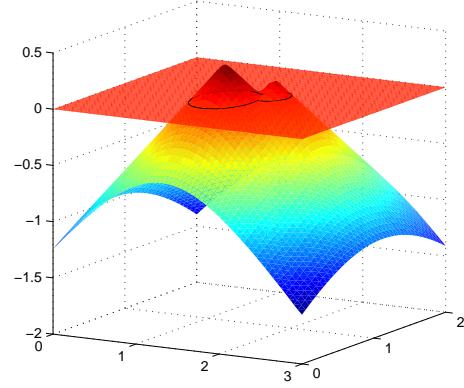
(c)



(d)



(e)



(f)

Figure 2.4: Level set description of two circular inclusions with radii 0.667 and 0.333, respectively, moving towards each other. Level set functions can be used to describe complex topologies in a fixed mesh.

approximate material properties such as in density methods.

Regularization techniques are used in a level set optimization problem to control the geometry of the design. Among these techniques are perimeter or curvature minimization [22, 70, 124]. The advantage of the level set method is that we possess a crisp definition of the phase interface, and therefore can integrate over the shape boundary  $\Gamma_{\phi=0}$ . These techniques will be studied in Section 6.

### 2.4.1 Topology optimization with the level set method

The topology of the level set field is modified by update schemes that use the sensitivities of the design variables. Several approaches exist, such as the Hamilton-Jacobi equation [125]. This work will focus on a mathematical programming approach, where the nodal values of the discrete level set field are defined as functions of the optimization design variables. Like in the filtering method of density approaches, we will define a linear filter

$$\phi(\mathbf{s}) = \frac{\sum_{i=1}^N w_i s_i}{\sum_{i=1}^N w_i} \quad (2.16)$$

with

$$w_i = \max(0, r_\phi - \|\mathbf{x}_i - \mathbf{x}\|) \quad (2.17)$$

where  $\phi(\mathbf{s})$  is the level set function at a point  $\mathbf{x}$ ,  $\mathbf{x}_i$  is the location of the node at which the design variable  $i$  is defined,  $w_i$  is the factor of point  $\mathbf{x}$  with respect to the design variable  $i$ ,  $r_\phi$  is the filter radius, and  $N$  is the number of nodes in the design domain. However, unlike the filter in the density approaches, this filter cannot control the minimum feature size, as discussed in Section 4.6.2.

The reader is referred to [23, 35] for a more detailed overview of the level set method and topology optimization approaches.

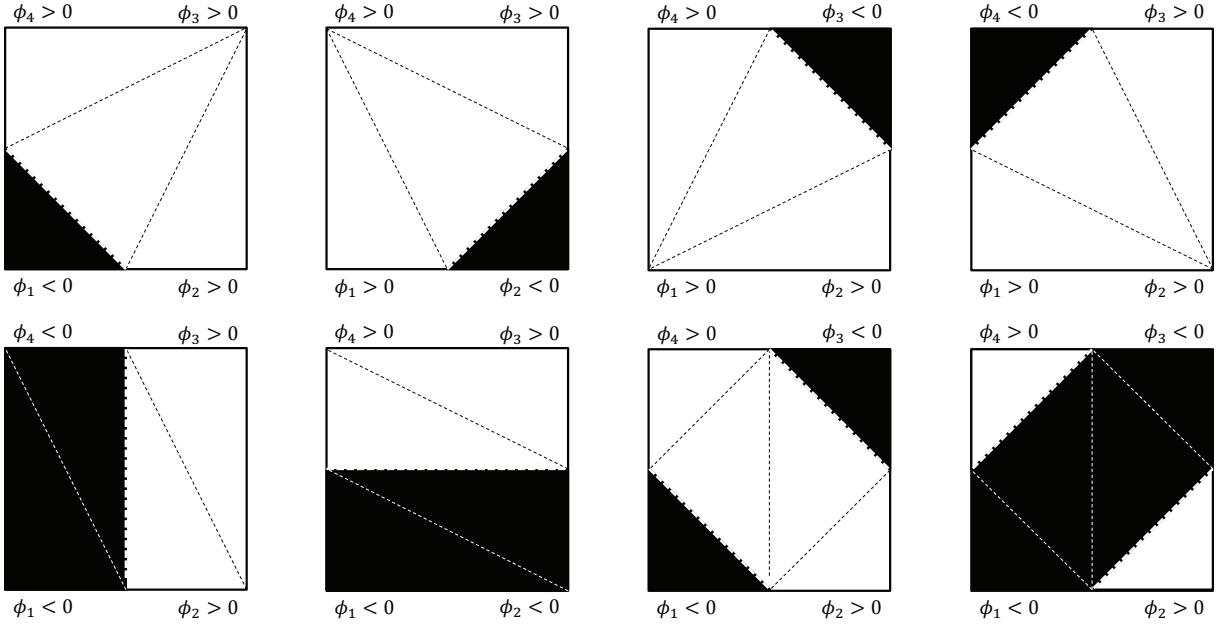


Figure 2.5: A two dimensional finite element has eight possible decompositions based on the nodal level set values.

## 2.5 Extended finite element method

The extended finite element method (XFEM) is an immersed boundary technique that works on fixed meshes. The XFEM was built upon the concept of partition of unity developed by [5], and it was originally used to model crack propagation [6]. Due to its ability to approximate discontinuous solutions, the method has been applied in a variety of discontinuous partial differential equations, such as multimaterial structural mechanics [113, 115], fluid-structure interaction [37], and multi-phase flows [19]. The method has also been applied to shape optimization by [25, 73, 74], and to topology optimization by [47, 60, 63, 70, 113, 120].

The XFEM decomposes the cut elements into subdomains and interfaces that it uses to integrate the weak form of the governing equations. Figure 2.5 illustrates this by showing the possible decompositions of a two dimensional finite element based on its nodal level set values.

The XFEM can model discontinuities in the solution field by augmenting the standard finite element function space with additional degrees-of-freedom, denoted “enriched degrees-

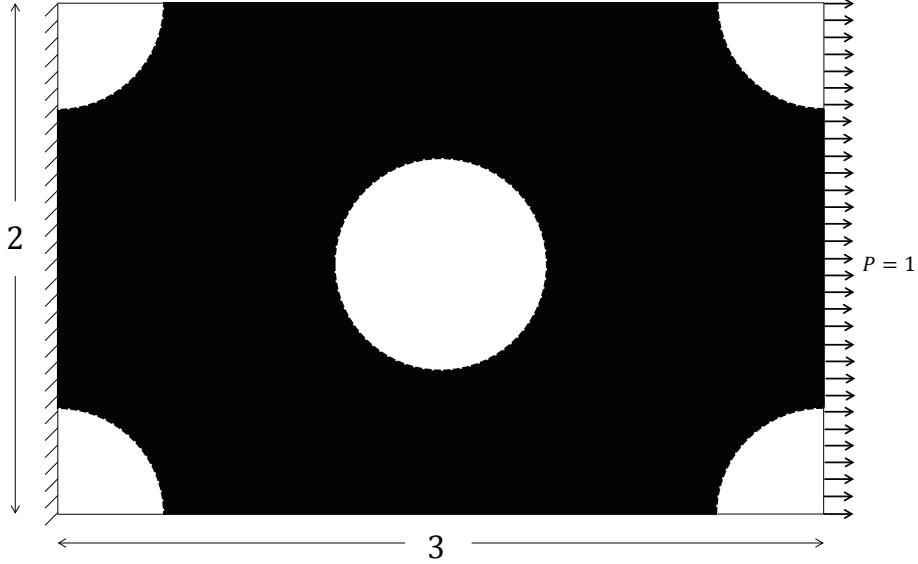


Figure 2.6: Structural problem setup. The domain contains multiple level set inclusions.

of-freedom”. To illustrate this with a quick example, consider an XFEM model that consists of a two dimensional mesh with four linear elements. The level-set distribution in Figure 2.6 leads to the intersection pattern shown in Figure 2.7. The nodes on the left are clamped, and the right edge is subject to a constant pressure load. The node at the center of the mesh in Figure 2.7 will use different degrees-of-freedom to interpolate the different subdomains, and avoid artificially coupling the disconnected phase regions.

The solution space is then interpolated using the generalized enrichment formulation of [42]:

$$\mathbf{u}(\mathbf{x}) = \sum_{m=1}^M \left( H(-\phi) \sum_{i=1}^n \mathbf{N}_i \mathbf{u}_{i,m}^A + H(\phi) \sum_{i=1}^n \mathbf{N}_i \mathbf{u}_{i,m}^B \right) \quad (2.18)$$

where  $m$  is the enrichment level,  $M$  is the maximum number of enrichment levels used for each phase,  $\mathbf{N}$  are the shape functions,  $\mathbf{u}_{i,m}^l$  is the vector of degrees-of-freedom values at node  $i$  for phase  $l = [A, B]$ ,  $\phi$  is the level set value evaluated at  $\mathbf{x}$ , and  $H$  denotes the Heaviside function. This is illustrated in Figure 2.8 for the structural problem of Figure 2.6.

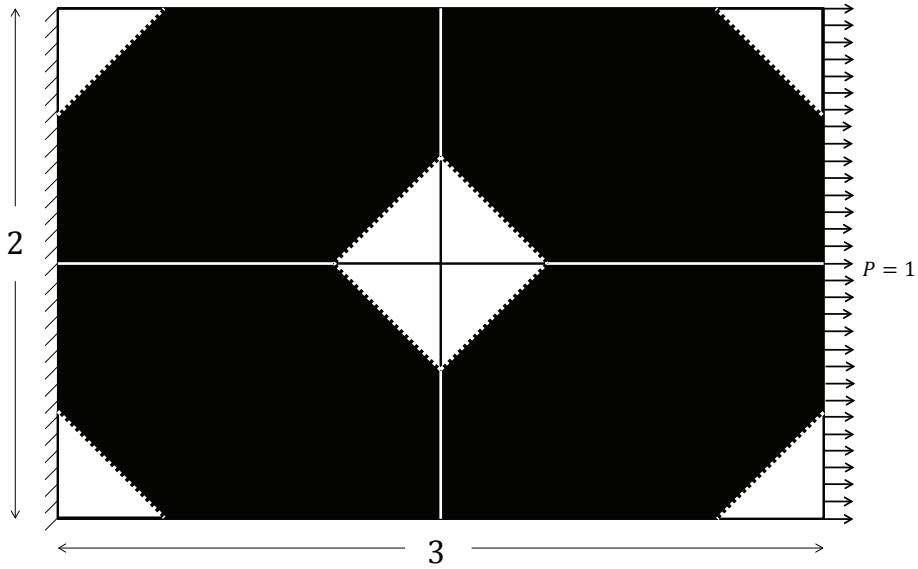


Figure 2.7: 4-element 2D mesh. Black areas: material phase 1, negative level-set value at the nodes; white areas: material phase 2, positive level-set value.

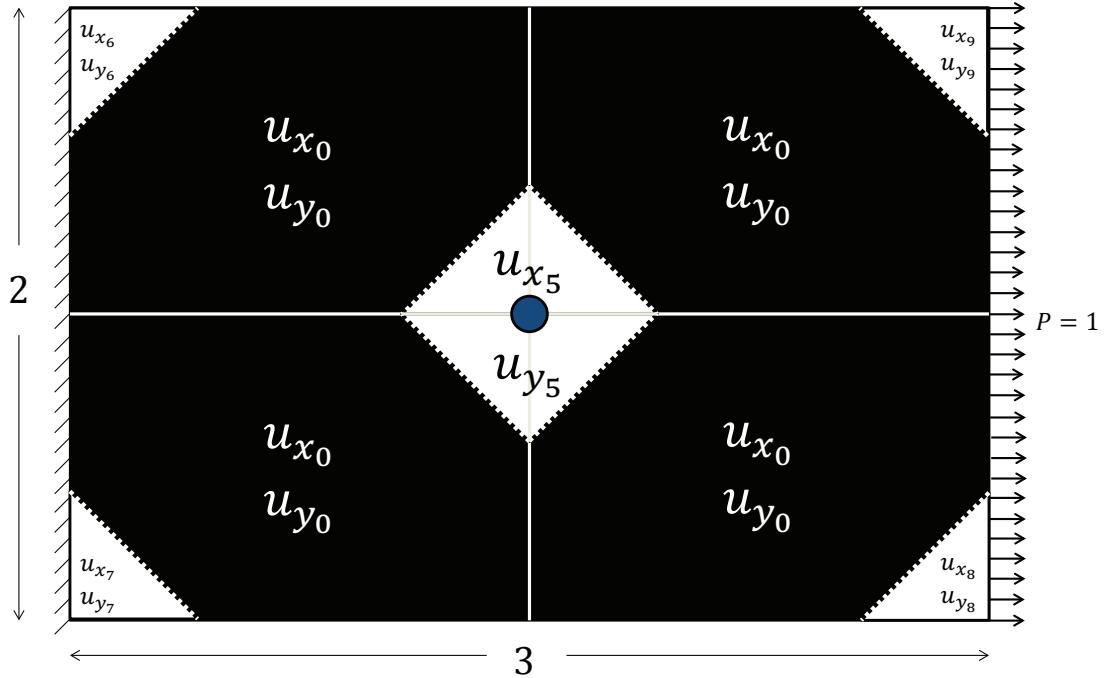


Figure 2.8: The center node, denoted by the color blue, uses different degrees-of-freedom to describe the disconnected phase regions. The subscripts denote the  $m$  enrichment level. The maximum number of enrichment levels used for each phase is  $M = 5$ . A value of 0 denotes the original finite element degrees-of-freedom, while other numbers indicate additional “enriched degrees-of-freedom”.

The Heaviside function  $H$  depends on the level set function and is defined as follows:

$$H(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases} \quad (2.19)$$

The Heaviside functions “turns on/off” the standard finite element interpolations in the particular phases. The approximation allows for discontinuities of the states  $\mathbf{u}$  along the phase boundaries. Therefore, the continuity of the solution at the interface between different phase regions must be enforced. Several techniques are available in the literature, such as stabilized Lagrange multipliers (Section 4) [16], and the Nitsche method (Section 5) [17]. More recent development include the face-oriented ghost penalty (Section 5) by [17, 93, 94], which aims at smoothing the gradients of the solution between cut elements. All three formulations will be studied and applied to topology optimization in this work.

For more details, refer to Sections 3, 4.3.2, and 4.5. For a review of the XFEM, the user is referred to [32].

# **Chapter 3**

## **A complete methodology for the implementation of XFEM inclusive models**

This document was written in 2012 as an internal implementation manual for future developers of our code, the Finite Element Multi-Disciplinary Optimization Code (femdoc).

This report offers a background into the eXtended Finite Element Method as a tool to solve the shortcomings of the classical Finite Element Method. An example of such is the numerical solution to problems with different material topologies (i.e. discontinuities). The XFEM uses level set functions to track the location of these discontinuities. The report also provides algorithms for locating these discontinuities and subsequently dividing the domain into sub-domains capable of integration. This report ultimately expounds upon how to effectively apply the local enrichment functions that the XFEM standard approximation requires at the nodes where the discontinuities occur.

The reader may skip this section, as it focuses on the algorithmic implementation of the framework and most of its content is intended for software developers.

## 3.1 Introduction

### 3.1.1 Finite element method

The Finite Element Method (FEM) is a numerical technique used for finding solutions to partial differential equations as well as to integral equations. Traditional finite element methods (FEM) requires meshing techniques that generate discrete representation of potentially complex geometry. Difficulties arise when using the traditional FEM for analyzing fracture mechanics: under traditional FEM, introducing a discontinuity in the mesh, changing the material topology or drastically changing the shape of the material requires a new mesh to ensure that the element edges align with the discontinuity [1]. The process is laborious and difficult [130]. The eXtended Finite Element Method (XFEM) arose as a solution to this impediment by applying enrichment functions at the position of the material interface or topology discontinuity instead of re-meshing the entire structure.

### 3.1.2 Discontinuity

A discontinuity can be defined as a rapid change in a field variable over a length negligible in size compared to the entire domain analyzed [1]. For example, in solid mechanics, strain and stress fields are discontinuous across material interfaces and displacements are discontinuous at cracks; in fluid dynamics, velocity and pressure fields are discontinuous at the boundary layer between two fluids. A discontinuity is classified as “weak” or “strong”. Weak discontinuities happen when a field variable (stress field, strain rate, etc.) has a kink, meaning the derivative has a jump. This can happen at boundary layers, or at a material or fluid interface. Strong discontinuities happen when the field quantity has a jump; this could include, for example, a crack in the structure [43].

### 3.1.3 eXtended Finite Element Method

The XFEM arose as a numerical technique capable of providing local enrichment functions at the position of the material interface, avoiding the need to re-mesh the entire structure while finding solutions for the discontinuous functions [32]. By doing this, XFEM appears to solve the shortcomings of the FEM by providing accurate solutions for complex problems in engineering that would be impossible to solve otherwise [1].

### 3.1.4 Level set method

Level set functions are used to model the motion of these discontinuities in the elements. A level set function is a numerical scheme where the discontinuity of interest is represented as the zero level set function. Basically, a level set function has a value of zero at the boundary of its closed curve and opposite signs on the interior and the exterior (Figure 3.1).

$$\begin{aligned}\phi(\mathbf{x}) &> 0 \quad \forall \mathbf{x} \in \Omega \setminus \partial\Omega \text{ (inside the region)} \\ \phi(\mathbf{x}) &= 0 \quad \forall \mathbf{x} \in \partial\Omega \text{ (on the boundary)} \\ \phi(\mathbf{x}) &< 0 \quad \forall \mathbf{x} \in D \setminus \Omega \text{ (outside the region)}\end{aligned}\tag{3.1}$$

The XFEM uses this method by placing the discontinuity at the boundary layer and giving the interface caused by the division positive and negative values. Since the zero level is interpreted as the discontinuity, new nodes (called pseudo-nodes) are created at this position and the enrichment of the FEM is produced at this location. This creates an advantage because instead of re-meshing the entire structure, a fixed Cartesian grid is used to divide the structure and the discontinuity into domains capable of integration. Once this mesh is settled, the XFEM will subsequently use either branched or Heaviside functions to enrich the nodes and solve for the problem.

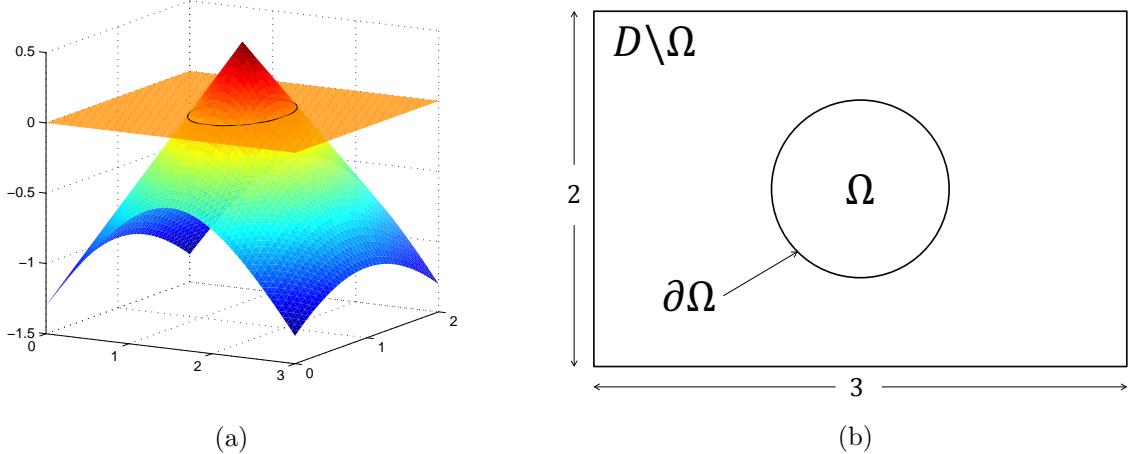


Figure 3.1: The zero level set isolevel of the level set function,  $\partial\Omega = \Gamma_{\phi=0}$ , in 3.1a divides the fixed mesh grid into different phase regions in 3.1b, where each phase may represent a different material or a different physics.

The Level Set Method (LSM) and the XFEM have a sort of natural coupling to solve problems with discontinuities. While the LSM is used to model the discontinuity and update its motion at each calculation, the XFEM is used to solve the problem and determine the direction of the discontinuity [105].

### 3.1.5 Delaunay triangulation

In order to apply the LSM and the XFEM to solve a problem, a framework for dividing arbitrarily complex geometries into integrable domains must be developed. The Delaunay triangulation is a critical step in this process. The Delaunay triangulation is the subdivision of a geometric object into triangles for 2D geometry and tetrahedra for 3D (Figure 3.2). This particular triangulation has the property that the circumcircle of any triangle in the triangulation does not contain the vertices of other triangles or its own in its interior (Figure 3.3) [57]. Because triangle and tetrahedra are integrable elements, the XFEM method can then be applied.

For an inclusion-based XFEM model (inclusion meaning the zero level function is always

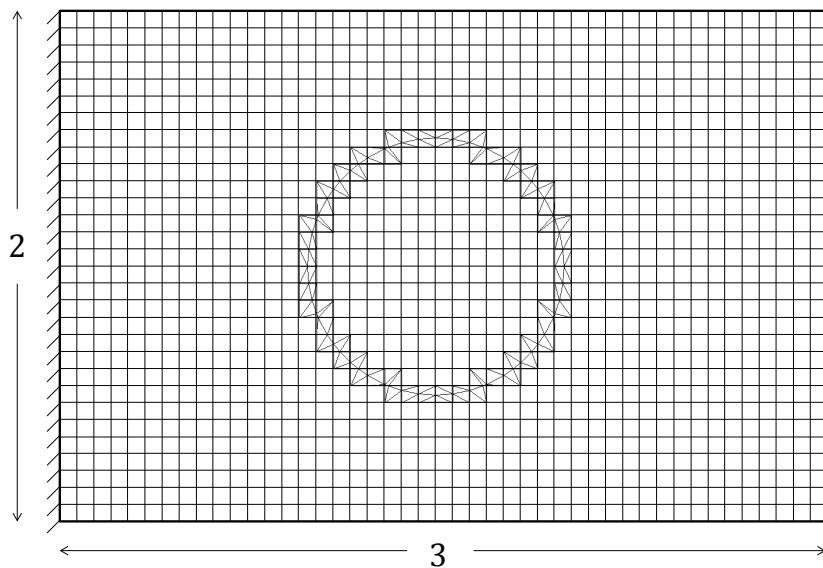


Figure 3.2: The Delaunay formulation triangulates QUAD4 finite elements cut by the level set zero isolevel from Figure 3.1b into TRI3 elements.

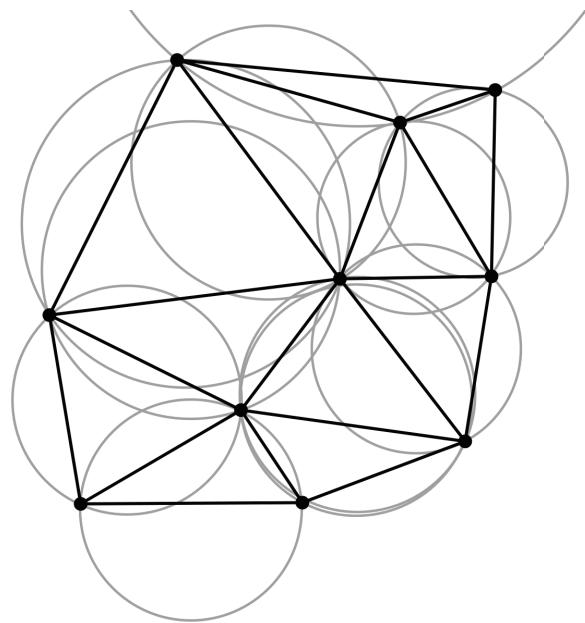


Figure 3.3: Delaunay circumcircles - A set of points can be uniquely triangulated in a way that the points form circumcircles.

a closed curve), there are only 8 triangulation configurations in 2D, while 127 different ones in 3D. Due to the low number of cases in 2D, a tabulation of the triangulation is performed instead of using the Delaunay triangulation. Figure 3.4 shows the different cases for 2D, while Figure 3.5 shows the triangulation of a 3D element with four different discontinuities.

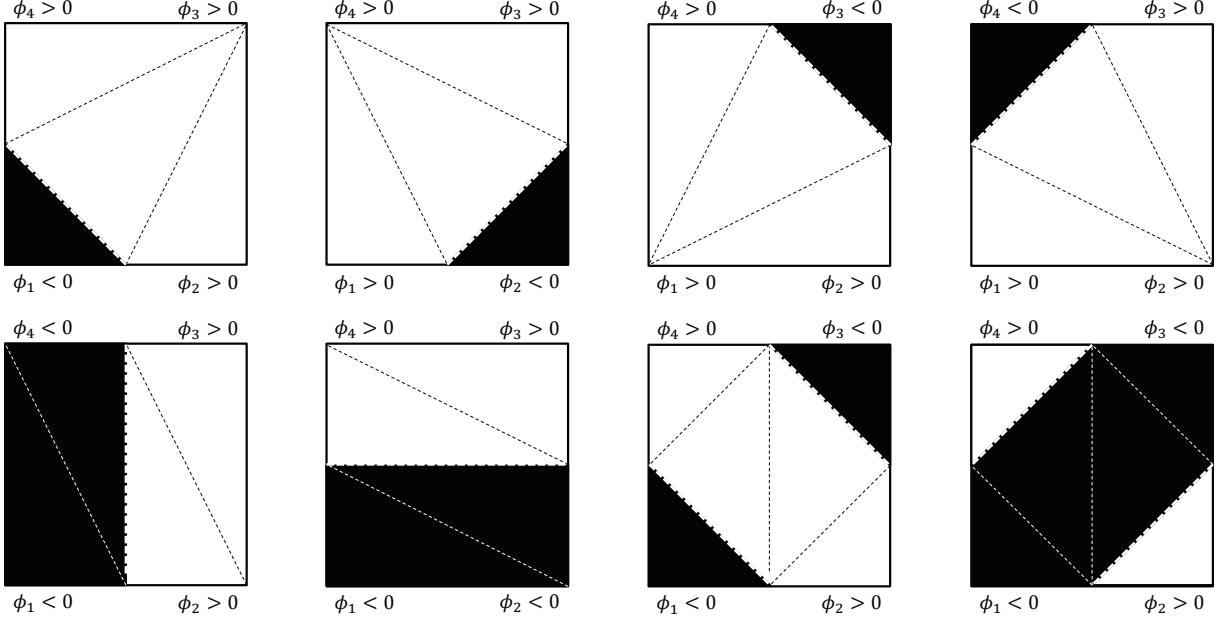


Figure 3.4: There are only 8 different triangulation configurations for an inclusion-based XFEM model.

## 3.2 Implementation

### 3.2.1 Summary

This document outlines the procedure for building an XFEM model for a given distribution of the level-set function. The XFEM model consists of:

- Intersection points along elemental edges.
- XFEM elements sub-divided into cells for integrating the weak form of the governing equations within the individual sub-domains belonging to a particular material phase.

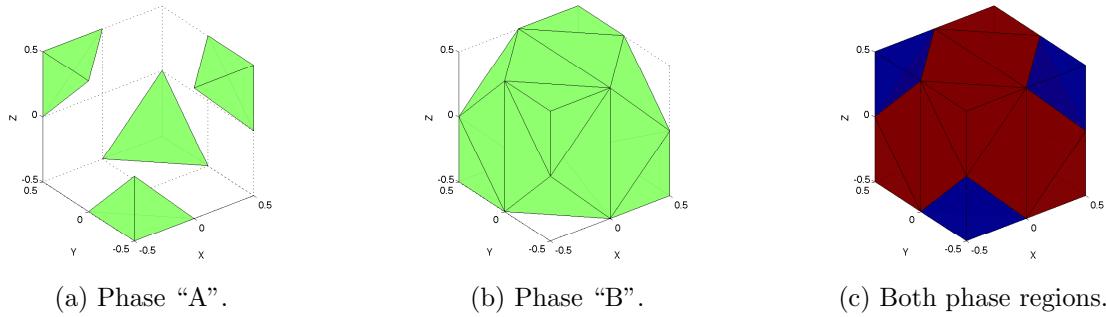


Figure 3.5: Triangulation in 3D is more complex and we use the Delaunay triangulation algorithm to perform the computation. This element with 4 discontinuities has 4 pseudo-elements from material phase 1 and 20 from material phase 2.

- Enrichment tables that define the nodal enriched degrees of freedom used to interpolate the solution within a cell.
  - Parallel implementation of building XFEM model.

### 3.2.2 Glossary

**computational mesh** – standard FE mesh that defines the nodal degrees of freedom.

**model** – physical entity, contains information about the XFEM elements and the level-set functions.

**main phase (phase)** – phase indicating a particular material phase.

**sub-phase** – the domain of a main phase can be decomposed into multiple sub-phases.

**intersection point** – intersection created by the zero level-set curve cutting through an edge.

**point** – geometrical entity with information about coordinates, connected cells and edges.

**cell** – geometrical entity, a collection of points, owns a list of edges too.

**edge** – geometrical entity with information about the points on its ends and its connected cells.

**Delaunay triangulation** – triangulation of our elements using their corner nodes and in-

tersection points.

**pseudo-element** – cells created by the triangulation of the regular element.

**nodal cluster** – set of elements (and their nodes) connected to a node consistency nodes nodes shared by multiple elements within nodal cluster.

### 3.2.3 Procedure overview

The main steps are:

1. Build point-to-cells connectivity list (list of cells connected to a point) by looping over all cells; needs to be built only once.
2. Build edge table in mesh (list of the cell edges that stores connectivity to points and cells) by looping over all cells; needs to be built only once.
3. Build table of nodes belonging to a nodal cluster (first-order neighbors of a node; defined as all nodes belonging to elements connected to a node) by looping over all elements for a node using point-to-cell table; needs to be built only once.
4. Build edge intersection points by looping over all edges; points are stored in mesh; however, the coordinates of the intersections are copied to the XFEM element; needs to be built for each instance of a level-set distribution.
5. Delaunay triangulation of each cell based on edge intersection; needs to be built for each instance of a level-set distribution.
6. Build table of phases and sub-phases for each triangle/tetrahedron (pseudo-cells) for each triangulated element; needs to be performed for each instance of level set distribution.
7. Build enrichment table that defines which nodal degrees of freedom are used to interpolate a field within a pseudo-element.

- Determine which degrees of freedom are used in the model.

### 3.2.4 Implementation algorithms

Consider the following XFEM model which consists of a 4-element mesh in 2D; the nodes on the left are clamped and the right edge is subject to a constant pressure load. The level-set distribution in Figure 3.6 leads to the intersection pattern shown in Figure 3.7. The mesh in Figure 3.8 shows the indices of the nodes and the cells.

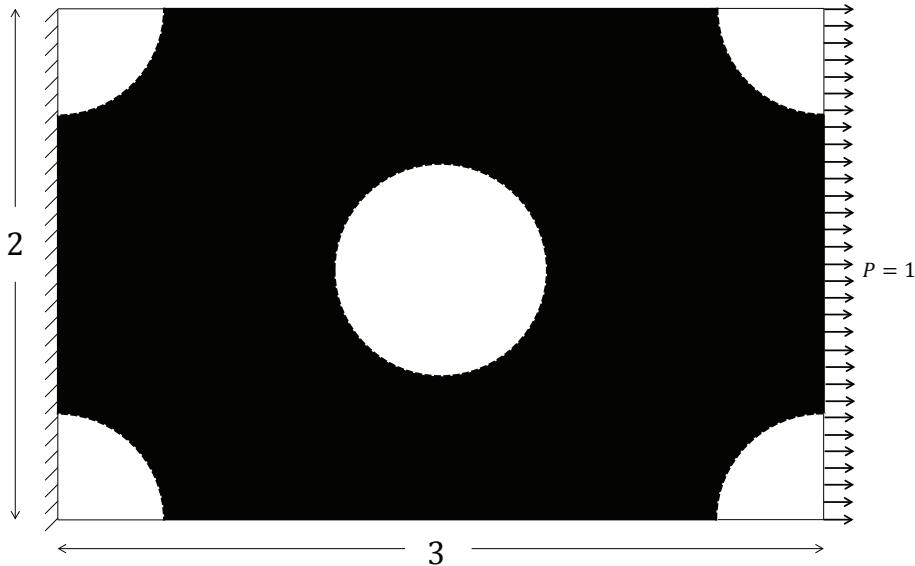


Figure 3.6: Structural problem setup. The domain contains multiple level set inclusions.

#### Point-to-cells connectivity table

To build a point-to-cell table for each point in our computational mesh we loop over all base cells in the computational mesh (base cells are all cells that are not side-set cells). For each base cell we loop over all points and store the current cell index with point index. This leads to the point-to-cell connectivity table (see Table 3.1):

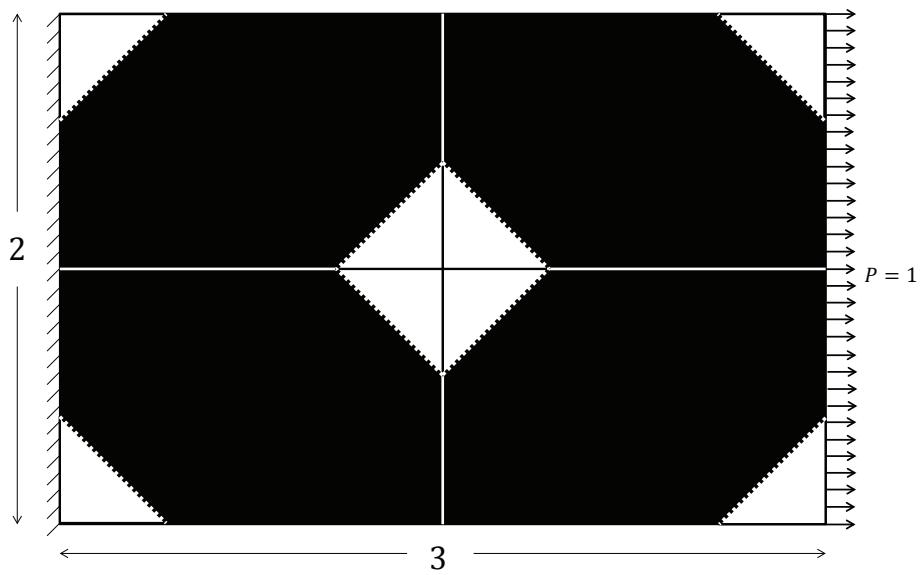


Figure 3.7: 4-element 2D mesh. Black areas: material phase 1, negative level-set value at the nodes; white areas: material phase 2, positive level-set value.

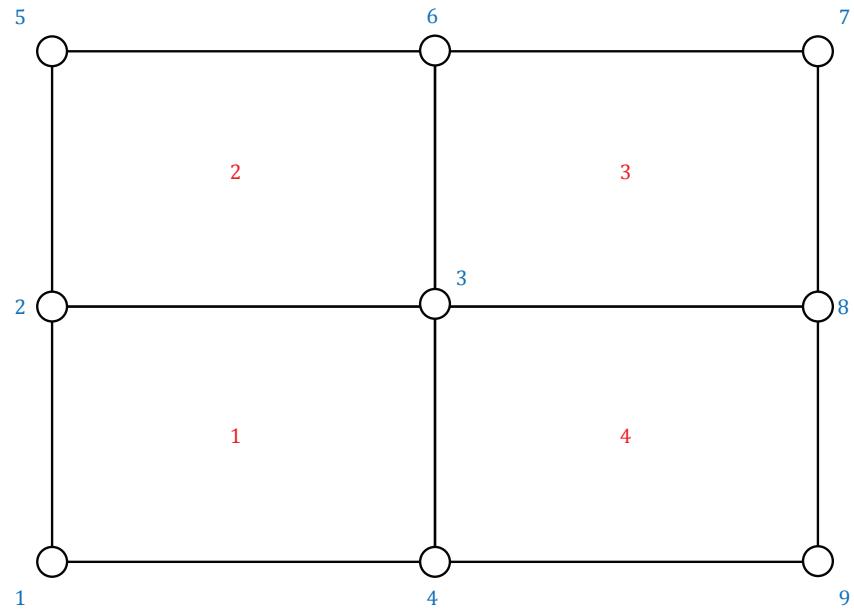


Figure 3.8: 4-element 2D mesh. Red numbers represent the global element identifiers, blue numbers represent the global node identifiers.

Point Id	Number of cells connected	Cell Ids
1	1	1
2	2	1,2
3	4	1,2,3,4
4	2	1,4
5	1	2
6	2	2,3
7	1	3
8	2	3,4
9	1	4

Table 3.1: Point ID to cell IDs connectivity table

## Edge table

To generate an edge table in our computational mesh we initially loop over all base cells. For each cell, we determine the number of edges and loop over all edges in the cell. For each edge, we check with the current element whether the edge has been created. In case the edge does not exist yet, we store the following edge information:

- Ids of end point of edge.
- Ids of cells to which element is connected.

We determine the cells which are connected to an edge via the intersection of cells connected to the end points of the edge, using the point-to-cell table. The following edge table is stored with the computational mesh.

At this point, each global edge knows the point Ids on its ends and the cell Ids of the elements it is connected to. We can use this information to create a map that links the global edge to the local edge index per element. Each element has an internal edge order list pre-built that indicates the order in which its edges are organized. For example, a QUAD4 element will have the following internal edge order list: 0 1, 1 2, 2 3, 3 0. The element also knows the point Ids that it owns (if not directly, point Ids can be obtained through the nodes). Using these two lists, we can compute which point Ids lay in each of the element's internal edges. By matching the point Ids of the global edge to the point Ids of the internal

element edges, we can make a map that tells us which internal edge in an element corresponds to a global edge. This list is stored with the edge in the same manner that cell IDs are stored. Cell IDs and internal edge numbers should have a one to one correspondence.

For example, for the 4-element cluster above we would obtain 12 global edges (see Table 3.2).

Global Edge Id	Point Ids	Cell Ids	Local Edge Num
1	1, 4	1	0
2	3, 4	1 4	1, 3
3	2, 3	1 2	2, 0
4	1, 2	1	3
5	4, 9	4	0
6	8, 9	4	1
7	3, 8	3 4	0, 2
8	2, 5	2	3
9	5, 6	2	2
10	3, 6	2 3	1, 3
11	6, 7	3	2
12	7, 8	3	1

Table 3.2: Edge to points IDs and cells IDs connectivity table

The table shows, for example, that the global edge 2 is defined by the point IDs 3 and 4 and is connected to cells with IDs 1 and 4. For cell id 1, it is the local edge 1 (counting from zero CCW starting at the bottom) and for cell id 4 it is local edge 3 (Figure 3.9).

## Nodal clusters

To determine the enrichment level used to interpolate fields within the XFEM elements we need to determine the first-order nodal neighbors of a node. This is the set of nodes that belong to the elements connected to a node. In addition, we need to identify the nodes that belong to two or more elements; we refer to these nodes as consistency nodes. Consistency nodes are simply the nodes in a nodal cluster (nodes of the connected elements of a main node) that are shared by more than one element in the nodal cluster. This information can be obtained by looping over the connected elements of a node, and obtaining the node list

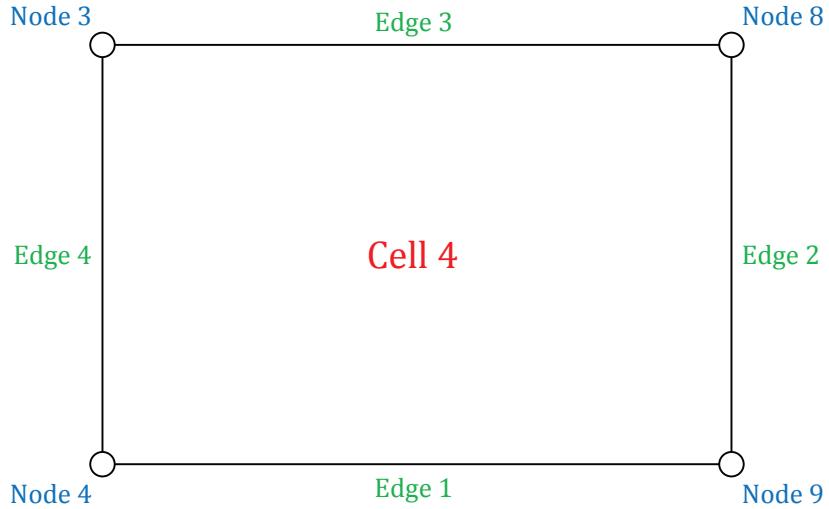


Figure 3.9: Edge representation in a QUAD4 element. Green edges represent the local edge index at the element.

for each element.

