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

by

Carlos Hernán Villanueva Pérez

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

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

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Mechanical Engineering

2016

This thesis entitled:
Topology Optimization using the Level Set and eXtended Finite Element Methods:
Theory and Applications
written by Carlos Hernán Villanueva Pérez
has been approved for the Department of Mechanical Engineering

Prof. Kurt Maute

Prof. Alireza Doostan

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Villanueva Pérez, Carlos Hernán (Ph.D., Mechanical Engineering)

Topology Optimization using the Level Set and eXtended Finite Element Methods:

Theory and Applications

Thesis directed by Prof. Kurt Maute

Computational design optimization provides designers with automated techniques to develop novel and non-intuitive optimal designs. Topology optimization is a design optimization technique that allows for the evolution of a broad variety of geometries in the optimization process. Traditional density-based topology optimization methods often lack a sufficient resolution of the geometry and physical response, which prevents direct use of the optimized design in manufacturing and the accurate modeling of the physical response of boundary conditions. The goal of this thesis 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 methodology is presented as an alternative to density-based optimization approaches, and is able to accommodate a broad range of engineering design problems. The framework presents state-of-the-art methods for immersed boundary techniques to stabilize the systems of equations and enforce the boundary conditions, and is studied with applications in 2D and 3D linear elastic structures, incompressible flow, and energy and species transport problems to test the robustness and the characteristics of the method. A comparison of the framework against density-based topology optimization approaches is studied with regards to convergence, performance, and the capability to manufacture the designs. Furthermore, the ability to control the shape of the design to operate within manufacturing constraints is developed and studied. The analysis capability of the framework is validated quantitatively through comparison against previous benchmark studies, and qualitatively through its application to topology optimization problems. The design optimization problems converge to intuitive designs and resembled well the results from previous 2D or density-based studies.

Contents

Chapter

Appendix

Tables

Table

Figures

Figure

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 is referred to as the LSM-XFEM optimization method, and is presented as an alternative to homogenization optimization approaches. Methodologies to accurately enforce boundary conditions and to maintain stability of the system of equations are studied. The framework is applied in linear elastic structures, incompressible flow, and energy and species transport problems to examine the robustness and the characteristics of the method. The ability to control the shape of the design to operate within manufacturing constraints and to predict relevant physical phenomena is also considered.

1.1 Motivation

Topology optimization approaches seek the optimal material distribution of a body within a given design domain. Originally, topology optimization methods were developed primarily to create conceptual designs of engineering systems in the early stages of the design process [Bendsøe and Sigmund, 2003; Rozvany, 2009]. Before the advent of Computational Fluid Dynamics (CFD) and the Finite Element Method (FEM) in the late 1960s, the design of these systems was driven by experimental studies. However, with the increase of computational power and the development of improved numerical schemes, several problems in structural mechanics and fluid dynamics could be simulated and analyzed numerically. Consequently, in recent decades, topology optimization has gained traction as a practical computational design technique. The method is appealing because, unlike shape optimization, it minimizes the influence of the initial design with its ability to produce both shape and topological changes during the optimization process.

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 3D 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 [Ning and Pellegrino, 2012; Meisel et al, 2013].

Traditional topology optimization methods, namely homogenization schemes [Bendsøe and Kikuchi, 1988], aim at finding the conceptual design, but often lack a sufficient resolution of the geometry and physical response. The lack of these two characteristics prevents direct use of the optimized design in manufacturing, and an accurate modeling of the physical response of boundary conditions. To overcome these limitations, this thesis studies the viability and characteristics of the eXtended Finite Element Method in combination with the Level Set Method as a topology optimization framework for 2D and 3D design problems.

The specific objectives of this thesis are then: (i) to develop a robust LSM-XFEM topology optimization scheme; (ii) to compare the LSM-XFEM optimization scheme with traditional homogenization methods, such as SIMP [Bendsøe, 1989], and study the advantages and disadvantages of our formulation; and (iii) to explore the characteristics of the methodology through cases studies in linear elasticity, incompressible flow, and energy and species transport examples. We aim to expand and study the LSM-XFEM topology optimization framework with respect to the following characteristics: (i) **genericity**: we will measure genericity by the ability of the framework to be applicable to a broad range of 2D and 3D problems with different physical phenomena, and to material-void and material-material problems; (ii) **efficiency**: we will measure efficiency by the ability of the method to reduce the computational cost against traditional homogenization methods, and by the convergence of the optimized geometry with mesh refinement; and (iii) **robustness**: we will measure the robustness of the method by its ability to control the shape of the design to operate within manufacturing constraints, its ability to predict relevant physical phenomena and to accurately enforce boundary conditions, and its application to real-life design problems.

1.2 Background

This section presents a brief overview on optimization problems and traditional density-based topology optimization methods. The information presented in this chapter is sufficient such that the reader can understand the work done prior to this thesis and the motivation for our study, but it is not intended to be comprehensive. References are provided for the reader who wishes to see more details on the topics.

1.2.1 Optimization

An optimization problem is a type of problem in which you seek to find the best solution from the set of all feasible solutions, with respect to certain optimization criteria. The class of optimization problems considered in this thesis involve the design of engineering systems. Therefore, the solution to the problem is described by a set of design variables. There are two categories of optimization problems, and their classification depends on whether the design 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 Nemhauser and Wolsey [1988] for a review.

The optimization problems are formulated with respect to an objective and one or more constraints for some desired functionality. The objective and constraints are defined in terms of design criteria, such as drag, power dissipation, fluid volume, etc. These design criteria can depend explicitly on the state and optimization variables (e.g. drag) or only on the optimization variables (e.g. fluid volume). As the state variables may depend on time, the design criteria are either evaluated over a time integral or at a given instance in time. The formulation for our optimization problems looks as follows:

$$\begin{aligned} \min_{\boldsymbol{s}} \quad \mathcal{Z} &= \int_{t_1}^{t_2} z\left(\boldsymbol{s}, \boldsymbol{u}\left(t\right)\right) \, \mathrm{d}t \ , \\ \text{s.t.} \quad & \int_{t_1}^{t_2} g_i\left(\boldsymbol{s}, \boldsymbol{u}\left(t\right)\right) \leq 0 \quad i = 1 \ldots N_g \ , \\ \boldsymbol{s} \in \boldsymbol{S} &= \left\{\mathbb{R}^{N_{\boldsymbol{s}}} | s_i^L \leq s_i \leq s_i^U, i = 1 \ldots N_s \right\} \ , \\ \boldsymbol{u}\left(t\right) \in \boldsymbol{U} &= \left\{\mathbb{R}^{N_{\boldsymbol{u}}} | \boldsymbol{R}\left(\boldsymbol{s}, \boldsymbol{u}\left(t\right)\right) = 0 \right\} \ , \end{aligned} \tag{1.1}$$

where $\boldsymbol{s}$ is the vector of optimization variables, of size N_s, and $\boldsymbol{u}\left(t\right)$ is the vector of time-dependent state variables, of size $N_{\boldsymbol{u}}$. The objective function $\mathcal{Z}$ is the integral of the time-dependent function z over the interval $[t_1, t_2]$. The function g_i is the i-th optimization constraint, and N_g is the number of inequality constraints. The optimization variables s_i are bounded by lower and upper limits, s_i^L and s_i^U, respectively. The state variables satisfy the residual of the governing equations, $\boldsymbol{R}\left(\boldsymbol{s}, \boldsymbol{u}\left(t\right)\right) = 0$. In this thesis, the optimization problem is considered converged if the design variables or the optimization objective only exhibit small changes over a number of optimization iterations, and if all constrainst are satisfied.