### Determine intersection points

The intersection points are defined by the zero level-set values. We loop over all edges defined in the edge table created in Step 2 and compute the intersection points along the edge using the nodal level-set information of the edge endpoints. The coordinates of the intersection points are then sent to all elements connected to the edge and stored in the corresponding XFEM element. (see Figure 3.10). NOTE: Since edges only know of the cells connected to it, we use a cell-to-element map in order to be able to send this information to the XFEM elements. This is owned by the model.

### Delaunay triangulation and assignment of main and sub-phases to pseudo-elements

We loop over all elements in the model and, if intersected, we perform a Delaunay triangulation, using the corner nodes and the edge intersection points stored with the elements in Step 3. The Delaunay triangulation requires only the coordinates of the element corner and edge intersection points. The Delaunay triangulation will return a list of triangles for

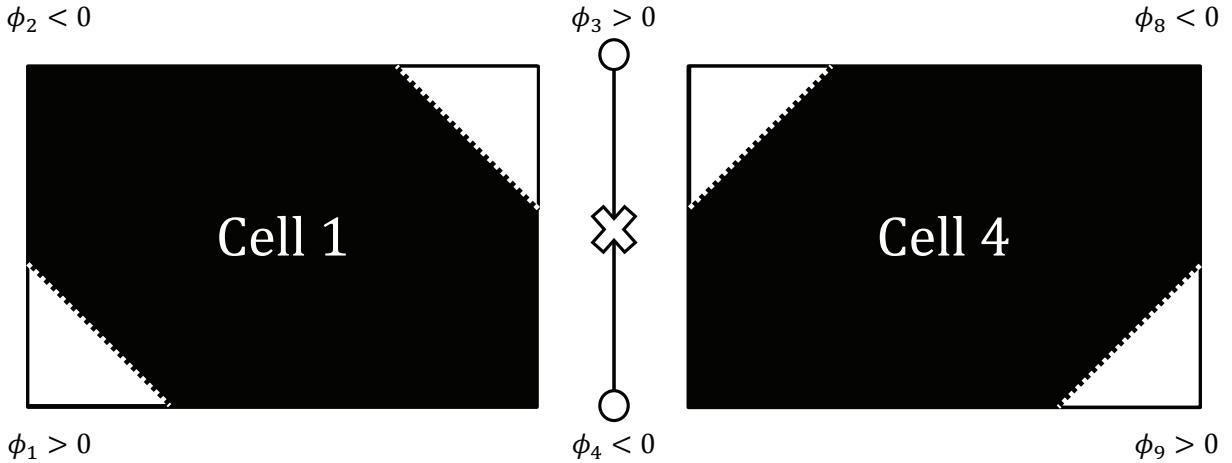


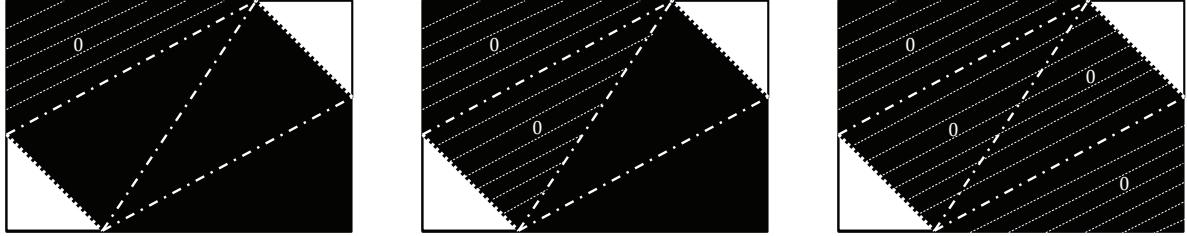
Figure 3.10: Mapping of the intersection points to the elements. An edge contains two nodes that have level set values of opposite sign. The location of the intersection point is labeled with an “X” in the figure. The information of the intersection point is sent to all the neighboring elements of the edge.

2D or tetrahedrons for 3D problems.

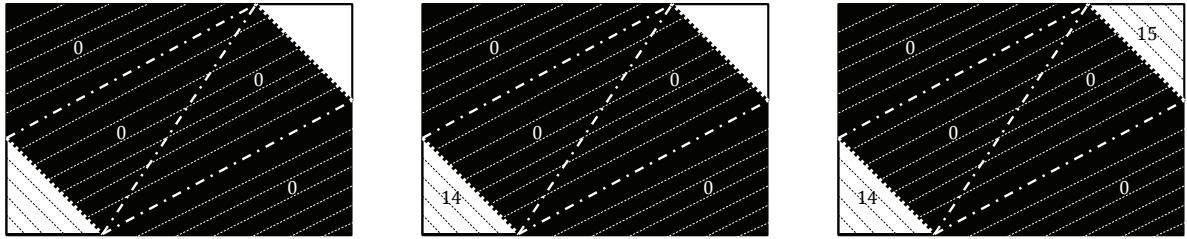
### Main phase and sub-phase

We assign a main and sub-phase to each triangle/tetrahedron. The main phase is determined based on the average main-phase value of the pseudo-element. In case the average is zero, we apply an exception rule (TBD). The sub-phase information is based on the connectivity of pseudo-elements which belong to the same main phase and is computed via a flood-fill algorithm. To this end we collect the pseudo-elements into a pseudo-mesh; each triangulated XFEM element has its own pseudo-mesh which consists of the points and the connectivity of the pseudo-elements. The main steps (Figure 3.11) of the flood-fill algorithm used are:

- Build edge table for pseudo-elements of current XFEM element, analogue to Step 3.2.4.
- Loop over all elements that have not been assigned a sub-phase:
  - Find unprocessed element; recursively find neighbors with same main-phase; assign lowest unassigned sub-phase to elements found in search process.



- (a) Start with the first subphase  
(b) Look for connected cells  
(c) Move on to the next cell in value of the first main phase. Do so through edges. If two cells share the list that shares the same main by selecting the first cell in the tri- the same main phase and are phase value, and continue checking angulation list without a value as- connected, then they have the the connectiv- signed.  
(d) Once the connectivity for the (e) Select the first subphase of the (f) Once the connectivity for a sub- specific subphase of the main phase second main phase. Repeat steps 1 phase is computed, increase the have the same main phase but dif- through 3. subphase value within the main ferent subphase. If there are not phase. Look for cells in the trian- any cells with the same main phase, gulation list that have not been move on the next main phase.



- (e) Select the first subphase of the (f) Once the connectivity for a sub- specific subphase of the main phase second main phase. Repeat steps 1 phase is computed, increase the have the same main phase but dif- through 3. subphase value within the main ferent subphase. If there are not any cells with the same main phase, move on the next main phase.

Figure 3.11: Subphase computation algorithm. Refer to section 3.2.4 for a more detailed description.

For Figure 3.11, we would get the list in table 3.3:

Triangle number	Main Phase	Sub-Phase
1	1	0
2	1	0
3	1	0
4	1	0
5	2	1
6	2	2

Table 3.3: Main-phase and sub-phase table for pseudo-elements.

We have 6 different triangles generated by Delaunay triangulation. By convention, the order of the triangles is determined after the triangulation so that all phase 1 ones are located at the top, followed by the phase 2 triangles. Each triangle is assigned a main phase and sub-phase (see Table 3.3); the assignment is stored, using a one-to-one map by the XFEM element.

### Nodal enrichments for pseudo-elements

To interpolate fields in the XFEM elements we need to determine which nodal enrichments are used within each pseudo element. The enrichments need to be chosen such that the interpolations are continuous across adjacent elements within the same main phase and unique within pseudo elements of the main sub-phase but topologically disconnected.

The main concept of the procedure is to clearly separate element-level and node-level operations. This separation enables the parallelization of the procedure. The first step is to determine the enrichment levels a particular node will use to interpolate fields in the triangulated elements with a particular sub-phase. This step is done by looping over all nodes. The result of this loop is a map that links the sub-phase information of an element to an enrichment level for each node of the element. In a second step we loop over all elements to update the enrichment levels of pseudo-element based on the map built previously.

Looping over all nodal clusters, we build the following node-element table. The entries in the table are the sub-phases at the nodes within each element. The consistency node numbers are marked by a “C”. Consistency nodes can be identified by nodes which have entries in more than one column; so they can be identified easily on the fly.

Two conditions, consistency and uniqueness, must be satisfied to ensure the assigned sub-phases are consistent across the nodal cluster. The consistency condition is satisfied if all sub-phases in a row are the same. The uniqueness condition is satisfied if the sub-phase of each set of connected nodes is unique in the cluster.

The initial node-element table (see Table 3.4) shows that the consistency condition is not

Nodes/Elements	1	2	3	4
1	14			
2 "C"	0	0		
3 "C"	15	14	14	15
4 "C"	0			0
5		15		
6 "C"		0	0	
7			15	
8 "C"			0	0
9				14

Table 3.4: Initial node-element table.

satisfied since node 3 is inconsistent. Also, nodes 5 and 7, for example, should be assigned a unique sub-phase. Therefore the uniqueness condition is also not satisfied. To ensure both conditions we build a second table and where we iteratively correct the sub-phase until all conditions are satisfied. Note that the consistency and uniqueness checks are not needed for a 1 element cluster. The correction procedure is as follows:

1. Initialize a list of all sub-phase possible; mark them as unused; initialize a list of checked nodes. Mark all nodes as unchecked; build list of consistency nodes.
2. Ensure consistency: Repeat the following steps until all consistency nodes are checked.
  - Select node: Start with the center node, which must be a consistency node (remember that this consistency and uniqueness check will not be applied for a one-element cluster). Otherwise select the first unchecked consistency node.
  - Select sub-phase: Select the lowest unused sub-phase. Mark the selected sub-phase as used (keep consistency of sub-phase to the respective main phase).
  - Identify connected consistency nodes and connected unique nodes:
    - For each element to which the selected node belongs, connected nodes have the same sub-phase as the selected node has for this particular element. Select the connected nodes which may be either consistency or unique nodes.

- In order to identify all connected consistency and unique nodes, search the connected elements for additional connected consistency and unique nodes.
- The above process requires recursively (a) searching for nodes within an element with the same sub-phase and (b) identifying elements that share consistency nodes. Note: the sub-phase id might change between elements if the sub-phases are not consistent yet for a consistency node.
- Assign sub-phase: Assign the selected sub-phase to all connected nodes identified in the search process described above. For any value that needs to be changed, flip sub-phases for that element. Mark connected nodes as checked.

Nodes/Elements	1	2	3	4
1	14 → 15			
2 “C”	0	0		
3 “C”	15 → 14	14	14	15→ 14
4 “C”	0			0
5		15		
6 “C”		0	0	
7			15	
8 “C”			0	0
9				14→ 15

Table 3.5: Flipping the enrichment levels to keep consistency.

3. Ensure uniqueness: Loop through the remaining unchecked nodes. For each element, collect the unique nodes with an unused sub-phase, assign the next unused sub-phase to these nodes and check the sub-phase as being used. Here nodes 1, 5, 7, and 9 are assigned the sub-phase 16, 17, and 18, respectively (node 1 was flipped initially, but it was not marked as checked). The outcome of the above correction procedure is Table 3.6:
4. Using the original and the corrected node-element tables we can build the map for the central point (here node ID 3). See Table 3.7. The rows correspond to the elements, the columns list the original sub-phases, and the entries are the enrichment levels used

Nodes/Elements	1	2	3	4
1	15			
2 "C"	0	0		
3 "C"	14	14	14	14
4 "C"	0			0
5		16		
6 "C"		0	0	
7			17	
8 "C"			0	0
9				18

Table 3.6: The result of the enrichment algorithm. Nodes that are shared across elements receive the same enrichment level.

by the central node. When building this table we check that the map is consistent within itself (the same sub-phases within each are assigned to the same enrichments).

For Node 3: Element / Enrichment level	0	14	15
1	0	15	14
2	0	14	16
3	0	14	17
4	0	18	14

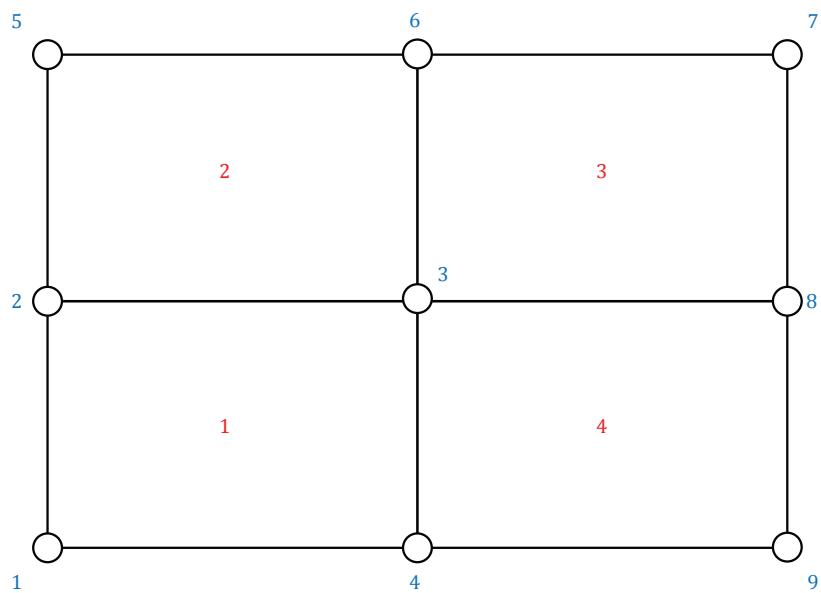
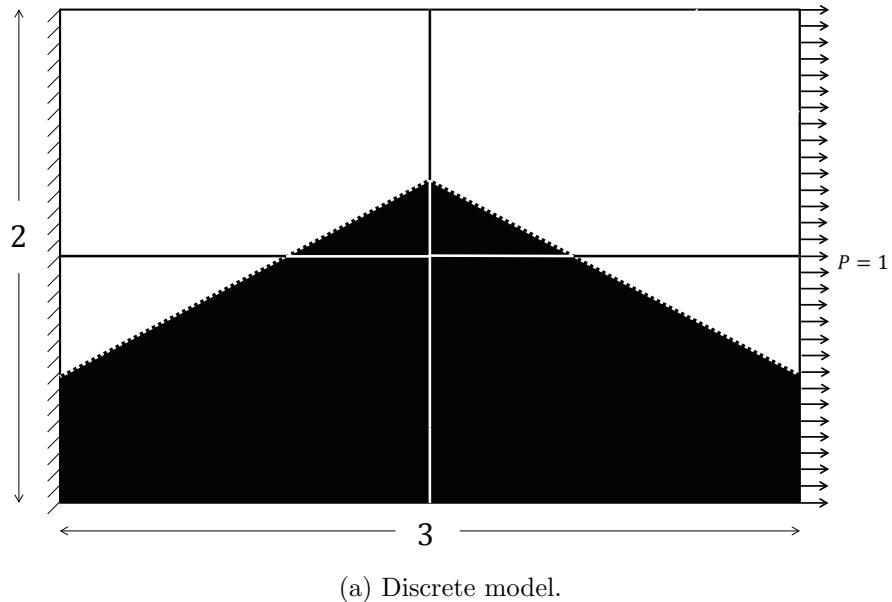
Table 3.7: Element to enrichment table for node ID 3. This table shows the initial enrichments node 3 received during the sub-phase algorithm and the enrichment levels after the enrichment algorithm.

- With this map we can loop over all elements and assign enrichment levels for each node to each pseudo-element, based on their sub-phase information.

### Determine degrees-of-freedom used

The last step in the algorithm is to flag which enrichment levels each node will use for interpolation. For example, in our test case, node 3 will have enrichment levels 0, 14, 15, 16, 17, 18 active.

## Implementation example 2



(b) Mesh information.

Figure 3.12: XFEM implementation configuration 2.

Nodes/Elements	1	2	3	4
1	0			
2 "C"	14	14		
3 "C"	0	0	0	0
4 "C"	0			0
5		14		
6 "C"		14	14	
7			14	
8 "C"			14	14
9				0

Table 3.8: Initial node-element table for configuration 2.

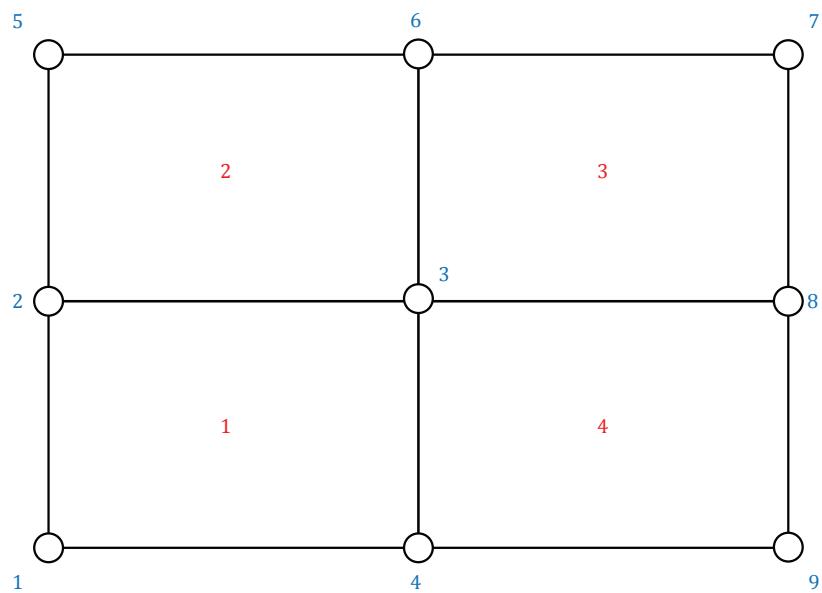
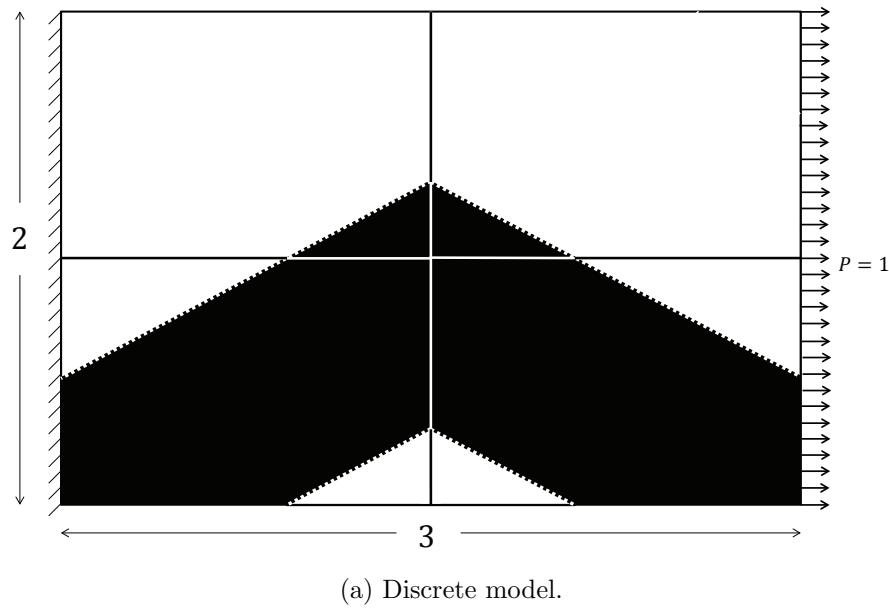
Nodes/Elements	1	2	3	4
1	0			
2 "C"	14	14		
3 "C"	0	0	0	0
4 "C"	0			0
5		14		
6 "C"		14	14	
7			14	
8 "C"			14	14
9				0

Table 3.9: Final node-element table for configuration 2.

For Node 3: Element / Enrichment level	0	14
1	0	14
2	0	14
3	0	14
4	0	14

Table 3.10: Enrichment level map for configuration 2.

### Implementation example 3



(b) Mesh information.

Figure 3.13: XFEM implementation configuration 3.

Nodes/Elements	1	2	3	4
1	0			
2 "C"	15	14		
3 "C"	0	0	0	0
4 "C"	14			0
5		14		
6 "C"		14	14	
7			14	
8 "C"			14	15
9				0

Table 3.11: Initial node-element table for configuration 3.

Nodes/Elements	1	2	3	4
1	0			
2 "C"	14	14		
3 "C"	0	0	0	0
4 "C"	15			15
5		14		
6 "C"		14	14	
7			14	
8 "C"			14	14
9				0

Table 3.12: Final node-element table for configuration 3.

For Node 3: Element / Enrichment level	0	14	15
1	0	15	14
2	0	14	
3	0	14	
4	0	15	14

Table 3.13: Enrichment level map for configuration 3.

### 3.2.5 Solving the problem

In the previous section, our algorithms determined which additional enriched degrees-of-freedom each node requires to account for the discontinuities in the elements. We will study our enrichment strategy with a heat conduction problem. To capture the discontinuities

along the phase boundaries, we enrich the standard finite element approximation with additional shape functions. We adopt the generalized enrichment strategy of Makhija and Maute [64] which resolves consistently the temperature fields in the presence of small features and does not suffer from artificially coupling disconnected phases.

$$u(\mathbf{x}) = \sum_{m=1}^M \left( H(-\phi) \sum_{i=1}^n \mathbf{N}_i u_{i,m}^A + H(\phi) \sum_{i=1}^n \mathbf{N}_i u_{i,m}^B \right) \quad (3.2)$$

where  $m$  is the enrichment level,  $M$  is the maximum number of enrichment levels used for each phase,  $\mathbf{N}$  are the shape functions,  $u_{i,m}^l$  is the vector of nodal temperature values at node  $i$  for phase  $l = [A, B]$ ,  $\phi$  is the level set value evaluated at the integration point, and  $H$  denotes the Heaviside function.

The Heaviside function  $H$  depends on the level set function and is defined as follows:

$$H(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases} \quad (3.3)$$

The Heaviside functions “turns on/off” the standard finite element interpolations in the particular phases. The approximation allows for discontinuities of the temperatures along the phase boundaries. Therefore the continuity is enforced weakly via the stabilized Lagrange multiplier method.

### 3.2.6 Preconditioner

When a sub-domain of a material phase is too small (around  $O(\epsilon^{1/2})$ ), the Jacobian matrix will be ill-conditioned. To solve this shortcoming, it is necessary to scale the matrix with another preconditioning matrix. This scaling matrix will be a function of the level-set field.

The preconditioner  $\mathbf{T}$  will have a scaling value for each degree of freedom in the problem. To obtain these values, we check which enriched degrees of freedom each node uses. Then,

we proceed to compute the integral of the shape function for the node with respect to the material sub-domain that requires said enriched degree-of-freedom. Because a node will have different scaling values across multiple elements, there are four preconditioner implementations available to compute the scaling value in a nodal cluster:

- Maximum value of integrals of shape functions
- Sum of values of integrals of shape functions
- Maximum value of integrals of the derivatives of the shape functions
- Sum of values of integrals of the derivatives of the shape functions

The third preconditioner approach will be used in this study. The matrix  $\mathbf{T}$  is a diagonal matrix built by integrating the spatial derivatives of the shape functions over the nodal support of nodes connected to an intersected element. The diagonal components of the matrix are defined as:

$$\mathbf{T}_{i,m}^l = \left( \max_{e \in E_i} \frac{\int_{\mathcal{D}_i^e} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x}) d\mathbf{x}}{\int_{\mathcal{D}_i^e} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x}) d\mathbf{x}} \right)^{-1/2} \quad (3.4)$$

where  $\mathbf{T}_{i,m}^l$  corresponds to the degree-of-freedom  $\mathbf{u}_{i,m}^l$ ,  $i$  is the node index,  $l = [A, B]$  is the material phase,  $m$  is the enrichment level,  $E_i$  is the set of elements connected to node  $i$ , and  $\mathcal{D}_i^e$  is the element domain of phase  $l$ . The components of the matrix increase as the region of influence of a degree-of-freedom decreases. The entries  $\mathbf{T}_{i,m}^l$  of nodes  $i$  that are not connected to at least one intersected element are set to one.

To avoid numerical issues due to large values for the components of  $\mathbf{T}$ , the degrees of freedom associated with the diagonal entry  $\mathbf{T}_{i,m}^l$  are constrained to zero if the following condition is satisfied:

$$\mathbf{T}_{i,m}^l \geq T_{tol} \quad (3.5)$$

where  $T_{tol}$  is  $10^9$  for this study.

At this point, there is one scaling value for each degree-of-freedom in the system. These scaling values are applied to the solution vector before the computation of the residual. After the residual is computed, they are both unscaled and then the new solution vector is computed.

### 3.3 Corroboration and results

#### 3.3.1 Methodology

Two formulations were used to corroborate the results of the XFEM implementation.

Equation 3.6 computes the difference in solutions at the discontinuity. Since the model we have implemented is based on inclusions and not crack propagation, this interface error should approach zero as the mesh gets finer.

$$\sqrt{\frac{\sum_{\text{element}} \sum_{\text{interface}} \int u^+ - u^- d\Gamma_i}{\sum_{\text{element}} \sum_{\text{interface}} \int d\Gamma_i}} \quad (3.6)$$

This equation computes the interface “jump” across all interfaces and elements in the model, then scales it with respect to the perimeter or area of the interface, and finally takes the square root.

Equation 3.7 compares the relative difference between the XFEM solution and the FEM solution.

$$\sqrt{\frac{\int u_{\text{XFEM}} - u_{\text{FEM}} d\Omega}{\int u_{\text{FEM}} d\Omega}} \quad (3.7)$$

$u_{\text{XFEM}}$  represents the XFEM solution, while  $u_{\text{FEM}}$  represents the FEM solution.

XFEM was used to solve a thermal problem with the configuration of Figure 3.14. The same problem was ran using the classical FEM. The FEM problem used two different types of elements and its mesh was refined until the solution reached convergence.

The mesh has a width of 20 units and a height of 20 units. The problem has Dirichlet boundary conditions on the sides. The temperature is prescribed to 0 on the left side and

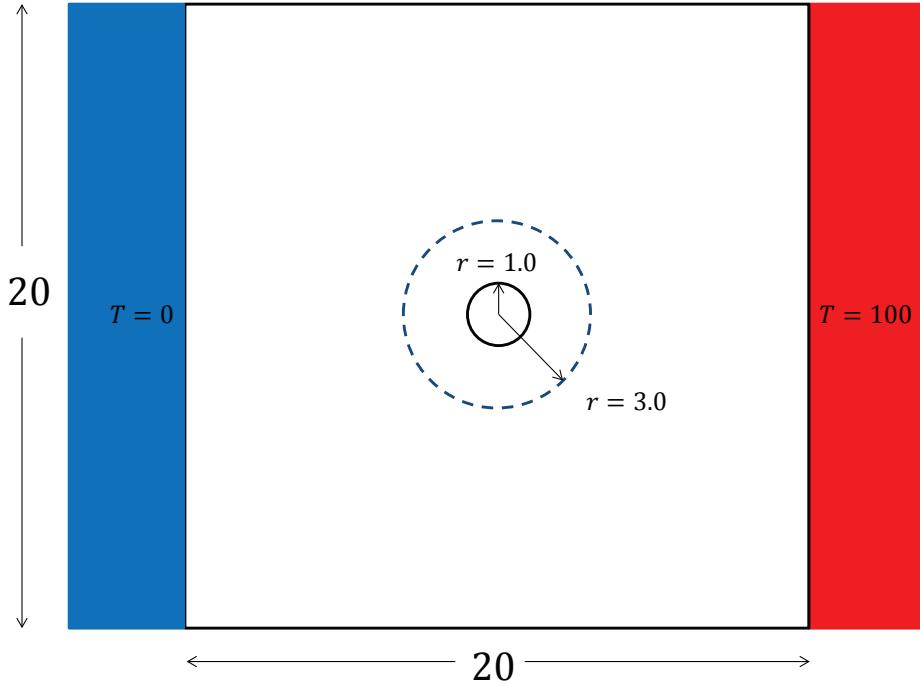


Figure 3.14: Diffusion problem setup.

100 on the right side. There is an inclusion at the center of the model. This inclusion is a different material with a different thermal conductivity than the material phase 1 domain.

The test consisted in modifying the diameter of the circle from 2 units to 6 units in 500 steps using different mesh refinements, different conductivity ratios and different preconditioners formulations.

### 3.3.2 Tests

#### Mesh refinement sweep

The mesh size was the variable in this test, while the conductivity ratio between both materials remained fixed at 10. No preconditioner scaling was applied. The different mesh sizes used were  $20 \times 20$ ,  $30 \times 30$ ,  $40 \times 40$ , and  $50 \times 50$ .

Figure 3.15 shows that as the mesh is refined, the interface error converges to zero.

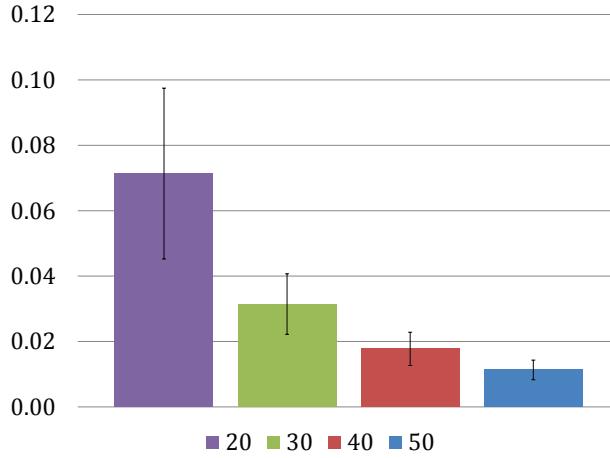


Figure 3.15: Mesh refinement sweep interface error.

Figure 3.16 shows that as the mesh is refined, the difference of the XFEM solution with respect to the FEM solution decreases. The larger difference for the  $50 \times 50$  mesh is due to the sampling and different mesh sizes used for the XFEM and FEM problems. A different mesh resampling size fixed the issue in other tests.

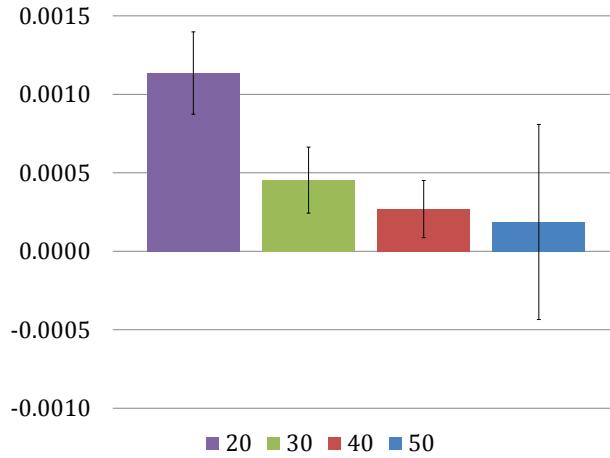


Figure 3.16: Mesh refinement sweep L2 error.

### Conductivity ratio sweep

The conductivity ratio between the different materials was the variable in this test. The mesh size was  $30 \times 30$  and the preconditioner formulation used the maximum spatial derivative of the shape functions. The different conductivity ratios used were 0.1, 10, 100, and

1000.

Figure 3.17 shows that when the material conductivity is the same for both materials (a “quasi-FEM” problem), the interface error is in the order of  $O(\epsilon)$ . However, the greater the difference in material properties at an interface, the larger the interface jump is.

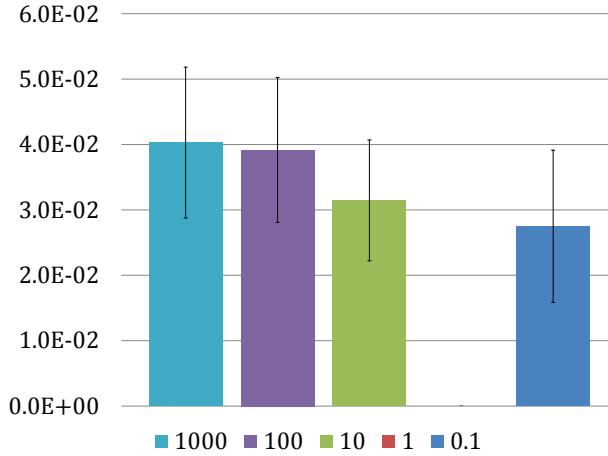


Figure 3.17: Conductivity refinement sweep interface error.

For the L2 computation, only FEM solutions with conductivity ratios of 10, 100, and 1000 were computed. Figure 3.18 shows that the difference in solutions is very small  $O(10^{-4})$ .

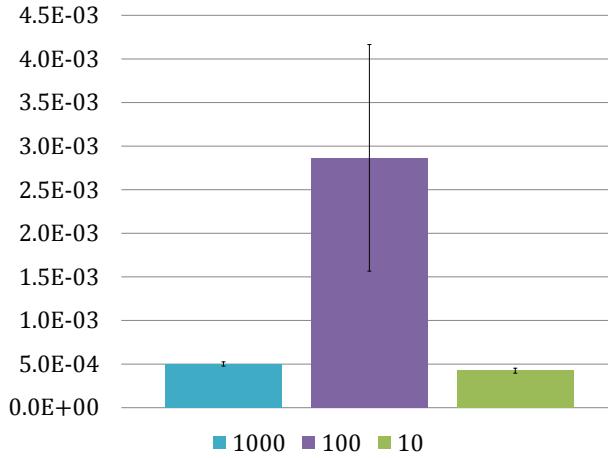


Figure 3.18: Conductivity refinement sweep L2 error.

## Condition number comparison

These tests were performed to compare the condition number of the global Jacobian matrix when the scaling was applied. The mesh size was  $30 \times 30$ , the conductivity ratio was 10 and the preconditioner formulation used the maximum spatial derivative of the shape functions. A direct solver and a GMRES iterative solver were used and compared.

Figure 3.19 shows that the Jacobian matrix has a condition number in the order of  $10^{15}$  when no scaling is applied <sup>1</sup>, while Figure 3.20 shows that the condition number decreases to the order of  $10^4$  when scaling is applied.

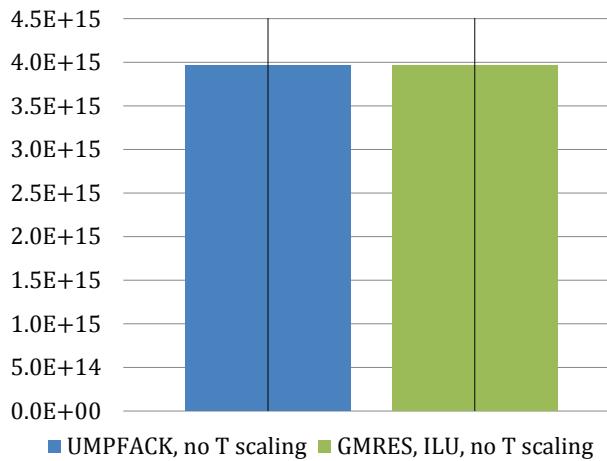


Figure 3.19: Condition number comparison - no pre-conditioner.

## 3.4 Conclusions

The framework developed in this project will help eliminate the need to re-mesh a model when discontinuities present in the domain. The program is capable of dividing an element into integrable sub-domains, calculating its topology, its enrichment information and computing the normal vector and Gauss points required for integration. The program is also

<sup>1</sup>We use the GMRES solver provided by the Trilinos linear algebra package to solve for the linear system. The solver uses an ILU preconditioner on top of the XFEM preconditioner of this study. However, the condition number of the matrix, after the ILU preconditioner is applied, is not provided by the Trilinos package. Because of that, the direct and iterative solver yield the same condition number. In reality, the GMRES option may have a lower condition number due to the ILU preconditioner.

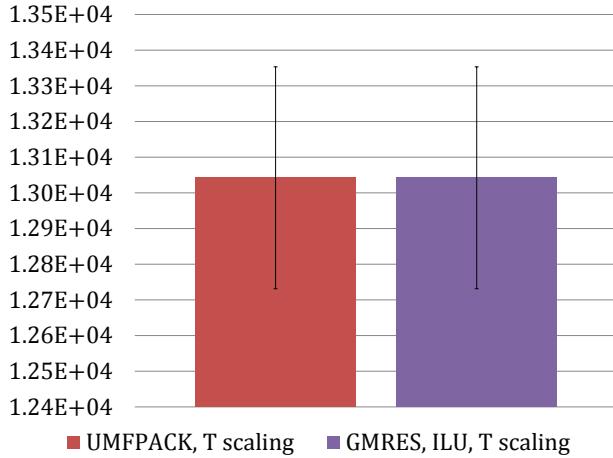


Figure 3.20: Condition number comparison - with pre-conditioner.

capable of solving XFEM problems with different topologies in 3D.

Results showed that the differences in solutions with a classical FEM problem for a two dimensional heat conduction model are small.

XFEM produced Jacobian matrices with high condition numbers, but the application of a preconditioner as a function of the level set field solved this shortcoming.

## 3.5 Appendix A: Delaunay Triangulation code

This code written in Matlab is the first attempt of the author to perform a Delaunay triangulation in an element based on the level set function values at the corner nodes. To triangulate different discontinuities change the *levs* variable: each entry corresponds to the value of the level set function at a node. The  $e_x$ ,  $e_y$ , and  $e_z$  vector variables contain the coordinates of the corner nodes and can be modified to change the shape of the element.

### 3.5.1 main.m

```

1 function [] = main()
2 % Main program
3 % Modify ex, ey, and ez to change shape of element
4 % Change levs to change level-set configuration
5
6 % Global x,y,z coordinates of element.

```

```

7 % This will form a 3D cubic element cube.
8 ex = [-0.5 -0.5 -0.5 -0.5 +0.5 +0.5 +0.5 +0.5];
9 ey = [-0.5 -0.5 +0.5 +0.5 -0.5 -0.5 +0.5 +0.5];
10 ez = [+0.5 -0.5 -0.5 +0.5 +0.5 -0.5 -0.5 +0.5];
11
12 %% Extract a particular case (set k manually).
13
14 % Initial test for a particular levelset function.
15 % Randn function will return a m x n matrix of random positive and negative
16 % numbers.
17 hexsect = cell(127,8);
18 k = round(1 + (127-1).*rand);
19 % levs = randn(1,8);
20 % levs = [-1 -1 -1 -1 -1 -1 -1 -1];
21 levs = [+1 -1 +1 -1 -1 +1 -1 +1];
22 [nsct,isct,xsct,ysct,zsct,levs] = xfem8isct(ex,ey,ez,levs);
23 [xtet,ytet,ztet,ctet,ptet,pnd,plist,Tetp1,Tetp2,tet1G,tet1L,tet2G,tet2L,size_phasel,size_phase2,volumel,
    volume2,total_volumel,total_volume2] = xfem8tet(isct, xsct, ysct, zsct, ex, ey, ez, levs);
24 hexsect{k,2} = xtet;
25 hexsect{k,3} = ytet;
26 hexsect{k,4} = ztet;
27 hexsect{k,5} = ctet;
28 hexsect{k,6} = ptet;
29 hexsect{k,7} = pnd;
30 hexsect{k,8} = plist;
31 [max(plist) min(plist)];
32
33 figure(1)
34 tetramesh(Tetp1,[xtet(1,:)',ytet(1,:)',ztet(1,:)'],-ones(size(Tetp1,1),1));
35 xlabel('X')
36 ylabel('Y')
37 zlabel('Z')
38 grid
39 axis equal
40
41 figure(2)
42 tetramesh(Tetp2,[xtet(1,:)',ytet(1,:)',ztet(1,:)'],ones(size(Tetp2,1),1));
43 xlabel('X')
44 ylabel('Y')
45 zlabel('Z')
46 grid
47 axis equal
48
49 figure(3)
50 tetramesh(ctet,[xtet(1,:)',ytet(1,:)',ztet(1,:)'],ptet);
51 xlabel('X')
52 ylabel('Y')
53 zlabel('Z')
54 axis equal
55
56 figure(4)
57 plot3(xtet(1,:),ytet(1,:),ztet(1,:),'X')
58 xlabel('X')
59 ylabel('Y')
60 zlabel('Z')
61 axis equal

```

```

62
63 end

```

### 3.5.2 xfem8isct.m

```

1 function [nsct ,isct ,xsct ,ysct ,zsct ,levs] = xfem8isct(ex,ey,ez,levs)
2
3 % Intersection of hex8 based on level-set values.
4 %
5 % Input: ex : global x-coordinates of nodes of 3D element
6 % ey : global y-coordinates of nodes of 3D element
7 % ez : global z-coordinates of nodes of 3D element
8 % levs : level set values randomly generated
9 %
10 % Output: nsct : number of intersected edges
11 % isect : vector flags of intersected edges
12 % xsct : x-coordinates of intersections in global and local coordinates
13 % ysct : y-coordinates of intersections in global and local coordinates
14 % zsct : z-coordinates of intersections in global and local coordinates
15
16 % Map of node connections across edges in a 3D element.
17 % Edgmap is a 12 x 2 matrix.
18 edgmap = [ 1 2; 2 3; 3 4; 4 1; 1 5; 2 6; 3 7; 4 8; 5 6; 6 7; 7 8; 8 5];
19
20 % Values of nodes in master element (local coordinates).
21
22 xp = [-1 -1 -1 -1 1 1 1 1];
23 yp = [-1 -1 1 1 -1 -1 1 1];
24 zp = [1 -1 -1 1 1 -1 -1 1];
25
26 % Set initial value of intersected edges to zero.
27 % Create a zero 12 x 1 matrix to flag edges with intersected edges.
28 % Create a zero 12 x 2 matrix to record x, y, z coordinates in global and
29 % local coordinates.
30 nsct = 0;
31 isct = zeros(12,1);
32 xsct = zeros(12,2);
33 ysct = zeros(12,2);
34 zsct = zeros(12,2);
35
36 for i = 1:12
37     % ic1 and ic2 return the values of the first and second columns of
38     % edgmap, respectively, for a determined row. ic1 and ic2 represent two
39     % nodes connected by a cube edge.
40     ic1 = edgmap(i,1);
41     ic2 = edgmap(i,2);
42     % The level set values at those nodes are multiplied to determine
43     % intersection.
44     if levs(ic1)*levs(ic2) < 0      % If so, then there is an intersection.
45         nsct = nsct+1;              % Increase number of intersected edges by one.
46         isct(i) = 1;                % Flag edge as having an intersection in isct matrix.
47         sctr = -levs(ic1)/(levs(ic2)-levs(ic1));    % Dimensionless location (ratio) of intersection.
48         % Insection in glb coordinates
49         xsct(i,1) = ex(ic1)+sctr*(ex(ic2)-ex(ic1));
50         ysct(i,1) = ey(ic1)+sctr*(ey(ic2)-ey(ic1));

```

```

51     zsct(i,1) = ez(ic1)+sctr*(ez(ic2)-ez(ic1));
52 % Intersection in local coordinates
53     xsct(i,2) = xp(ic1)+sctr*(xp(ic2)-xp(ic1));
54     ysct(i,2) = yp(ic1)+sctr*(yp(ic2)-yp(ic1));
55     zsct(i,2) = zp(ic1)+sctr*(zp(ic2)-zp(ic1));
56 end
57 end
58
59 display('Number of intersected edges: ');
60 disp(nsct);
61 display('Edges with intersections: ');
62 disp(isct');
63 display('x-coordinates of intersections in global coordinates: ');
64 disp(xsct(:,1));
65 display('x-coordinates of intersections in local coordinates: ');
66 disp(xsct(:,2));
67 display('y-coordinates of intersections in global coordinates: ');
68 disp(ysct(:,1));
69 display('y-coordinates of intersections in local coordinates: ');
70 disp(ysct(:,2));
71 display('z-coordinates of intersections in global coordinates: ');
72 disp(zsct(:,1));
73 display('z-coordinates of intersections in local coordinates: ');
74 disp(zsct(:,2));
75
76 end

```

### 3.5.3 xfem8tet.m

```

1 function [xtet,ytet,ztet,ctet,ptet,pnd,plist,Tetp1,Tetp2,tet1G,tet1L,tet2G,tet2L,size_phase1,size_phase2,
2 volumel,volume2,total_volumel,total_volume2] = xfem8tet(isct,xsct,ysct,zsct,ex,ey,ez,levs)
3 % Intersection of hex8 based on level-set values.
4 %
5 % Input: nsct : number of intersected edges
6 %        isect : vector flags intersected edges
7 %        xsect : x-coordinates of intersections in global and local coordinates
8 %        ysect : y-coordinates of intersections in global and local coordinates
9 %        zsect : z-coordinates of intersections in global and local coordinates
10 %       ex : global x-coordinates of nodes
11 %       ey : global y-coordinates of nodes
12 %       ez : global z-coordinates of nodes
13 %       levs : levs
14 %
15 % Output: xtet : x-coordinates of nodes of triangulated element
16 %        ytet : y-coordinates of nodes of triangulated element
17 %        ztet : z-coordinates of nodes of triangulated element
18 %        ctet : connectivity of tets in triangulated element
19 %        ptet : phase of each tetrahedron
20 %        pnd : nodal levelset values (actual values), # of nodes
21 %        plist : phase of each node (in terms of -1, 1 and 0)
22 %        Tetp1 : nodes of tetrahedrons for main phase 1
23 %        Tetp2 : nodes of tetrahedrons for main phase 2
24 %        tet1G : global coordinates of tetrahedron for phase 1
25 %        tet1L : local coordinates of tetrahedron for phase 1

```

```

26 %          tet2G : global coordinates of tetrahedron for phase 1
27 %          tet2L : local coordinates of tetrahedron for phase 2
28 %          size_phase1 : number of tetrahedrons in phase 1
29 %          size_phase2 : number of tetrahedrons in phase 2
30 %          volume1    : volume of a tetrahedron in phase 1
31 %          volume2    : volume of a tetrahedron in phase 2
32 %          total_volume1 : total volume of all tetrahedrons in phase 1
33 %          total_volume2 : total volume of all tetrahedrons in phase 2
34
35 %% Identify coordinates and phases to triangulate
36
37 % Nodes in local coordinates
38 xp = [-1 -1 -1 -1  1  1  1  1];
39 yp = [-1 -1  1  1 -1 -1  1  1];
40 zp = [1  -1 -1  1  1  -1 -1  1];
41
42 % Find intersected edges. Order of edges depends on configuration of edgmap
43 % isct flagged intersected edges with a 1
44 itr = find(isct>0);
45 % Sort nodes by main phase. Determine which nodes have positive or negative
46 % level-set values.
47 ip1 = find(levs<0);
48 ip2 = find(levs>0);
49
50 % Id-numbers of node at edge intersections, i.e. create pseudonodes
51 ips = 9:8+length(itr);
52
53 % Create phase vector for triangulated element. Vector contains level-set
54 % values at the 8 original nodes plus zero values for the new pseudonodes
55 pnd = [levs zeros(1,length(itr))];
56
57 % Create nodes for main phase 1
58 % Coordinates where nodes are negative plus coordinates of intersections in
59 % global coordinates
60 exp1 = [ex(ip1) xsct(itr,1)];
61 eyp1 = [ey(ip1) ysct(itr,1)];
62 ezp1 = [ez(ip1) zsct(itr,1)];
63 % Nodes and pseudonodes numbers of main phase 1
64 ipx1 = [ip1 ips];
65
66 % Create nodes for main phase 2
67 % Coordinates where nodes are positive plus coordinates of intersections in
68 % global coordinates
69 exp2 = [ex(ip2) xsct(itr,1)];
70 eyp2 = [ey(ip2) ysct(itr,1)];
71 ezp2 = [ez(ip2) zsct(itr,1)];
72 % Nodes and pseudonodes numbers of main phase 2
73 ipx2 = [ip2 ips];
74
75 % Combine triangulation points of main phase 1 and 2 in local and global
76 % coordinates
77 xtet = [ex xsct(itr,1);xp xsct(itr,2)];
78 ytet = [ey ysct(itr,1);yp ysct(itr,2)];
79 ztet = [ez zsct(itr,1);zp zsct(itr,2)];
80
81 % Same as xtet, ytet, ztet, but only with global coordinates. Useful for

```

```

82 % triangulation below.
83 exp = [ex xsct(itr,1)'];
84 eyp = [ey ysct(itr,1)'];
85 ezp = [ez zsct(itr,1)'];
86
87 %% Triangulate main phase 1 and 2 together
88 % If we triangulate main phase 1 and 2 separately ,tetrahedron will
89 % superpose for some level-set combinations. We need to triangulate the
90 % entire element as a whole.
91
92 Tp = DelaunayTri(exp',eyp',ezp');
93 Tetp = Tp.Triangulation(:,,:);
94
95 %% Separate triangulation into main phases 1 and 2
96
97 Tetp1 = zeros(1,4);
98 Tetp2 = zeros(1,4);
99 index1 = 1;
100 index2 = 1;
101 % If tetrahedron contains a negative node, it is phase 1. Phase 2,
102 % otherwise.
103 for i = 1:size(Tetp,1)
104     if pnd(Tetp(i,1)) < 0||pnd(Tetp(i,2)) < 0||pnd(Tetp(i,3)) < 0||pnd(Tetp(i,4)) < 0
105         Tetp1(index1,:) = Tetp(i,:);
106         index1 = index1 + 1;
107     else
108         Tetp2(index2,:) = Tetp(i,:);
109         index2 = index2 +1;
110     end
111 end
112
113 %% Display coordinates of phase 1 tetrahedrons
114 % Display coordinates of phase 1 tetrahedrons in global coordinates
115 % Display volume of each phase 1 tetrahedron
116 total_volume1 = 0;
117 display('The global coordinates for the phase 1 tetrahedrons are: ')
118 for i = 1:size(Tetp1,1)
119     tet1G = [xtet(1,Tetp1(i,1)),ytet(1,Tetp1(i,1)),ztet(1,Tetp1(i,1));xtet(1,Tetp1(i,2)),ytet(1,Tetp1(i,2)),
120         ,ztet(1,Tetp1(i,2));xtet(1,Tetp1(i,3)),ytet(1,Tetp1(i,3)),ztet(1,Tetp1(i,3));xtet(1,Tetp1(i,4)),
121         ,ytet(1,Tetp1(i,4)),ztet(1,Tetp1(i,4))];
122     disp(tet1G)
123     % Calculate volume. Algorithm: For 4 points in tetrahedron P,Q,R,S,
124     % volume = abs(det([Q-P;R-Q;S-R]))/6
125     a = tet1G(2,:)-tet1G(1,:);
126     b = tet1G(3,:)-tet1G(2,:);
127     c = tet1G(4,:)-tet1G(3,:);
128     volume1 = abs(det([a;b;c]))/6;
129     total_volume1 = total_volume1 + volume1;
130     display('The volume of phase 1 tetrahedron is: ')
131     disp(volume1)
132 end
133
134 %% Display coordinates of phase 1 tetrahedrons in local coordinates
135 display('The local coordinates for the phase 1 tetrahedrons are: ')
136 for i = 1:size(Tetp1,1)

```

```

135     tet1L = [xtet(2,Tetp1(i,1)),ytet(2,Tetp1(i,1)),ztet(2,Tetp1(i,1));xtet(2,Tetp1(i,2)),ytet(2,Tetp1(i,2)),ztet(2,Tetp1(i,2));xtet(2,Tetp1(i,3)),ytet(2,Tetp1(i,3)),ztet(2,Tetp1(i,3));xtet(2,Tetp1(i,4)),ytet(2,Tetp1(i,4)),ztet(2,Tetp1(i,4))];
136     disp(tet1L)
137 end
138
139 % Display total volume of phase 1 tetrahedrons
140 display('The total volume of phase 1 tetrahedrons is: ')
141 disp(total_volume1)
142
143 %% Display coordinates of phase 2 tetrahedrons
144 % Display coordinates of phase 2 tetrahedrons in global coordinates
145 % Display volume of each phase 2 tetrahedron
146 total_volume2 = 0;
147 display('The global coordinates for the phase 2 tetrahedrons are: ')
148 for i = 1:size(Tetp2,1)
149     tet2G = [xtet(1,Tetp2(i,1)),ytet(1,Tetp2(i,1)),ztet(1,Tetp2(i,1));xtet(1,Tetp2(i,2)),ytet(1,Tetp2(i,2)),ztet(1,Tetp2(i,2));xtet(1,Tetp2(i,3)),ytet(1,Tetp2(i,3)),ztet(1,Tetp2(i,3));xtet(1,Tetp2(i,4)),ytet(1,Tetp2(i,4)),ztet(1,Tetp2(i,4))];
150     disp(tet2G)
151     % Calculate volume.
152     a = tet2G(2,:)-tet2G(1,:);
153     b = tet2G(3,:)-tet2G(2,:);
154     c = tet2G(4,:)-tet2G(3,:);
155     volume2 = abs(det([a;b;c]))/6;
156     total_volume2 = total_volume2 + volume2;
157     display('The volume of phase 2 tetrahedron is: ')
158     disp(volume2)
159 end
160
161 % Display coordinates of phase 2 tetrahedrons in local coordinates
162 display('The local coordinates for the phase 2 tetrahedrons are: ')
163 for i = 1:size(Tetp2,1)
164     tet2L = [xtet(2,Tetp2(i,1)),ytet(2,Tetp2(i,1)),ztet(2,Tetp2(i,1));xtet(2,Tetp2(i,2)),ytet(2,Tetp2(i,2)),ztet(2,Tetp2(i,2));xtet(2,Tetp2(i,3)),ytet(2,Tetp2(i,3)),ztet(2,Tetp2(i,3));xtet(2,Tetp2(i,4)),ytet(2,Tetp2(i,4)),ztet(2,Tetp2(i,4))];
165     disp(tet2L)
166 end
167
168 % Display total volume of phase 2 tetrahedrons
169 display('The total volume of phase 2 tetrahedrons is: ')
170 disp(total_volume2)
171
172 %% Display number of tetrahedrons in each phase.
173 size_phase1 = size(Tetp1(:,1));
174 size_phase1 = size_phase1(1);
175 size_phase2 = size(Tetp2(:,1));
176 size_phase2 = size_phase2(1);
177 display('The number of tetrahedrons in phase 1 is: ')
178 disp(size_phase1);
179 display('The number of tetrahedrons in phase 2 is: ')
180 disp(size_phase2);
181
182 %% Locate triangle interfaces - new algorithm
183 % Use function ismember to compare array vectors
184 % Display only result if three nodes repeat

```

```

185
186 display('The interfaces can be found on the triangles with nodes: ')
187 for i = 1:size(Tetp1,1)
188     for j = 1:size(Tetp2,1)
189         r = ismember(Tetp1(i,:),Tetp2(j,:));
190         Tetp1_tri = Tetp1(i,:);
191         Tetp1_tri = Tetp1_tri(r);
192         if size(Tetp1_tri,2) == 3
193             disp(Tetp1_tri)
194         end
195     end
196 end
197
198 %% Combine triangulation of nodes in ctet
199 % Size of ptet depends on number of tetrahedrons
200 % Phase 1 tetrahedrons have value of -1, phase 2 value of 1 in ptet
201
202 % Original:
203 % ctet=[Tetp1;Tetp2];
204 % ptet=[-ones(size(Tetp1,1),1);ones(size(Tetp2,1),1)];
205
206 % New method:
207 % Provide two options for triangulation
208 % Option 1: based on volume
209 % If one phase is significantly larger than the other, switch tetramesh
210 if total_volume2 <= (0.2*(total_volumel + total_volume2))
211     ctet=[Tetp2;Tetp1];
212     ptet=[-ones(size(Tetp2,1),1);ones(size(Tetp1,1),1)];
213 else
214     ctet=[Tetp1;Tetp2];
215     ptet=[-ones(size(Tetp1,1),1);ones(size(Tetp2,1),1)];
216 end
217
218 % Option 2: based on user's choice
219 % option = input('Choose triangulation option 1 or 2: ');
220 % if option == 1
221 %     ctet=[Tetp2;Tetp1];
222 %     ptet=[-ones(size(Tetp2,1),1);ones(size(Tetp1,1),1)];
223 % elseif
224 %     ctet=[Tetp1;Tetp2];
225 %     ptet=[-ones(size(Tetp1,1),1);ones(size(Tetp2,1),1)];
226 % end
227
228 %% Check connectivity between phases. PART of ORIGINAL CODE
229
230 % Obsolete? plist produces the same value as pnd
231 % Check connectivity for main phase 1 and creat sub-phase information
232
233 % Matrix plisp1 formed of (nodes + pseudonodes) x 1 elements with value of
234 % -2. ipx1 represents location of phase 1 nodes + pseudonodes. At these
235 % locations, values are replaced by -1
236 plisp1 = -2*ones(size(xtet,2),1);
237 plisp1(ipx1) = -1;
238
239 domp1 = 0;
240 % While there are phase 1 nodes

```

```

241 while ismember(-1,plisp1) > 0
242     domp1 = domp1 + 1;
243     % Find where the negative nodes are in the nodes + pseudonodes vector
244     % Value are progressively changed by 0, one at a time
245     ppp = find(plisp1 == -1);
246     plisp1(ppp(1)) = 0;
247     % Locate where values were changed to zero and create new variable ppp
248     % pid represents the node where value has transformed into zero
249     while ismember(0,plisp1) > 0
250         ppp = find(plisp1 == 0);
251         pid = ppp(1);
252         for it = 1:size(ctet,1)
253             % If the node at pid belongs to Tetp1 and is negative
254             if ismember(pid,ctet(it,:)) > 0 && ptet(it) < 0
255                 plisp1(ctet(it,:)) = max(0,plisp1(ctet(it,:)));
256             end
257         end
258         % At the location of phase 1 nodes + pseudonodes, the value will be
259         % replaced to 1.
260         plisp1(pid) = domp1;
261     end
262 end
263
264 % Check connectivity for main phase 2 and creat sub-phase information
265 % Functions the same as previous routine, but for phase 2
266 plisp2 = -2*ones(size(xtet,2),1);
267 plisp2(ipx2) = -1;
268
269 domp2 = 0;
270 while ismember(-1,plisp2) > 0
271     domp2 = domp2+1;
272     ppp = find(plisp2 == -1);
273     plisp2(ppp(1)) = 0;
274     while ismember(0,plisp2) > 0
275         ppp = find(plisp2 == 0);
276         pid = ppp(1);
277         for it = 1:size(ctet,1)
278             if ismember(pid,ctet(it,:)) > 0 && ptet(it) > 0
279                 plisp2(ctet(it,:)) = max(0,plisp2(ctet(it,:)));
280             end
281         end
282         plisp2(pid) = domp2;
283     end
284 end
285
286 % Build phase list including subphase information
287 % Replace all pseudonodes by 0
288 plisp1(ips) = 0;
289 plisp2(ips) = 0;
290
291 % Find location of the pseudonodes
292 idp1 = find(plisp1>0);
293 idp2 = find(plisp2>0);
294
295 % Create a zero (nodes + pseudonodes) x 1 matrix plist
296 plist = zeros(size(xtet,2),1);

```

```

297 % Create list that shows which original nodes belong to phase I and II
298 plist(idp1) = -plisp1(idp1);
299 plist(idp2) = plisp2(idp2);
300
301 end

```

### 3.5.4 number\_configurations.m

```

1 function [] = number_configurations()
2 % This function computes all possible combinations of level-set function
3 % values at the corner nodes to obtain the number of 3D triangulation
4 % combinations possible.
5
6 % x,y,z coordinates of element.
7 % This will form a 3D cubic element cube.
8 ex = [0 1 1 0 0 1 1 0];
9 ey = [0 0 1 1 0 0 1 1];
10 ez = [0 0 0 0 1 1 1 1];
11
12 %% Sweep over all possible level-set configurations.
13
14 % Twelve edges. Maximum number of configurations:
15 maxc = 2^12;
16 icase = zeros(maxc,1);
17 hexsect = cell(127,8);
18
19 % Set initial configuration, k
20 % This loop calculates the total number of possible configurations of
21 % edge intersections
22 % Routine uses simple negative/positive level set values
23 k = 0;
24 for i1 = -1:2:1
25     for i2 = -1:2:1
26         for i3 = -1:2:1
27             for i4 = -1:2:1
28                 for i5 = -1:2:1
29                     for i6 = -1:2:1
30                         for i7 = -1:2:1
31                             for i8 = -1:2:2
32                                 levs = [i1 i2 i3 i4 i5 i6 i7 i8];
33                                 [nsct,isct,xsct,ysct,zsct,levs] = xfem8isct(ex,ey,ez,levs);
34                                 % isct is a 12 x 1 matrix representing each edge of the 3D cube element
35                                 % Since each edge can have one or zero intersections, the isct vector
36                                 % becomes a binary display
37                                 cbin = sprintf('%d%d%d%d%d%d%d%d%d%d%d',isct(1),isct(2),isct(3),isct(4),
38                                 isct(5),isct(6),isct(7),isct(8),isct(9),isct(10),isct(11),isct(12));
39                                 % cbn is a binary number displayed as a string, and converted into the
40                                 % decimal cdec.
41                                 cdec = bin2dec(cbin);
42                                 % If icase is equal to zero, it means it is a new unique configuration and
43                                 % therefore,
44                                 % number of total configurations k should
45                                 % be increased by one
46                                 if icase(cdec+1) == 0 && nsct > 0
47                                     k = k+1;

```

```

44      [xtet,ytet,ztet,ctet,ptet,pnd,plist,Tetp1,Tetp2,tet1G,tet1L,tet2G,
45      tet2L] = xfem8tet(isct,xsct,ysct,zsct,ex,ey,eZ,levs);
46      hexsect{k,1} = cdec;
47      hexsect{k,2} = xtet;
48      hexsect{k,3} = ytet;
49      hexsect{k,4} = ztet;
50      hexsect{k,5} = ctet;
51      hexsect{k,6} = ptet;
52      hexsect{k,7} = pnd;
53      hexsect{k,8} = plist;
54      [max(plist) min(plist)];
55      icase(cdec+1)=1;
56      end
57      end
58      end
59      end
60      end
61      end
62      end
63 end
64
65 fprintf('Number of configurations = %d\n',k);
66
67 end

```

# **Chapter 4**

## **Density and level set XFEM schemes for topology optimization of 3D structures**

As the capabilities of additive manufacturing techniques increase, topology optimization provides a promising approach to design geometrically sophisticated structures which can be directly manufactured. Traditional topology optimization methods aim at finding the conceptual design but often lack a sufficient resolution of the geometry and structural response, needed to directly use the optimized design for manufacturing. To overcome these limitations, this paper studies the viability and characteristics of the eXtended Finite Element Method (XFEM) in combination with the Level-Set Method (LSM) for topology optimization of three dimensional structural design problems. The LSM describes the geometry by defining the nodal level set values via explicit functions of the optimization variables. The structural response is predicted by a generalized version of the XFEM. The LSM-XFEM approach is compared against results from a traditional Solid Isotropic Material with Penalization (SIMP) method for two-phase “solid-void” and “solid-solid” problems. The numerical results demonstrate that the LSM-XFEM approach can describe crisply the geometry and

predict the structural response of complex three-dimensional structures with acceptable accuracy even on coarse meshes. However, the LSM-XFEM studied here lacks a robust and intuitive formulation to control the minimum feature size, and the optimization results may depend on the initial design.

## 4.1 Introduction

Recent advances in additive manufacturing allow the precise placement of one or multiple materials at micrometer resolution with essentially no restrictions on the geometric complexity of the spatial arrangement. Complex three dimensional solids can be created with highly non-regular material distributions in a near optimal fashion, enabling the fabrication of structures with enhanced performance. Topology optimization has emerged as a promising approach to utilize the benefits of additive manufacturing [71, 80]. Structural topology optimization seeks to find the optimal geometry and/or the material layout of a body within a given design domain. The geometry is represented by the spatial distribution of two or more material phases; in structural problems, one of these material phases may represent void.

Originally topology optimization methods were developed primarily to create conceptual designs in the early stage of the design process [11, 90]. Later, topology optimization was applied to directly design micro-electro-mechanical systems (MEMS), utilizing the ability of thin-film fabrication techniques, such as photolithography in combination with chemical etching, to create geometrically complex devices at low cost [98, 99]. As the MEMS design problem is essentially two-dimensional, traditional approaches were sufficient to achieve the necessary geometric resolution at acceptable computational cost. Motivated by the availability of affordable additive fabrication methods for three dimensional structures, this paper focuses on topology optimization of three dimensional structures and introduces a new approach for finding optimized designs with high geometric resolution on rather coarse

computational meshes.

Most approaches for structural topology optimization are density methods. For a two-phase problem, the density is considered a design variable and can assume intermediate values between the density of the material phase “A” and the density of the material phase “B”. The most popular density method is the Solid Isotropic Material with Penalization method introduced by Bendsøe [7] and Zhou and Rozvany [129]. It features great versatility, robustness, efficiency, and ease of implementation for a broad range of applications [21, 101].

Density methods typically describe the boundaries between the material phases either via intermediate density values or by discrete material distributions leading to jagged boundaries. In both cases, the enforcement of boundary and interface conditions at the material interface is hampered and may result in non-physical responses, such as premature yielding [69]. Often this issue can be mitigated by mesh refinement or adaptive re-meshing [67, 68]. However, for problems that require an accurate description of the boundaries, such as boundary layer problems in fluids and skin-depth issues in electromagnetics, it was reported that density methods fail [101].

The shortcomings of density methods have promoted the development of the Level Set Method (LSM) for topology optimization. The LSM allows a crisp representation of the phase boundaries and the accurate enforcement of boundary conditions on fixed meshes. The material interface in the LSM is described implicitly by the iso-contours of a Level Set Function (LSF), usually the zero level-set contour [3, 95, 117].

The LSF is typically discretized by the same mesh used for the physical field and is updated in the optimization process via the solution of the Hamilton-Jacobi equations. Alternatively, the parameters of the discretized LSF are defined as explicit functions of the optimization variables, and the resulting parameter optimization problem is solved by standard nonlinear programming methods [23]. The key challenges for the LSM include (a) controlling the spatial gradients of the LSF in the vicinity of the zero-level set contour to avoid ill-conditioning of the optimization problem, (b) controlling local feature sizes, and

(c) accelerating the convergence of the geometry in the optimization process. For a detailed discussion of the LSM, the reader is referred to the comprehensive review by Dijk et al [23] and Gain and Paulino [35].

In LSMs, the structural geometry can be represented in the mechanical model via an Ersatz material approach, immersed boundary techniques, or by adaptive geometry conforming meshes. The first two approaches allow the use of fixed, design independent meshes while the last approach requires local or global re-meshing as the structural geometry evolves in the design process.

Using an Ersatz material approach, the void phase is modeled by a soft material and the material properties in elements intersected by the zero-level set contour are interpolated between the “void” and solid phase, proportional to the volumes of the individual phases. However, this approach faces the same issues as density methods in regards to enforcing boundary conditions across the material interface. Alternative approaches to model the mechanical response include generalized and adaptive finite element schemes such as the Super-Imposed Finite Element Method (SFEM) [118], the eXtended finite element method (XFEM) [53, 73, 120], and local re-meshing schemes [126].

In this paper, we focus on the LSM in combination with the XFEM. The XFEM does not require a mesh that conforms to the material interfaces and reduces the complexity of mesh construction. Spatial discontinuities in the structural response are captured by augmenting the standard finite element interpolations with additional shape functions. This approach is similar to the SFEM [118] as the solution is obtained by super-imposing the standard and enriched shape functions. However, unlike the SFEM, the XFEM can combine multiple types of shape functions and thus allows for greater flexibility.

The XFEM builds upon the partition of unity concept developed by Babuška and Melenk [5]. The XFEM was originally proposed by Belytschko and Black [6] to model crack propagation. The reader is referred to Yazid et al [127] for an overview of the application of XFEM to problems in fracture mechanics. The XFEM has been used for a variety of problems in

computational mechanics, such as fluid-structure interaction [36, 37], multi-phase flows [33], and nano-scale heat transfer [58]. A general overview of the method is presented by Fries and Belytschko [32].

Duysinx et al [25] originally introduced the XFEM into shape optimization using a simplified XFEM formulation. This formulation does not use additional enrichment functions and is limited to problems where one of the material phases represents void and the geometric configuration is “simple”, i.e. does not contain geometric features that are smaller than the size of two elements [64]. In this instance, the weak form of the governing equations is only integrated over the solid material in each element. In addition, if the interface between the material phases is traction free, this simplified version of the XFEM only differs from the traditional finite element method with respect to the domain of integration. The simplified XFEM version was applied to structural shape optimization, for example, by Duysinx et al [25], van Miegroet et al [74], and van Miegroet and Duysinx [73], and to topology optimization of two and three dimensional structures by Herrero et al [47] and Li et al [60], respectively.

An XFEM approach based on a standard enrichment strategy allows to model two-phase problems with a simple intersection pattern. This formulation is considered, for example, by Wei et al [120] to solve “solid-void” structural topology optimization problems, modeling the “void” phase as a soft material. Maute et al [70] used a standard enrichment strategy to discretize the phonon Boltzmann transport equation and optimize the thermal conductivity of nano-composites. However, this enrichment strategy is not guaranteed to consistently approximate the state variable fields for configurations with complex intersection patterns.

An enhanced version of the XFEM was proposed by Makhija and Maute [64], who presented a generalized enrichment strategy based on the step enrichment of Hansbo and Hansbo [42] and applied it to two dimensional structural topology optimization. This formulation captures consistently the mechanical response for complex geometries and intersection patterns for general multi-phase problems.

This paper will expand the combination of the LSM and the XFEM onto general two-phase problems in three dimensions. We will compare results of the proposed LSM-XFEM framework with SIMP results for structural topology optimization examples. The numerical results will show that the LSM-XFEM combination is a promising approach for three dimensional problems and allows for the use of coarse meshes to represent the structural geometry and to describe the structural response with acceptable accuracy.

The main challenges of expanding the previous LSM with XFEM approaches to three dimensions stem from the increased complexity of possible intersections patterns. Such patterns include elements that are intersected multiple times and elements containing only a small volume of a particular phase. In contrast to Li et al [60], we will adopt the generalized enrichment strategy of Makhija and Maute [64] to (a) accurately model the structural response on complex three dimensional patterns and (b) solve solid-solid material distribution problems. Further, we will expand the preconditioning scheme of Lang et al [56] onto three dimensions to mitigate ill-conditioning issues in the XFEM analysis problems due to elements with small volume fractions.

The main contributions of the paper are: (a) we present a numerically robust and computationally viable approach for solving general two-phase, three dimensional topology optimization problems, and (b) we provide a direct comparison of LSM-XFEM and SIMP results for three dimensional problems, highlighting key features of the two methods.

The remainder of this paper is structured as follows: the geometry models of the LSM-XFEM and SIMP methods are described in Section 4.2. The mechanical model and the XFEM formulation are summarized in Section 4.3. Details of the LSM-XFEM and SIMP optimization approaches are presented in Section 4.4. Section 4.5 highlights specific computational challenges of the XFEM approach for three dimensional problems. Section 4.6 presents structural topology optimization examples, comparing the LSM-XFEM and SIMP approaches. Section 4.7 summarizes the main conclusions drawn from this study.

## 4.2 Geometry Modeling

In topology optimization, the geometry of a body is defined via its material distribution. In density methods, such as SIMP, the material distribution is discretized by finite elements, with either elemental or nodal parameters defining the distribution within the element. Most often the same mesh is used to approximate the density distribution and the structural response. Alternatively, the state and density fields can be discretized by different meshes with different refinement levels; see for example the Multi-resolution Topology Method (MTOP) by Nguyen et al [77]. The optimization variables define analytically or by means of auxiliary partial differential equations the nodal or element density parameters [101]. For two-phase problems, the density is continuously varied between “0” (phase “A”) and “1” (phase “B”). Implicit or explicit penalization schemes, optionally combined with projection methods, are used to encourage “0-1” solutions [40, 100].

The crispness of the interface geometry, as described via the optimized material distribution, depends on (a) the formulation of the optimization problem, i.e. the objective and constraints, (b) regularization techniques, such as density or sensitivity filters, and (c) the optimization algorithm. In general, the resolution of the phase boundaries increases as the mesh is refined. For three dimensional problems, often coarse meshes are used to limit the computational costs. In this case, the boundary geometry either lacks crispness due to the presence of elements with intermediate densities or is approximated by a spatially discontinuous material distribution, leading to jagged interfaces.

Alternatively, the material distribution can be described via a level set function  $\phi(\mathbf{x})$ . Typically the zero level set contour defines the phase boundaries. Considering a two-phase problem, the interface  $\Gamma_{A,B}$  is defined as follows:

$$\begin{aligned}\phi(\mathbf{x}) &< 0, \quad \forall \mathbf{x} \in \Omega_A, \\ \phi(\mathbf{x}) &> 0, \quad \forall \mathbf{x} \in \Omega_B, \\ \phi(\mathbf{x}) &= 0, \quad \forall \mathbf{x} \in \Gamma_{A,B},\end{aligned}\tag{4.1}$$

where the vector  $\mathbf{x}$  collects the spatial coordinates,  $\Omega_A$  is the domain of material phase “A”,  $\Omega_B$  is the domain of material phase “B”, and  $\Gamma_{A,B}$  defines the material interface between phase “A” and phase “B”. For example, to model a sphere in a three dimensional mesh centered at  $(x_c, y_c, z_c)$ , the value of the level set function at a grid point  $(x_i, y_i, z_i)$  is:

$$\phi_i = (x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 - r^2, \quad (4.2)$$

where  $r$  represents the radius of the sphere, and the sign of  $\phi_i$  at each node determines if the node is inside or outside the sphere.

The level set field is typically discretized by shape functions with either local or global support [23]. In the simplest and most common approach, the level-set field is approximated on the same mesh used for discretizing the governing equations. In this study, we follow this approach and define the nodal level-set values explicitly as functions of the optimization variables (see 4.4.1). The resulting parameter optimization problem is solved by a standard nonlinear programming method.

While LSMs provide a crisp definition of the phase boundaries, they require seeding the initial design with inclusions and/or introducing inclusions in the course of the optimization process, for example via topological derivatives [15, 29, 82, 103]. An example of an initial design with a regular pattern of spherical inclusions is shown in Fig. 4.1. The images in the upper row and the image in the lower left corner show only phase “A”. The material layout of both phases is depicted in the lower right image. This layout can be generated by superposing Eq. 4.2 for spheres at uniformly spaced center locations. The optimization results of the LSM are typically dependent on the initial layout. Furthermore, it is non-trivial to generate an initial design that satisfies geometric design constraints, such as volume or perimeter constraints. In our experience, starting from an initial design that satisfies the constraints prevents the optimization process from converging to a design with poor performance and numerical artifacts, such as disconnected, free-floating material.

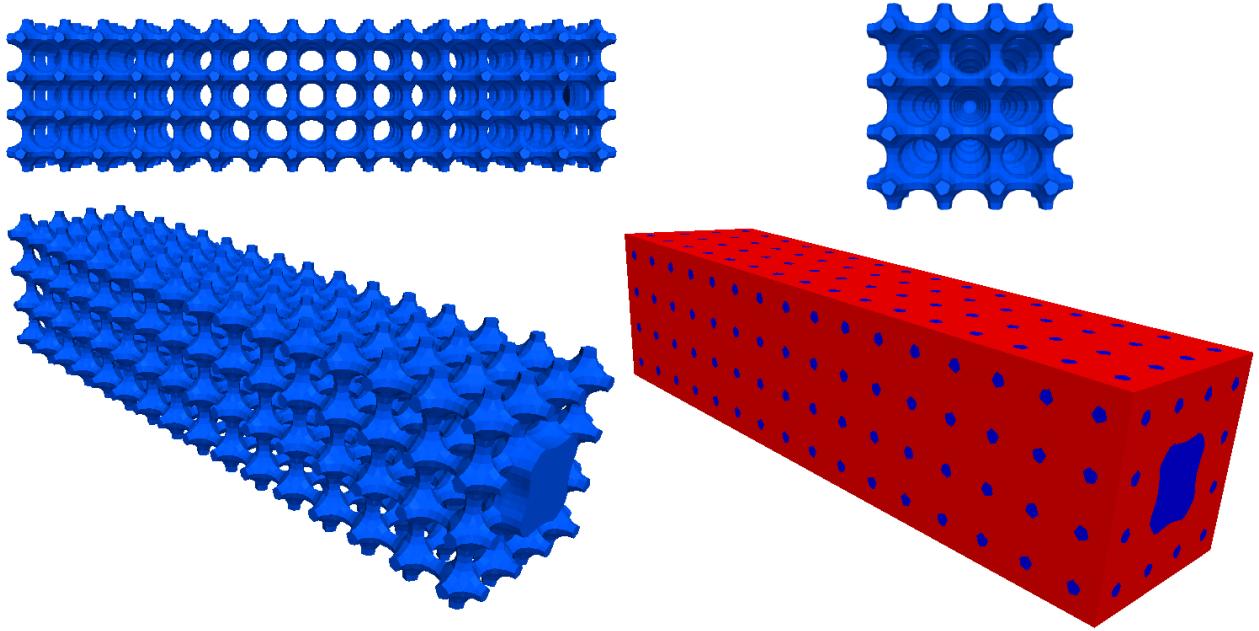


Figure 4.1: Initial design with array of spherical inclusions for the cantilever beam example of Section 4.6.3; blue represents phase “A” material and red represents phase “B”.

## 4.3 Structural Analysis

In this section, we briefly discuss the structural model and the XFEM analysis used in this paper. The governing equations modeling a linear elastic structural response are presented first, followed by a summary of the XFEM discretization and analysis.

### 4.3.1 Structural Model

This paper considers the topology optimization of structures using the LSM approach described above and the XFEM to predict the structural response, assuming infinitesimal strains, a linear elastic material behavior, and static conditions. We consider the two-phase problem depicted in Fig. 4.2, where  $\Gamma_N$  denotes the surface where traction forces are applied,  $\Gamma_D$  denotes the surface with prescribed displacements, and  $\mathbf{n}$  is the normal at the material interface pointing from phase “A” to phase “B”. The weak form of the governing equations

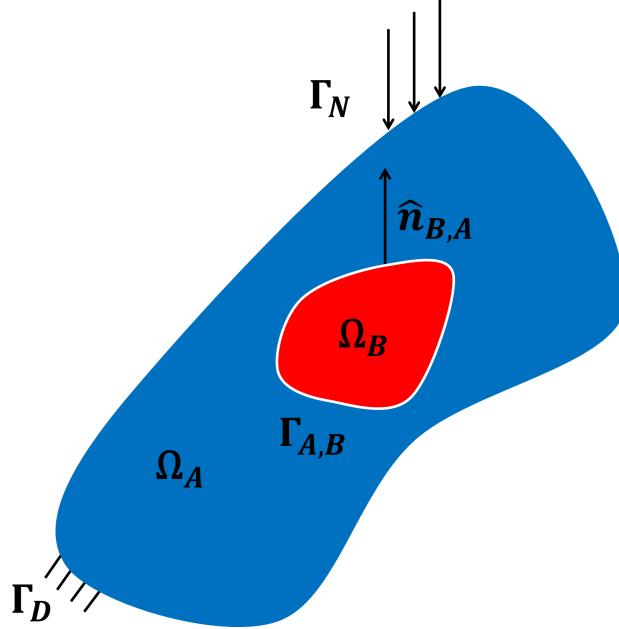


Figure 4.2: Two-phase problem.

can be decomposed into the following terms:

$$W = W_S + W_L + W_k = 0, \quad (4.3)$$

where  $W_S$  collects the contributions from the static equilibrium, including body forces and surface tractions,  $W_L$  models the interface conditions along the phase boundaries for “solid-solid” problems, and  $W_k$  is due to a fictitious spring model to pin free floating material in “solid-void problems”.