There are a variety of optimization algorithms to solve the optimization problem in (1.1). These algorithms range from simple formulations, like random walk and steepest descent to more complex ones, like the Globally Convergent Method of Moving Asymptotes (GCMMA) [Svanberg, 2002]. For a reference on other optimization mathematical tools, such as the Sequential Quadratic Programming (SQP) or the Sparse Nonlinear OPTimizer (SNOPT) [Gill et al, 2002], the reader is referred to the work of Bendsøe and Sigmund [2003]. Optimization algorithms can also be grouped by their use of the gradient information. Gradient-free methods neglect gradient information, and therefore do not require continuous, differentiable objective functions. However, the computational cost of these methods increases as the number of design variables grows larger. Sigmund [2011] showed that these methods quickly become impractical as mesh refinement increases. On the other hand, gradiend-based methods use the sensitivities of the objective and constraint functions with respect to the design variables to choose a search direction and minimize the functionals towards a local or global minima. Gradient-based methods address the scaling concerns of gradient-free methods, using the gradients to choose good search directions. Sigmund [2011] showed that gradient-based methods typically require an order of magnitude less design iterations as compared to gradient-free algorithms. In the following section, we will explore the computation of the gradients for gradient-based optimization problems.

1.2.2 Sensitivity Analysis

The gradients of the objective functional and the design constraints with respect to the design variables are required by gradient-based methods. In this section, we derive the sensitivities of the objective as an example, which is a function of both the design variables and the time-dependent state variables. The calculation of the derivatives for the constraint functions follows the exact same approach.

The optimization functions considered in this thesis can be written in discretized form as:

$$\mathcal{Z} = \sum_{n=N_t^1}^{N_t^2} \left(z^{(n)} \left(\boldsymbol{s}, \boldsymbol{u}^{(n)} \left(\boldsymbol{s} \right) \right) \right) \ , \tag{1.2}$$

where the time steps N_t^1 and N_t^2 correspond to the time intervals $[t_1, t_2]$ introduced in (1.1). The derivative of the objective function with respect to the optimization variable s_i is decomposed into an explicit and an implicit term such that:

$$\frac{\mathrm{d}\mathcal{Z}}{\mathrm{d}s_i} = \frac{\partial \mathcal{Z}}{\partial s_i} + \sum_{n=N_t^1}^{N_t^2} \frac{\partial z^{(n)}}{\partial \boldsymbol{u}^{(n)}}^{\mathsf{T}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial s_i} \ , \tag{1.3}$$

where the vector of time-dependent state variables at time step n, $\boldsymbol{u}^{(n)}$, satisfies the residual of the weak form of the governing equations:

$$\boldsymbol{R}^{(n)} \left(\boldsymbol{s}, \boldsymbol{u}^{(n)} \left(\boldsymbol{s} \right) \right) = 0 \ . \tag{1.4}$$

We can compute the derivative of the residual function with respect to the design variables as:

$$\frac{\mathrm{d}\boldsymbol{R}^{(n)}}{\mathrm{d}s_i} = \frac{\partial \boldsymbol{R}^{(n)}}{\partial s_i} + \frac{\partial \boldsymbol{R}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial s_i} \ , \tag{1.5}$$

which, assuming that the governing equations are satified, can be solved such that:

$$\frac{\partial \boldsymbol{u}^{(n)}}{\partial s_i} = - \left(\frac{\partial \boldsymbol{R}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \right)^{-1} \frac{\partial \boldsymbol{R}^{(n)}}{\partial s_i} \ . \tag{1.6}$$

Inserting (1.6) into (1.3), we obtain:

$$\frac{\mathrm{d}\mathcal{Z}}{\mathrm{d}s_i} = \frac{\partial \mathcal{Z}}{\partial s_i} - \sum_{n=1}^{N_t} \frac{\partial z^{(n)}}{\partial \boldsymbol{u}^{(n)}}^{\mathsf{T}} \left(\frac{\partial \boldsymbol{R}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \right)^{-1} \frac{\partial \boldsymbol{R}^{(n)}}{\partial s_i} \ . \tag{1.7}$$

This form contains an inverse matrix, $\left(\partial \boldsymbol{R}^{(n)}/\partial \boldsymbol{u}^{(n)}\right)^{-1}$, which needs to be handled carefully to ensure efficiency. Given that these matrices may be large due to the number of state variables, the inverse needs to be considered as a linear solve. Two methods exist to handle this problem: the **direct** method and the **adjoint** method.

Using the direct method, we introduce a variable $\boldsymbol{\xi}$ and the implicit term in (1.3) is rewritten as:

$$\sum_{n=N_t^1}^{N_t^2} \frac{\partial z^{(n)}}{\partial \boldsymbol{u}^{(n)}}^{\mathsf{T}} \boldsymbol{\xi}^{(n)} \ , \tag{1.8}$$

such that:

$$\left(\frac{\partial \boldsymbol{R}^{(n)}}{\partial \boldsymbol{u}^{(n)}} \right) \boldsymbol{\xi}^{(n)} = \frac{\partial \boldsymbol{R}^{(n)}}{\partial s_i} \ . \tag{1.9}$$

On the other hand, the implicit term in (1.3) can also be computed by the adjoint method as follows:

$$\sum_{n=N_t^1}^{N_t^2} \frac{\partial z^{(n)}}{\partial \boldsymbol{u}^{(n)}}^{\mathsf{T}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial s_i} = \sum_{n=0}^{N_t^2} \boldsymbol{\lambda}^{(n)\mathsf{T}} \frac{\partial \boldsymbol{R}^{(n)}}{\partial s_i} \ , \tag{1.10}$$

where $\boldsymbol{\lambda}^{(n)}$ are the adjoint states at time step n. Note that the scalar product of the adjoint vector and the derivative of the residual with respect to the design variables, s_i, is summed from the initial time step at $n = 0$ through $n = N_t$. Opposite to the forward analysis, and assuming a 1-step backward differentiation scheme in time, the adjoint state variables are computed *backwards* as follows:

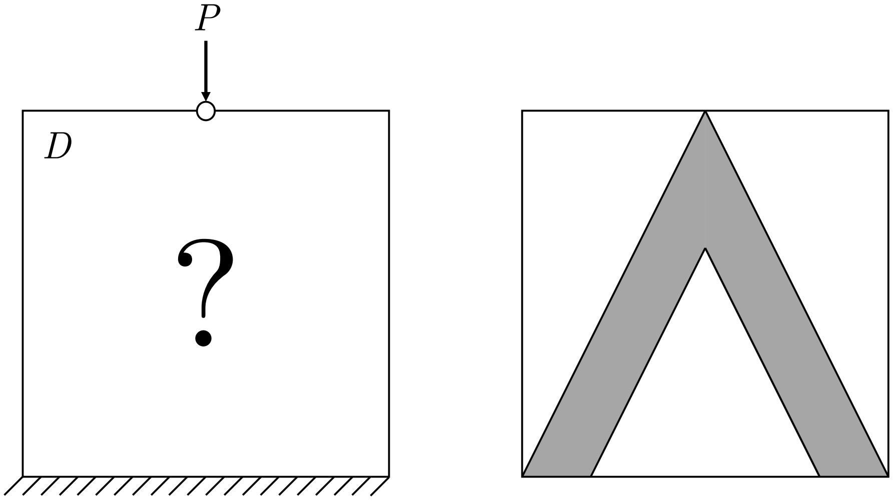
$$\frac{\partial \boldsymbol{R}^{(n)}}{\partial \boldsymbol{u}^{(n)}}^{\mathsf{T}} \boldsymbol{\lambda}^{(n)} = -\frac{\partial z^{(n)}}{\partial \boldsymbol{u}^{(n)}} + \left. \frac{\partial \boldsymbol{R}^{(n+1)}}{\partial \dot{\boldsymbol{u}}^{(n+1)}} \frac{\partial \dot{\boldsymbol{u}}^{(n+1)}}{\partial \boldsymbol{u}^{(n+1)}}^{\mathsf{T}} \right|_{\boldsymbol{u}^{(n+1)}} \boldsymbol{\lambda}^{(n+1)} \ , \tag{1.11}$$

for $n = N_t \ldots 0$. The initial solution for the state variables, $\boldsymbol{\lambda}^{(N_t+1)}$, is 0.

The direct method requires a number of linear solves that is equal to the number of design variables, N_s, while the adjoint method requires a number of solves equal to the number of objective and constraint functions, $(1 + N_g)$. If the number of design variables is small relative to the number of functions, the direct method should be used. If the number of design variables is large relative to the number of functions, the adjoint method should be used. In the following sections, we will delve into the approaches available to solve the optimization problem using this gradient information.

1.2.3 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. The goal is to minimize some objective functional over the design domain with respect to the design variables $s_i(\boldsymbol{x})$.



(a) Optimization problem setup.

(b) Optimized material layout.

Figure 1.1: Topology optimization example. The objective is to minimize the compliance of the mechanism.

A quick example is described in Figure 1.1. The objective is to minimize the compliance of the mechanism, subject to a maximum 30% volume utilization of the design domain. In the example, the optimization problem modifies the material layout of the body until it finds an optimal geometry at a local or global minima that satifies the volume constraint.

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 [Bendsøe and Sigmund, 2003; Rozvany, 2009]. 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 [Michell, 1904]. While topology optimization focused for decades on structural design, it has recently found its application in a wide range of physical disciplines [Bendsøe et al, 2005], including acoustics [Yoon et al, 2007], wave propagation [Sigmund and Jensen, 2003], and electromagnetics [Labbé et al, 2009; Shim et al, 2008]. For a more detailed review on topology optimization, please refer to Bendsøe and Sigmund [2003] and Eschenauer and Olhoff [2001].

Topology optimization problems generally require a large number of design variables (roughly equal to the number of nodes or elements) and for nonlinear or transient analysis, the objective function can be expensive to compute. Consequently, gradient-based methods become the most suitable choice for topology optimization problems. For topology optimization problems with non-trivial and multiple constraints, the Method of Moving Asymptotes (MMA) [Svanberg, 1987], and its globally convergent counterpart, the Globally Convergent Method of Moving Asymptotes (GCMMA) [Svanberg, 2002] have become the algorithms of choice. The work of this document will utilize the GCMMA algorithm. Two concerns arise, nevertheless, which are the differentiability and the calculation of gradients. Certain physical models can lead to a discontinuous response of the optimization criteria because of a topology change, which causes the objective functional to become non-differentiable or discontinuous, and the derivative information can drive gradient-based algorithms in a “bad” direction. This can become a problem and hinder the convergence of the optimization problem.

Gradients need to be computed in an efficient time. Finite differencing the objective and constraint functionals is an option; however, it is not computationally efficient for a large number of design variables. As topology optimization problems typically have a large number of design variables, the sensitivities for the optimization problems presented in this document will be computed using the adjoint method (1.10). We adopt the discrete adjoint formulation for nonlinear fluid and coupled systems of Kreissl and Maute [2011]. In this work, the derivative of the optimization functions with respect to the state variables, $\partial z^{(n)}/\partial \boldsymbol{u}^{(n)}$, is computed analytically. The partial derivative of the residual function of the weak form of the governing equations with respect to the design variables is computed by a central finite difference scheme.

The eXtended Finite Element Method presented in this thesis is discontinuous by nature. This behavior can cause the partial derivative of the residual equation to become non-differentiable or discontinuous, and the derivative information can drive the optimization algorithm in a "bad" direction. This can hinder the convergence of the optimization problem. This issue will be discussed later, where we develop and study a finite difference scheme to compute the partial derivatives with respect to the design variables within the context of the LSM-XFEM optimization framework.

1.2.4 Density-based Topology Optimization

Structural topology optimization, specifically topology optimization of continuum structures, is in its mathematical nature one of the most challenging optimization problems [Bendsøe and Sigmund, 2003]. However, in 1988, Bendsøe and Kikuchi [1988] 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 [Bendsøe, 1989]. 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 equations, and therefore is capable of producing internal holes in the design domain without an *a priori* knowledge of them. Among homogenization methods, the density-based formulations quickly gained traction as effective optimization tools.

In 1989, Bendsøe [1989] and Zhou and Rozvany [1991] introduced the "Solid Isotropic Material with Penalization" (SIMP) method. In the SIMP approach, the design variables represent the artificial densities, $\rho_i(\boldsymbol{x})$, of a group of elements in a fixed finite element grid (our design domain D), and their material properties are parametrized in terms of a set of material interpolation functions such that intermediate values are penalized. The optimization problem will vary the design variables in order to minimize the design objective, while satisfying the design constraints. In a structural topology optimization problem, we can represent a design variable at a point i as solid material by setting $\rho_i = s_i = 1.0$, and as void by setting $\rho_i = s_i = 0.0$. Figure 1.2 shows the setup for a topology optimization problem where the objective is to minimize the compliance, subject to a maximum volume fraction of 50% for the solid phase to suppress trivial solutions. Figure 1.3 shows the changes in the design during the optimization process, where the design variables at the elements are represented as solid (black), void (white), or in-between (grey). Grey areas represent the boundary between the solid and void phases.