The weak form of the structural equilibrium equations is:

$$\begin{aligned} W_S = & \int_{\Omega_A} \boldsymbol{\eta} : \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega + \int_{\Omega_B} \boldsymbol{\eta} : \boldsymbol{\sigma}(\mathbf{u}) \, d\Omega - \int_{\Omega_A} \mathbf{v} \cdot \mathbf{b} \, d\Omega \\ & - \int_{\Omega_B} \mathbf{v} \cdot \mathbf{b} \, d\Omega - \int_{\Gamma_N} \mathbf{v} \cdot \mathbf{f} \, d\Gamma_N, \end{aligned} \quad (4.4)$$

where  $\mathbf{v}$  is the kinematically admissible test function,  $\boldsymbol{\eta}$  is the strain tensor associated with the test function  $\mathbf{v}$ ,  $\mathbf{u}$  is the displacement vector,  $\boldsymbol{\sigma}$  is the stress tensor,  $\mathbf{b}$  is the applied body force, and  $\mathbf{f}$  is the external traction applied along  $\Gamma_N$ .

To enforce continuity of the displacements along the phase boundaries, the static equilibrium equations are typically augmented by either an enhanced Lagrange multiplier or penalty formulations, such as the stabilized Lagrange multiplier and the Nitsche method. Note that the standard Lagrange multiplier approach is not suitable for the XFEM as it suffers from numerical instabilities. The reader is referred to Stenberg [104], Juntunen and Stenberg [48], and Dolbow and Harari [24] for more details.

In this paper, we enforce displacement continuity along phase boundaries for “solid-solid” problems via the following stabilized Lagrange multiplier method [64]:

$$W_L = - \int_{\Gamma_{A,B}} [\mathbf{v}] \cdot \boldsymbol{\lambda} \, d\Gamma_{A,B} + \gamma \int_{\Gamma_{A,B}} \boldsymbol{\mu} \cdot [\mathbf{u}] \, d\Gamma_{A,B} + \int_{\Gamma_{A,B}} \mu \cdot (\boldsymbol{\lambda} - \bar{\boldsymbol{\sigma}} \cdot \mathbf{n}_{A,B}) \, d\Gamma_{A,B}, \quad (4.5)$$

$$[\mathbf{u}] = \mathbf{u}^{(A)} - \mathbf{u}^{(B)}, \quad [\mathbf{v}] = \mathbf{v}^{(A)} - \mathbf{v}^{(B)}, \quad (4.6)$$

$$\bar{\boldsymbol{\sigma}} = \frac{1}{2} (\boldsymbol{\sigma}^{(A)} + \boldsymbol{\sigma}^{(B)}), \quad (4.7)$$

where  $\boldsymbol{\lambda}$  is the Lagrange multiplier, and  $\boldsymbol{\mu}$  is the associated test function. The higher the weight  $\gamma$  is, the better the interface condition is satisfied, at the cost of numerical stability.

In “solid-void” topology optimization problems, free-floating solid particles surrounded by void material may emerge, leading to a singular analysis problem. This issue does not exist in an Ersatz material approach as the void phase is modeled via a soft material. A similar approach can be applied to the XFEM to suppress singularities and the “void” phase can be modeled via a soft material [120]. However, this approach requires accounting for the interface contributions (4.5) and integrating the governing equations over the void phase. To avoid the associated complexity and computational costs, we extend the approach of Makhija and Maute [64] onto three dimensions and assume that the solid phase is supported by weak fictitious springs. This model leads to the following contribution to the governing

equations, assuming that phase “A” is the solid phase:

$$W_k = \int_{\Omega_A} k \mathbf{v} \cdot \mathbf{u} d\Omega, \quad (4.8)$$

where  $k$  denotes the stiffness of the distributed system of springs.

For a more detailed explanation of this XFEM formulation, the reader is referred to the paper by Makhija and Maute [64].

### 4.3.2 Discretization

To capture the discontinuities in the strain and stress fields along the phase boundaries, we enrich the standard finite element approximation with additional shape functions. We adopt the generalized enrichment strategy of Makhija and Maute [64] which resolves consistently the displacement fields in the presence of small features and does not suffer from artificially coupling disconnected phases. Considering a two-phase problem, the displacement field is approximated as follows:

$$\mathbf{u}(\mathbf{x}) = \sum_{m=1}^M \left( H(-\phi) \sum_{i=1}^n \mathbf{N}_i \mathbf{u}_{i,m}^A + H(\phi) \sum_{i=1}^n \mathbf{N}_i \mathbf{u}_{i,m}^B \right), \quad (4.9)$$

where  $m$  is the enrichment level,  $M$  is the maximum number of enrichment levels used for each phase,  $\mathbf{N}$  are the shape functions,  $\mathbf{u}_{i,m}^l$  is the vector of nodal displacement components at node  $i$  for phase  $l = [A, B]$ ,  $\phi$  is the level set value evaluated at the integration point, and  $H$  denotes the Heaviside function. The enrichment level is chosen such that the displacements in disconnected volumes of the same phase are interpolated by separate sets of degrees of freedom. When interpolating the level set field by element-wise linear functions, a maximum of 14 enrichment levels is needed in three dimensions. This enrichment strategy will be revisited in Section 4.5.

The Heaviside function  $H$  depends on the level set function and is defined as follows:

$$H(z) = \begin{cases} 1 & z > 0, \\ 0 & z \leq 0, \end{cases} \quad (4.10)$$

The Heaviside functions “turns on/off” the standard finite element interpolations in the particular phases. The approximation (4.9) allows for discontinuities of the displacements along the phase boundaries. Therefore the continuity is enforced weakly via the stabilized Lagrange multiplier method (4.5).

Following a Bubnov-Galerkin scheme, we test the governing equations with the same subspace as we use for the trial functions; see Eq. 4.9. The weak form of the governing equations is integrated numerically over the individual phases, using the Delaunay triangulation of the element along the phase boundaries.

### 4.3.3 Preconditioner

As described above, the degrees of freedom  $\mathbf{u}_{i,m}^l$  interpolate the structural displacements in topologically connected subdomains of phase  $l$  in the elements connected to node  $i$ . As the total volume of these subdomains vanishes, the discretized structural model becomes increasingly ill-conditioned; i.e. the condition number of the stiffness matrix rapidly increases. This phenomenon is more pronounced in three dimensional problems than in two dimensional ones.

To mitigate this ill-conditioning issue, we expand the geometric preconditioning scheme of Lang et al [56], which was introduced and studied for two dimensional heat conduction and flow problems, onto three dimensional problems in structural mechanics. The goal of this preconditioning scheme is to balance the influence of all degrees of freedom in the system, as the volumes in which the subset of these degrees of freedoms interpolates the solution

approach zero. To this end, we introduce the following projection:

$$\tilde{\mathbf{u}} = \mathbf{T}\mathbf{u}, \quad (4.11)$$

where  $\mathbf{u}$  is the vector of displacement degrees of freedom according to Eq. 4.9,  $\mathbf{T}$  is a transformation matrix, and  $\tilde{\mathbf{u}}$  is the solution vector in the transformed space. The residual,  $\tilde{\mathbf{R}}$ , and stiffness matrix,  $\tilde{\mathbf{K}}$ , in the transformed space are defined as:

$$\tilde{\mathbf{R}} = \mathbf{T}^T \mathbf{R} \quad \tilde{\mathbf{K}} = \mathbf{T}^T \mathbf{K} \mathbf{T}, \quad (4.12)$$

where the residual,  $\mathbf{R}$ , and the stiffness matrix,  $\mathbf{K}$ , result from integrating the weak form of the governing equations using the XFEM approximation (4.9).

The preconditioner  $\mathbf{T}$  is a diagonal matrix built by integrating the spatial derivatives of the shape functions over the nodal support of nodes connected to an intersected element. The diagonal components of the matrix are defined as:

$$\mathbf{T}_{i,m}^l = \left( \max_{e \in E_i} \frac{\int_{\mathcal{D}_i^e} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x}) d\mathbf{x}}{\int_{\mathcal{D}_i^e} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x}) d\mathbf{x}} \right)^{-1/2}, \quad (4.13)$$

where  $\mathbf{T}_{i,m}^l$  corresponds to the degree of freedom  $\mathbf{u}_{i,m}^l$ ,  $i$  is the node index,  $l = [A, B]$  is the material phase,  $m$  is the enrichment level,  $E_i$  is the set of elements connected to node  $i$ , and  $\mathcal{D}_i^e$  is the element domain of phase  $l$ . The components of the matrix increase as the region of influence of a degree of freedom decreases. The entries  $T_{i,m}^l$  of nodes  $i$  that are not connected to at least one intersected element are set to one.

To avoid numerical issues due to large values for the components of  $\mathbf{T}$ , the degrees of freedom associated with the diagonal entry  $\mathbf{T}_{i,m}^l$  are constrained to zero if the following condition is satisfied:

$$\mathbf{T}_{i,m}^l \geq T_{tol}, \quad (4.14)$$

where  $T_{tol}$  is a specified tolerance. As studies by Lang et al [56] have shown, the above preconditioning scheme is rather insensitive to the value of  $T_{tol}$  and is typically set to a value larger than  $10^8$ . For more details on this formulation, the reader is referred to the paper by Lang et al [56].

## 4.4 Optimization Model

The design optimization problems considered in this paper can be written as follows:

$$\begin{aligned} & \min_{\mathbf{s}} \mathcal{F}(\mathbf{s}, \mathbf{u}(\mathbf{s})), \\ \text{s.t. } & \begin{cases} \mathbf{s}, & \text{subject to design constraints } \mathcal{G}_j \leq 0, \\ \mathbf{u}, & \text{solves } W = 0 \text{ for a given } \mathbf{s}, \end{cases} \end{aligned} \quad (4.15)$$

where  $\mathbf{s}$  denotes the vector of design variables,  $\mathcal{F}$  the objective function, and  $\mathcal{G}_j$  the  $j$ -th design constraint. In general, the objective and constraints depend on the optimization and state variables. The optimization problem (4.15) is solved by nonlinear programming methods, and the gradients of the objective and constraint functions are computed via the adjoint method.

In this paper, we compare the proposed LSM-XFEM approach against the well-known SIMP method, augmented by a projection scheme. In the following subsections, we briefly outline the models that define the discretized level set field and the material properties as function of the optimization variables in the LSM-XFEM and the SIMP method, respectively.

#### 4.4.1 XFEM

The nodal values of the discretized level set field are defined as analytical functions of the optimization variables via the following linear filter:

$$\phi^n(\mathbf{s}) = \frac{\sum_{i=1}^P \mathbf{w}_i^n \mathbf{s}_i}{\sum_{i=1}^P \mathbf{w}_i^n}, \quad (4.16)$$

with

$$\mathbf{w}_i^n = \max(0, r_\phi - \| \mathbf{x}_i - \mathbf{x}_n \|), \quad (4.17)$$

where  $\phi^n$  is the level set value at node  $n$ ,  $\mathbf{x}_n$  is the position vector of node  $n$ ,  $\mathbf{x}_i$  is the location of the node at which the design variable  $i$  is defined,  $\mathbf{w}_i^n$  is the weight of node  $n$  with respect to design variable  $i$ ,  $r_\phi$  is the filter radius, and  $P$  is the number of nodes in the computational mesh.

The above linear filter was used previously in the studies of Kreissl and Maute [53] and Makhija and Maute [64], and was shown to improve the convergence rate in the optimization process. However, in contrast to density or sensitivity filters used in SIMP methods, the filter above is not guaranteed to control the minimum feature size. This issue will be revisited in Section 4.6.

#### 4.4.2 SIMP

Here, the material distribution is parameterized by nodal density values,  $\rho_i$ , which are treated as optimization variables, i.e.  $\rho_i = \mathbf{s}_i$ . Following the work of Guest et al [40], we compute the elemental density by combining a linear density filter and a projection scheme as follows:

$$\rho^e(\mathbf{s}) = \frac{\sum_{i=1}^E \mathbf{w}_i^e \mathbf{s}_i}{\sum_{i=1}^E \mathbf{w}_i^e}, \quad (4.18)$$

with

$$\mathbf{w}_i^e = \max(0, r_\rho - \| \mathbf{x}_i - \mathbf{x}_e \|), \quad (4.19)$$

where  $\rho^e$  is the elemental density of element  $e$ ,  $\mathbf{x}_e$  is the position vector of the centroid of element  $e$ ,  $\mathbf{x}_i$  is the location of the node at which the design variable  $i$  is defined,  $\mathbf{w}_i^e$  is the factor of element  $e$  with respect to design variable  $i$ ,  $r_\rho$  is the filter radius, and  $E$  is the number of elements in the computational mesh.

Guest et al [40] proposed a density projection method to reduce the volume occupied by material with intermediate densities. The projection is based on a smoothed Heaviside function and applied to the elemental densities as follows:

$$\hat{\rho}^e(\mathbf{s}) = 1 - e^{-\beta\rho^e(\mathbf{s})} + \rho^e(\mathbf{s})e^{-\beta} \quad (4.20)$$

where  $\hat{\rho}^e$  is the projected elemental density, and the parameter  $\beta \geq 0$  controls the crispness of the projection. For  $\beta = 0$  the projection turns into an identity operator, i.e.  $\hat{\rho}^e = \rho^e$ .

The Young's modulus,  $E$ , is defined as a function of the density,  $\hat{\rho}^e$ , using the standard SIMP interpolation:

$$E(\mathbf{x}) = E_B + |E_A - E_B| \hat{\rho}^e(\mathbf{s})^p \quad (4.21)$$

where  $E_A$  and  $E_B$  are the Young's moduli for material phase "A" and "B", and  $p$  is the SIMP penalization factor. To model a "solid-void" optimization problem,  $E_B$  is set to value much smaller than  $E_A$ .

The filter (4.18) prevents the formation of features smaller than  $r_\rho$ , typically at the cost of generating intermediate density values along the phase boundaries. This effect is mitigated by the projection (4.20) which, in the limit for  $\beta \rightarrow \infty$ , maps non-zero  $\rho^e$  values into "1". The reader is referred to the papers by Guest et al [40] and Guest et al [41] for further details of the scheme presented above, and to Sigmund and Maute [101] for a comprehensive discussion of projection schemes.

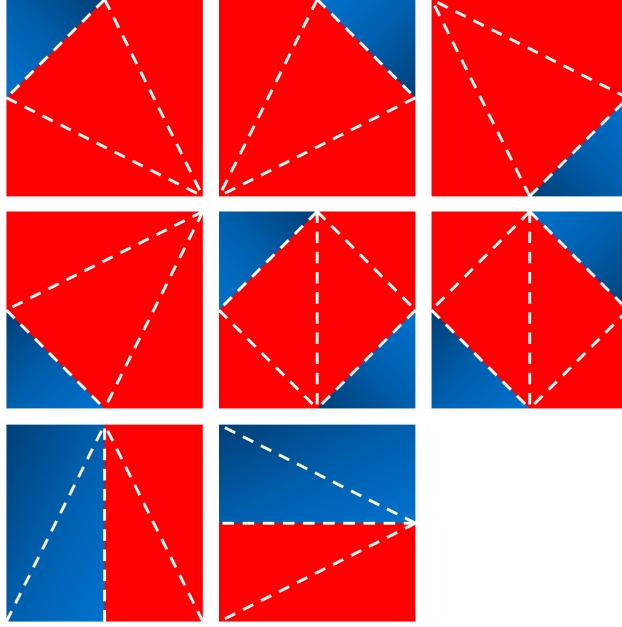


Figure 4.3: Intersection patterns for a two-dimensional QUAD4 element.

## 4.5 Computational Considerations

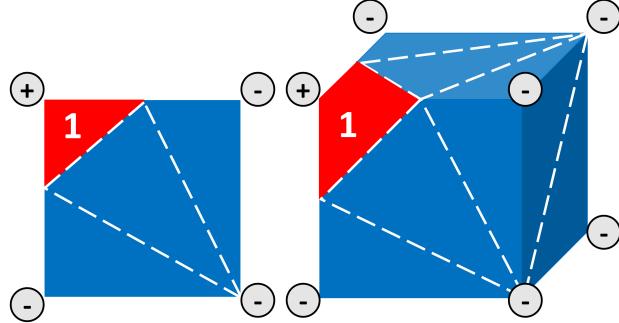


Figure 4.4: Initial assignment of enrichment levels.

Expanding the LSM-XFEM combination onto three dimensional problems faces both algorithmic and computational challenges which are briefly discussed below. The XFEM requires integrating the weak form of the governing equations separately in each phase. To this end, an element intersected by the zero level set contour is subdivided. For two dimensional problems and using a linear interpolation of the level set field within an element, there are only 8 intersection configurations which can be tabulated; see Fig. 4.3. In three dimensions, there are 127 intersection configurations. To handle this complexity, we compute

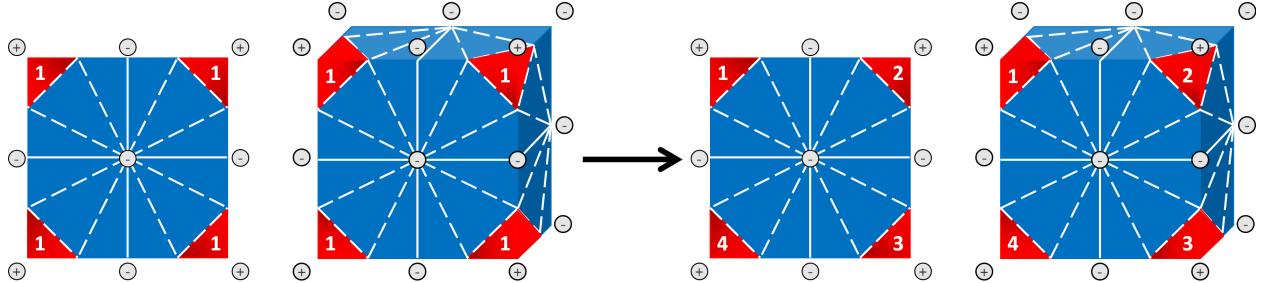


Figure 4.5: Initial and final assignment of enrichment levels considering a cluster of elements.

the intersection point of the zero level set contour with the element edges and use a Delaunay triangulation to subdivide the element. In numerical experiments, this approach has proven robust and computationally inexpensive.

Previous studies on topology optimization for three dimensional structures with the XFEM [60] have employed a simplified enrichment scheme which is limited to “solid-void” problems and may suffer from artificial coupling of disconnected material. Our work overcomes these issues by adopting the generalized enrichment scheme summarized in Section 4.3.2. The key challenge of this scheme is to identify the enrichment levels needed to consistently interpolate the displacements in elemental subdomains with the same phase.

To this end, the subdomains in all elements connected to a node need to be considered. This naturally leads to an algorithm which loops over all nodes and, in an inner loop, over all elements connected to the current node. As this approach processes an element repeatedly, the following simple and efficient two-step scheme is introduced:

1. A temporary, elemental enrichment level is assigned to the subdomains in each element. Recall that the enrichment level defines the set of degrees of freedom used to interpolate the displacements in an elemental subdomain. Because this assignment is done individually for each element, the continuity of the interpolation across elements is not guaranteed. Fig. 4.4 shows the triangulation and enrichment level for the red phase in two and three dimensions.
2. The nodal enrichment levels are constructed to ensure that the displacement field is

interpolated continuously across elements, and by a different set of shape functions for each disconnected elemental subdomain of the same phase. To this end, the cluster of elements connected to a node is considered, and the elemental enrichment levels assigned in step 1 are adjusted to satisfy the continuity and consistency conditions.

This process is illustrated in Fig. 4.5. The node of interest is the one located in the center of the element cluster. In step 1 each subdomain of the red phase is assigned an enrichment level of  $m = 1$ . Applying this enrichment level to the degrees of freedom for the red phase at the center node would incorrectly couple the displacement fields in the red phase subdomains. Analyzing the element cluster around the center nodes shows that these subdomains are disconnected and individual enrichment level are assigned.

Topology optimization in three dimensions leads to FEM or XFEM models with a large number of degrees of freedom, and typically requires using iterative solvers and parallel computing. The stabilized Lagrange multiplier formulation of the interface conditions (4.5) leads to a non-symmetric stiffness matrix. Numerical experiments have shown that the XFEM problems considered in this study can be robustly and efficiently solved by a generalized minimal residual (GMRES) method preconditioned by incomplete LU (ILU) factorization. Note that the ILU preconditioner operates on the projected XFEM system (4.12).

## 4.6 Numerical Examples

We study the features of the proposed LSM-XFEM topology optimization approach with numerical examples. The LSM-XFEM results of “solid-void” and “solid-solid” problems are compared against the ones of the SIMP approach outlined in Section 4.4.2. In all examples we seek to minimize the strain energy subject to a constraint on the volume of the stiff phase. This problem formulation is chosen because it is well studied in the literature and the numerical experiments can be easily repeated. The following numerical studies will provide insight into (a) the convergence of the geometry and the structural response as the

meshes are refined and (b) the influence of regularization techniques on the optimized results, such as the filter radii in (4.16) and (4.18), and perimeter constraints.

In all examples, the optimization problems are solved by the Globally Convergent Method of Moving Asymptotes (GCMMA) of Svanberg [109]. The sensitivities are computed by the adjoint method. The design domains are discretized by 8-node linear elements. The linear systems of the forward and adjoint problems are solved by a parallel implementation of the GMRES method [46]. The problems are preconditioned by an ILU factorization with a fill of 2.0 and an overlap of 1.0. The convergence tolerances for both, the GCMMA and the GMRES solver, are chosen sufficiently low such that the optimization results do not depend on the tolerance values. In the SIMP problems, the parameters  $p$  and  $\beta$  are kept constant in the optimization process, i.e. no continuation approach is used. In the LSM-XFEM examples, the spring stiffness value,  $k$ , is  $10^{-6}$ . Geometric and material parameters are given in non-dimensional and self-consistent units.

While the LSM-XFEM results can be directly used to fabricate the structure, for example by 3-D printing, the SIMP results need to be post-processed. From a practitioner perspective, only the post-processed SIMP results should be compared against the LSM-XFEM results. To this end, we post-process the SIMP results with a lumping method that uses the iso-contours of the density distribution. To obtain a strict “0-1” density distribution with smooth phase boundaries, we construct iso-contours for different threshold values,  $\rho_T$ , from the nodal density values,  $\rho_i$ ; see Section 4.4.2. The volume enclosed by the iso-contour with  $\rho \geq \rho_T$  is considered solid; the remaining volume is considered “void”. We select the threshold value that results in the smallest strain energy and for which the volume constraint is satisfied. The structural response of the design for different  $\rho_T$  values is analyzed conveniently with the XFEM. We refer to this post-processing approach as iso-contour density lumping (IDL).

To gain further insight into the crispness of the SIMP results and the influence of the post-processing methods above on their performance, we measure the volume fraction,  $\bar{\rho}$ ,

occupied by elemental densities with  $0 < \hat{\rho}^e < 1$  as follows:

$$\bar{\rho} = \frac{1}{\int_{\Omega_D} d\Omega_D} \int_{\Omega_D} \hat{\rho}^e (1 - \hat{\rho}^e) d\Omega_D \quad (4.22)$$

where  $\Omega_D$  denotes the design domain.

#### 4.6.1 Cube with center load

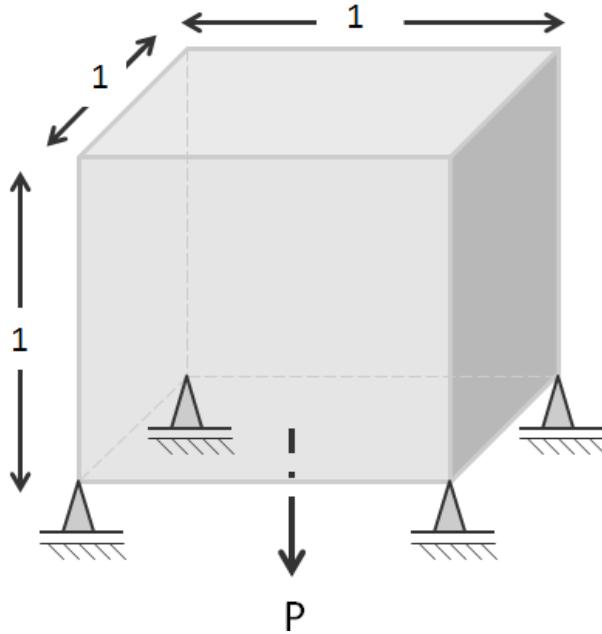


Figure 4.6: Cube with center load.

We consider the “solid-void” optimization problem depicted in Fig. 4.6. With this example we will illustrate the basic features of the LSM-XFEM approach for three dimensional problems and show that the proposed LSM-XFEM approach and the SIMP formulation may exhibit comparable convergence behaviors as the mesh is refined.

The  $1 \times 1 \times 1$  cubical design domain is pinned at its four bottom corners in the vertical direction and a unit force is applied at the center of the bottom face. The Young’s modulus of the stiff phase is set to 1 and the Poisson ratio to 0.3. The maximum volume of the stiff phase is 10%. We compare LSM-XFEM and SIMP results for two mesh sizes:  $24 \times 24 \times 24$

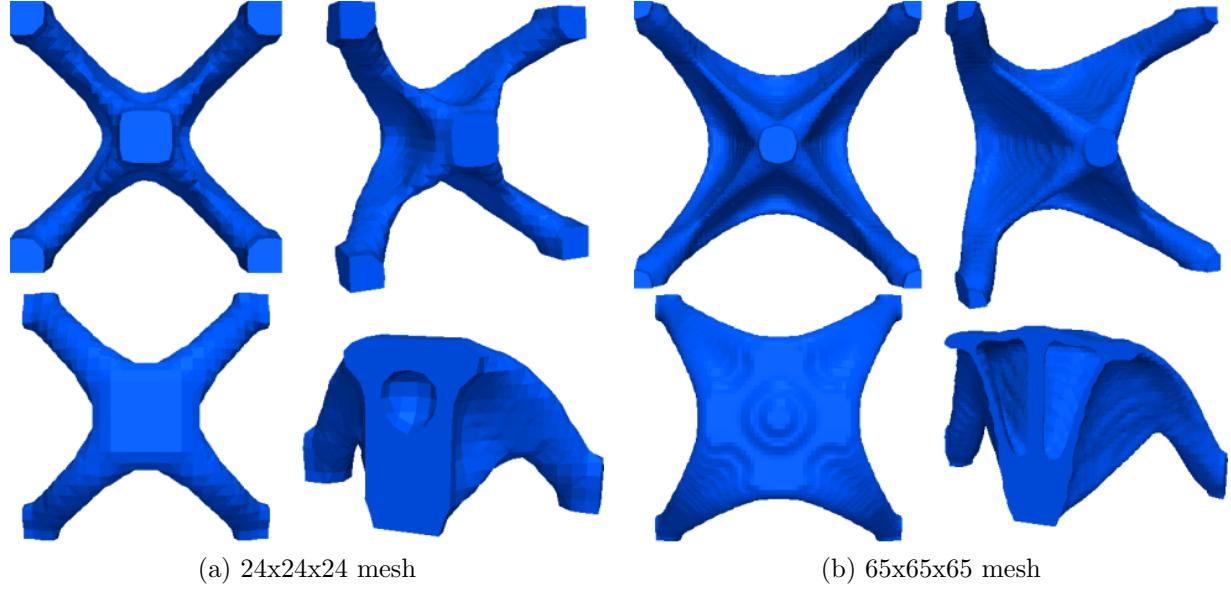


Figure 4.7: SIMP results for cube with center load problem; clockwise: bottom, side, top, and clipped views.

and  $65 \times 65 \times 65$ . The problem is solved by analyzing the entire design domain, i.e. we do not restrict the solution to a symmetric design.

First, we apply the SIMP approach with a penalization factor of  $p = 3$ . The Young's modulus of the void phase is set to  $E_B = 10^{-9}$ . The size of the smoothing radius is mesh dependent, and is set to  $r_\rho = 3.2$  for the coarse mesh and  $r_\rho = 1.182$  for the fine mesh; the projection parameter is set to  $\beta = 0$ . Note that the smoothing radius is intentionally set relative to the element size ( $1.6 \times$  element edge length). While this approach does not ensure mesh-independent optimization results, it still prevents the formation of checkerboard patterns and provides insight into the dependency of the geometry resolution of SIMP as the mesh is refined. The design domain is initialized with a uniform material distribution of  $\rho_i = 0.1$ . The optimized material distributions are shown in Figure 4.7 where material with a density lower than  $\rho_i < 0.75$  is considered void. The strain energies are reported in Tab. 4.1. For both meshes the volume constraint is active in the converged designs. As expected, the optimized geometry is smoother and the strain energy is lower for the refined mesh. The SIMP results for the coarse and fine mesh are post-processed with the IDL

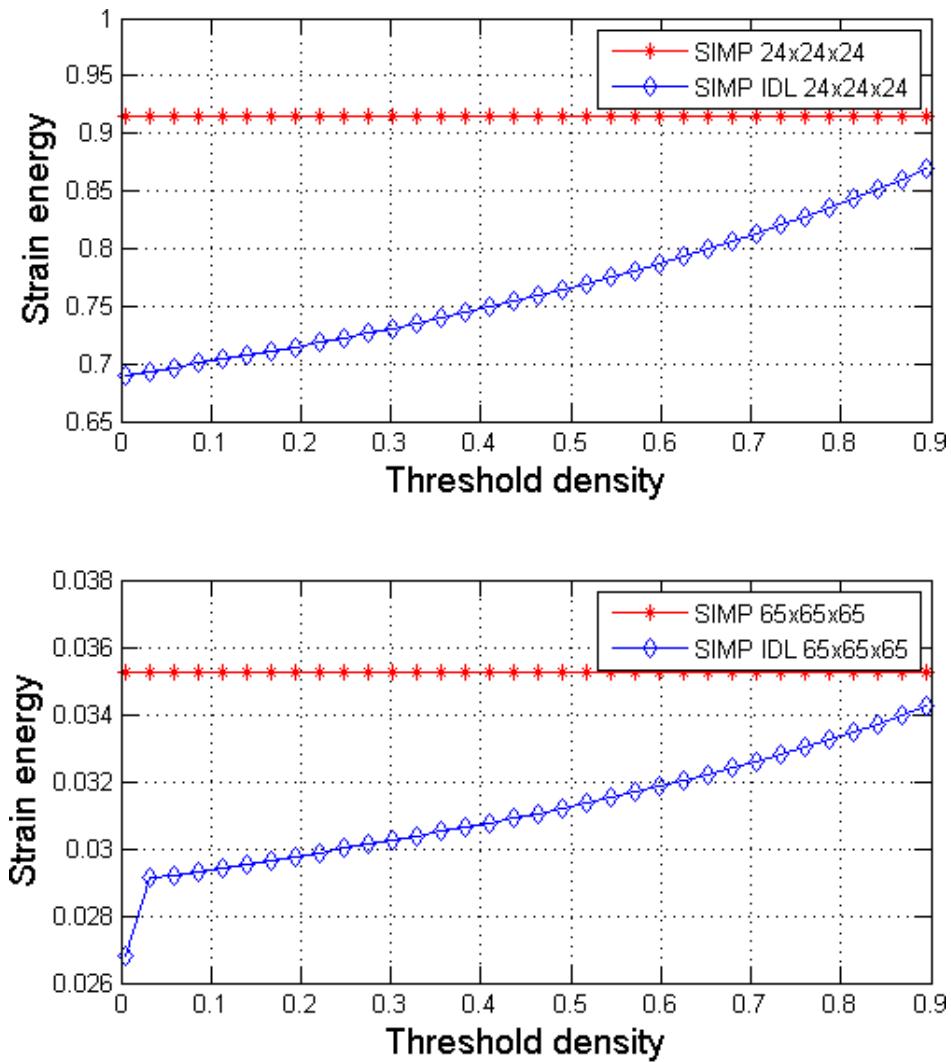


Figure 4.8: IDL post-processing of SIMP results for cube with center load problem.

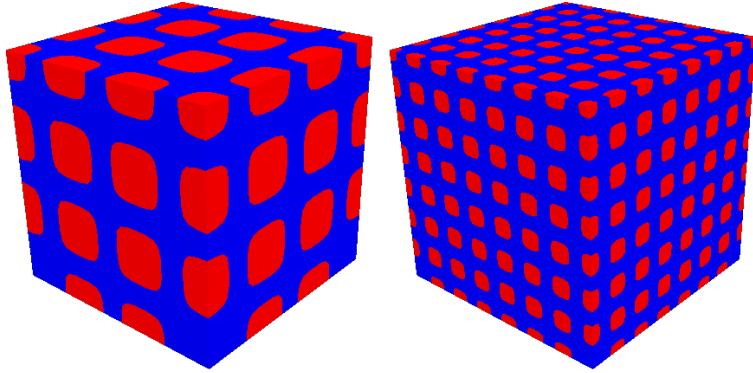


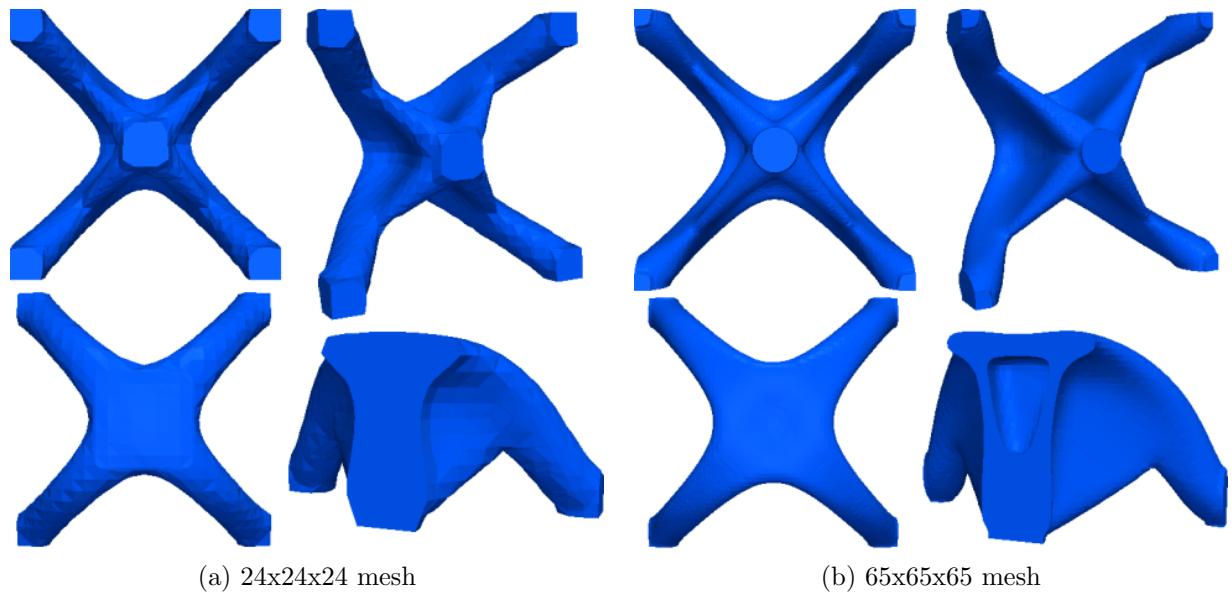
Figure 4.9: Initial level set configurations for cube with center load problem.

	Mesh size	Strain energy
SIMP	$24 \times 24 \times 24$	9.1456e-01
	$65 \times 65 \times 65$	3.5244e-02
XFEM	$24 \times 24 \times 24$	1.0082e+00
	$65 \times 65 \times 65$	3.5519e-02

Table 4.1: Comparison of strain energies of SIMP and LSM-XFEM results for cube with center load problem; the corresponding designs are shown in Figs. 4.7 and 4.10.

approach described above. The strain energies for varying threshold values,  $\rho_T$ , are plotted in Fig. 4.8. The volume constraint is met for  $\rho_T = 0.78$  for the coarse mesh and  $\rho_T = 0.44$  for the fine mesh. For these threshold values, the strain energies of SIMP-IDL designs are 4.8939% and 12.1625% lower than the ones of the raw SIMP results for the coarse and fine meshes, respectively. The value of  $\rho_T$  is higher for the coarse mesh because it cannot converge to a design with void inclusions. In both cases the strain energies of the post-processed results match well the SIMP predictions as the density distributions converged well to “0-1” solutions.

The volume fractions of intermediate densities (4.22) are 0.285 and 0.0189 for the coarse and fine mesh, respectively. The post-processed designs have lower strain energies because the post-processing counteracts the effect of the density filter (4.18). The same optimization problem is solved with the proposed LSM-XFEM approach. The smoothing radius is set to  $r_\phi = 3.2$  for the coarse mesh and  $r_\phi = 1.182$  for the fine mesh. No perimeter constraint is imposed. We seed the initial design with two different configurations of void



(a) 24x24x24 mesh

(b) 65x65x65 mesh

Figure 4.10: LSM-XFEM results for cube with center load problem; clockwise: bottom, side, top, and clipped views.

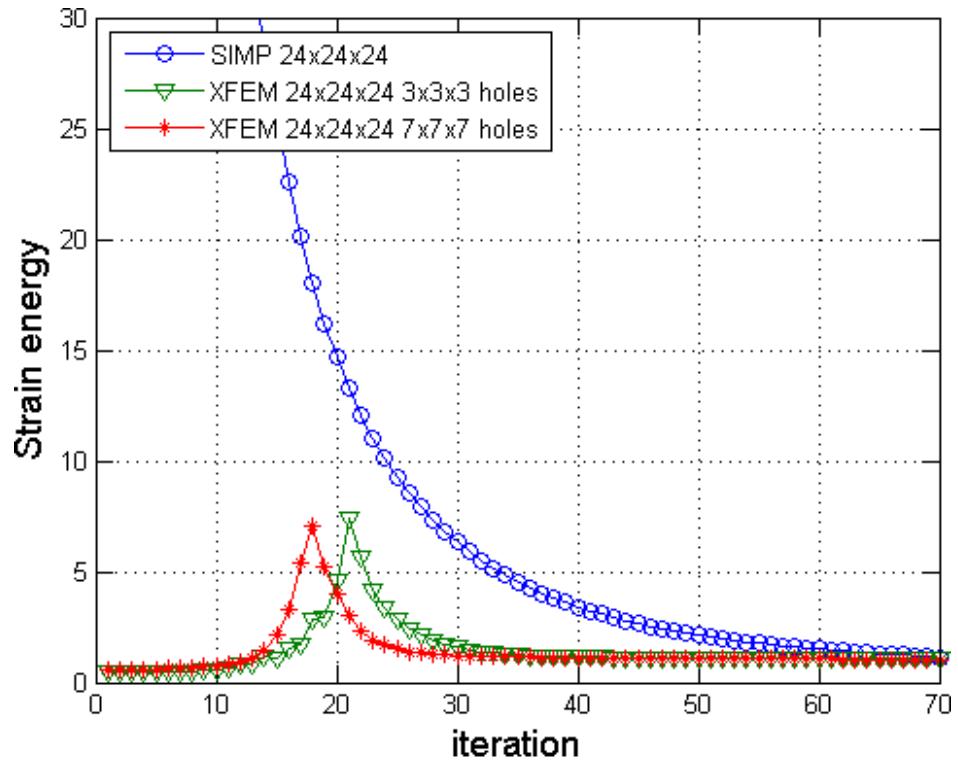


Figure 4.11: Evolution of strain energies in the optimization process for SIMP and LSM-XFEM approaches.

inclusions to study the influence of the initial layout on the optimization results. For both configurations we start from an equally spaced array of square-shaped holes with rounded corners, by modifying Eq. 4.2 into the following:

$$\phi_i = (x_i - x_c)^{10} + (y_i - y_c)^{10} + (z_i - z_c)^{10} - r^{10}. \quad (4.23)$$

One configuration has  $3 \times 3 \times 3$  equally spaced holes with radius 5.50, the other  $7 \times 7 \times 7$  holes with radius 2.0, as shown in Fig. 4.9. In both cases, the volume constraint is not satisfied with the initial design. Note that no inclusions are placed at the four bottom corners where the boundary conditions are applied.

Both level set configurations converge to nearly indistinguishable designs and strain energy values, for both the coarse and fine meshes. The optimized designs are shown in Fig. 4.10. The strain energies of the optimized designs are given in Tab. 4.1. The convergence history for the coarse meshes in SIMP and LSM-XFEM is shown in Fig. 4.11.

For the example considered here, the SIMP and LSM-XFEM results match well, both in regards to the geometry and the strain energy values. The LSM-XFEM approach shows a faster convergence as the mesh is refined. Comparing the optimized geometries, the SIMP results contain more structural features for both mesh resolutions. For example, considering the fine mesh, the SIMP method generates two small holes in the webs connecting the supports to the load point, while the LSM-XFEM approach leads to only one larger hole, independent of the initial design configuration. However, these small differences have only a minor impact on the structural performance, i.e. the strain energy, of the optimized designs.

Considering the conceptual structural layout, both, the SIMP and the LSM-XFEM approach, display only minor mesh dependencies for the problem studied here. Although the strain values show significant differences, the optimized geometries obtained with the coarse and fine meshes differ insignificantly for the SIMP and LSM-XFEM approach. The following example will demonstrate a less benign convergence and identify more pronounced differences

between the SIMP and LSM-XFEM methods.

#### 4.6.2 Cuboid under torsion

The second “solid-void” example is taken from Nguyen et al [78] and reveals differences in the SIMP and LSM-XFEM approaches. We will show that, without imposing a mesh-independent minimum feature size constraint, the proposed LSM-XFEM approach may converge to a design with a significantly lower strain energy than the SIMP method employed in this paper. However, we will also illustrate that our LSM-XFEM approach suffers from a lack of a robust and intuitive shape control technique.

The design domain is a cuboid of size  $4 \times 1 \times 1$ , as shown in Fig. 4.12. A torque moment is generated via 4 unit loads acting at the centers of the edges of the top face. The design domain is clamped at the bottom face. The Young’s modulus is set to 1.0 and the Poisson ratio to 0.3. The volume of the stiff phase is constrained to 10% of the total volume. The problem is solved on the full mesh.

#### Mesh convergence study

The optimization problem is solved with the SIMP approach for four different mesh sizes:  $40 \times 10 \times 10$ ,  $60 \times 15 \times 15$ ,  $80 \times 20 \times 20$ , and  $120 \times 30 \times 30$ . The Young’s modulus of the void phase is set to  $E_B = 10^{-9}$ . The design domain is initialized with a uniform material distribution of  $\rho_i = 0.1$ . The penalization factor is  $p = 3$ . First we consider a projection parameter of  $\beta = 0$  and scale the smoothing radius with the element size:  $r_\rho = 1.6 \times$  the element edge length.

The optimized material distributions are shown in Fig. 4.13 where material with a density lower than  $\rho_i < 0.35$  is considered void. The strain energies are reported in Tab. 4.2 and display the expected decrease in strain energy as the mesh is refined. For all meshes the volume constraint is active in the converged designs. As the mesh is refined, the evolution of the SIMP results shows an interesting discontinuity which is typically not observed for two

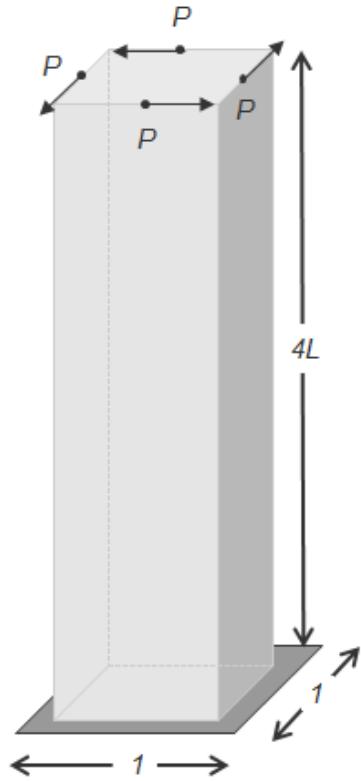
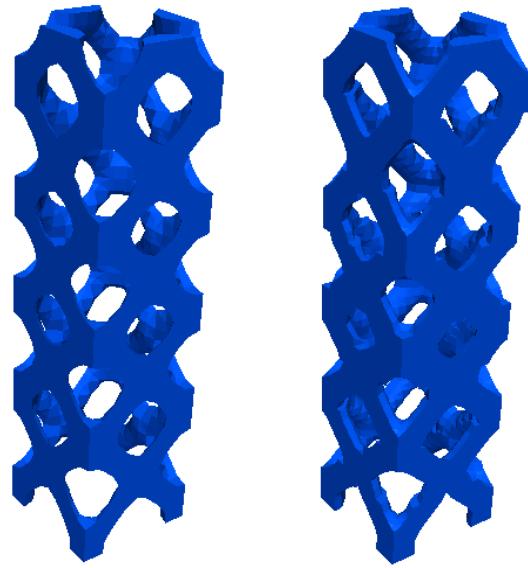


Figure 4.12: Cuboid under torsion model.

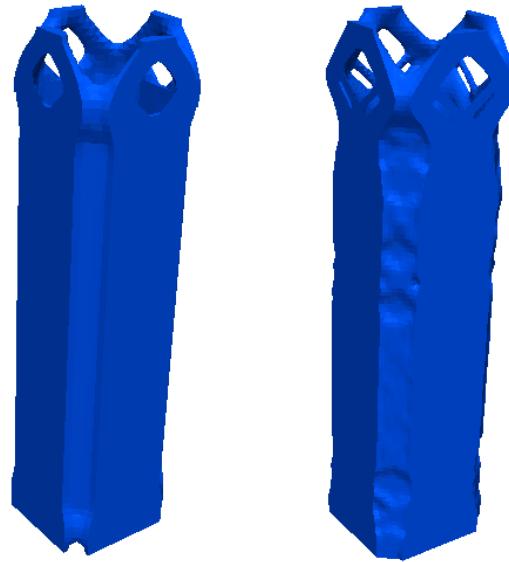
	Mesh size	Strain energy
SIMP	$40 \times 10 \times 10$	7.5195e+03
	$60 \times 15 \times 15$	4.2076e+03
	$80 \times 20 \times 20$	4.0298e+03
	$120 \times 30 \times 30$	2.6555e+03

Table 4.2: Strain energies of SIMP results for cuboid under torsion problem; SIMP parameters:  $p = 3$ ,  $r_\rho = 1.6$  of the element edge length, and  $\beta = 0$ ; the corresponding designs are shown in Fig. 4.13.



(a) 40x10x10

(b) 60x15x15



(c) 80x20x20

(d) 120x30x30

Figure 4.13: SIMP results of cuboid under torsion problem for different levels of mesh refinement; SIMP parameters:  $p = 3$ ,  $r_\rho = 1.6$  of the element edge length,  $\beta = 0$ .

	Mesh size	Strain energy
LSM-XFEM	$40 \times 10 \times 10$	$8.7551e + 02$
	$120 \times 30 \times 30$	$9.8262e + 02$

Table 4.3: Strain energies of LSM-XFEM results for cuboid under torsion problem; LSM-XFEM parameters:  $r_\phi = 1.6$  of the element edge length, no perimeter constraint; the corresponding designs are shown in Fig. 4.14.

dimensional problems. The optimized material layout switches abruptly from a grid-type structure, which conceptually agrees with the results of Nguyen et al [78], to a hollow square prism design. In contrast to two dimensional structures, where refining the mesh with a mesh-dependent filter radius leads to an ever increasing number of holes, in this example the opposite is the case. As the filter radius drops below a threshold, it is more advantageous to form a continuous thin outer wall rather than a grid-type structure. This behavior is a direct consequence of the combination of SIMP penalization and density smoothing. We will revisit this issue again later.

The LSM-XFEM results for a smoothing radius of  $r_\phi = 1.6 \times$  the element edge length are shown in Fig. 4.14. No perimeter constraint is applied to this problem. Here only the results for the coarsest and the finest meshes of the SIMP study above are shown. Note that in contrast to the SIMP results, the LSM-XFEM approach leads to conceptually equivalent design on both meshes. Refining the mesh only improves some local details. This feature is due to the ability of the LSM to represent thin structural features on coarse meshes. The thicknesses of the walls at half the height of the design domain are 0.0288 for the coarse mesh and 0.0276 for the fine mesh. The strain energies of the LSM-XFEM results are given in Tab. 4.3. The strain energy for the fine mesh is slightly larger than the one of the coarse mesh. This effect is due to the tendency of coarse finite element discretization over predicting the stiffness. The differences between the SIMP and LSM-XFEM results are significant. Although the discrepancy in strain energy decreases as the mesh is refined, the difference is large even for the two finer meshes where the SIMP and LSM-XFEM designs are similar. As the following investigation will show, the poorer performance of the SIMP

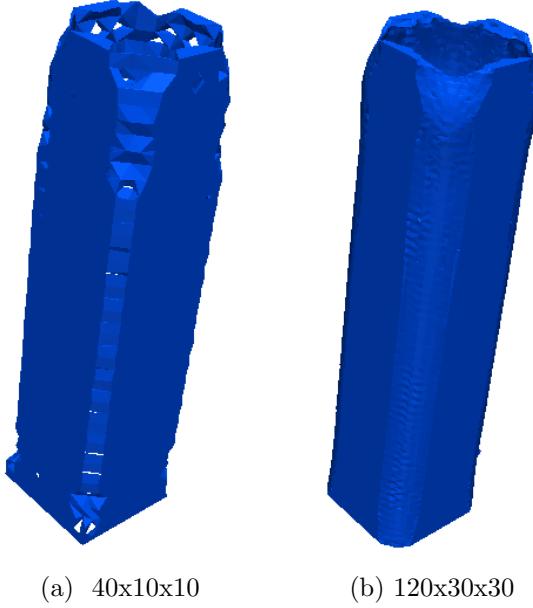


Figure 4.14: LSM-XFEM results of cuboid under torsion problem for two levels for mesh refinement; LSM-XFEM parameters:  $r_\phi = 1.6$  of the element edge length, no perimeter constraint.

results is primarily caused by the density filter, which prevents the material distribution to converge to a “0-1” result. First we study the influence of the projection scheme (4.20) on the SIMP results for the most refined mesh. The optimized material distributions for  $\beta = 4.0$  and  $\beta = 8.0$  are shown in Fig. 4.15, where material with a density lower than  $\rho_i < 0.35$  is considered void. For convenience the result for  $\beta = 0.0$  is shown again. Table 4.4 reports on the strain energies and the volume fractions of intermediate densities,  $\bar{\rho}$ , as the projection parameter,  $\beta$ , is increased. The higher  $\beta$ , the lower  $\bar{\rho}$  and the lower the strain energy, approaching the one of the LSM-XFEM result. Note that as  $\beta$  increases, the more holes emerge. The thickness of the walls for  $\beta = 8$  is 0.0434, which is smaller than the value for  $\beta = 0$ , 0.0447, and closer to the LSM-XFEM value. Instead of enforcing a better convergence toward a “0-1” solution by increasing the projection parameter  $\beta$ , we post-process the SIMP results for  $\beta = 0$  by the IDL post-processing method. Figure 4.16 shows the strain energy of the post-processed design over the threshold density,  $\rho_T$ , for the coarsest and the finest

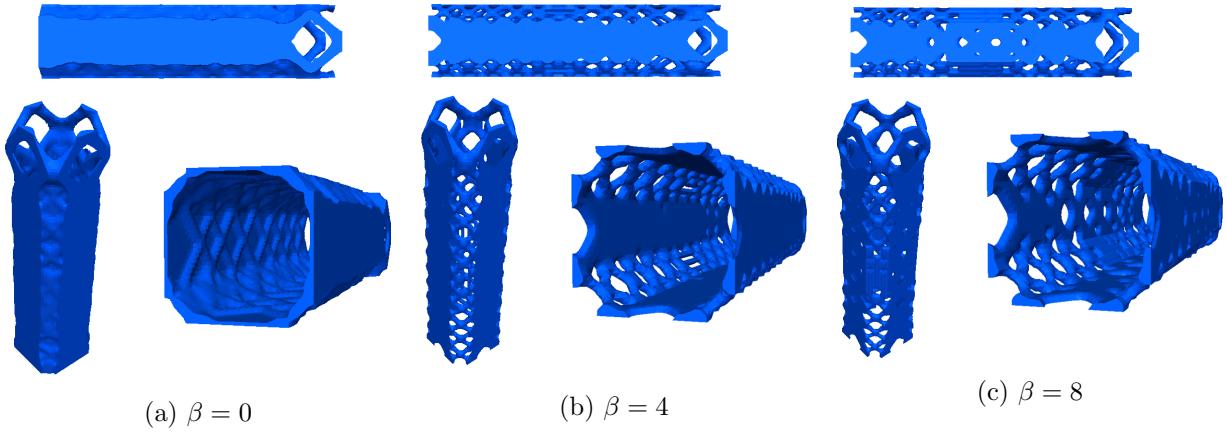


Figure 4.15: SIMP results for different projection parameters  $\beta = [0.0, 4.0, 8.0]$ ; mesh size:  $120 \times 30 \times 30$ .

	$\beta$ projection	Strain energy	$\bar{\rho}$ utilization
SIMP	0	2.6555e+03	4.0688e-02
	4	2.0264e+03	2.5131e-02
	8	1.9039e+03	1.9509e-02

Table 4.4: Strain energies of SIMP results for different projection parameters  $\beta = [0.0, 4.0, 8.0]$ ; the corresponding designs are shown in Fig. 4.15.

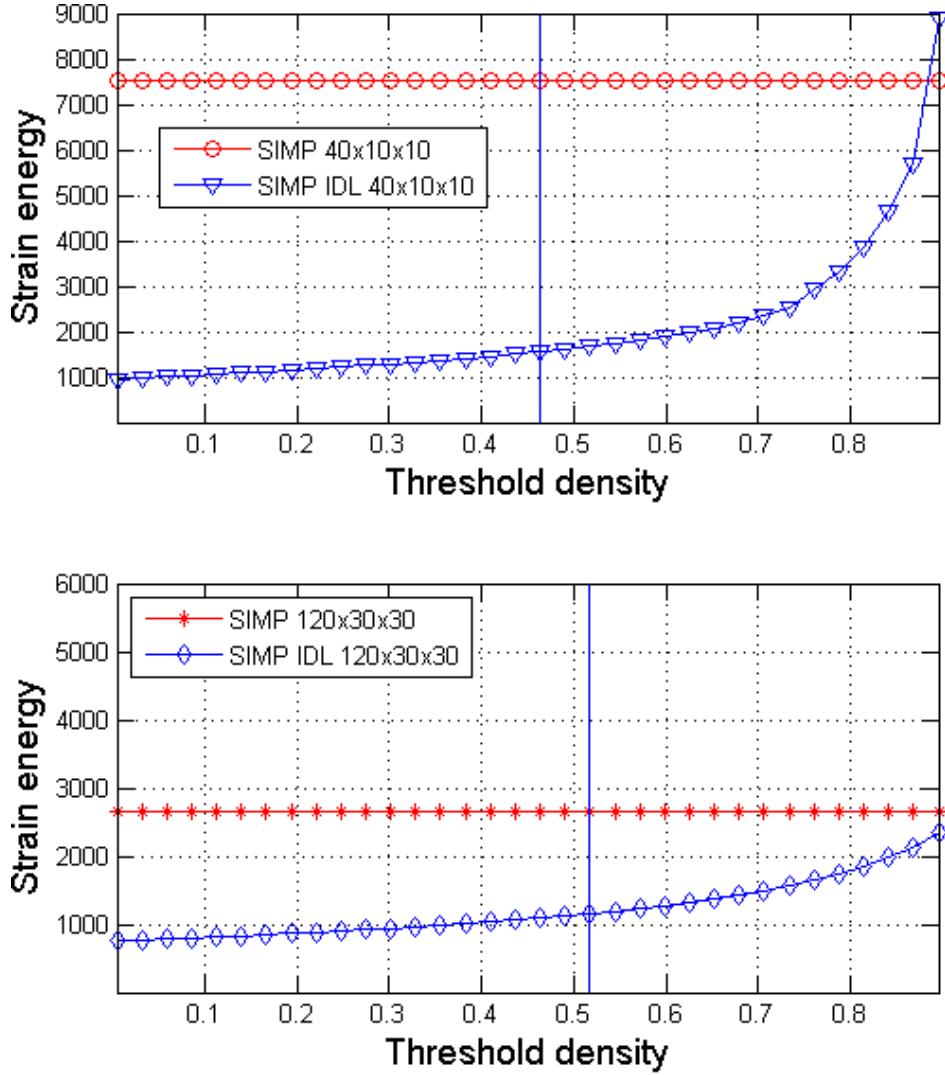


Figure 4.16: IDL post-processing of SIMP results for cuboid under torsion problem; the vertical lines mark the threshold values at which the volume constraint is satisfied.

mesh. The volume constraint is satisfied for a threshold value of  $\rho_T = 0.4634$  for the coarse mesh, and  $\rho_T = 0.5174$  for the fine mesh. For these threshold values, the strain energies of the SIMP-IDL designs are 70.6564% and 62.8167% lower than the ones of the raw SIMP results for the coarse and fine meshes, respectively. The associated strain energies are given in Tab. 4.5. The strain energy of the post-processed results of the fine mesh is rather similar to the result obtained for SIMP with  $\beta = 8.0$  in Tab. 4.4 and the LSM-XFEM results in Tab. 4.3. For the coarse mesh, the strain energy is well below the raw SIMP results from Tab. 4.2 but still above the results for the LSM-XFEM approach in Tab. 4.3. As we will see

	Mesh	$\rho_T$	Strain energy
SIMP	$40 \times 10 \times 10$	0.4634	$1.5601e + 03$
	$120 \times 30 \times 30$	0.5174	$1.1530e + 03$

Table 4.5: Strain energies of the SIMP-IDL designs of cuboid under torsion problem; the corresponding designs are shown in Fig. 4.13.

below, this is because of the larger smoothing radius which prevents the formation of smaller features and thinner walls.

### Feature size control

The mesh refinement study above suggests that the results of the LSM-XFEM approach are less sensitive to mesh refinement than the SIMP method without mesh-independent filtering. Geometric features, such as the thin walls, can be represented on coarse and fine meshes, independent of their size. This observation is in agreement with studies for two-dimensional problems, see for example Kreissl and Maute [53], but the phenomena is more pronounced and of greater importance for three dimensional problems. The lesser mesh sensitivity of the LSM-XFEM approach is in general a desired feature. In addition, however, the ability to control the minimum feature size is of importance for many applications, for example to account for manufacturing constraints and costs. The following study will show that the proposed LSM-XFEM approach currently lacks the ability to efficiently and intuitively control the local feature size. We first show that applying the same absolute filter radius in the SIMP formulation efficiently controls the feature size. Figure 4.17(a) shows the SIMP results on the  $120 \times 30 \times 30$  mesh for a projection parameter  $\beta = 0$ , a penalization factor of  $p = 3$ , and a smoothing radius of  $r_\rho = 0.16$  which is the same radius applied earlier for the coarsest mesh in Fig. 4.13(a). Comparing the SIMP results in Fig. 4.17(a) and Fig. 4.13(a) confirms the finding of numerous studies [11] that the SIMP approach leads to the same conceptual layout independent of the mesh refinement level if a mesh-independent filter is used. The strain energy of the design in Fig. 4.17(a) is given in Tab. 4.6. A similar effect is not observed in the LSM-XFEM approach when we apply the same filter radius,  $r_\phi$ ,

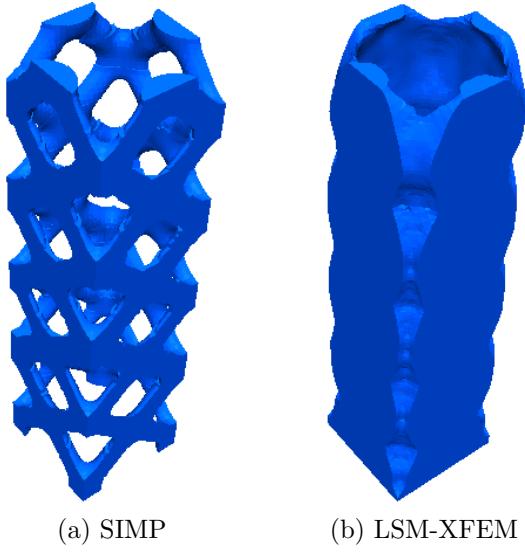


Figure 4.17: SIMP and LSM-XFEM results for larger smoothing radius; mesh size:  $120 \times 30 \times 30$ .

	Mesh size	Strain energy
SIMP	$120 \times 30 \times 30$	$6.5772e + 03$
LSM-XFEM	$120 \times 30 \times 30$	$8.2077e + 02$

Table 4.6: Strain energies of SIMP and LSM-XFEM results for cuboid under torsion problem using a mesh independent filter; the corresponding designs are shown in Fig. 4.17.

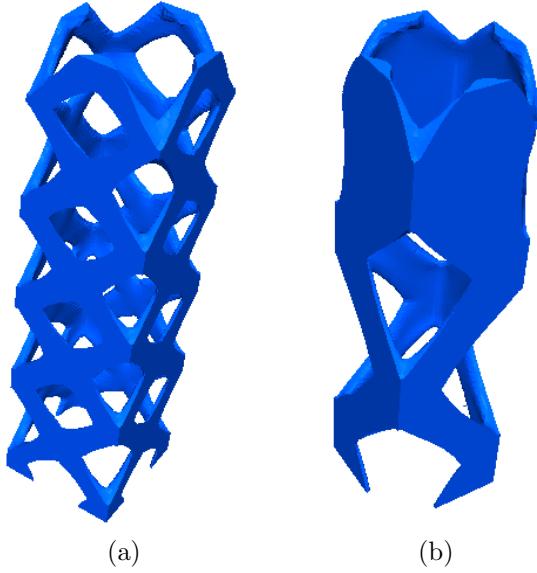


Figure 4.18: LSM-XFEM results for cuboid under torsion problem using a perimeter constraint: (a) LSM-XFEM restarted from SIMP and (b) LSM-XFEM restarted from LSM-XFEM result; mesh size:  $120 \times 30 \times 30$ .

used earlier for the coarse mesh to the fine mesh. Figure 4.17(b) shows the outcome of this procedure. The overall design is unchanged, and increasing the smoothing radius results in a less smooth design. The strain energy of this design is reported in Tab. 4.6. To control the overall structural complexity in the LSM, the formulation of the optimization problem (4.15) is often augmented by a perimeter constraint [23]. While this approach does not directly control the minimum feature size, reducing the maximum feasible perimeter often removes small features which do not alter much the structural performance. To study the influence of a perimeter constraint on the torsion problem, we perform the following two numerical experiments on the  $120 \times 30 \times 30$  mesh using the LSM-XFEM approach. We measure the perimeter of the SIMP result shown in Fig. 4.17(a) and impose this value as an upper bound on the perimeter. One problem uses the SIMP result in Fig. 4.17(a) as the initial design, and the other one uses the LSM-XFEM result in Fig. 4.14(b). The results are shown in Fig. 4.18 and the strain energies are given in Tab. 4.7.

Depending on the initial designs, the LSM-XFEM problems converge to different designs.

	Initial design	Strain energy
LSM-XFEM	SIMP	$1.3321e + 03$
LSM-XFEM		$1.3185e + 03$

Table 4.7: Strain energies of LSM-XFEM results for cuboid under torsion problem using a perimeter constraint and different initial designs; the corresponding final designs are shown in Fig. 4.18.

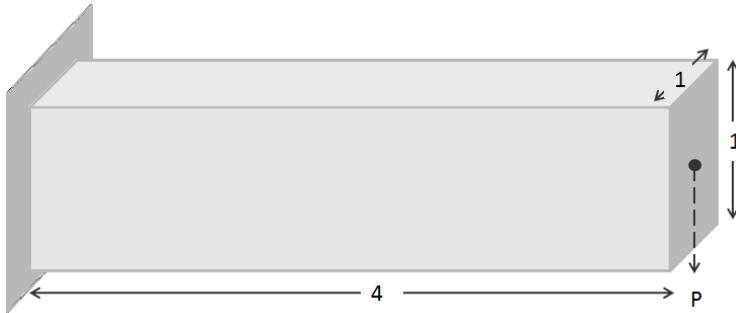


Figure 4.19: Initial setup for two-phase cantilever beam problem.

While the design in Fig. 4.18(b) displays a truss-like design in the bottom half of the design domain, the perimeter constraint does not prevent the formation of thin walls in the upper half. The thickness of the walls in the upper half of the design is 0.0188. Thus, the perimeter constraint does not control the local feature size. The design in Fig. 4.18(a) resembles closely the SIMP result from which it was restarted. However, considering the strain energy in Tab. 4.7, this design has a larger strain energy than the one in Fig. 4.18(b).

The study above has shown that neither smoothing the level set field nor imposing a perimeter constraint allows controlling the minimum feature size. Further, the effect of a perimeter constraint is non-intuitive as the result in Fig. 4.18(b) shows. The design has more structural features than the design without perimeter constraint in Fig. 4.14(b).

### 4.6.3 Two-phase Cantilevered Beam Design

The examples in the two previous subsections were concerned with “solid-void” problems. Here we study a “solid-solid” problem to demonstrate the applicability of the proposed LSM-XFEM approach to this class of problems. Note that the simplified XFEM formulation

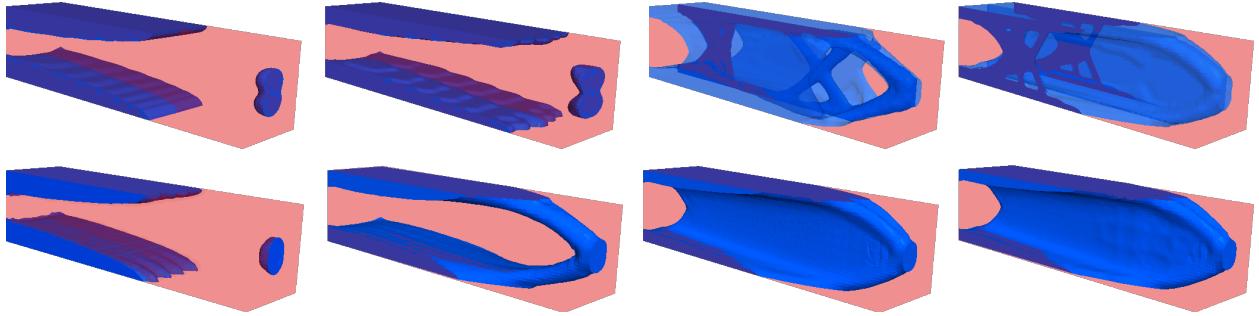


Figure 4.20: SIMP and LSM-XFEM results for different stiffness ratios: (from left to right)  $E_B = 0.5E_A$ ,  $E_B = 0.1E_A$ ,  $E_B = 0.01E_A$ ,  $E_B$  is void; SIMP results (top row); LSM-XFEM results (bottom row).

discussed in Section 4.1 is not applicable to such problems. The generalized enrichment strategy of Section 4.3.2 is required and the interface conditions of Eq. 4.5 need to be satisfied. We study the optimal two-phase layout of a  $4 \times 1 \times 1$  cantilevered beam subject to a tip load; see Fig. 4.19. The stiff phase “A” has Young’s modulus of  $E_A = 1.0$ ; three values of Young’s moduli for the soft phase are considered:  $E_B = [0.5, 0.1, 0.01]$ . Both phases have a Poisson ratio of 0.3. The maximum volume of the stiff phase is limited to 30% of the total volume. The design domain is discretized by  $120 \times 30 \times 30$  elements. Because of the symmetry condition, only one half of the cuboid is numerically analyzed. We compare the SIMP and LSM-XFEM results.

The optimization problem is solved by a SIMP approach with a penalization factor of  $p = 3$ , a smoothing radius of  $r_\rho = 0.05333$  ( $1.6 \times$  element edge length) and the projection parameter of  $\beta = 0$ . The design domain is initialized with a uniform material distribution of  $\rho_i = 0.3$ . The optimized material distributions are shown in Fig. 4.20 where material with a density lower than  $\rho_i < 0.25$  is transparent. The strain energies are reported in Tab. 4.8.

The LSM-XFEM results for a smoothing radius of  $r_\phi = 0.05333$  ( $1.6 \times$  the element edge) are shown in Fig. 4.20 and the strain energies are given in Tab. 4.8. Considering the full design domain, the level set field is initialized with a  $16 \times 4 \times 4$  array of equally spaced holes with radius of 0.1050. The initial design is shown in Fig. 4.1 and satisfies the volume constraint for the stiff phase. Note that the interface condition is enforced via the stabilized

	Stiffness ratio	Strain energy
SIMP	$E_B = 0.50E_A$	$4.4081e - 05$
	$E_B = 0.10E_A$	$6.2862e - 05$
	$E_B = 0.01E_A$	$7.8627e - 05$
	$E_B$ is void	$7.6721e - 05$
LSM-XFEM	$E_B = 0.50E_A$	$4.3221e - 05$
	$E_B = 0.10E_A$	$5.9192e - 05$
	$E_B = 0.01E_A$	$6.4448e - 05$
	$E_B$ is void	$6.6283e - 05$

Table 4.8: Strain energies of SIMP and LSM-XFEM results for different stiffness ratios; the corresponding designs are shown in Fig. 4.20.

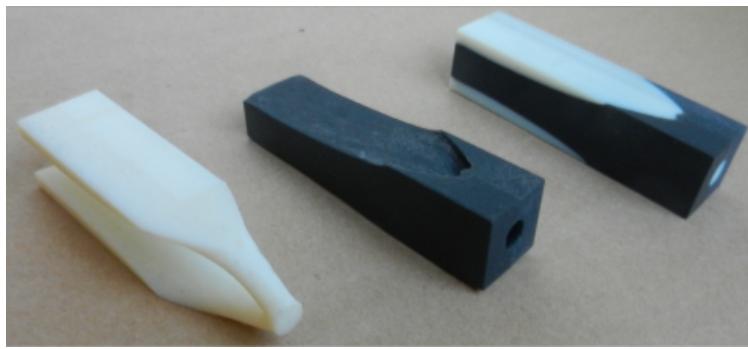


Figure 4.21: LSM-XFEM optimized two-phase design for  $E_B = 0.1E_A$  realized by 3D printing.

Lagrange multiplier method (4.5) with an element wise constant Lagrange multiplier,  $\lambda$ , and a consistency factor of  $\gamma = 10$  ( $E_A + E_B$ ).

Comparing the SIMP and LSM-XFEM results, the same trends can be observed for this “solid-solid” problem as for the “solid-void” ones studied earlier. The LSM-XFEM approach leads to three dimensional structures with thinner walls and higher stiffness. In contrast, the SIMP method generates truss-type structures, in particular if the discretization is too coarse and the optimum wall thickness is less than the size of an element.

For illustration purposes only, we show a realization of the LSM-XFEM optimized design for  $E_B = 0.1E_A$  in Fig. 4.21. The structure was fabricated with a polyjet 3D printing process on a Connex Objet 260 printer. White material represents phase “A”, black represents phase “B”. The left and center pieces show the individual phases printed separately, the printed two-phase design is shown on the right.

## 4.7 Conclusions

In this paper we presented an optimization approach combining a level set method (LSM) for describing the geometry and an extended finite element method (XFEM) for predicting the structural response. Building upon generalized enrichment and preconditioning schemes, previously developed for two-dimensional problems, the proposed optimization scheme was applied to two-phase “solid-void” and “solid-solid” problems in three dimensions. In all examples, the strain energy was minimized subject to a volume constraint on the stiff phase. The results of the LSM-XFEM approach with and without perimeter constraints were compared with the ones of a SIMP method which employs density filtering and projection.

The numerical studies suggest that the LSM-XFEM method features an improved convergence as the mesh is refined and is able to represent thin-walled structures on coarse meshes. The SIMP approach may require a strong projection to achieve clear “0-1” results with comparable strain energies. While density filtering is an efficient and intuitive method to control the local feature size, neither level set smoothing nor imposing a perimeter constraint achieves a similar effect on LSM-XFEM results.

The current lack of a feature size control and the significant improved complexity of the LSM-XFEM formulation limit the attractiveness of this scheme. However, for problems where a high mesh resolution is not tolerable and/or interface conditions need to be enforced with high accuracy, the LSM-XFEM approach might be an interesting alternative to SIMP-type methods. The advantages of the LSM-XFEM problem have been shown by Kreissl and Maute [53] for fluid problems at high Reynolds numbers in two dimensions. The authors plan to study three dimensional flow problems with the LSM-XFEM approach in the future.

# Chapter 5

## Draft: Level Set XFEM Topology Optimization of 3D Navier-Stokes and Scalar Transport Problems

This paper studies level set topology optimization of incompressible Stokes and Navier-Stokes flow, and scalar transport problems in three-dimensions. The paper expands on previous work on the LSM-XFEM by [63, 113] to three-dimensional fluid and scalar transport problems. The focus of the study is to analyze the correct application of boundary conditions at the level set interface to be able to model high Reynolds number flow in optimization problems. Accurate enforcement of boundary conditions will be done through the Nitsche method [17], and the ghost penalty formulation [94]. Using these approaches, we will generate a robust optimization approach capable of handling multiple discontinuities at the cut elements, and modeling coupled physics such as fluid flow and advection-diffusion problems.

As the level set method can accurately describe the discontinuities in the design domain, and the capabilities of additive manufacturing techniques increase, our work will study a curvature regularization technique to increase the smoothness of the level set interface in or-

der to create better-looking designs. Our work illustrates that the LSM-XFEM optimization approach is an ideal tool to generate optimized geometries that can be easily manufactured using the capabilities of three-dimensional printing.

## 5.1 Methodology

We consider an incompressible Navier-Stokes problem in a two-phase domain with  $\Omega = \Omega^+ \cup \Omega^-$ , where  $\Omega^k$  denotes the domain occupied by phase  $k \in \{+, -\}$ . The interface between the phases is denoted by  $\Gamma_{\text{int}} = \Omega^+ \cap \Omega^-$ . The domain  $\Omega$  is governed by the incompressible Navier-Stokes equations, which are given in convective form as

$$\rho^k \frac{\partial \mathbf{u}^k}{\partial t} + \rho^k \mathbf{u}^k \cdot \nabla \mathbf{u}^k + \nabla p^k - 2\mu^k \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}^k) = \rho^k \mathbf{g} \quad \text{in } \Omega^k \quad (5.1)$$

$$\nabla \cdot \mathbf{u}^k = 0 \quad \text{in } \Omega^k \quad (5.2)$$

where  $\rho^k$  and  $\mu^k$  represent the density and the dynamic viscosity of the fluid for phase  $k$ .  $\mathbf{u}^k$  is the velocity and  $p^k$  is the pressure in the phase  $k$ .  $\boldsymbol{\varepsilon} = \frac{1}{2} (\nabla \mathbf{u}^k + (\nabla \mathbf{u}^k)^T)$  is the rate-of-deformation tensor.  $\mathbf{g}$  is the gravity vector.

Initial conditions are applied in  $\Omega^k$ , and boundary conditions on the outer part of  $\Gamma^k$ , where  $\Gamma^k$  represents the boundary of the domain  $\Omega$ .

At time  $t = 0$ , a divergence-free velocity field is defined as

$$\mathbf{u}^k(\mathbf{x}, t = 0) = \mathbf{u}_0^k(\mathbf{x}) \quad \Omega^k \quad (5.3)$$

Dirichlet and Neumann boundary conditions are applied on  $\Gamma_D^k$  and  $\Gamma_N^k$  as

$$\mathbf{u}^k = \mathbf{u}_D^k \quad \text{on } \Gamma_D^k \quad (5.4)$$

$$\boldsymbol{\sigma}(\mathbf{u}^k, p^k) \cdot \mathbf{n}^k = \mathbf{h}^k \quad \text{on } \Gamma_N^k \quad (5.5)$$

where  $\boldsymbol{\sigma}(\mathbf{u}^k, p^k) = -p^k \mathbf{I} + 2\mu^k \boldsymbol{\varepsilon}(\mathbf{u}^k)$  represents the Cauchy stress tensor. The boundary of the subdomain  $\Omega^k$  is represented with  $\Gamma^k = \Gamma_D^k \cup \Gamma_N^k \cup \Gamma_{\text{int}}$ .

Additional interface conditions must be used to enforce continuity of the solution at the interface of the subdomains:

$$[\mathbf{u}] = 0 \quad \text{on } \Gamma_{\text{int}} \quad (5.6)$$

where  $[\cdot]$  represents the jump operator across a phase boundary as

$$[\mathbf{u}] = \mathbf{u}^+ - \mathbf{u}^- \quad (5.7)$$

Surface tension effects are balanced by a jump in the traction:

$$[\boldsymbol{\sigma}(\mathbf{u}, p) \cdot \mathbf{n}_{\text{int}}] = -\gamma \kappa \mathbf{n}_{\text{int}} \quad \text{on } \Gamma_{\text{int}} \quad (5.8)$$

where  $\gamma$  is the surface tension coefficient, and  $\kappa = \nabla \cdot \mathbf{n}_{\text{int}}$  is the curvature of the phase boundary.

A Streamline Upwind Petrov/Galerkin (SUPG) [111] and a Pressure Stabilizing Petrov/Galerkin (PSPG) [14] schemes are used to alleviate numerical instabilities and resolve the subgrid-scale components.

Assuming discrete trial functions spaces  $\mathbf{u}_h^k \in V_{\mathbf{u},h}^k$  for the velocity, and  $p_h^k \in V_{p,h}^k$  for the pressure, as well test functions  $\mathbf{v}_h^k \in \hat{V}_{\mathbf{v},h}^k$  for the velocity, and  $q_h^k \in \hat{V}_{p,h}^k$  for the pressure, we seek  $(\mathbf{u}_h^k, p_h^k) \in V_{\mathbf{u},h}^k \times V_{p,h}^k$  such that

$$\begin{aligned} & \int \mathbf{v}_h^k \rho^k \frac{\partial \mathbf{u}_h^k}{\partial t} d\Omega^k + \int \mathbf{v}_h^k \rho^k \mathbf{u}_h^k \cdot \nabla \mathbf{u}_h^k d\Omega^k - \int \nabla \cdot \mathbf{v}_h^k p_h^k + d\Omega^k + \int 2\boldsymbol{\varepsilon}(\mathbf{v}_h^k) \mu^k \boldsymbol{\varepsilon}(\mathbf{u}_h^k) d\Omega^k + \\ & \int \rho^k \mathbf{u}_h^k \cdot \nabla \mathbf{v}_h^k \tau_M^k \mathbf{r}_{M,h}^k d\Omega^k + \int \nabla \cdot \mathbf{v}_h^k \tau_C^k \mathbf{r}_{C,h}^k d\Omega^k + \int \nabla q_h^k \tau_M^k \mathbf{r}_{M,h}^k d\Omega^k - \\ & \int q_h^k \nabla \cdot \mathbf{u}_h^k d\Omega^k + \int \mathbf{v}_h^k \boldsymbol{\sigma}(\mathbf{u}_h^k, p_h^k) \cdot \mathbf{n}^k d\Gamma_{\text{int}} = \int \mathbf{v}_h^k \rho^k \mathbf{g} d\Omega^k + \int \mathbf{v}_h^k \mathbf{h}^k d\Gamma_N^k \end{aligned} \quad (5.9)$$

where  $\mathbf{r}_{M,h}^k$  is the discrete residual of the momentum equation:

$$\mathbf{r}_{M,h}^k = \rho^k \frac{\partial \mathbf{u}_h^k}{\partial t} + \rho^k \mathbf{u}_h^k \cdot \nabla \mathbf{u}_h^k + \nabla p_h^k - 2\mu^k \nabla \cdot \boldsymbol{\varepsilon}(\mathbf{u}_h^k) - \rho^k \mathbf{g} \quad (5.10)$$

and  $\mathbf{r}_{C,h}^k$  is the discrete residual of the continuity equation:

$$\mathbf{r}_{C,h}^k = \nabla \cdot \mathbf{u}_h^k \quad (5.11)$$

$\tau_M^k$  and  $\tau_C^k$  represent the SUPG and PSPG stabilization parameters, respectively.

We apply Nitsche's method [44, 81] for the weak imposition of the velocity continuity described in Equation 5.7, and face-oriented ghost-penalty stabilization terms for each subdomain in the interface zone [17, 93].