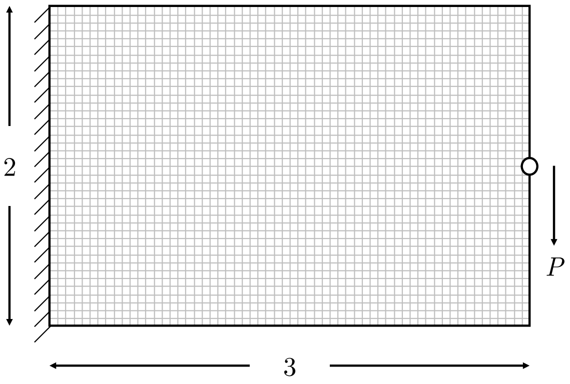
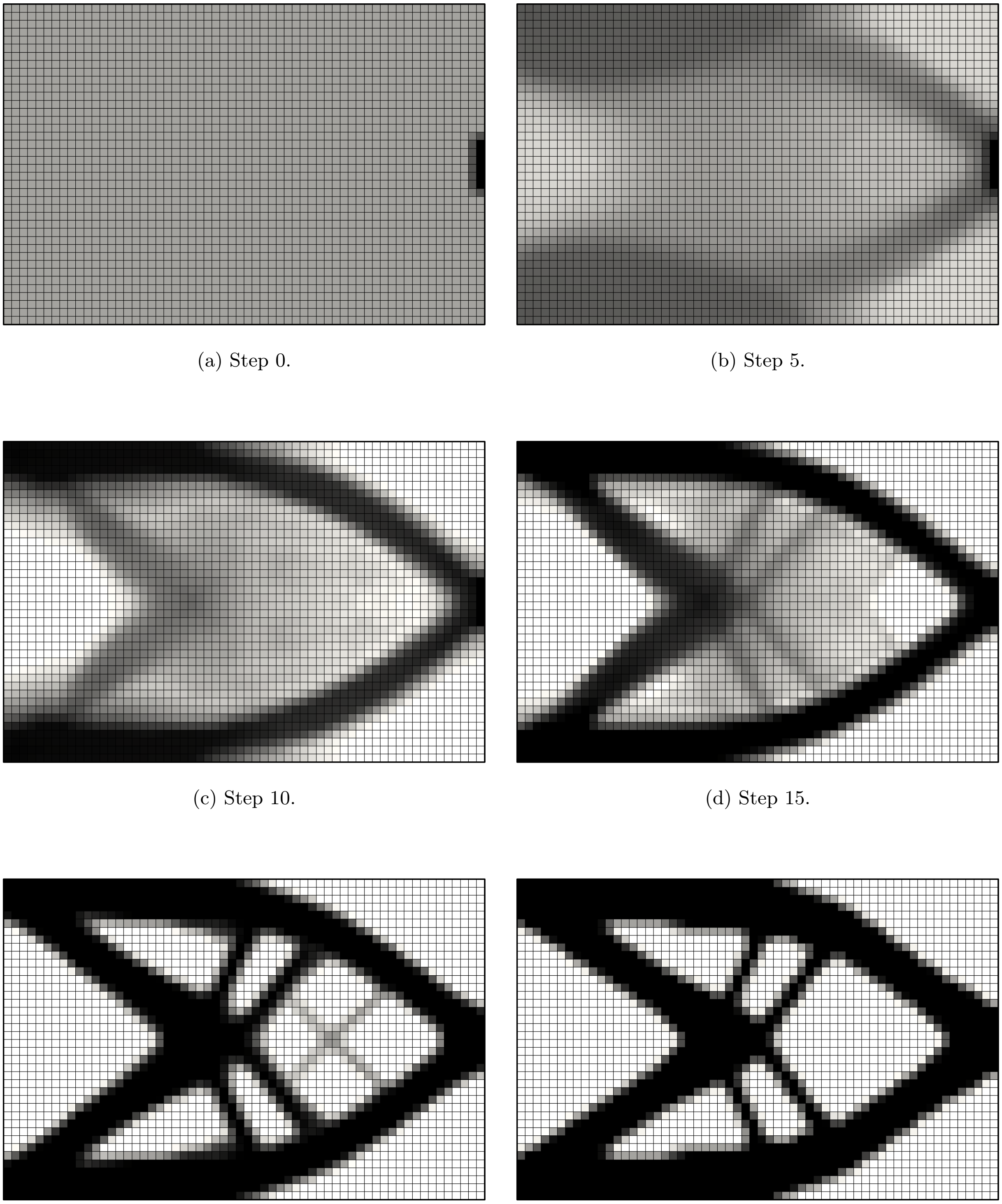


Figure 1.2: Setup of a structural topology optimization problem. A mesh of size $3L \times 2L$, with 60×40 quadrilateral bilinear elements is anchored to a wall on its left side, and subject to a point load on its right side.

1.2.5 Smoothing Filter

An additional numerical scheme is necessary to prevent numerical instabilities in density-based topology optimization. This is referred to as the filtering method [Guest et al, 2004]. For example, in structural topology optimization, the smoothed density, rather than being a function



(a) Step 0.

(b) Step 5.

(c) Step 10.

(d) Step 15.

(e) Step 25.

(f) Step 50.

Figure 1.3: Design iterations during the optimization process. The objective is to minimize the compliance of the structure, subject to a maximum volume fraction of 50% for the solid phase.

of a single design variable, can instead be computed with a linear smoothing filter as follows:

$$\tilde{\rho}_i\left(\boldsymbol{s}\right) = \frac{\sum\limits_{j=1}^{N_n} w_{ij} s_j}{\sum\limits_{j=1}^{N_n} w_{ij}} \ , \tag{1.12}$$

with:

$$w_{ij} = \max\left(0, r_\rho - \left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|\right) \ , \tag{1.13}$$

where $\tilde{\rho}_i\left(\boldsymbol{s}\right)$ is the smoothed density at a point $\boldsymbol{x}_i$, s_j is equivalent to the density ρ_j at a point $\boldsymbol{x}_j$, $\boldsymbol{x}_j$ is the location of the node at which the design variable j is defined, w_{ij} is the factor of point $\boldsymbol{x}_i$ with respect to the design variable j, r_ρ is the smoothing filter radius, and N_n is the number of nodes in the design domain.

The filter in (1.12) prevents the formation of features smaller than r_ρ, and serves as a minimum feature size control. However, this comes at the cost of forming intermediate densities along the material interface. Methods for penalizing intermediate densities have been proposed by Fuchs et al [2005], Sigmund [2007], and Stolpe and Svanberg [2001]. Guest et al [2004] 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 is applied to the densities as follows:

$$\hat{\rho}_i\left(\tilde{\rho}_i\right) = 1 - e^{-\beta\tilde{\rho}_i} + \tilde{\rho}_i e^{-\beta} \ , \tag{1.14}$$

where $\hat{\rho}_i$ 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 $\hat{\rho}_i = \tilde{\rho}_i$. As we increase β, intermediate densities are penalized towards the value of 1.0, as shown in Figure 1.4. Note, however, that if the objective and/or constraints of the optimization problem find the intermediate densities to be benefitial, the optimization algorithm will ignore the effects of this projection scheme. The reader is referred to Guest et al [2004], Guest et al [2011], Sigmund [2007], Xu et al [2010], and Wang et al [2011] for more details on projection schemes.

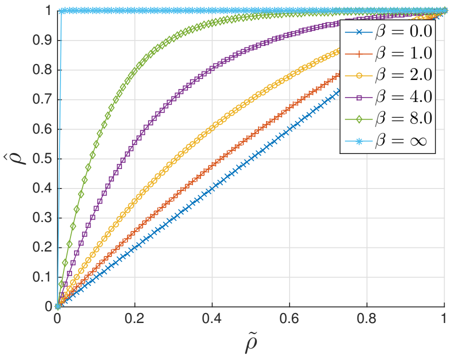


Figure 1.4: The density Heaviside projection for various magnitudes of β.