Analogous to the structural topology optimization problem of [113], we introduce fictitious springs to pin free floating material in “fluid-void” problems.

$$\mathbf{r}_{P,h}^k = \mathbf{v}_h^k k_p p_h^k \quad (5.12)$$

where  $k_p$  denotes the stiffness of the distributed system of springs.

## 5.2 Numerical examples

We study our LSM-XFEM approach with numerical examples for incompressible Navier-Stokes and scalar transport problems in three dimensions. Four numerical examples will be used: a) a pipe bend, b) a diffuser, c) a micromixer, and d) a fluid duct with heat source.

In all examples, the optimization problems are solved by the Globally Convergent Method of Moving Asymptotes (GCMMA) of [108, 109]. The sensitivities are computed by the adjoint method. The design domains are discretized by eight-node linear elements. The linear systems of the forward and adjoint problems are solved by a parallel implementation

of the GMRES method [46] The problems are preconditioned by an ILU factorization with a fill of 2.0 and an overlap of 1.0. The convergence tolerances for both, the GCMMA and the GMRES solver, are chosen sufficiently low such that the optimization results do not depend on the tolerance values. The pressure spring stiffness value,  $k_p$ , is  $10^{-6}$ . Geometric and material parameters are given in non-dimensional and self-consistent units.

### 5.2.1 Optimization criteria

This section details the objective and constraints criteria used in the optimization problems.

The pressure drop criteria measures the difference of the average total pressure between the inlet and the outlet:

$$F_{\text{PD}} = \frac{\int p_h^f + \frac{\rho^f |\mathbf{u}_h^f|^2}{2} d\Gamma_{\text{in}}}{\int d\Gamma_{\text{in}}} - \frac{\int p_h^f + \frac{\rho^f |\mathbf{u}_h^f|^2}{2} d\Gamma_{\text{out}}}{\int d\Gamma_{\text{out}}} \quad (5.13)$$

where the superscript  $f$  represents the value over the fluid phase.

To measure the maximum difference between a current scalar field and a target scalar value, we use the Kreisselmeier-Steinhauser function [51]:

$$F_{\text{TSV}} = \frac{1}{\beta^f} \ln \int e^{\beta(T_h^f - T_{\text{ref}})^2} d\Gamma \quad (5.14)$$

where a larger value of  $\beta^f$  increases the enforcement of the function, but may result in numerical overflow.

The volume of the fluid and void (or solid) phase regions is computed as:

$$F_{\text{volume},f} = \int d\Omega^f \quad (5.15)$$

$$F_{\text{volume},s} = \int d\Omega^s \quad (5.16)$$

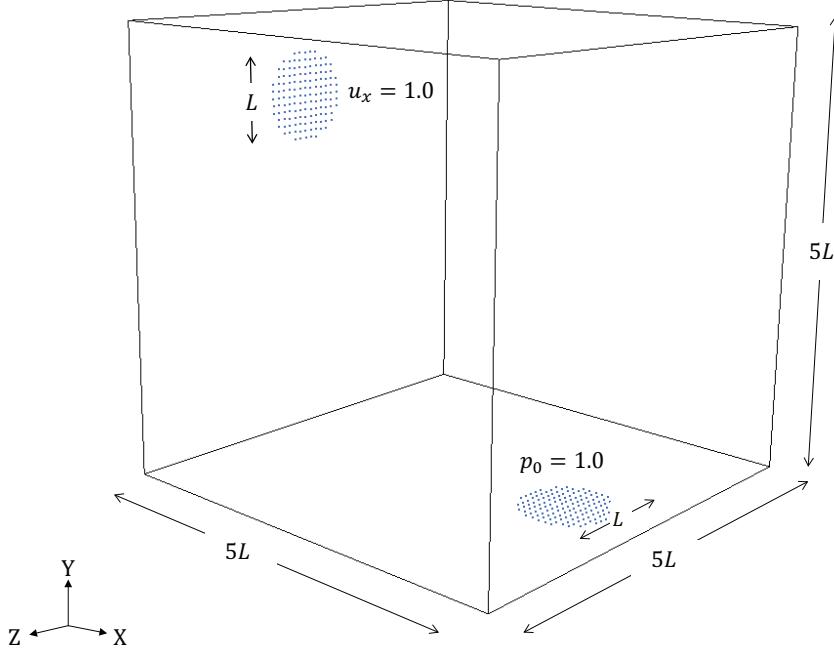


Figure 5.1: Setup for the pipebend problem. A parabolic inlet flow in the  $x$ -direction, with a maximum value of 1.0 enters the design domain through a circular inclusion of diameter  $L$  in the minimum YZ facet. The outlet is modeled as a circle with diameter  $L$  and is located in the minimum  $XZ$  facet with a constant pressure of 1.0.

The perimeter criteria is computed as:

$$F_{\text{perimeter}} = \int d\Gamma_{\text{int}} \quad (5.17)$$

### 5.2.2 Pipebend and Diffuser

We consider the bend and diffuser “fluid-void” problems depicted in Figures 5.1 and 5.2. The problems were adapted to three dimensions from their two dimensional equivalents found in [13, 53, 63]. This example seeks to study the capability of our LSM-XFEM approach to model three-dimensional “fluid-void” incompressible Stokes and Navier-Stokes flows, with low and high Reynolds numbers. In contrast to density methods, our framework does not model fluid in the void phase. The design variables are fixed at the inlet and outlet to be in the fluid phase to ensure the correct enforcement of the boundary conditions. A velocity

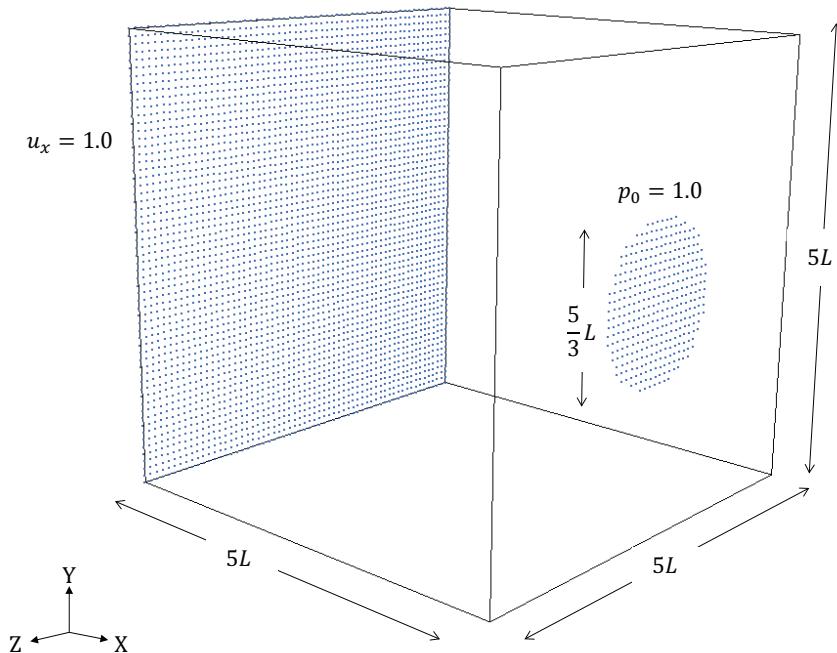


Figure 5.2: Setup for the diffuser problem. A parabolic inlet flow in the  $x$ -direction, with a maximum value of 1.0 enters the design domain through the entire minimum YZ facet. The outlet, modeled as a circle of diameter  $\frac{5}{3}L$  is located in the maximum YZ facet with a constant pressure of 1.0.

Outlet pressure	1.0
Reynolds number	1.0
Inlet velocity	$u_{h_x}^f = 1.0$
Element length	$h = 0.083333$
Maximum volume fraction	$\bar{v}_f = 0.25$
Pressure difference penalty	$w_{\text{pres}} = 1.0$
Perimeter penalty	$w_{\text{per}} = 0.01$
Smoothing filter	$d = 2.4 \cdot h$

Table 5.1: Parameters for bend problem.

Outlet pressure	1.0
Reynolds number	1.0
Inlet velocity	$u_{h_x}^f = 1.0$
Element length	$h = 0.083333$
Maximum volume fraction	$\bar{v}_f = 0.5$
Pressure difference penalty	$w_{\text{pres}} = 1.0$
Perimeter penalty	$w_{\text{per}} = 0.01$
Smoothing filter	$d = 2.4 \cdot h$

Table 5.2: Parameters for diffuser problem.

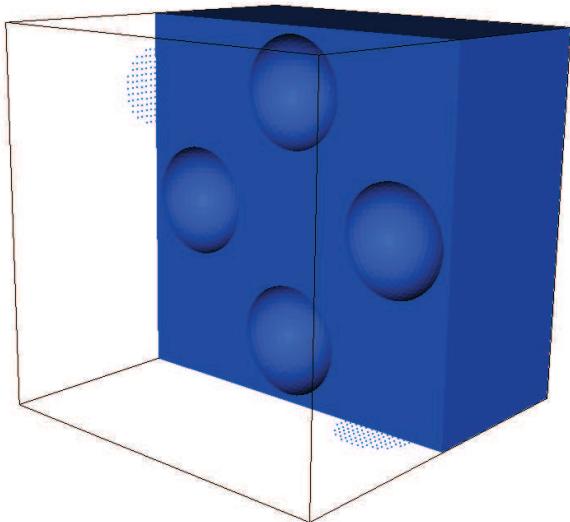
of 1 in the  $x$ -direction is prescribed at the inlet, and a pressure of 1 is prescribed at the outlet. For both, the bend and diffuser problems, we will model a Stokes flow with  $Re = 1$ , a Navier-Stokes flow with  $Re = 1$ , and a Navier-Stokes flow with a  $Re = 100$ . The initial designs for the bend and diffuser are shown in Figures 5.3 and 5.4. The complete list of parameters for the problem setup are shown in Tables 5.1 and 5.2. The objective for the bend and diffuser problems is defined as:

$$\mathcal{F}_{\text{bend/diffuser}} = w_{\text{pres}} F_{\text{PD}} + w_{\text{per}} F_{\text{perimeter}} \quad (5.18)$$

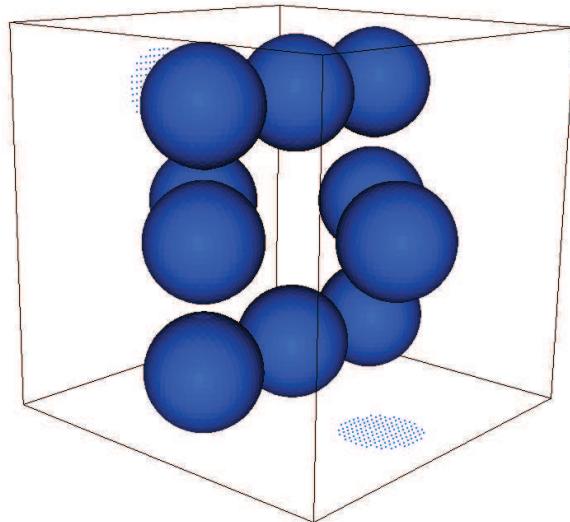
where  $w_{\text{pres}}$  and  $w_{\text{per}}$  are problem dependent parameters to penalize the pressure difference and perimeter objectives.

The volume fraction of the fluid phase is constrained as:

$$\mathcal{G}_{\text{bend/diffuser}} = \frac{F_{\text{volume},f}}{\bar{v}_f (F_{\text{volume},f} + F_{\text{volume},s})} - 1 \quad (5.19)$$

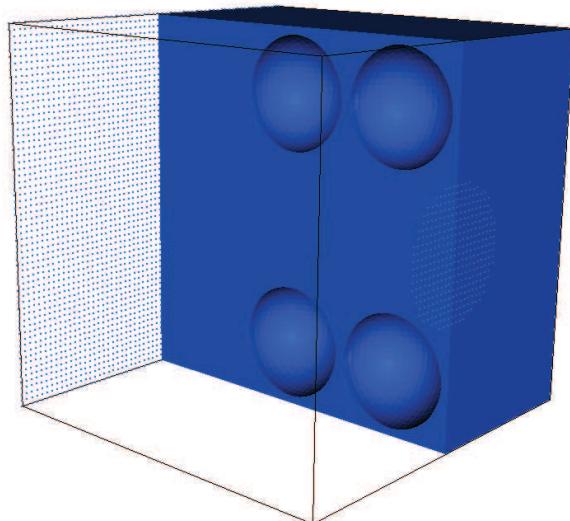


(a) Symmetrical clip along the XY facet of the fluid phase.

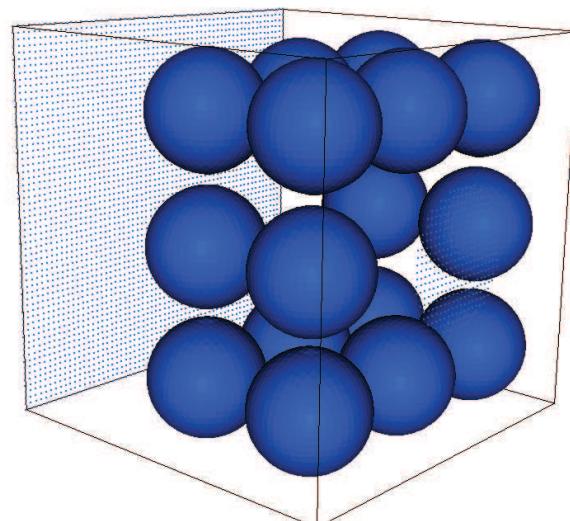


(b) Void phase.

Figure 5.3: Initial designs for the pipe bend problem.



(a) Symmetrical clip along the XY facet of the fluid phase.



(b) Void phase.

Figure 5.4: Initial designs for the diffuser problem.

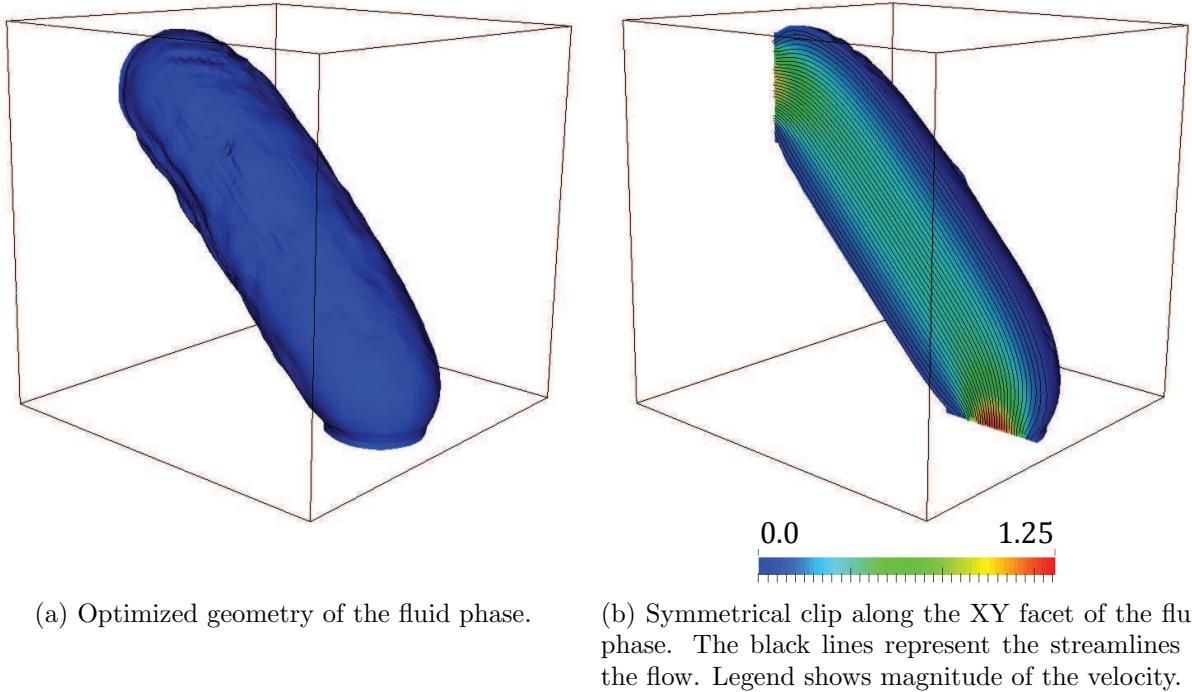


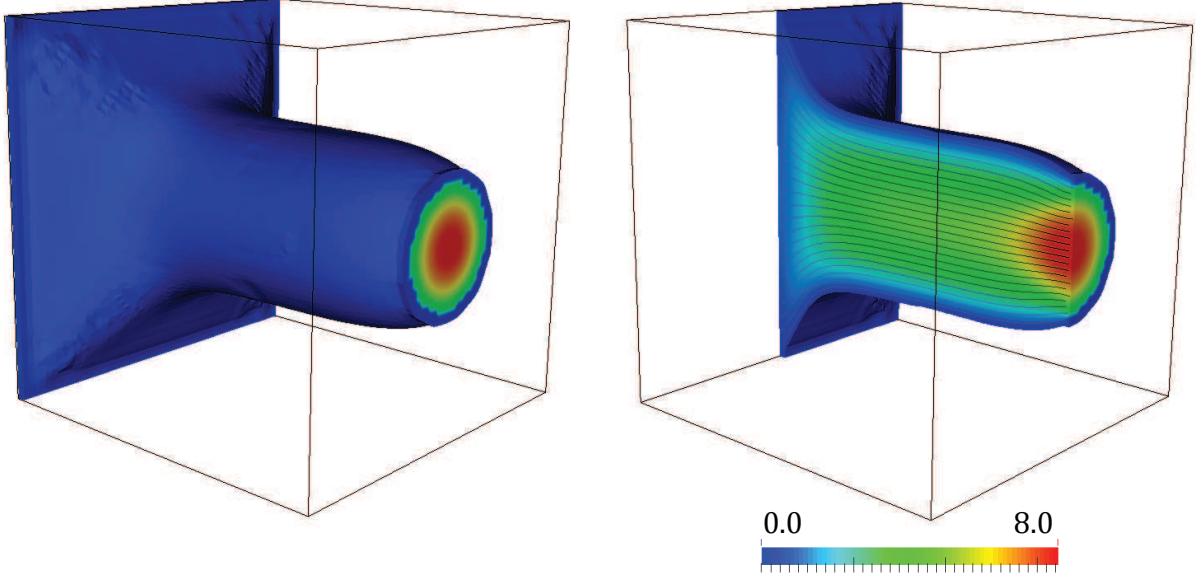
Figure 5.5: Results for the Stokes and Navier-Stokes pipebend optimization problems with a Reynolds number of 1.

where  $\bar{v}_f$  is the maximum volume fraction of the fluid phase.

Results for the Stokes and Navier-Stokes flows for a Reynolds number of 1, for the bend and diffuser problems, yield the same optimized results and are shown in Figures 5.5 and 5.6.

### 5.2.3 Micromixer

This example studies the optimization of a micromixer. This problem has been previously studied by using a homogenization method with a Brinkman penalization by [65] and by using a LSM-XFEM approach in two dimensions by [63]. The problem setup is defined in Figure 5.7. A fluid with a scalar field value of 1.0 and fluid with a value of 0.0 enter the domain through a circular inclusion on the minimum YZ facet and leave through another circular inclusion on the maximum YZ facet. Both fluids are ideally miscible and have identical flow properties. The optimization problem consists in finding the optimal channel geometry that



(a) Optimized geometry of the fluid phase.

(b) Symmetrical clip along the XY facet of the fluid phase. The black lines represent the streamlines of the flow. Legend shows magnitude of the velocity.

Figure 5.6: Results for the Stokes and Navier-Stokes diffuser optimization problems with a Reynolds number of 1.

maximizes the mixing of the two fluids. The objective of the optimization problem is defined as

$$\mathcal{F}_{\text{micromixer}} = F_{\text{TSV}} \quad (5.20)$$

with  $\beta^f = 100$  and  $T_{\text{ref}}^f = 0.5$ , which minimizes the maximum deviation of the scalar values of both fluids to the desired mixing value of  $T_{\text{ref}}^f$ . Unlike the previous examples, a perimeter penalty is not used, as it would penalize the design towards a flat channel which would affect the maximization of the mixing.

Constraints are imposed on the maximum pressure drop to ensure the channel remains connected from the inlet to the outlet, and on the volume fraction of the fluid phase.

$$\mathcal{G}_{\text{micromixer},1} = \frac{F_{\text{PD}}}{\Delta p_{\text{ref}}} - 1 \quad (5.21)$$

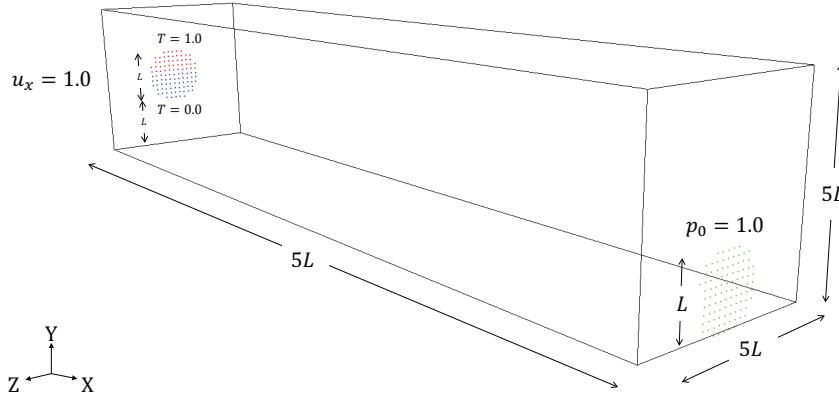


Figure 5.7: Setup for the micromixer problem. A parabolic inlet flow in the  $x$ -direction, with a maximum value of 1.0 enters the design domain through a circular inclusion of diameter  $L$  through the minimum YZ facet. The upper area of the inlet has a scalar value of 1.0, while the lower area has a value of 0.0. The outlet is located in the maximum YZ facet with a constant pressure of 1.0.

Outlet pressure	1.0
Reynolds number	1.0
Thermal conductivity	$k_{xx}^f = 0.001$
Inlet velocity	$u_{hx}^f = 1.0$
Element length	$h = 0.166667$
Maximum volume fraction	$\bar{v}_f = 0.5$
Maximum pressure drop	$\Delta p_{\text{ref}} = 1$
Smoothing filter	$d = 2.4 \cdot h$

Table 5.3: Parameters for diffuser problem.

$$\mathcal{G}_{\text{micromixer},2} = \frac{F_{\text{volume},f}}{\bar{v}_f (F_{\text{volume},f} + F_{\text{volume},s})} - 1 \quad (5.22)$$

The rest of the parameters for the problem are defined in Table 5.3. The optimized geometry for the Stokes and Navier-Stokes flows for a Reynolds number of 1 yield the same optimized result, and is shown in Figure 5.8. The scalar field with a value of  $T = 1.0$ , the “red” fluid, and the field with a value of  $T = 0.0$ , the “blue” fluid, mix to create “green” fluid with a value of  $T = 0.5$ . The final design increases the length of the path traveled by the fluid, enhancing the mixing by creating multiple channels in all directions, similar to a “twizzler”.

Streamlines of the fluid flow are shown in Figure 5.9.

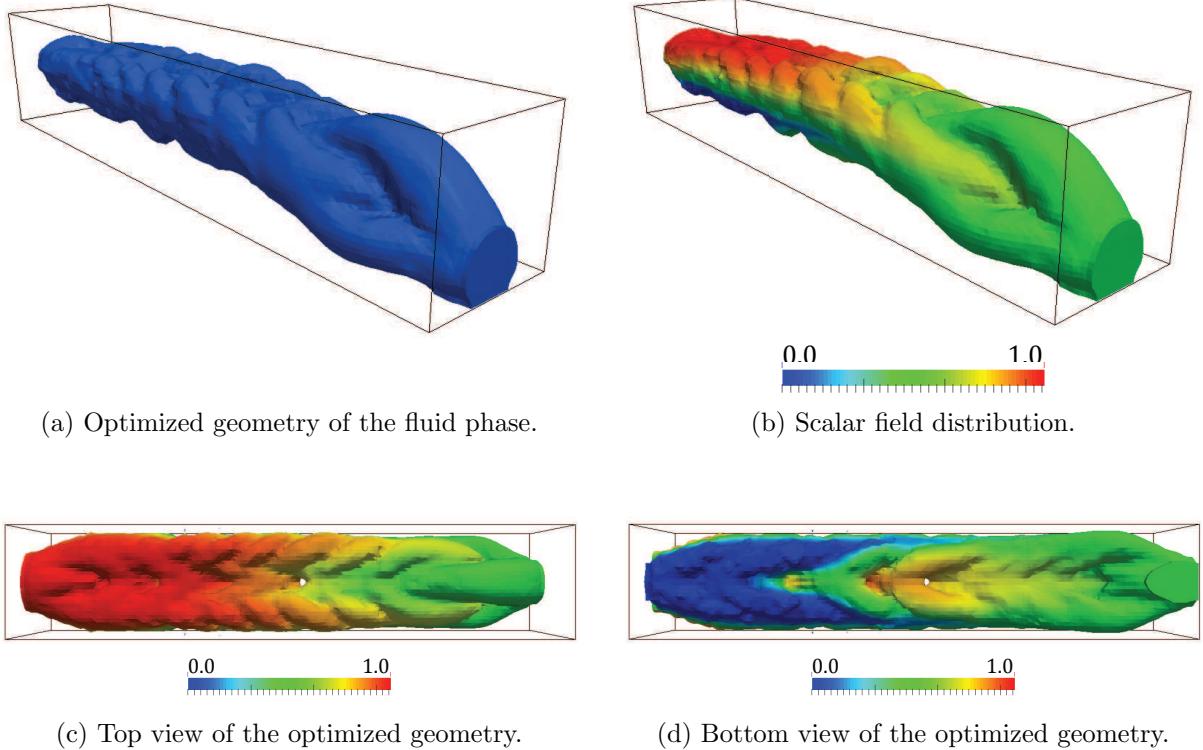
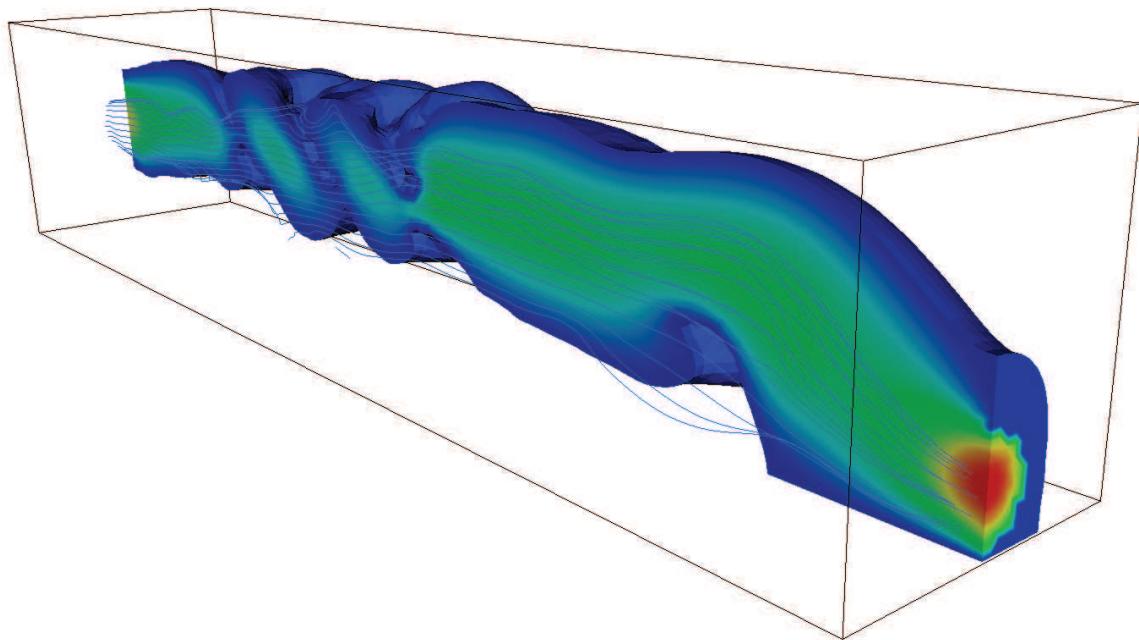
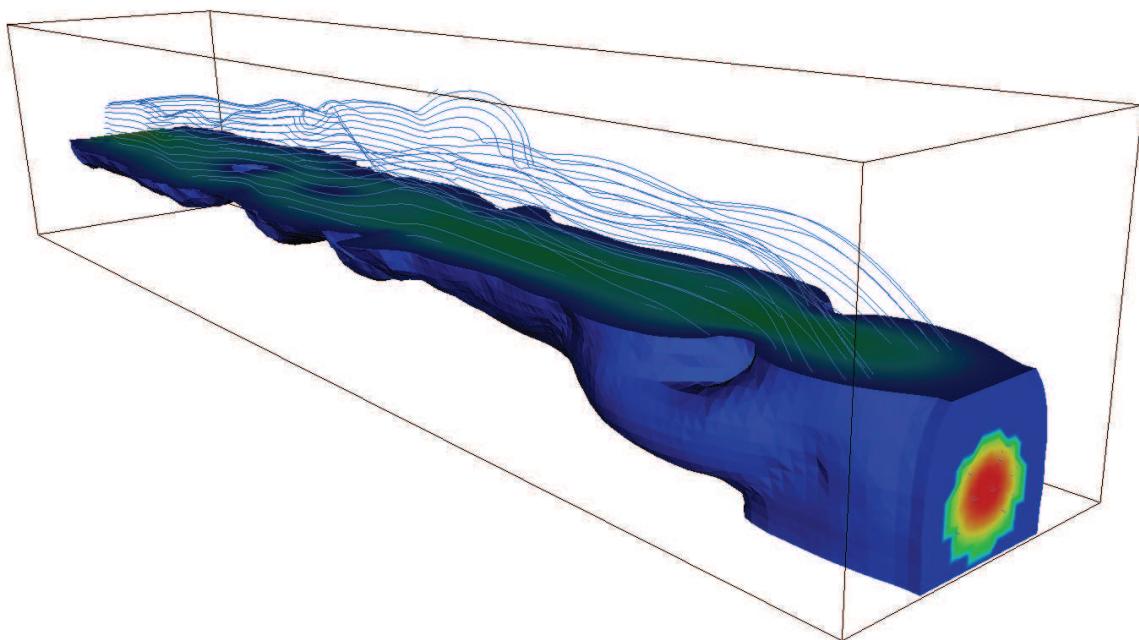


Figure 5.8: Results for the Stokes and Navier-Stokes micromixer optimization problems with a Reynolds number of 1 and a thermal conductivity of  $k_{xx}^f = 0.001$ . Legend shows the magnitude of the scalar field.



(a) Streamlines along the XY facet.



(b) Streamlines along the XZ facet.

Figure 5.9: Streamlines for the Stokes and Navier-Stokes micromixer optimization problems with a Reynolds number of 1 and a thermal conductivity of  $k_{xx}^f = 0.001$ .

### 5.3 Conclusions

The LSM-XFEM framework was extended to three-dimensional incompressible Navier-Stokes flows and scalar transport optimization problems. Preliminary results show that high Reynolds number flows cannot be modeled accurately without the ghost penalty formulation.

To conclude this work, the author will implement the face-oriented ghost-penalty to formulate a robust optimization framework and be able to solve high Reynolds number flows. Curvature minimization approaches will be studied to generate smooth surfaces that facilitate the manufacturing of the optimized geometries via three dimensional printing.

# Chapter 6

## Draft: Topology Optimization Approaches for the Curvature Minimization of Level Set Isocontours

Curvature measures the smoothness of a surface. This paper studies the application of curvature minimization schemes to the zero level set isocontour. The objective is to maximize the smoothness of the level set surface of an optimized geometry. Rather than maximizing the smoothness of the surface using postprocessing methods like in [61], the curvature minimization is inserted directly into the optimization problem.

### 6.1 Curvature

Curvature is a mathematical concept that describes how sharply a curve bends; in other words, curvature measures how much a geometric object deviates from being flat. For example, the curvature of a straight line is identically zero, while curves that bend very little have small curvature and curves that bend sharply have large curvature.

The concept of curvature dates back to the Greek when Euclid provided a definition of the straight line as one which lies evenly with the points on itself; however, it was not

until Newton that the concept of curvature was defined [76]. As stated above, the basic idea is that some curves are straight, while some are curved. Among the different concepts behind curvature, they all agree that a circle has the same curvature at all the points of its circumference. Given that the smaller the radius, the greater the curvature, we can say that curvature is a quantity inversely proportional to the radius of the circle. Then, we can measure the curvature of any curve by measuring the curvature of the circle lying nearest to it [20].

A more formal definition of curvature states that if we move along a curve, the direction of the tangent vector will not change as long as the curve is flat. However, its direction will change if the curve bends. The more the curve bends, the more the direction of the tangent vector changes. Therefore, to study curvature we must study how the tangent vector changes as we move along a curve. But, because we are only interested in the direction of the tangent vector, not its magnitude, we will consider the unit tangent vector.

Let  $C$  be a smooth curve with position vector  $\mathbf{r}(s)$  where  $s$  is the arc length parameter. The curvature  $\kappa$  of  $C$  is defined to be:

$$\kappa = \kappa_t = \left\| \frac{d\mathbf{t}}{ds} \right\| \quad (6.1)$$

where  $\mathbf{t}$  is the unit tangent vector.

Assuming the unit normal vector,  $\mathbf{n}$  is orthonormal to the unit tangent vector, we can redefine curvature as:

$$\kappa = \kappa_n = \left\| \frac{d\mathbf{n}}{ds} \right\| \quad (6.2)$$

Other definitions of curvature as a function of the normal unit vector can be found in [62, 115, 116], where the mean curvature measure was used.

$$H = \|\nabla \cdot \mathbf{n}\| \quad (6.3)$$

### 6.1.1 Comparison of curvature and mean curvature in $\mathbb{R}^2$

Mathematically, Equations 6.3 and 6.2 are equivalent for geometrical objects in  $\mathbb{R}^2$ . We will show this by computing the curvature and mean curvature of a circle in the examples below.

#### Curvature of circle

In this example, we will compute the curvature of a circle with radius  $a$  to use as an analytical solution to our curvature measures of 6.4. We can parametrize the curve of the circle as a function of the arclength,  $s$ , as:

$$\mathbf{r}(s) = \left\{ a \cos \frac{s}{a}, a \sin \frac{s}{a} \right\} \quad (6.4)$$

Then, the unit normal vector,  $\mathbf{n}$ , is defined as:

$$\mathbf{n}(s) = \frac{\frac{d\mathbf{r}}{ds}}{\left\| \frac{d\mathbf{r}}{ds} \right\|} \quad (6.5)$$

where  $\frac{d\mathbf{r}}{ds}$  is computed as:

$$\begin{aligned} \frac{d\mathbf{r}}{ds} &= \left\{ a \left( -\frac{1}{a} \sin \frac{s}{a} \right), a \left( \frac{1}{a} \cos \frac{s}{a} \right) \right\} \\ &= \left\{ -\sin \frac{s}{a}, \cos \frac{s}{a} \right\} \end{aligned} \quad (6.6)$$

Thus, the unit normal vector,  $\mathbf{n}$ , is:

$$\mathbf{n}(s) = \left\{ -\sin \frac{s}{a}, \cos \frac{s}{a} \right\} \quad (6.7)$$

and its derivative with respect to the arclength is:

$$\frac{d\mathbf{n}}{ds} = \left\{ -\frac{1}{a} \cos \frac{s}{a}, -\frac{1}{a} \sin \frac{s}{a} \right\} \quad (6.8)$$

Finally, the curvature  $\kappa_n$  is:

$$\kappa_n = \left\| \frac{d\mathbf{n}}{ds} \right\| = \frac{1}{a} \quad (6.9)$$

We can conclude that the curvature of the circle is the inverse of its radius.

### Mean curvature of circle

Following the circle parametrization of Equation 6.4, we can compute the mean curvature (6.3) of a circle as follows:

$$\begin{aligned} H &= \left\| \frac{\partial n_x}{\partial x} + \frac{\partial n_y}{\partial y} \right\| \\ &= \left\| \frac{\partial n_x}{\partial s} \frac{\partial s}{\partial x} + \frac{\partial n_y}{\partial s} \frac{\partial s}{\partial y} \right\| \end{aligned} \quad (6.10)$$

From Equation 6.4, we parametrized  $x$  as  $a \cos \frac{s}{a}$  and  $y$  as  $a \sin \frac{s}{a}$ . Therefore

$$\begin{aligned} \frac{\partial s}{\partial x} &= \frac{x}{\sqrt{x^2 + y^2}} = a \frac{\cos \frac{s}{a}}{a} \\ \frac{\partial s}{\partial y} &= \frac{y}{\sqrt{x^2 + y^2}} = a \frac{\sin \frac{s}{a}}{a} \end{aligned} \quad (6.11)$$

From Equation 6.6, the derivative of the normal unit vector with respect to the arclength  $s$  is:

$$\begin{aligned} \frac{\partial n_x}{\partial s} &= -\frac{1}{a} \cos \frac{s}{a} \\ \frac{\partial n_y}{\partial s} &= -\frac{1}{a} \sin \frac{s}{a} \end{aligned} \quad (6.12)$$

Finally, the mean curvature is

$$H = \left\| -\frac{1}{a} \cos \frac{s}{a} \left( a \frac{\cos \frac{s}{a}}{a} \right) - \frac{1}{a} \sin \frac{s}{a} \left( a \frac{\sin \frac{s}{a}}{a} \right) \right\| = \frac{1}{a} \quad (6.13)$$

We can conclude that the curvature 6.2 and mean curvature 6.3 formulations are the same for a geometrical object in  $\mathbb{R}^2$ .

## Curvature of sinusoidal wave

This same methodology can be applied to compute the curvature of the sinusoidal wave:

$$\mathbf{r}(s) = \{s, A \sin \pi s\} \quad (6.14)$$

where  $A$  is the amplitude of the sinusoidal wave. The curvature is then:

$$\kappa_n = \pi^2 \left( \frac{A^2(1 + \cos(\pi s)^2) \sin(\pi s)^2}{(1 + A\pi \cos(\pi s)^2)^3} \right)^2 \quad (6.15)$$

The results obtained in the numerical examples will be compared to these analytical formulations to attest the accuracy of our curvature measure.

Curvature and mean curvature are equivalent measures for geometrical objects in  $\mathbb{R}^2$ . Note, however, that this assertion is not applicable to three-dimensional objects.

## 6.2 Curvature Squared

[26, 79, 83] used the square power of the curvature  $\kappa_n$  as their measurement formulation. Because the shape of interest is the zero level set isocontour, we will measure curvature in our design domain by integrating  $\kappa_n^2$  over  $\Gamma_{\phi=0}$ .

$$\int_{\phi=0} \kappa_n^2 d\Gamma = \int_{\phi=0} \left( \left\| \frac{d\mathbf{n}}{ds} \right\| \right)^2 d\Gamma \quad (6.16)$$

## 6.3 Level set normal unit vector

As shown in section 6.1, we need to define the normal vector of the level set field in order to measure curvature along the  $\Gamma_{\phi=0}$  isocontour. By definition of equation 2.15, we define the normal of the level set field as:

$$\mathbf{n} = \frac{\nabla \phi}{\|\nabla \phi\|} \quad (6.17)$$

For this curvature study, we will use several variations of the normal unit vector 6.17 to compute the curvature.

### 6.3.1 $\mathbf{n}_\phi$

Our first definition of a normal unit vector, which we will denote as  $\mathbf{n}_\phi$ , will enforce equation 6.17 in a strong form:

$$\mathbf{n}_\phi = \frac{\nabla\phi}{\|\nabla\phi\|} \quad (6.18)$$

The normal will be interpolated at the interface  $\Gamma_{\phi=0}$  from the nodally defined level set field, using the elemental shape functions. Note that this formulation does not guarantee continuity of the level set normals at the nodes. For example, Figure 6.1a shows a circular inclusions of radius 0.05 in a domain of size  $3L \times 2L$ , and Figure 6.1b shows the discretized domain with  $45 \times 30$  elements. Figure 6.1c zooms into the inclusion area and shows how the value of  $\mathbf{n}_\phi$  at node is interpolated differently from each one of its four adjacent elements.

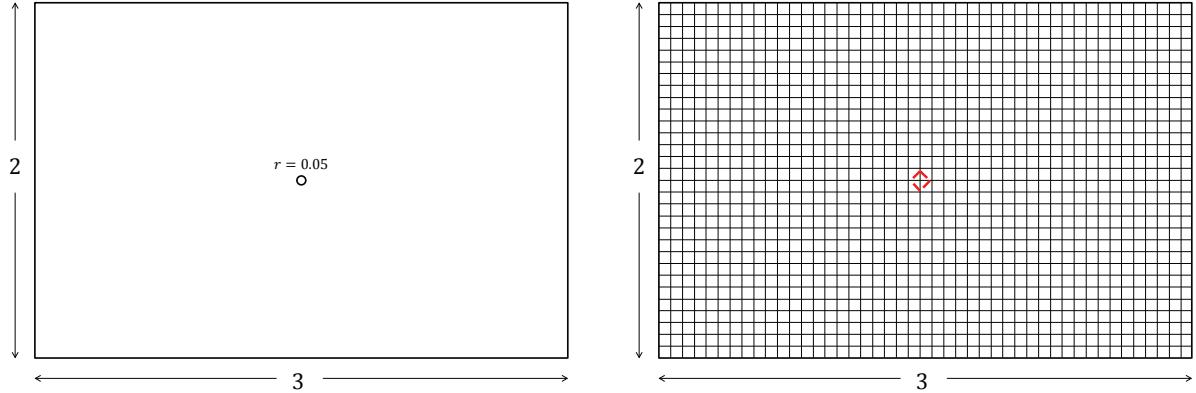
### 6.3.2 $\mathbf{n}_u$

We will denote our second formulation as  $\mathbf{n}_u$ , and we will enforce Equation 6.17 weakly:

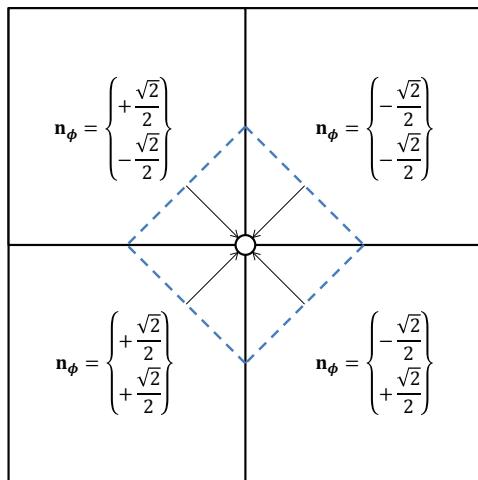
$$\int_\Omega \delta\mathbf{n}_u (\mathbf{n}_u \|\nabla\phi\| - \nabla\phi) \, d\Omega = 0 \quad (6.19)$$

As our solution will be computed using the XFEM, we need to ensure continuity of the  $\mathbf{n}_u$  degrees-of-freedom at the interface. We will impose a penalty formulation for the normal unit vector such that  $\mathbf{n}_u^+ = \mathbf{n}_u^-$  at  $\Gamma_{\phi=0}$ :

$$\begin{aligned} \gamma_u \int_{\Gamma^+} \delta\mathbf{n}_u^+ (+\mathbf{n}_u^+ - \mathbf{n}_u^-) \, d\Gamma^+ &= 0 \\ \gamma_u \int_{\Gamma^-} \delta\mathbf{n}_u^- (-\mathbf{n}_u^+ + \mathbf{n}_u^-) \, d\Gamma^- &= 0 \end{aligned} \quad (6.20)$$



(a) Physical model for a level set circular inclusion with a radius of 0.05 in a domain of size  $3L \times 2L$ .  
(b) Discrete model for a level set circular inclusion with a radius of 0.05 in a domain with  $45 \times 30$  elements. The red line indicates the location of  $\Gamma_{\phi=0}$ .



(c)  $\mathbf{n}_\phi$  interpolated at a node from each of its four adjacent elements.

Figure 6.1: Discontinuities in the  $\mathbf{n}_\phi$  field at a node.

where  $\gamma_u$  is a scaling factor. Equation 6.18 is computed locally at the element level and does not ensure a continuous normal unit vector field at the nodes, as shown in Figure 6.1. This alternative formulation, 6.19, seeks to study the effect on the curvature measure of using a continuous field.

### 6.3.3 $\mathbf{n}_\psi$

Our last formulation with regards to the level set normal unit vector will be similar to Equation 6.20. However, this formulation will first project the level set field using a smooth Heaviside function. For this goal, we will use the hyperbolic tangent function introduced by [84] and [85]:

$$\psi(\phi) = \frac{1}{2} \left( \tanh \left( \frac{\phi}{2\epsilon} \right) + 1 \right) \quad (6.21)$$

where  $\epsilon$  determines the thickness of the profile. By using this projection, the interface is no longer defined by the  $\phi = 0$  isocontour, but rather by  $\psi = 0.5$ . Figure 6.2 shows the original level set function and their hyperbolic tangent projection for different values of  $\epsilon$ . The new weak form looks as follows:

$$\int_{\Omega} \delta \mathbf{n}_\psi (\mathbf{n}_\psi \|\nabla \psi\| - \nabla \psi) \, d\Omega = 0 \quad (6.22)$$

with interface conditions:

$$\begin{aligned} \gamma_\psi \int_{\Gamma^+} \delta \mathbf{n}_\psi^+ (+\mathbf{n}_\psi^+ - \mathbf{n}_\psi^-) \, d\Gamma^+ &= 0 \\ \gamma_\psi \int_{\Gamma^-} \delta \mathbf{n}_\psi^- (-\mathbf{n}_\psi^+ + \mathbf{n}_\psi^-) \, d\Gamma^- &= 0 \end{aligned} \quad (6.23)$$

where  $\gamma_\psi$  is the scaling factor.

There are two ways to compute  $\nabla \psi$  for Equation 6.22. The first approach will compute  $\psi(\phi)$  at the nodes and then use the element shape functions to interpolate to the integration

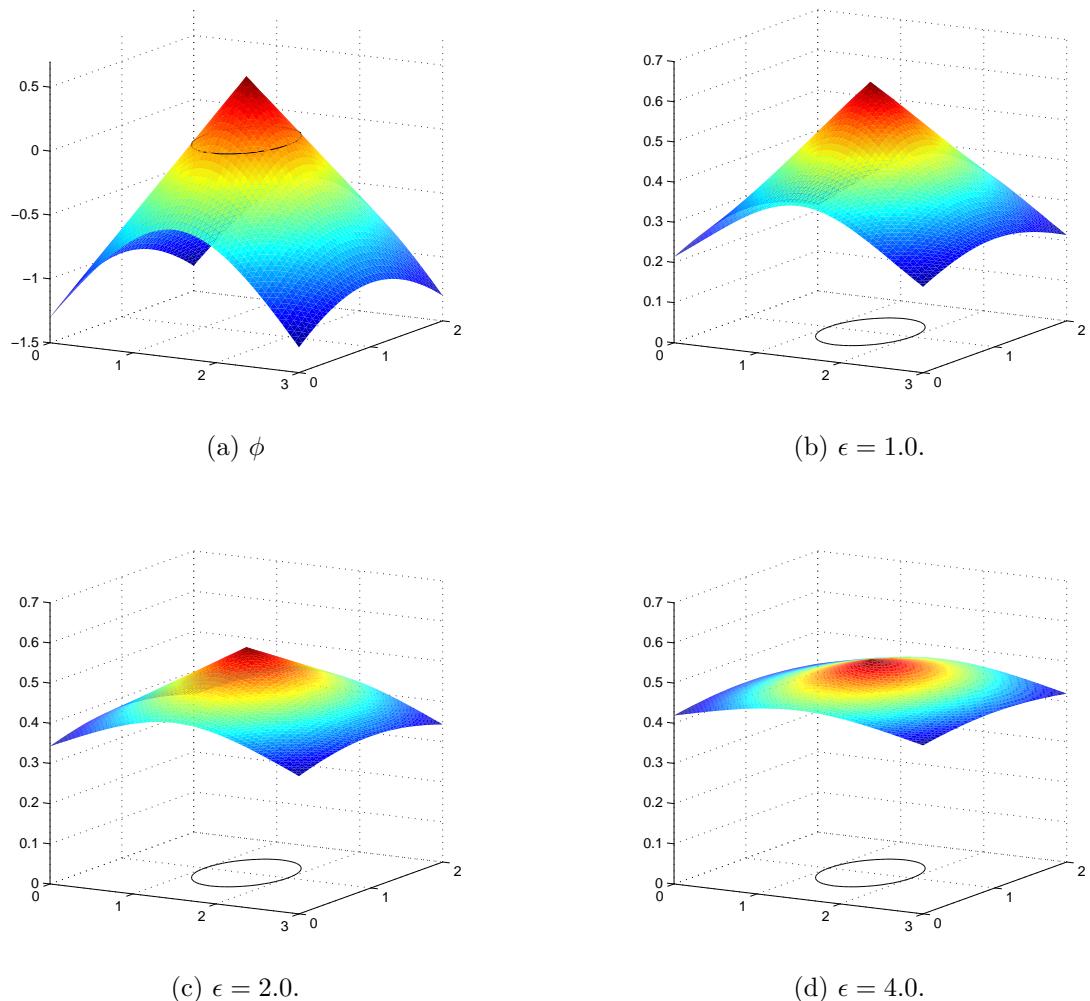


Figure 6.2: A level set field for a circular inclusion of radius 0.5, and their respective hyperbolic tangent projections for different values of  $\epsilon$ .

point as:

$$\nabla\psi = \frac{\partial\psi}{\partial\mathbf{x}} \quad (6.24)$$

Our second approach will first interpolate the  $\phi$  field, and then apply chain rule to get  $\nabla\psi$  as:

$$\nabla\psi = \frac{\partial\psi}{\partial\phi} \frac{\partial\phi}{\partial\mathbf{x}} \quad (6.25)$$

where  $\frac{\partial\psi}{\partial\phi}$  is computed analytically. To differentiate between the two approaches, we will refer to the first formulation as  $\mathbf{n}_{\psi_\phi}$ , and  $\mathbf{n}_{\psi_{gp}}$  for the second.

### 6.3.4 $\mathbf{n}_g$

The final formulation will consider the interface of the cut elements as geometrical objects, and compute the normal unit vector  $\mathbf{n}_g$  from the coordinates of these objects. This approach will not be used with Equation 6.2, but rather to an equivalent formulation by measuring strain energy.

The square power of the curvature is an equivalent measure to strain energy. The use of  $\mathbf{n}_g$  allows us to model the level set interface as a set of structural linear beam elements, and can solve a finite element problem and compute the strain energy of the system. Our measure of curvature squared is then:

$$\int_{\phi=0} \kappa_n^2 d\Gamma = \frac{1}{2} \int_{\phi=0} \boldsymbol{\sigma} : \boldsymbol{\epsilon} d\Gamma \quad (6.26)$$

To illustrate the application of this approach, consider a level set function that describes a sinusoidal wave 6.32 with an amplitude  $A_s$  of 0.50, as shown in Figure 6.3. The red dot in the figure indicates one of the many points at which the level set function intersects an element. To compute the strain energy at this location, we define a search radius  $r_\kappa$ , and look for all the cut elements that share the same interface and that are within distance of this radius, as shown in Figure 6.4. From these elements, we extract the interfaces and build a set of

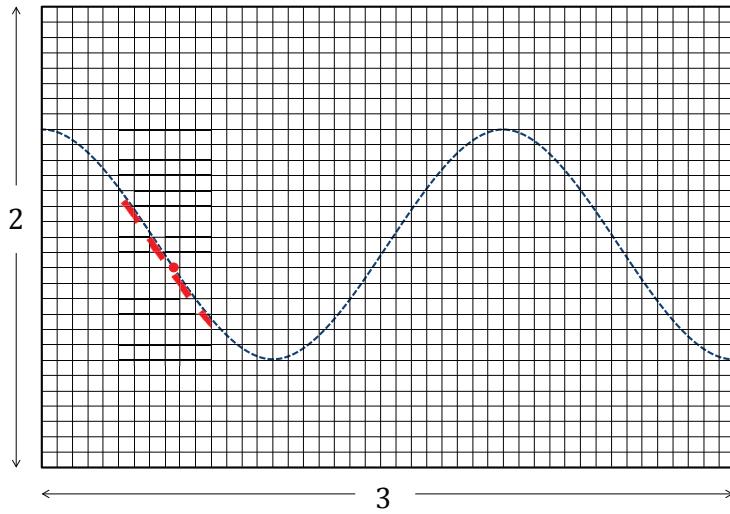


Figure 6.3: Sinusoidal wave with an amplitude of 0.50. Mesh discretization is  $45 \times 30$  elements. The red dot indicates one of the points at which the level set function intersects an element. The red dotted line indicates all cut elements that lie on the same interface of the red dot within a search radius  $r_\kappa = 0.4$ .

structural linear beam elements, as shown in Figure 6.5. The beam elements are assumed to be at rest; therefore, the initial displacements for  $u_x$  and  $u_y$  are equal to zero. For all internal nodes, we define two rotational degrees-of-freedom, one for each element connected to a node. End nodes only have a single rotational degree-of-freedom, as shown in Figure 6.5. We assume the rotations at the internal nodes are continuous:

$$\theta_z^L - \theta_z^R = \alpha_{CR} \quad (6.27)$$

where  $\theta_z^L$  is the rotation of the left adjacent element around the  $z$  axis,  $\theta_z^R$  is the rotation of the right one, and  $\alpha_{CR}$  is computed for every node as a function of the geometrical normal

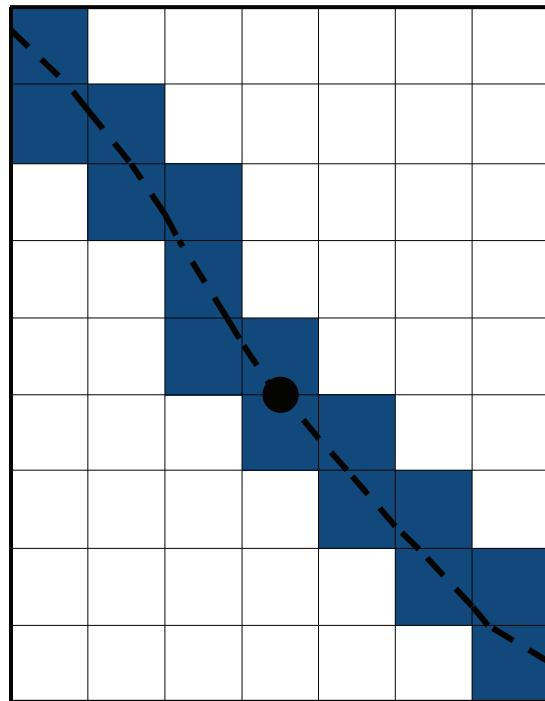


Figure 6.4: Zoomed image for the cut elements within the search radius  $r_\kappa = 0.4$ . Blue elements represent the cut element that are on the same interface.

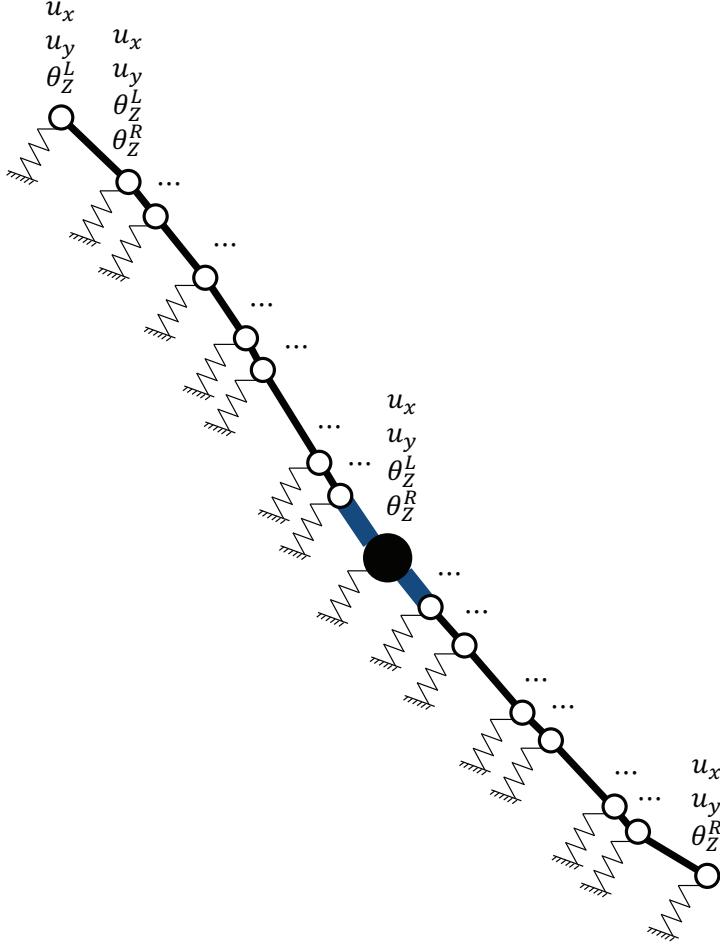


Figure 6.5: Discretization of the structural linear beam elements. Inner nodes have two separate rotational degrees-of-freedom, left and right, one for each adjacent element. Nodes at the end of the interface only have a single rotational degree-of-freedom. All nodes rest on a field of artificial springs.

of its adjacent elements, as shown in Figure 6.6:

$$\begin{aligned}
 \mathbf{n}_g^C &= \frac{\mathbf{n}_g^L + \mathbf{n}_g^R}{2} \\
 \mathbf{n}_g^{LC} &= \mathbf{n}_g^L \times \mathbf{n}_g^C \\
 \mathbf{n}_g^{CR} &= \mathbf{n}_g^C \times \mathbf{n}_g^R \\
 \alpha_{CR} &= \frac{\mathbf{n}_g^{LC}}{\sqrt{1 - \mathbf{n}_g^{LC}^2}} + \frac{\mathbf{n}_g^{CR}}{\sqrt{1 - \mathbf{n}_g^{CR}^2}}
 \end{aligned} \tag{6.28}$$

The finite element problem solves for the displacement and rotational degrees-of-freedom,

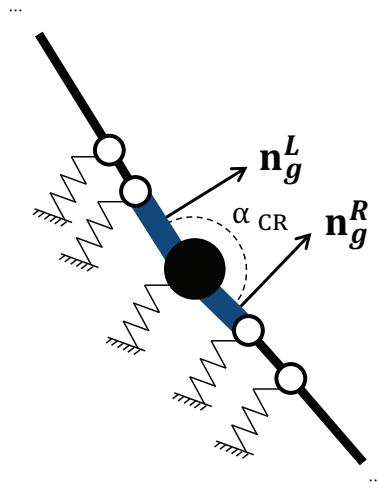


Figure 6.6: Computation of  $\alpha_{CR}$ .  $\alpha_{CR}$  is a function of the geometrical normal unit vectors of the adjacent elements.

and the Lagrange multipliers. Despite the fact that the finite element problem consists of multiple elements, the strain energy of the point of interest is only computed on the adjacent elements as:

$$\Pi^C = \frac{1}{2} (\Pi^L + \Pi^R) \quad (6.29)$$

where  $\Pi^C$  is the strain energy of the intersection point. The reason for this approach is that a larger number of elements yield a more accurate description of the rotations at the point of interest; however, once we get the solution, we are only interested in the strain energy of the adjacent elements because we will perform this procedure for all intersection points in the domain.

Initial computations showed that the strain energy of the system is highly sensitive to the discretization of the level set function. If the projection of a point in the function onto the mesh is off by a small amount due to discretization errors, it leads to large changes in the computed curvature when compared to its analytical value. To ameliorate this issue, we introduce a field of artificial springs with stiffness coefficient  $k_\kappa$  to allow the points to move to what their correct location should be given the enforcement of the rotations, as shown in

Figure 6.5.

## 6.4 Methodology

The squared curvature of two geometrical objects, a circle and a sinusoidal wave, will be studied. The objective of this study is to find the accuracy of our curvature measure by comparing the results to the analytical solutions we obtained in Section 6.1.1. We will use the results of this section to select the most promising approaches and later minimize the curvature of the level set interface.

### 6.4.1 Sweep setup

For our circle sweep, we will model the level set field with the following equation:

$$\phi_i = r_c - \sqrt{x_i^2 + y_i^2} \quad (6.30)$$

where  $\phi_i$  represents a level set value at a node, and  $x_i$  and  $y_i$  are the spatial coordinates. We will vary the radius of the circle  $r_c$  from a value of 0.25 to 0.95 in 100 steps , as shown in Figure 6.7. From Equations 6.9 and 6.16, the curvature will be equal to:

$$\int_{\phi=0} \kappa_n^2 d\Gamma = \frac{2\pi}{r_c} \quad (6.31)$$

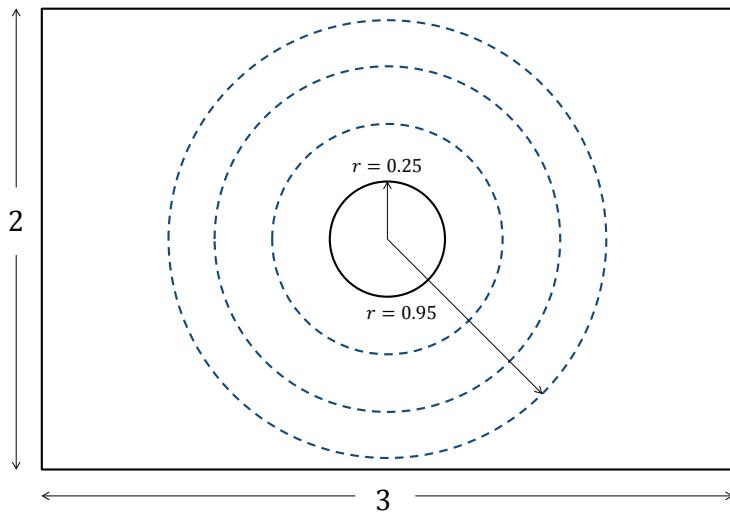


Figure 6.7: Setup for sweeping over the radius of a circle ranging from 0.25 to 0.95.

The sinusoidal wave will be modeled by the following equation:

$$\phi_i = y_i - A_s \sin \pi x_i \quad (6.32)$$

where we will sweep over the amplitude of the sinusoidal wave,  $A_s$ , from 0.25 to 0.75, as shown in Figure 6.32.

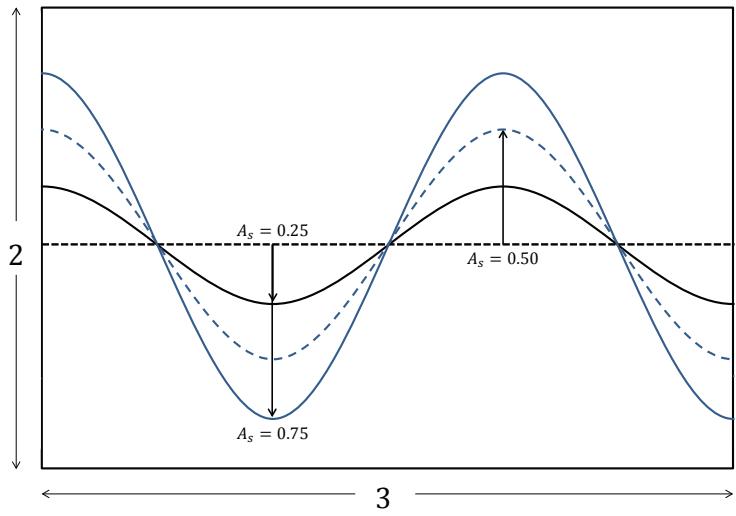


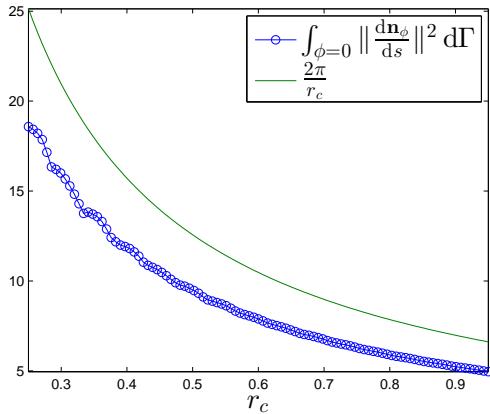
Figure 6.8: Setup for sweeping over the amplitude of a sinusoidal wave ranging from 0.25 to 0.75.

The mesh has spatial dimensions of  $3L \times 2L$ , and will be divided into  $45 \times 30$  elements. The finer version of the mesh will have  $180 \times 120$  elements.

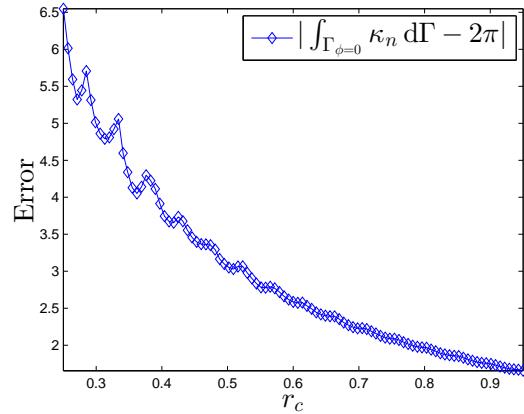
#### 6.4.2 Preliminary sweep results

##### $\mathbf{n}_\phi$ results

The results for the circular sweep with the  $\mathbf{n}_\phi$  formulation and mesh  $45 \times 30$  are shown in Figure 6.9.



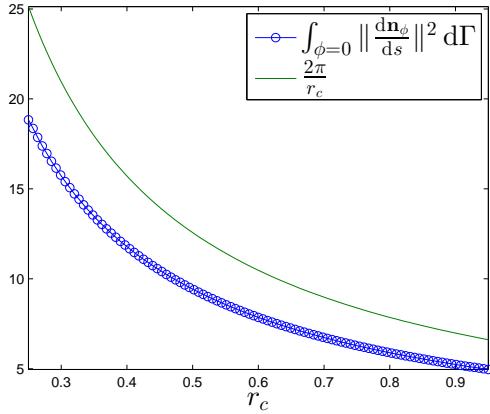
(a)



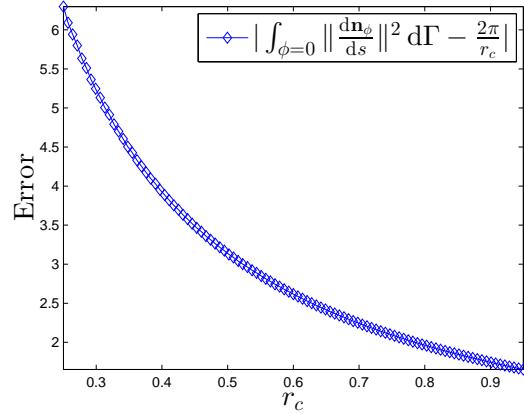
(b)

Figure 6.9: Squared curvature and absolute error for the circular sweep using the  $\mathbf{n}_\phi$  formulation, with mesh  $45 \times 30$ .

The finer mesh results are shown in Figure 6.10.



(a)



(b)

Figure 6.10: Squared curvature and absolute error for the circular sweep using the  $\mathbf{n}_\phi$  formulation, with mesh  $180 \times 120$ .

### $\mathbf{n}_u$ results

Using the  $\mathbf{n}_u$  formulation, we get the results displayed in Figures 6.11 and 6.12 for a circle and sinusoidal wave, respectively. The mesh size is  $45 \times 30$ . The scaling factor,  $\gamma_u$ , is 100.

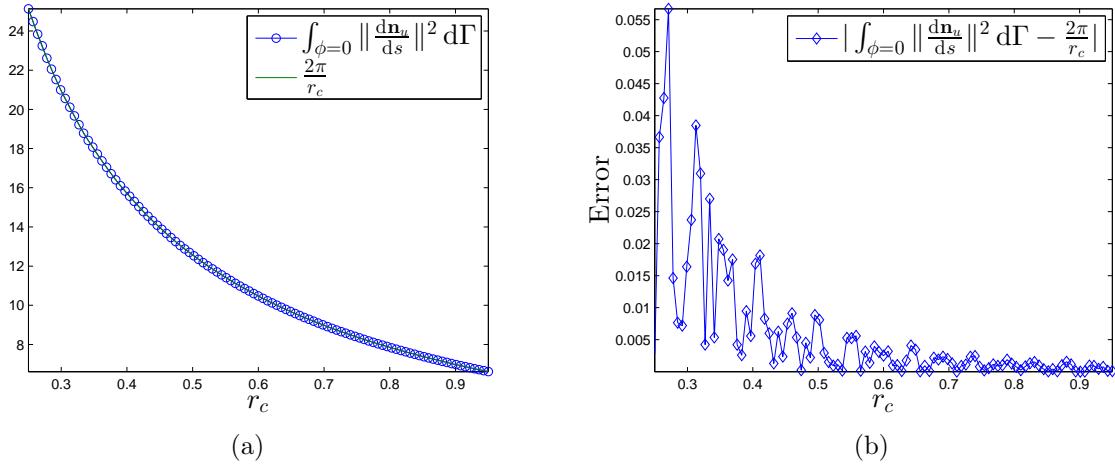


Figure 6.11: Squared curvature and absolute error for the circular sweep using the  $\mathbf{n}_u$  formulation, with mesh  $45 \times 30$ .

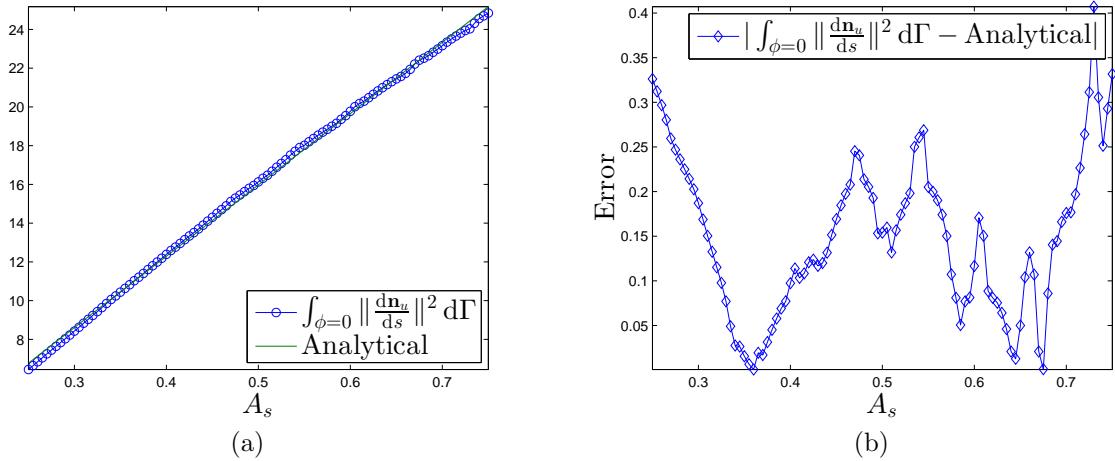


Figure 6.12: Squared curvature and absolute error for the sinusoidal wave sweep using the  $\mathbf{n}_u$  formulation, with mesh  $45 \times 30$ .

### $\mathbf{n}_\psi$ results

Measuring the curvature using the  $\mathbf{n}_\psi$  formulation requires us to define the parameter  $\epsilon$ . We will examine the influence of  $\epsilon = 1.0$ ,  $\epsilon = 2.0$ , and  $\epsilon = 4.0$ , and the benefits of using  $\nabla\psi = \frac{\partial\psi}{\partial\mathbf{x}}$  or  $\nabla\psi = \frac{\partial\psi}{\partial\phi} \frac{\partial\phi}{\partial\mathbf{x}}$ .

---

Point load	$P = 1$
Young's modulus	$E_{xx} = 1$
Element length	$h = 0.066667$
Maximum volume fraction	$\bar{v}_s = 0.5$
Smoothing filter	$d = 1.6 \cdot h$

---

Table 6.1: Parameters for the curvature optimization study.

### Preliminary conclusions of the sweep study

Figures 6.9 and 6.10 show that using  $\mathbf{n}_\phi$  causes oscillations in the measurement if the mesh is not fine enough. Using a projection scheme in the  $\mathbf{n}_u$  and  $\mathbf{n}_v$  formulations yields a smoother curvature measure and decreases the error.

## 6.5 Preliminary optimization results

For all the problems studied in this section, we will model a minimal compliance “solid-void” structural linear optimization problem:

$$\mathcal{F} = \frac{1}{2} w_\Pi \int \boldsymbol{\sigma} : \boldsymbol{\varepsilon} \, d\Omega^s + w_\kappa \int_{\phi=0} \kappa_n \, d\Gamma \quad (6.33)$$

where the strain energy of the structural problem is computed over the solid phase,  $\Omega^s$ , and  $w_\Pi$  and  $w_\kappa$  represent the penalties for the compliance and curvature penalties, respectively. The optimization problem is subject to a volume fraction constraint:

$$\mathcal{G} = \frac{\int d\Omega^s}{\bar{v}_s (\int d\Omega^s + \int d\Omega^v)} - 1 \quad (6.34)$$

where  $\Omega^v$  is the domain of the void region, and  $\bar{v}_s$  is the maximum volume fraction of the solid phase.

The problem setup is shown in Figure 6.13. A list of common parameters for all problems is shown in Table 6.1. The parameters  $w_\Pi$  and  $w_\kappa$ , and the initial design, will be problem dependent values. For all results, the initial design and its respective level set field will be

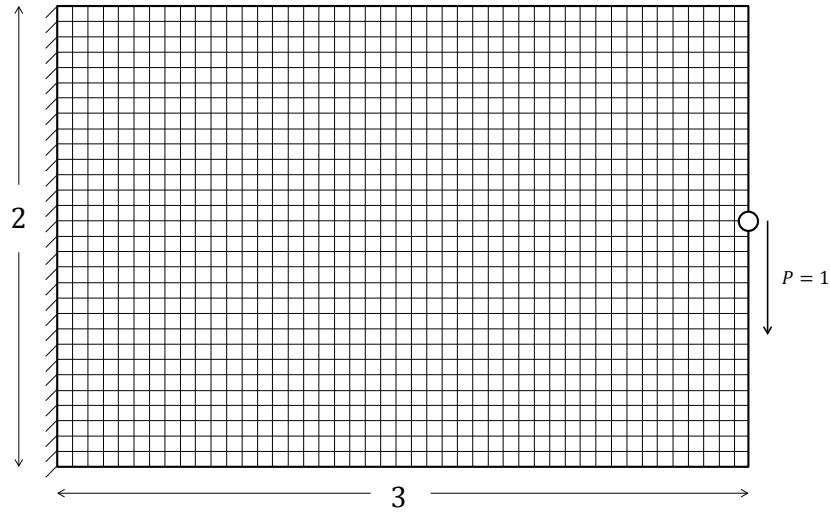


Figure 6.13: Setup for the curvature optimization study. A mesh of size  $3L \times 2L$ , with  $45 \times 30$  quadrilateral linear elements is anchored to the wall on its left side, and subject to a point load on its right side.

displayed in the top left corner of the figures.

$$6.5.1 \quad \kappa_n^2 = \left\| \frac{d\mathbf{n}_u}{ds} \right\|^2$$

The results for equation 6.2 with formulation  $\mathbf{n}_u$  are shown in Figures 6.14 and 6.15.

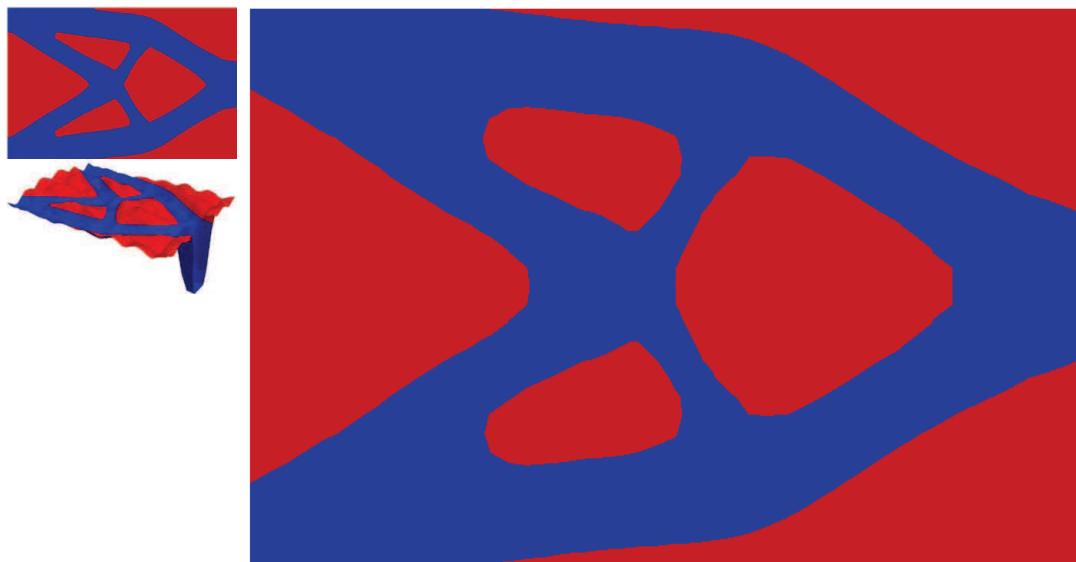


Figure 6.14: Optimized geometry for computing strain energy over the interface 6.2 with formulation  $\mathbf{n}_u$ .  $w_{\Pi} = 1$  and  $w_{\kappa} = 0.1$ .

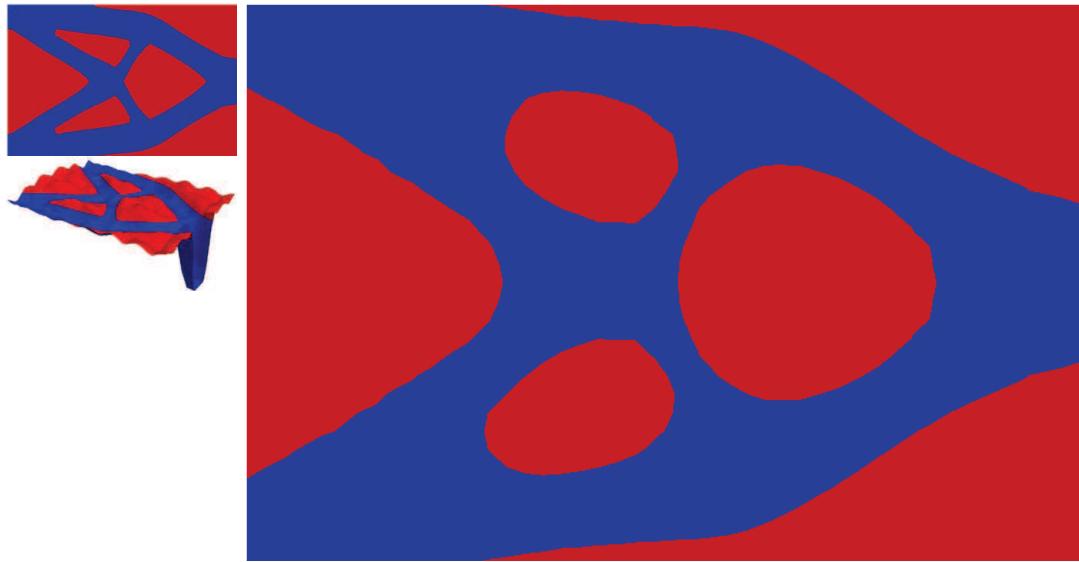


Figure 6.15: Optimized geometry for computing strain energy over the interface 6.2 with formulation  $\mathbf{n}_u$ .  $w_{\Pi} = 1$  and  $w_{\kappa} = 1$ .

### 6.5.2 $\kappa_n^2 = \frac{1}{2}\boldsymbol{\sigma} : \boldsymbol{\varepsilon}$ with $\mathbf{n}_g$

The results for Equation 6.26 with formulation  $\mathbf{n}_g$  are shown in Figures 6.16 and 6.17.