1.2.6 Structural Topology Optimization

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 $p > 1$, in order to guarantee a well-posed optimization problem [Bendsøe and Sigmund, 1999]. The structural stiffness for a solid-void problem can then be formulated as:

$$E\left(\boldsymbol{x}\right) = \hat{\rho}^{p} E^{(0)} \ , \tag{1.15}$$

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 [Rozvany et al, 1994]; this is the so-called continuation method [Sigmund and Petersson, 1998]. It was shown that if one uses $p > 3$, we approach a black-and-white binary-like material distribution [Bendsøe and Sigmund, 2003]. That is why the density approach has been referred to as a *pixelated* geometric model.

The SIMP method can be expanded to model a multimaterial optimization problem. For example, applying a "rule of mixture", we can model a two-phase material in structural topology optimization by modifying (1.15) as:

$$E\left(\boldsymbol{x}\right) = \hat{\rho}^{p} E_1 + \left(1 - \hat{\rho}^{p}\right) E_2 \ , \tag{1.16}$$

where E_1 represents the stiffness of the first material, and E_2 represents the stiffness of the second material. Notice that if we model the second material phase as void, and set $E_2 = 0$ we recover the original equation from (1.15). 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. $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, etc.), as shown in Wang and Wang [2004] and Park and Sutradhar [2015]. In general, the SIMP method requires $(n-1)$ design variable fields for n distinct material phases [Wang and Wang, 2004].

1.2.7 Fluid Flow Topology Optimization

Several applications require finding the optimal geometries of systems to improve the performance of internal and external flows [Maute, 2014]. Adopting the concept of density methods, Borrvall and Petersson [2003] 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 exerted by the porous media:

$$f_i = -\alpha\left(\boldsymbol{x}\right) v_i \ . \tag{1.17}$$

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

The coefficient α can be interpolated from the design variables as:

$$\alpha\left(\boldsymbol{x}\right) = \alpha_{max}\gamma\left(\boldsymbol{x}\right) \ . \tag{1.18}$$

The parameter α_{max} should be large enough such that the term f_i in (1.17) sufficiently penalizes the flow velocity to $u_i = 0$. Kondoh et al [2012] set α_{max} to:

$$\alpha_{max} = \left(1 + \frac{1}{Re}\right)\chi \ , \tag{1.19}$$

where Re is the Reynolds number, and χ is set to a very large value, i.e. 10^{+04} [Borrvall and Petersson, 2003].

This linear interpolation, however, produces large gradients in the fluid flow, which cause numerical issues and may lead the optimization problem to converge to a local minimum. Borrvall and Petersson [2003] introduced a convex interpolation to ameliorate this issue:

$$\alpha\left(\boldsymbol{x}\right) = \alpha\left(\gamma\left(\boldsymbol{x}\right)\right) = \alpha_{max} + \gamma\left(\alpha_{min} - \alpha_{max}\right)\frac{1+\alpha_p}{\gamma+\alpha_p}\ , \tag{1.20}$$

where α_p is a constant penalty factor. Figure 1.5 shows the influence of the α_p term. In general, we want to choose α_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 Kreissl and Maute [2011], α_p was chosen to be 0.01 with favorable results. α_{min} is set to zero, such that at its minimum, $\alpha\left(\boldsymbol{x}\right) = 0$ recovers the original Navier-Stokes equations.

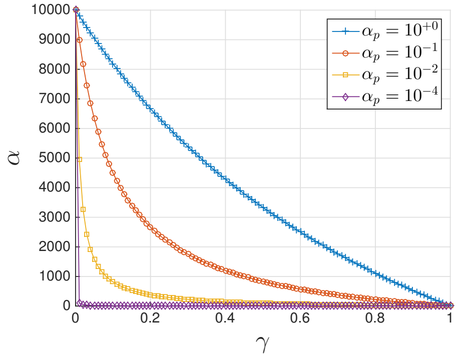


Figure 1.5: Influence of the interpolation penalty α_p, for $\alpha_{min} = 0$ and $\alpha_{max} = 10^{+04}$.

For more details on topology optimization of Stokes and Navier-Stokes flows, the reader is referred to Maute [2014].

1.3 Discussion

The concept of relating some artificial densities to the stiffness in structural problems can be expanded to other physics disciplines, as discussed in Section 1.2.7. The SIMP method can be used to describe the material properties in thermal conductivity, magnetic permeability, porosity, etc.; and consequently, it has found its way to a wide range of applications. The method requires a relatively small 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 of the material interface. This characteristic prevents the method from suffering of 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 [2001] detailing a 99-line SIMP code implemented in Matlab, as well as his web-based topology optimization program [Tcherniak and Sigmund, 2001] 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 approach 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 not robust. This may result in non-physical responses, such as premature yielding [Maute et al, 1998] in structural mechanics, fluid flow penetrating solid material in low Reynolds number flow [Kreissl et al, 2011], and scalar fields diffusing through solid material at low Péclet number flow [Makhija et al, 2012]. This issue can be mitigated by representing the material interface more accurately either by mesh refinement or adaptive remeshing [Maute and Ramm, 1995, 1997]. However, the adaptive refinement of the finite element discretization by remeshing affects the convergence of the optimization process if a gradient-based optimization algorithm is applied [Schleupen et al, 2000]. Furthermore, 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 [Erentok and Sigmund, 2011; Yamasaki et al, 2011].

The material interpolation schemes used in the SIMP method present further disadvantages. Interpolating the density field into a physical property, such as the Young's modulus in structural mechanics (1.15), or a body force in fluid dynamics (1.17) reduces the genericity of the approach because each different physical phenomenon requires a new interpolation formulation. Additionally, using material interpolation to address multimaterial optimization problems is not a physics-based technique, and has been shown to violate the Hashin-Shtrikman bounds for low values of ρ_i and large values of p [Hashin and Shtrikman, 1962]. Therefore, modeling multiple material phases can become complicated [Yin and Ananthasuresh, 2001], and lead to an inefficient slow convergence due to the larger number of iterations required.

With regards to manufacturing the optimized designs, the SIMP approach displays additional downsides. For example, extracting the optimal geometry from the density distribution in the design domain is not trivial and requires additional postprocessing steps. This is counterproductive for rapid prototyping techniques.

The disadvantages of density-based optimization methods, and homogenization methods in general, prompted the development of this thesis, and the use of immersed boundary techniques such as the Level Set Method and the eXtended Finite Element Method as topology optimization tools.

1.4 The LSM-XFEM Framework

The goal of this thesis proposal is to introduce a unified topology optimization framework that uses the Level Set Method to describe the design geometry and the eXtended Finite Element Method to solve the governing equations and measure the performance of the design. The framework will be referred to as the LSM-XFEM optimization method.

A computational optimization framework consists of four areas of study: the optimization model, the design model, the analysis model, and the fabrication model, as shown in Figure 1.6. In this thesis, we will study the optimization model with respect to the formulation of the optimization problems; the design model with respect to the parametrization of the design geometry and the capability to control the shape of the design using the LSM, and the definition of the design variables; the analysis model with respect to the analysis of multiphysics problems using the XFEM;

and the fabrication model with respect to the capacity to meet manufacturing constraints.

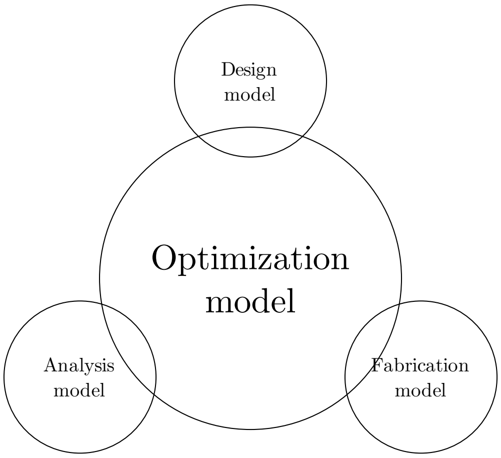


Figure 1.6: Computational approach for an optimization framework.

We will study the capabilities and characteristics of our topology optimization framework with 2D and 3D linear elasticity, laminar incompressible flow, species transport, and conjugate heat transfer problems. The analysis domain is represented as material-void or material-material problems. The geometry of the material interface is described by an explicit Level Set Method, where the parameters of a Level Set Function (LSF) are defined as functions of the optimization variables. The governing equations in the analysis domain are discretized in space by a Heaviside-step generalized formulation of the eXtended Finite Element Method, which preserves the crisp geometry definition of the LSM. A geometric preconditioner and face-oriented ghost-penalty terms are added for stability reasons and to improve the conditioning of the system. The structural behavior is modeled by a linear elasticity model. The fluid behavior is modeled by the incompressible Navier-Stokes equations, augmented with a Boussinesq approximation of the buoyancy forces to model conjugate heat transfer. The species transport in the fluid phase is modeled by an advection-diffusion equation. The species field in the solid phase is modeled by a linear diffusion model. The emergence of isolated fluid regions surrounded by solid in the flow problems or free-floating solid particles during the optimization process lead to a singular analysis problem. A novel auxiliary indicator field is modeled to avoid the ill-conditioning of the system by weakly imposing a pressure constraints on these isolated regions in the flow problems, and by adding a system of soft springs in the structural problems. The interface conditions at the material boundary are enforced weakly via stabilized Lagrange multipliers or Nitsche's method. The optimization problems are formulated with a nonlinear programming method; the flexibility of this scheme allows us to define additional design variables in addition to the parameters of the LSF to control the position, size, and shape of the inlets and outlets in the flow problems. The sensitivities are computed using the adjoint method. The numerical results demonstrate the applicability of the proposed method for 2D and 3D linear elasticity, and steady-state and transient coupled multiphysics laminar flow problems. Additional regularization techniques are added to the optimization framework to control the shape of the design and meet manufacturing constraints.

1.5 Outline

The thesis is structured as follows: Section 2 details our description of the geometry using the design variables and the LSM. In Section 3, we present the XFEM. Section 4 describes the problem setup, including the governing equations, the temporal discretization, the optimization criteria, and the sensitivity analysis. Section 5 describes the stability methods used in the context of the XFEM. Section 6 studies regularization techniques to control the shape of the designs by utilizing the level set field and the geometry of the material interface. Numerical examples are studied in Section 7. The conclusions drawn from this work are presented in Section 8. The contributions of this thesis are stated in Section 9. Future work is proposed in Section 10.

Chapter 2

Level Set Method

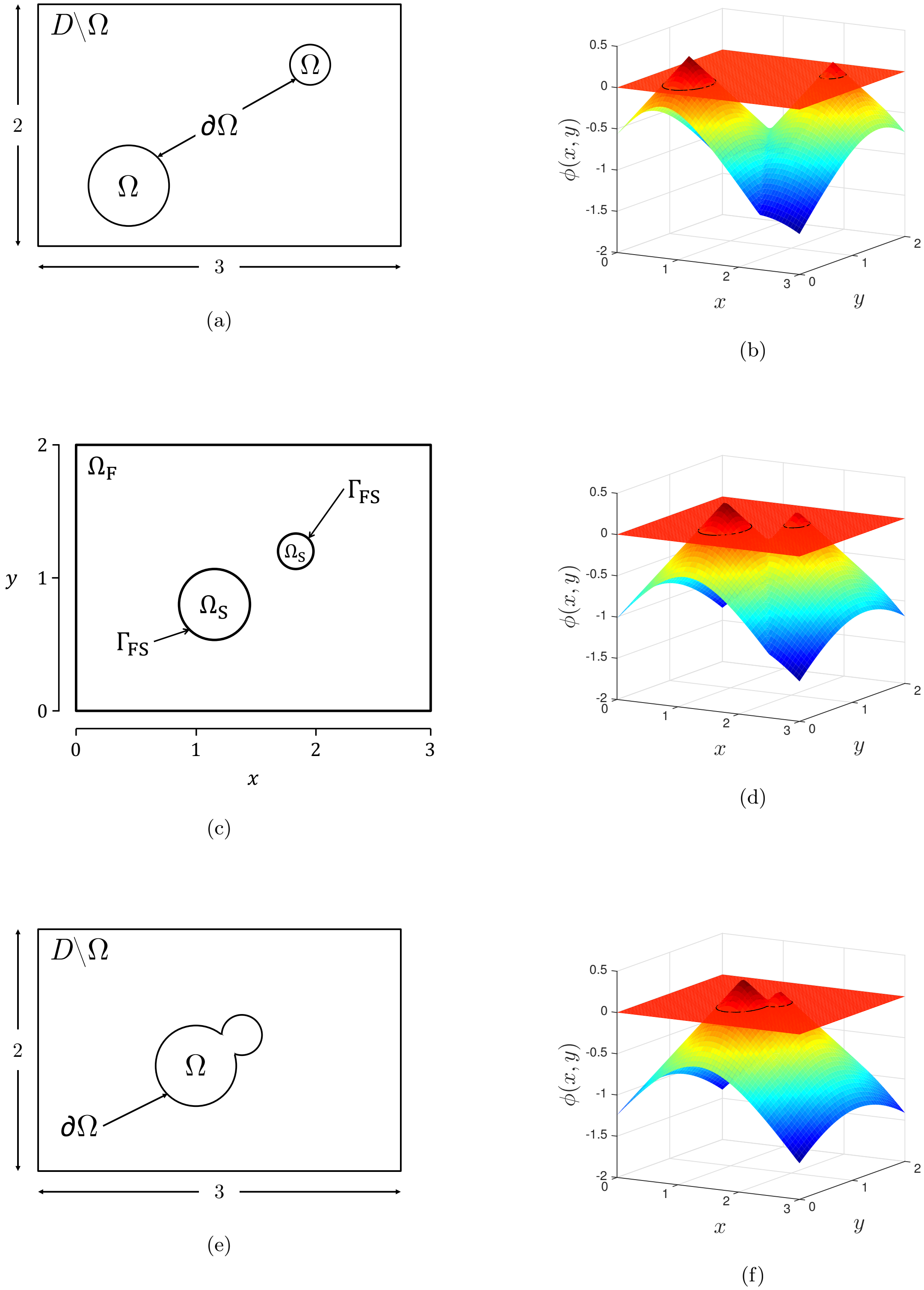
The shortcomings of density methods have promoted the development of immersed boundary methods for topology optimization. There are several immersed approaches in the literature (c.f. Parvizian et al [2012] and van Dijk et al [2013]). The Level Set Method applied to topology optimization arose as an immersed boundary 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 [Wang and Wang, 2004].