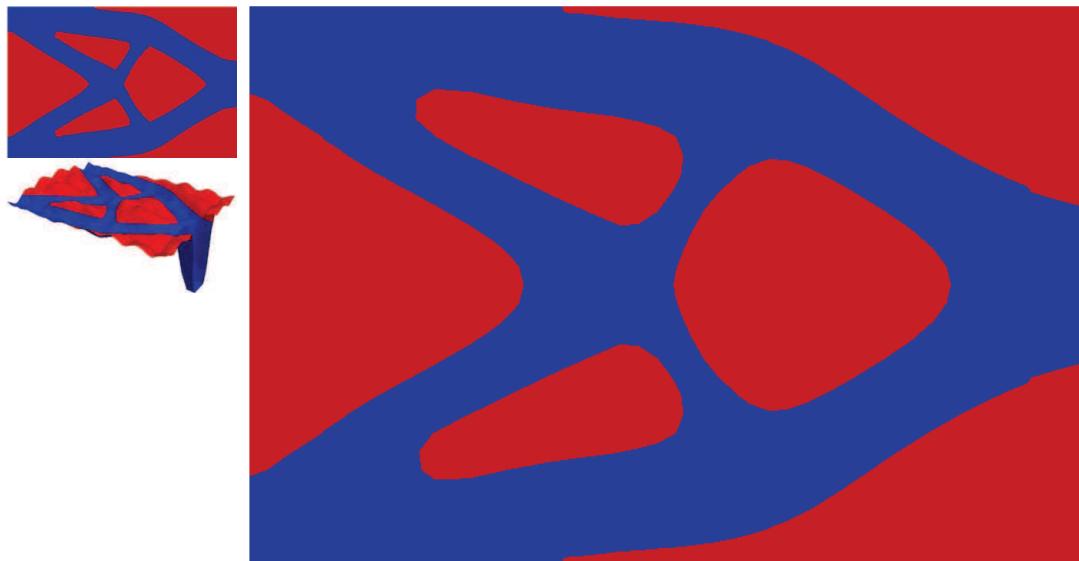


Figure 6.16: Optimized geometry for computing strain energy over the interface 6.26 with formulation  $\mathbf{n}_g$ .  $w_{\Pi} = 1$  and  $w_{\kappa} = 0.1$ .  $k_{\kappa} = 0.01$  and  $r_{\kappa} = 1.2$ .

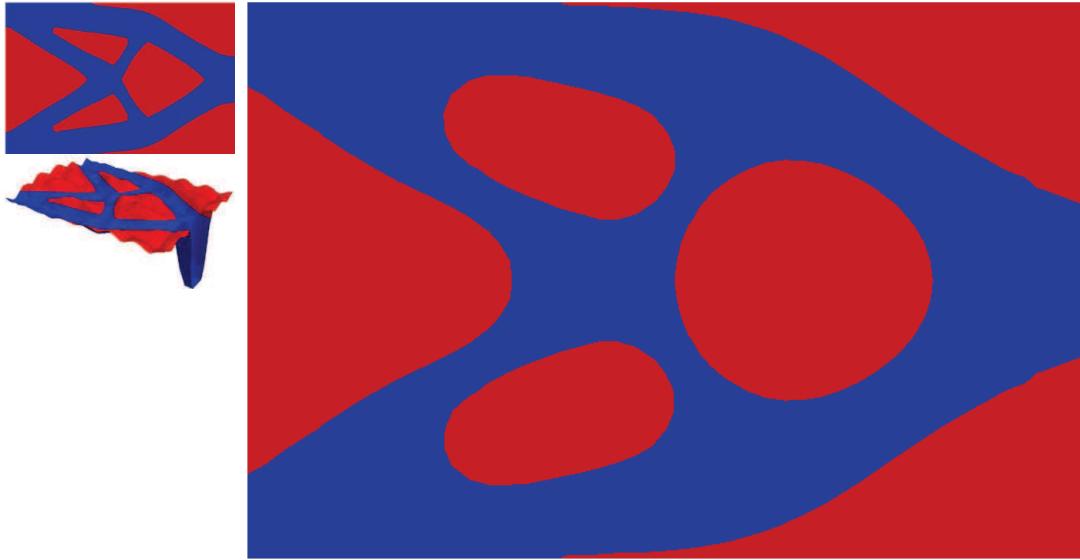


Figure 6.17: Optimized geometry for computing strain energy over the interface 6.26 with formulation  $\mathbf{n}_g$ .  $w_{\Pi} = 1$  and  $w_{\kappa} = 1$ .  $k_{\kappa} = 0.01$  and  $r_{\kappa} = 1.2$ .

## 6.6 Conclusions

Measuring curvature as a function of the level set unit normal yields results that are not smooth at the points where the level set function cuts the elements. Equating curvature to strain energy and measuring its value at the points of intersection yields smoother results, at the cost of increased sensitivity. However, inserting a field of artificial springs to allow for the points to displace reduces the sensitivities and yields smoother optimized geometries.

The author plans to finalize the sweep with  $\mathbf{n}_{\psi}$  and  $\mathbf{n}_g$  and perform further optimization runs.

# Chapter 7

## Draft: Modeling of Multiple Scalar Transport Fields for an Atomic Layer Deposition Machine

Atomic layer deposition (ALD) is a thin film deposition technique in which two or more chemicals react with the surface of a material. The deposition of the chemicals is done in a sequential manner.

The design of an ALD machine is shown in Figure 7.1. The conveyor belt moves the web on which the thin film will be deposited. The chemical reactants enter through the inlets at the top in a sequential manner. Figure 7.2 shows a detailed description of a cross-sectional view of the machine. As was shown in figures 7.1 and 7.2, there are several sections of the machine that have open boundaries. The interaction of the reactants ( $N_2$ , TMA,  $H_2O$ ) with unwanted outside sources, such as air, leads to imperfections in the deposition, as shown in figure 7.3

The goals of this study are to develop a profound understanding of the advective and diffusive transport phenomena which dominate the spatial and temporal distribution of reactants, and to create a novel computational environment for optimally designing the layout,

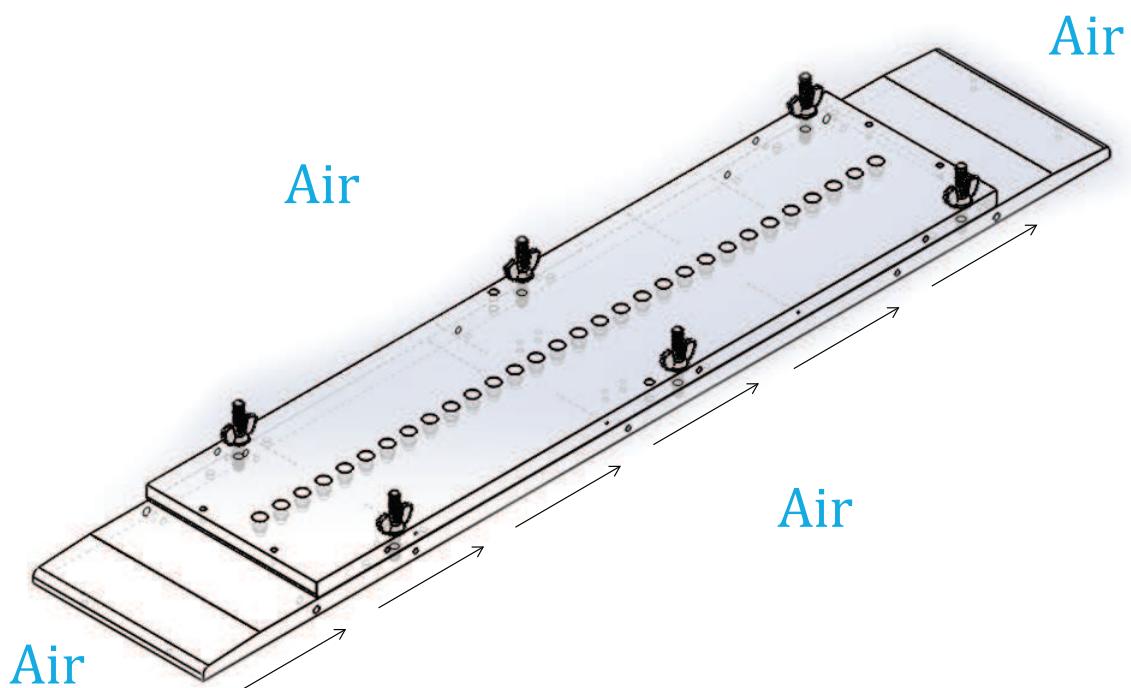


Figure 7.1: CAD model of the atomic layer deposition machine. The holes at the top represent the inlets through which the chemical reactants will enter. The conveyor belt at the bottom moves the web in the direction of the arrows as the chemicals are deposited. The machine has open boundaries which may cause reactions with air, and lead to imperfections in the deposition.

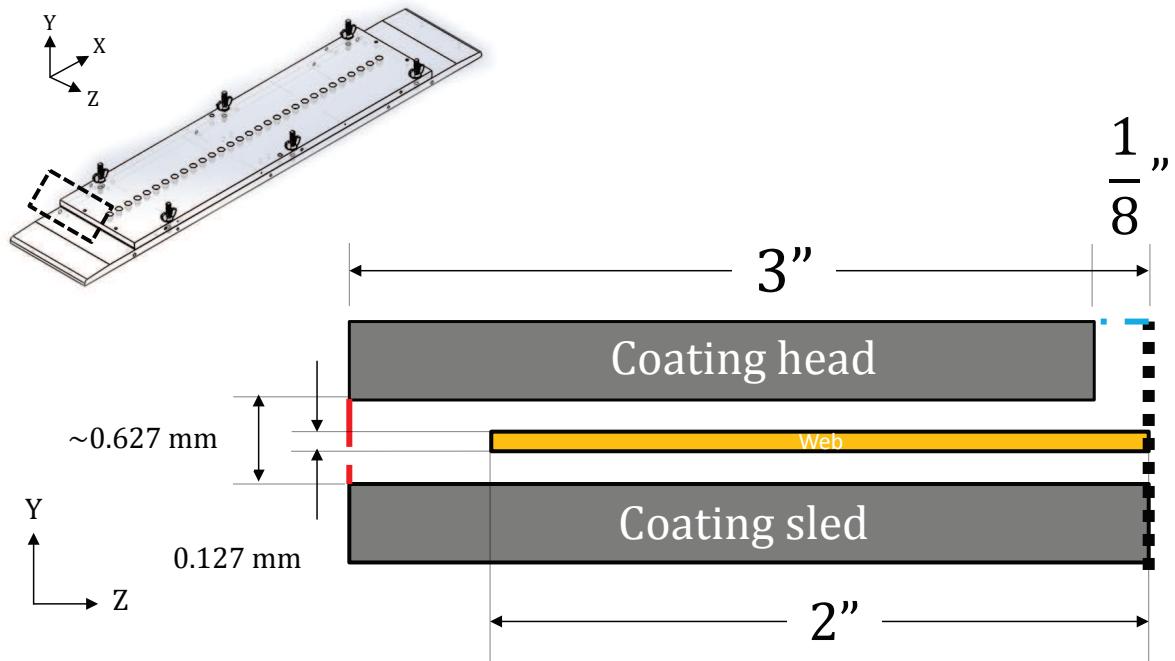


Figure 7.2: Cross-sectional view of the ALD machine. The black dotted line on the right represents a symmetry boundary condition. The coating head and sled are made of solid aluminum. The web on which the reactants are deposited is a polymer. The blue dotted line represents the gas inlets, and the red dotted line represents the open boundary conditions.

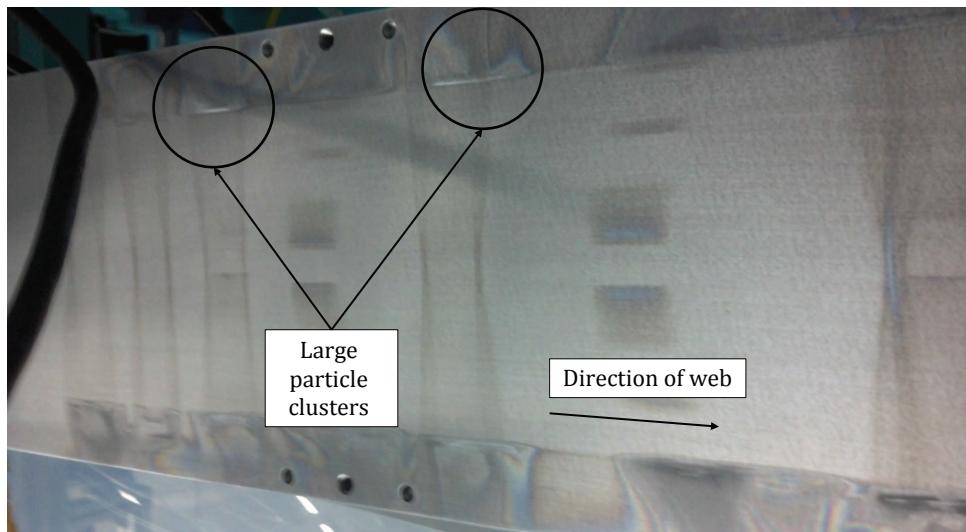


Figure 7.3: View of the coating sled after a deposition was performed. The picture shows that large particle clusters are formed at the open boundaries. Our hypothesis is that air reacts at the edges with the TMA reactant.

the geometry and the operating parameters of the gas source head assembly.

A computational fluid dynamics model was developed to predict the distribution of reactants. The model used an incompressible Navier-Stokes flow model which advects the concentration field of the reactants, augmented with a Brinkman model [13] for modeling the flow in porous media. This model was applied to a gas source head configuration which is used by ALD Nanosolutions and verified against experimental data. Various inlet configurations were studied. The Brinkman model was furthermore augmented to simulate a moving porous material rather than a static one. This model was applied to the moving porous substrate layer that transports the web on which the reactants are applied.

We created the foundation for a level-set based geometry optimization framework where the flow is predicted by the extended finite element method (XFEM). Expanding on previous work we expanded our level-set/XFEM optimization capabilities to three-dimensional problems, an essential capability to design the layout of the gas source head. The level-set XFEM model was expanded to handle “fluid-void” and “fluid-solid” problems in three dimensions, which was essential to model the interaction of the reactants with the porous substrates.

The Brinkman flow modeling research helped to understand how the porous substrate captures and prevents the transport of the reactants. Our initial work on modeling the penetration of reactants into porous substrate was targeted at gaining insight into the effects of the different timescales involved in the transport phenomena. Our current work with the Brinkman model targets the penetration and transport of the reactants into the porous substrate when the web is in motion. Our work with the level-set XFEM optimization framework aimed at developing robust methods to model “fluid-void” and “fluid-solid” problems in three dimensions and optimize the porous substrate in order to minimize the areas of mixing zones.

## 7.1 Conclusions

With respect to the CFD modeling, we have developed a framework to model fluid-structure interaction using a Brinkman model in which the structure has the capability to displace. This will play a key role in the verification of the experimental data obtained from ALD NanoSolutions. At the same time, our level-set XFEM framework allows us to model complex “fluid-fluid” and “fluid-solid” problems and optimize the area where the reactants mix, reducing imperfections in the atomic deposition and allowing for fast new design iterations.

We plan to expand our scalar transport model to simulate multiple scalar transport fields and research the interaction of the reactant with the outside sources, such as air.

We further plan to expand and apply our level-set/XFEM optimization framework to design the geometry of gas source heads.

# Chapter 8

## Conclusions

The extended finite element method used with a level set topology optimization approach is a promising tool for modeling multiphysics problems, and generating optimal designs. The strength of the approach lies in the capacity of the LSM to describe a crisp definition of the phase boundaries, while the XFEM serves as a tool to model discontinuities in the solution.

In Section 3, we studied the ability of the XFEM to decompose a design domain into sub-domains. These subdomains were generated by the intersection of the finite elements in the mesh with a discretized level set function. We then studied an algorithm to determine which enriched degrees-of-freedom to use to integrate the governing equations across disconnected phase regions. Our approach was tested with a two dimensional heat conduction problem. The results showed that our enrichment strategy avoids artificial coupling of disconnected phase regions.

In Section 4, we expanded the work of [64], and the algorithms and results from Section 3 to model three dimensional structural problems. The case studies in this section suggest that the LSM-XFEM framework features an improved convergence as the mesh is refined and is able to represent thin-walled structures on coarse meshes. Density approaches may require a projection scheme 2.5 to achieve clear “0-1” results with comparable strain energies. While the smoothing filter in the SIMP method 2.3 is an efficient and intuitive method to control

the local feature size, neither level set smoothing 2.3 nor imposing a perimeter constraint achieves a similar effect on LSM-XFEM results. We can conclude that the lack of feature size control and the significant improved complexity of the LSM-XFEM formulation, as shown in Section 3 limit the attractiveness of this scheme. However, for problems where a high mesh resolution is not tolerable and/or interface conditions need to be enforced with high accuracy, the LSM-XFEM framework approach might be an interesting alternative to density methods.

In Section 6, we studied a level set regularization technique in the form of a curvature measure. Preliminary results shows that the formulation helps in controlling the global geometry of the design, and generate smoother surfaces. This may prove useful in generating smooth surface meshes for three dimensional printing.

# Chapter 9

## Remaining Work

### 9.1 Ghost penalty formulation

Case studies in the literature show that the ghost penalty formulation leads to a better-posed linear system [17, 93, 94], and smooths the gradient of the solution at the phase interface. These studies have applied this formulation to two dimensional diffusion and incompressible Navier-Stokes flow problems. Our current work shows that we cannot apply our LSM-XFEM framework in high Reynolds number flows in three dimensions. This occurs because small intersection regions cause ill-conditioning of the system that our preconditioner cannot improve, and the stress jumps 5.8 across the phase regions is not enforced accurately. We plan to test if the ghost penalty formulation can be applied to three dimensional flows and provide convergence to our problems. If so, is it robust enough that it can handle the complex shapes the optimization process may generate?

### 9.2 Curvature

The sweep study will be finalized and further optimization runs will be executed. The squared curvature as strain energy 6.26 will be penalized such that low and high values are ignored, and rather intermediate values are reduced during the optimization process. Our

hypothesis is that this penalization approach will ameliorate the sensitivity of the curvature formulation to discretization errors. This penalization, along with the artificial springs, should lead to a well-posed optimization problem. We will study the characteristics of this curvature formulation in two dimensional structural optimization problems.

### **9.3 Atomic layer deposition**

The interaction of the reactants with unwanted outside sources, such as air, leads to imperfections in the deposition. We plan to expand our scalar transport model to simulate multiple scalar transport fields and research the interaction of the reactant with these outside sources. We further plan to expand and apply our LSM-XFEM optimization framework to design the geometry of gas source heads.

### **9.4 Timeline**

The study of curvature as a penalization criteria in an optimization problem will be finalized by April. The optimization of three dimensional incompressible Navier-Stokes and scalar transport problems with a face-oriented ghost penalization will be finalized by the beginning of summer. Summer and fall semester will be dedicated to the atomic layer deposition project, and the modeling of an advection-diffusion problem with multiple scalar fields. The expected graduation date is December 2015.

# Bibliography

- [1] Abdelaziz Y, Hamouine A (2008) A survey of the extended finite element. *Computers & Structures* 86(11):1141–1151
- [2] Allaire G, Jouve F, Toader A (2002) A level-set method for shape optimization. *Comptes Rendus Mathematique* 334(12):1125–1130
- [3] Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics* 194(1):363–393
- [4] Allaire G, Gournay Fd, Jouve F, Toader AM (2005) Structural optimization using topological and shape sensitivity via a level set method. *Control and Cybernetics* 34(1):59–80
- [5] Babuška I, Melenk JM (1997) The partition of unity method. *International Journal for Numerical Methods in Engineering* 40(4):727–758
- [6] Belytschko T, Black T (1999) Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering* 45(5):601–620
- [7] Bendsøe M (1989) Optimal shape design as a material distribution problem. *Structural and Multidisciplinary Optimization* 1(4):193–202
- [8] Bendsøe M, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering* 71(2):197–224

- [9] Bendsøe M, Sigmund O (1999) Material interpolation schemes in topology optimization. *Archive of Applied Mechanics* 69(9-10):635–654
- [10] Bendsøe M, Lund E, Olhoff N, Sigmund O (2005) Topology optimization-broadening the areas of application. *Control and Cybernetics* 34(1):7
- [11] Bendsøe MP, Sigmund O (2003) *Topology Optimization: Theory, Methods and Applications*. Springer
- [12] Borel PI, HarpÃth A, Frandsen LH, Kristensen M, Shi P, Jensen JS, Sigmund O (2004) Topology optimization and fabrication of photonic crystal structures. *Optics Express* 12(9)
- [13] Borrvall T, Petersson J (2003) Topology optimization of fluids in Stokes flow. *International Journal for Numerical Methods in Fluids* 41(1):77–107
- [14] Brooks A, Hughes T (1982) Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 32(1-3):199 – 259
- [15] Burger M, Hackl B, Ring W (2004) Incorporating topological derivatives into level set methods. *Journal of Computational Physics* 194(1):344–362
- [16] Burman E, Hansbo P (2010) Fictitious domain finite element methods using cut elements: I. a stabilized lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering* 199(41):2680–2686
- [17] Burman E, Hansbo P (2012) Fictitious domain finite element methods using cut elements: II. a stabilized nitsche method. *Applied Numerical Mathematics* 62(4):328–341
- [18] Cao Y, Li S, Petzold L (2002) Adjoint sensitivity analysis for differential-algebraic equations: algorithms and software. *Journal of Computational and Applied Mathematics* 140(1-2):19–35

matics 149(1):171 – 191, scientific and Engineering Computations for the 21st Century - Methodologies and Applications Proceedings of the 15th Toyota Conference

- [19] Chessa J, Belytschko T (2003) An extended finite element method for two-phase fluids. Journal of Applied Mechanics 70
- [20] Coolidge J (1952) The unsatisfactory story of curvature. American Mathematical Monthly pp 375–379
- [21] Deaton J, Grandhi R (2013) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Structural and Multidisciplinary Optimization pp 1–38
- [22] van Dijk N, Langelaar M, van Keulen F (2012) Explicit level-set-based topology optimization using an exact heaviside function and consistent sensitivity analysis. International Journal for Numerical Methods in Engineering 91(1):67–97
- [23] Dijk N, Maute K, Langelaar M, Keulen F (2013) Level-set methods for structural topology optimization: a review. Structural and Multidisciplinary Optimization 48(3):437–472
- [24] Dolbow J, Harari I (2009) An efficient finite element method for embedded interface problems. Int J Numer Meth Engng 78:229–252
- [25] Duysinx P, Mie groet L, Jacobs T, Fleury C (2006) Generalized shape optimization using x-fem and level set methods. In: IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials, Springer, pp 23–32
- [26] Dyn N, Hormann K, Kim SJ, Levin D (2001) Optimizing 3d triangulations using discrete curvature analysis. In: Lyche T, Schumaker LL (eds) Mathematical Methods for Curves and Surfaces, Vanderbilt University, chap Optimizing 3D Triangulations Using Discrete Curvature Analysis, pp 135–146

- [27] Erentok A, Sigmund O (2011) Topology optimization of sub-wavelength antennas. *Antennas and Propagation, IEEE Transactions on* 59(1):58–69
- [28] Eschenauer H, Olhoff N (2001) Topology optimization of continuum structures: a review. *Applied Mechanics Reviews* 54:331
- [29] Eschenauer H, Kobelev V, Schumacher A (1994) Bubble method for topology and shape optimization of structures. *Structural and Multidisciplinary Optimization* 8(1):42–51
- [30] Farhat C, Argrow B, Nikbay M, Maute K (2004) Shape optimization with f-function lobe balancing for reducing the sonic boom initial shock pressure rise. *International Journal of Aeroacoustics* 3:361–378
- [31] Frenzel M (2004) Topology optimization for wave problems. Master's thesis, University of Colorado at Boulder
- [32] Fries T, Belytschko T (2010) The extended/generalized finite element method: an overview of the method and its applications. *International Journal for Numerical Methods in Engineering* 84(3):253–304
- [33] Fries TP (2009) The intrinsic xfem for two-fluid flows. *Int J Numer Meth Fluids* 60(4):437–471
- [34] Fuchs M, Jiny S, Peleg N (2005) The srv constraint for 0/1 topological design. *Structural and Multidisciplinary Optimization* 30(4):320–326
- [35] Gain AL, Paulino GH (2013) A critical comparative assessment of differential equation-driven methods for structural topology optimization. *Structural and Multidisciplinary Optimization* 48(4):685–710
- [36] Gerstenberger A, Wall WA (2008) Enhancement of fixed-grid methods towards complex fluid-structure interaction applications. *Int J Numer Meth Fluids* 57(9):1227–1248

- [37] Gerstenberger A, Wall WA (2008) An extended finite element method/Lagrange multiplier based approach for fluid-structure interaction. Computer Methods in Applied Mechanics and Engineering 197:1699–1714
- [38] Giannakoglou K, Papadimitriou D (2008) Adjoint methods for shape optimization. In: Thvenin D, Janiga G (eds) Optimization and Computational Fluid Dynamics, Springer Berlin Heidelberg, pp 79–108
- [39] Gill PE, Murray W, Saunders MA (2002) Snopt: An sqp algorithm for large-scale constrained optimization. SIAM Journal on Optimization 12(4):979–1006
- [40] Guest J, Prévost J, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. International Journal for Numerical Methods in Engineering 61(2):238–254
- [41] Guest J, Asadpoure A, Ha SH (2011) Eliminating beta-continuation from heaviside projection and density filter algorithms. Structural and Multidisciplinary Optimization 44(4):443–453
- [42] Hansbo A, Hansbo P (2004) A finite element method for the simulation of strong and weak discontinuities in solid mechanics. Computer Methods in Applied Mechanics and Engineering 193(33-35):3523 – 3540
- [43] Hansbo A, Hansbo P (2004) A finite element method for the simulation of strong and weak discontinuities in solid mechanics. Computer methods in applied mechanics and engineering 193(33):3523–3540
- [44] Hansbo P (2005) Nitsche’s method for interface problems in computational mechanics. GAMM-Mitteilungen 28(2):183–206
- [45] Hashin Z, Shtrikman S (1962) On some variational principles in anisotropic and non-

homogeneous elasticity. *Journal of the Mechanics and Physics of Solids* 10(4):335 – 342

- [46] Heroux M, Bartlett R, Hoekstra VHR, Hu J, Kolda T, Lehoucq R, Long K, Pawlowski R, Phipps E, Salinger A, Thornquist H, Tuminaro R, Willenbring J, Williams A (2003) An Overview of Trilinos. *Tech. Rep.* SAND2003-2927, Sandia National Laboratories
- [47] Herrero D, Martínez J, Martí P (2013) An implementation of level set based topology optimization using gpu. In: 10th World Congress on Structural and Multidisciplinary Optimization
- [48] Juntunen M, Stenberg R (2009) Nitsches method for general boundary conditions. *Mathematics of Computation* 78:1353–1374
- [49] Kao C, Osher S, Yablonovitch E (2005) Maximizing band gaps in two-dimensional photonic crystals by using level set methods. *Applied Physics B: Lasers and Optics* 81(2):235–244
- [50] Kondoh T, Matsumori T, Kawamoto A (2012) Drag minimization and lift maximization in laminar flows via topology optimization employing simple objective function expressions based on body force integration. *Structural and Multidisciplinary Optimization* 45(5):693–701
- [51] Kreisselmeier G, Steinhauser R (1979) Systematic control design by optimizing a vector performance index. In: International Federation of Active Contrals Symposium on Computer Aided Design of Control Systems, Zurich, Switzerland
- [52] Kreissl S, Maute K (2011) Topology optimization for unsteady flow. *International Journal for Numerical Methods in Engineering* 87:1229–1253
- [53] Kreissl S, Maute K (2012) Levelset based fluid topology optimization using the

extended finite element method. Structural and Multidisciplinary Optimization 46(3):311–326

- [54] Kreissl S, Pingen G, Maute K (2011) An explicit level set approach for generalized shape optimization of fluids with the lattice boltzmann method. International Journal for Numerical Methods in Fluids 65(5):496–519
- [55] Labbé T, Glineur F, Dehez B (2009) Topology optimization method applied to the design of electromagnetic devices: focus on convexity issues. In: 8<sup>th</sup> International Symposium on Advanced Electromechanical Motion Systems and Electric Drives Joint Symposium, ELECTROMOTION 2009
- [56] Lang C, Makhija D, Doostan A, Maute K (2013) A simple and efficient preconditioning scheme for xfem with heaviside enrichments. Computational Mechanics
- [57] Lee DT, Schachter BJ (1980) Two algorithms for constructing a delaunay triangulation. International Journal of Computer & Information Sciences 9(3):219–242, j2: International Journal of Computer and Information Sciences
- [58] Lee P, Yang R, Maute K (2011) An Extended Finite Element Method for the Analysis of Submicron Heat Transfer Phenomena, Multiscale Methods in Computational Mechanics, Lecture Notes in Applied and Computational Mechanics, vol 55, Springer, pp 195–212
- [59] Legay A, Chessa J, Belytschko T (2006) An eulerianlagrangian method for fluidstructure interaction based on level sets. Computer Methods in Applied Mechanics and Engineering 195(1718):2070 – 2087, fluid-Structure Interaction
- [60] Li L, Wang M, Wei P (2012) Xfem schemes for level set based structural optimization. Frontiers of Mechanical Engineering 7(4):335–356

- [61] Lorensen WE, Cline HE (1987) Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH Comput Graph 21(4):163–169
- [62] Luo J, Luo Z, Chen L, Tong L, Wang M (2008) A semi-implicit level set method for structural shape and topology optimization. Journal of Computational Physics 227(11):5561–5581
- [63] Makhija D, Maute K (2014) Level set topology optimization of scalar transport problems. Structural and Multidisciplinary Optimization DOI 10.1007/s00158-014-1142-7
- [64] Makhija D, Maute K (2014) Numerical instabilities in level set topology optimization with the extended finite element method. Structural and Multidisciplinary Optimization 49(2):185–197
- [65] Makhija D, Pingen G, Yang R, Maute K (2012) Topology optimization of multi-component flows using a multi-relaxation time lattice boltzmann method. Computers & Fluids 67(0):104 – 114
- [66] Maute K (2014) Topology optimization of flows: Stokes and navier-stokes models. In: Rozvany G, Lewiski T (eds) Topology Optimization in Structural and Continuum Mechanics, CISM International Centre for Mechanical Sciences, vol 549, Springer Vienna, pp 409–419
- [67] Maute K, Ramm E (1995) Adaptive topology optimization. Structural and Multidisciplinary Optimization 10(2):100–112
- [68] Maute K, Ramm E (1997) Adaptive topology optimization of shell structures. AIAA Journal 35(11):1767–1773
- [69] Maute K, Schwarz S, Ramm E (1998) Adaptive topology optimization of elastoplastic structures. Structural and Multidisciplinary Optimization 15(2):81–91

- [70] Maute K, Kreissl S, Makhija D, Yang R (2011) Topology optimization of heat conduction in nano-composites. In: 9<sup>th</sup> World Congress on Structural and Multidisciplinary Optimization, Shizuoka, Japan
- [71] Meisel N, Gaynor A, Williams C, Guest J (2013) Multiple-material topology optimization of compliant mechanisms created via polyjet 3d printing. In: Twenty Forth Annual International Solid Freeform Fabrication Symposium An Additive Manufacturing Conference, August 12-14, 2013, Austin, TX
- [72] Michell AGM (1904) The limits of economy of material in frame-structures. *Philos Magazine* S6 8(47):589–597
- [73] van Miegroet L, Duysinx P (2007) Stress concentration minimization of 2d filets using x-fem and level set description. *Structural and Multidisciplinary Optimization* 33(4-5):425–438
- [74] van Miegroet L, Moës N, Fleury C, Duysinx P (2005) Generalized shape optimization based on the level set method. In: 6<sup>th</sup> World Congress of Structural and Multidisciplinary Optimization
- [75] Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization, vol 18. Wiley New York
- [76] Newton I, Colson J (1736) The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines.
- [77] Nguyen T, Song J, Paulino G (2010) Challenges and advances in system reliability-based optimization of structural topology. In: Structures Congress 2010, pp 480–491
- [78] Nguyen T, Paulino G, Song J, Le C (2012) Improving multiresolution topology optimization via multiple discretizations. *International Journal for Numerical Methods in Engineering* 92(6):507–530

- [79] Nieuwenhuis C, Töppe E, Gorelick L, Veksler O, Boykov Y (2013) Efficient regularization of squared curvature. CoRR abs/1311.1838
- [80] Ning X, Pellegrino S (2012) Design of lightweight structural components for direct digital manufacturing. In: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference
- [81] Nitsche J (1975) Über ein variationsprinzip zur loesung yon dirichlet-problemen bei verwendung yon teilraeumen, die keinen randbedingungen unterworfen sind. Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 36:9–15
- [82] Norato J, Bendsøe M, Haber R, Tortorelli D (2007) A topological derivative method for topology optimization. Structural and Multidisciplinary Optimization 33(4):375–386
- [83] Olsson C, Boykov Y (2012) Curvature-based regularization for surface approximation. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, pp 1576–1583
- [84] Olsson E, Kreiss G (2005) A conservative level set method for two phase flow. Journal of Computational Physics 210(1):225 – 246
- [85] Olsson E, Kreiss G, Zahedi S (2007) A conservative level set method for two phase flow {II}. Journal of Computational Physics 225(1):785 – 807
- [86] Osher S, Paragios N (2003) Geometric level set methods in imaging, vision, and graphics. Springer-Verlag New York Inc
- [87] Osher S, Santosa F (2001) Level set methods for optimization problems involving geometry and constraints: I. frequencies of a two-density inhomogeneous drum. Journal of Computational Physics 171(1):272 – 288
- [88] Osher SJ, Sethian JA (1988) Fronts propagating with curvature dependent speed:

algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics* 79:12–49

- [89] Park J, Sutradhar A (2014) A multi-resolution method for 3d multi-material topology optimization. *Computer Methods in Applied Mechanics and Engineering* (0):–
- [90] Rozvany G (2009) A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization* 37(3):217–237
- [91] Rozvany G, Zhou M, Sigmund O (1994) Optimization of topology. *Advances in design optimization* Chapman & Hall, London pp 340–399
- [92] Schleupen A, Maute K, Ramm E (2000) Adaptive fe-procedures in shape optimization. *Structural and Multidisciplinary Optimization* 19:282–302
- [93] Schott B, Wall W (2014) A new face-oriented stabilized xfem approach for 2d and 3d incompressible navier–stokes equations. *Computer Methods in Applied Mechanics and Engineering* 276:233–265
- [94] Schott B, Rasthofer U, Gravemeier V, Wall W (2014) A face-oriented stabilized nitsche-type extended variational multiscale method for incompressible two-phase flow. *International Journal for Numerical Methods in Engineering*
- [95] Sethian J, Wiegmann A (2000) Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics* 163(2):489–528
- [96] Shim H, Ho V, Wang S, Tortorelli D (2008) Topological shape optimization of electromagnetic problems using level set method and radial basis function. *CMES: Computer Modeling in Engineering & Sciences* 37(2):175–202
- [97] Sigmund O (2001) A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization* 21(2):120–127

- [98] Sigmund O (2001) Design of multiphysics actuators using topology optimization – Part I: One-material structures. *Computer Methods in Applied Mechanics and Engineering* 190(49–50):6577–6604
- [99] Sigmund O (2001) Design of multiphysics actuators using topology optimization - part II: Two-material structures. *Computer Methods in Applied Mechanics and Engineering* 190(49–50):6605–6627
- [100] Sigmund O (2007) Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization* 33(4-5):401–424
- [101] Sigmund O, Maute K (2013) Topology optimization approaches: A comparative review. *Structural and Multidisciplinary Optimization*
- [102] Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural and Multidisciplinary Optimization* 16(1):168–75
- [103] Sokolowski J, Zochowski A (1999) Topological derivatives for elliptic problems. *Inverse problems* 15:123
- [104] Stenberg R (1995) On some techniques for approximating boundary conditions in the finite element method. *Journal of Computational and Applied Mathematics* 63(1-3):139 – 148
- [105] Stolarska M, Chopp D, Moes N, Belytschko T (2001) Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering* 51(8):943–960
- [106] Stolpe M, Svanberg K (2001) An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization* 22(2):116–124

- [107] Svanberg K (1987) The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering* 24(2):359–373
- [108] Svanberg K (1995) A globally convergent version of MMA without linesearch. In: *Proceedings of the First World Congress of Structural and Multidisciplinary Optimization*, 28 May - 2 June 1995, Goslar, Germany, pp 9–16
- [109] Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM J on Optimization* 12(2):555–573
- [110] Tcherniak D, Sigmund O (2001) A web-based topology optimization program. *Structural and Multidisciplinary Optimization* 22(3):179–187
- [111] Tezduyar TE, Mittal S, Ray SE, Shih R (1992) Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering* 95:221–242
- [112] Vese LA, Chan TF (2002) A multiphase level set framework for image segmentation using the mumford and shah model. *International journal of computer vision* 50(3):271–293
- [113] Villanueva CH, Maute K (2014) Density and level set-xfem schemes for topology optimization of 3-d structures. *Comput Mech* 54(1):133–150
- [114] Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization* 43(6):767–784
- [115] Wang M, Wang X (2004) “Color” level sets: a multi-phase method for structural topology optimization with multiple materials. *Computer Methods in Applied Mechanics and Engineering* 193(6-8):469–496

- [116] Wang M, Wang X (2005) A level-set based variational method for design and optimization of heterogeneous objects. *Computer-Aided Design* 37(3):321–337
- [117] Wang MY, Wang X, Guo D (2003) A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering* 192(1-2):227–246
- [118] Wang S, Wang MY (2006) A moving superimposed finite element method for structural topology optimization. *Int J Numer Meth Engng* 65:1892–1922
- [119] Wchter A, Biegler LT (2006) On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming* 106(1):25–57
- [120] Wei P, Wang M, Xing X (2010) A study on X-FEM in continuum structural optimization using a level set model. *Computer-Aided Design* 42(8):708–719
- [121] Wilke DN, Kok S, Groenwold AA (2006) A quadratically convergent unstructured remeshing strategy for shape optimization. *International journal for numerical methods in engineering* 65(1):1–17
- [122] Xu S, Cai Y, Cheng G (2010) Volume preserving nonlinear density filter based on heaviside functions. *Structural and Multidisciplinary Optimization* 41(4):495–505
- [123] Yaji K, Yamada T, Kubo S, Izui K, Nishiwaki S (2015) A topology optimization method for a coupled thermalfluid problem using level set boundary expressions. *International Journal of Heat and Mass Transfer* 81(0):878 – 888
- [124] Yamada T, Izui K, Nishiwaki S, Takezawa A (2010) A topology optimization method based on the level set method incorporating a fictitious interface energy. *Computer Methods in Applied Mechanics and Engineering* 199(45-48):2876–2891
- [125] Yamasaki S, Nishiwaki S, Yamada T, Izui K, Yoshimura M (2010) A structural optimization method based on the level set method using a new geometry-based re-

initialization scheme. International Journal for Numerical Methods in Engineering 83(12):1580–1624

- [126] Yamasaki S, Nomura T, Kawamoto A, Sato K, Nishiwaki S (2011) A level set-based topology optimization method targeting metallic waveguide design problems. International Journal for Numerical Methods in Engineering 87(9):844–868
- [127] Yazid A, Abdelkader N, Abdelmajid H (2009) A state-of-the-art review of the x-fem for computational fracture mechanics. Applied Mathematical Modelling 33(12):4269 – 4282
- [128] Yin L, Ananthasuresh G (2001) Topology optimization of compliant mechanisms with multiple materials using a peak function material interpolation scheme. Structural and Multidisciplinary Optimization 23(1):49–62
- [129] Zhou M, Rozvany GIN (1991) The COC algorithm, part II: Topological, geometrical and generalized shape optimization. Computer Methods in Applied Mechanics and Engineering 89(1-3):309–336
- [130] Zienkiewicz O, Taylor RL, Zhou S (2005) Finite Element Method - Its Basis and Fundamentals, 4th edn, Elsevier, chap 9, pp 206–256