In topology optimization, the geometry of a design is defined by the vector of design variables. In our LSM-XFEM implementation, the parameters of a discretized Level Set Function (LSF) are defined as explicit functions of the design variables. In the following subsections, we describe the concepts behind the LSM, and the parametrization of the LSF with respect to the design variables.

2.1 Geometry Description

The LSM can describe the geometry of a body immersed in a domain, $\Gamma = \partial\Omega$, by the shape boundary of a higher dimensional LSF, $\phi\left(\boldsymbol{x}\right)$, where $\boldsymbol{x}$ denotes the vector of spatial coordinates. This shape boundary is usually represented by the zero level set of this function such that:

$$\Gamma = \left\{\boldsymbol{x} \mid \phi\left(\boldsymbol{x}\right) = 0\right\} . \tag{2.1}$$



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.1: LSM description of 2 circular inclusions with radii of 0.667 and 0.333, respectively, moving towards each other.

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

$$\phi\left(\boldsymbol{x}\right) > 0 \ , \quad \forall \boldsymbol{x} \in \Omega \backslash \partial\Omega \qquad \text{(inside the region)} \ , \tag{2.2}$$

$$\phi\left(\boldsymbol{x}\right) = 0 \ , \quad \forall \boldsymbol{x} \in \partial\Omega \qquad \text{(on the boundary)} \ , \tag{2.3}$$

$$\phi\left(\boldsymbol{x}\right) < 0 \ , \quad \forall \boldsymbol{x} \in D \backslash \Omega \qquad \text{(outside the region)} \ , \tag{2.4}$$

where D represents the design domain, either bounded or unbounded, and contains all possible admissible shapes of Ω, as shown in Figure 2.1. These functions can be used to describe complex geometries; a few non-trivial examples of analytically-given shapes are taken from Burman et al [2014], and are defined as:

- Doughnut:

$$\phi(x, y, z) = \left(R - \sqrt{x^2 + y^2}\right)^2 + z^2 - r^2 \ , \tag{2.5}$$

- Popcorn ([Annavarapu et al, 2012; Chern and Shu, 2007; Burman et al, 2014]):

$$\phi(x, y, z) = \sqrt{x^2 + y^2 + z^2} - r_0 - \sum_{k=0}^{11} A e^{-((x-x_k)^2 + (y-y_k)^2 + (z-z_k)^2)/\sigma^2} \ , \tag{2.6}$$

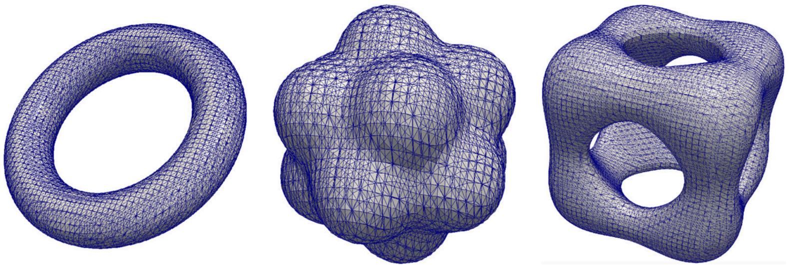
where:

$$(x_k, y_k, z_k) = \begin{cases} \frac{r_0}{\sqrt{5}} \left(2\cos\left(\frac{2k\pi}{5}\right), 2\sin\left(\frac{2k\pi}{5}\right), 1\right) & 0 \leq k \leq 4 \ , \\ \frac{r_0}{\sqrt{5}} \left(2\cos\left(\frac{(2(k-5)-1)\pi}{5}\right), 2\sin\left(\frac{(2(k-5)-1)\pi}{5}\right), 1\right) & 5 \leq k \leq 9 \ , \\ (0, 0, +r_0) & k = 10 \ , \\ (0, 0, -r_0) & k = 11 \ . \end{cases} \tag{2.7}$$

- Swiss cheese block:

$$\begin{aligned} \phi(x, y, z) = {} & \left(x^2 + y^2 - 4\right)^2 + \left(z^2 - 1\right)^2 \\ & + \left(y^2 + z^2 - 4\right)^2 + \left(x^2 - 1\right)^2 \\ & + \left(z^2 + x^2 - 4\right)^2 + \left(y^2 - 1\right)^2 \ . \end{aligned} \tag{2.8}$$

These shapes are illustrated in Figure 2.2.



(a) Doughnut (b) Popcorn (c) Swiss cheese

Figure 2.2: LSFs can be used to describe complex geometries.

Each phase region may represent a different material [Allaire et al, 2002; Wang and Wang, 2004; Osher and Santosa, 2001; Sethian and Wiegmann, 2000] or a different physics [Legay et al, 2006; Gerstenberger and Wall, 2008]. The LSM was first applied extensively in the field of imaging and computer vision [Osher and Paragios, 2003]. Eventually, the approach found its way to topology optimization, as the technique is well suited for the task: LSFs can form holes, split into multiple pieces, or merge with other functions [Wang and Wang, 2004; Allaire et al, 2002; Osher and Sethian, 1988].

Multiple functions can be used to model more than two phase regions. As with the LSM itself, the use of multiple functions originated in image processing [Vese and Chan, 2002]. This so-called "color" LSM requires m functions to model $n = 2^m$ different phase regions. For a reference on the method, the reader is referred to Wang and Wang [2004] and Wang and Wang [2005]. The work of this thesis is restricted, at the most, to two-material problems, and therefore, only one function will be used at any time.

2.2 Parametrization of the Level Set Function

The LSF is typically updated in the optimization process by solving the Hamilton-Jacobi equations [Yamasaki et al, 2010]. Methods that utilize this update scheme are denoted as implicit methods. An alternate approach, specifically the one utilized in this work, is to define the parameters of the discretized LSF as explicit functions of the optimization variables. The resulting parameter optimization problem is solved by standard nonlinear programming methods [van Dijk et al, 2013], such as the GCMMA algorithm [Svanberg, 2002].

In this thesis, we parametrize a discretized LSF to describe a combination of geometric primitives and to allow for the evolution of geometries in the optimization process. These geometric primitives serve as ports in the flow problems, and are described by cylinders. In both parametrization schemes, the value of the function at a point within an element is interpolated from the nodal values using standard trilinear finite element shape functions. In theory, the parametrization mesh may differ from the analysis mesh; however, for simplicity, we use the same mesh for the parametrization and the analysis. Note that because our shape functions are linear, an element edge can be intersected at most once by the fluid-solid interface, i.e. the point i at which $\phi_i\left(\boldsymbol{x}\right) = 0$. This interpolation scheme restricts the geometry resolution of the LSF to the size of a finite element, and may cause convergence issues in the optimization process if smaller features are generated. Hierarchical mesh refinement may ameliorate this issue; however, as these features become smaller, the required level of refinement might not be feasible for practical purposes. This issue has been discussed in Jenkins and Maute [2015] and Coffin and Maute [2015b]. A regularization scheme to discourage subelement-size features is presented in Section 6.2. However, as the scheme is not mesh independent a more advanced approach is developed and studied in Section 6.3.

2.2.1 Topology Optimization

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 [Yamasaki et al, 2010]. Instead, 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 (1.12), we define a linear filter:

$$\phi_i\left(\boldsymbol{s}\right) = \frac{\sum\limits_{j=1}^{N_n} w_{ij} s_j}{\sum\limits_{j=1}^{N_n} w_{ij}} \ , \tag{2.9}$$

with:

$$w_{ij} = \max\left(0, r_\phi - \|\boldsymbol{x}_i - \boldsymbol{x}_j\|\right) \ , \tag{2.10}$$

where $\phi_i\left(\boldsymbol{s}\right)$ is the level set value at a point $\boldsymbol{x}_i$, w_{ij} is the factor of point $\boldsymbol{x}_i$ with respect to the design variable j, r_ϕ is the smoothing filter radius, and N_n is the number of nodes in the design domain. The linear filter in (2.9) was used previously in the studies of Kreissl and Maute [2012] and Makhija and Maute [2014a], and was shown to accelerate the convergence of the geometry in the optimization process. The filter helps control the spatial gradients of the LSF in the vicinity of the zero level set to avoid ill-conditioning of the optimization problem. Furthermore, the filter may promote (but does not guarantee) smooth shapes of the phase boundaries; however, in contrast to density or sensitivity filters used in density-based methods, the filter above is not guaranteed to control the minimum feature size [Villanueva and Maute, 2014].

Numerical experiments have shown that filter radii in the range from 2.0 to 4.0 times the element length scale, h, yield an effective and efficient smoothing of the nodal design variables [Coffin and Maute, 2015b]. Larger values can actually be counterproductive, and can yield a design that, although at the global scale looks the same as a design with a smaller radius, is actually less smooth in the vicinity of the level set interface [Villanueva and Maute, 2014].

The reader is referred to van Dijk et al [2013] and Gain and Paulino [2013] for a more detailed overview of the Level Set Method in the context of topology optimization.

2.2.2 Geometric Primitives

The optimization problems are formulated with a nonlinear programming method. The flexibility of this scheme allows us to define optimization variables to describe the shape of geometric primitives.

In this thesis, we utilize this concept to model the ports in our flow problems as 3D cylinders. The design variables describe the position and size of the cylinders; each cylinder is defined by its local coordinate system, $\tilde{\boldsymbol{x}}$, and its radius. The level set value of the j-th cylinder, $\phi_{c,j}\left(\tilde{\boldsymbol{x}}\right)$, is defined as:

$$\phi_{c,j}\left(\tilde{\boldsymbol{x}}\right) = r_{c,j} - \sqrt{\left(\tilde{x} - \tilde{x}_{c,j}\right)^2 + \left(\tilde{y} - \tilde{y}_{c,j}\right)^2} \ , \tag{2.11}$$

where $\tilde{x}$ and $\tilde{y}$ are the in-plane coordinates of the port, $r_{c,j}$ is the radius of the cylinder, and $\tilde{x}_{c,j}$ and $\tilde{y}_{c,j}$ are the coordinates at its center. Note that this formulation only allows the port to move within the plane on which it was initially placed. The level set value ϕ_i is defined by approximating the minimum level set value among all ports using a Kreisselmeier-Steinhauser function, which ensures the differentiability of the formulation with respect to the cylinder parameters. Furthermore, the function (2.11) overrides the value given by (2.9) at the ports. For more details and examples, the reader is referred to Coffin and Maute [2015a].

2.3 Mechanical Model

Several methods exist to describe how the geometry and the material distribution described by the LSF are represented in the mechanical model. The Ersatz material method [Wang et al, 2003; Allaire et al, 2005] interpolates the physical properties of a fictitious material by either using element-wise constant material fractions or by mapping the level set field directly to a point [Yaji et al, 2015] (see Figure 2.3). In a fluid-solid problem, the Ersatz material approach models the solid phase as a body force using Brinkman penalization. While the Ersatz material approach eases the computational complexity, the method faces the same issues as density methods in regards to enforcing boundary conditions across the material interface.

An alternative to the Ersatz interpolation is to repeatedly generate new meshes that align with the geometry of the zero level set, as shown in Figure 2.4. However, generating an entirely new body-fitted mesh typically suffers from robustness and efficiency, particularly for 3D problems. It was also shown by Schleupen et al [2000] and Wilke et al [2006] that this method affects the

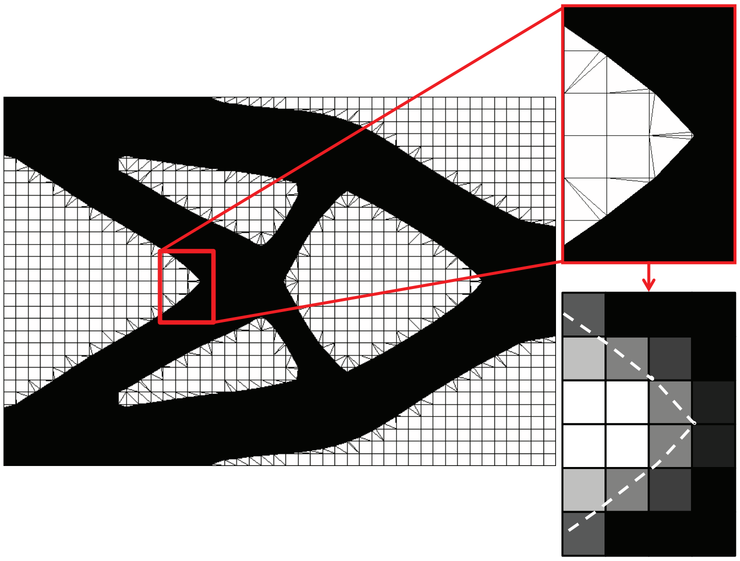


Figure 2.3: In an Ersatz material interpolation approach, the material properties of each finite element are interpolated proportional to the volumes of the phase regions.

convergence of the optimization process.

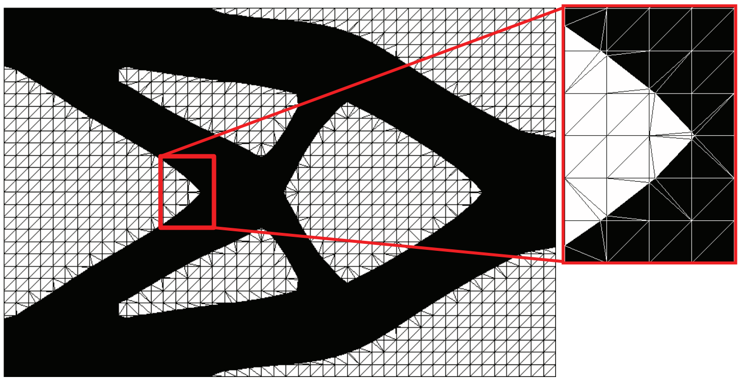


Figure 2.4: Remeshing the design domain such that elements align with the zero level set is an alternative to the Ersatz material interpolation approach.

In this work, we utilize the XFEM to describe the material distribution in the mechanical model. The XFEM is an immersed boundary technique that does not require a mesh that conforms to the phase boundary. The XFEM decomposes the elements cut by the zero level set into subdomains and interfaces to integrate the weak form of the governing equations. This approach avoids the need to interpolate the material properties such as in density methods because each subdomain has a distinct phase. We study the XFEM in more detail in Section 3.

2.4 Discussion

There are key challenges for the LSM in the context of topology optimization, such as: (i) controlling the spatial gradients of the LSF in the vicinity of the zero level set to avoid ill-conditioning of the optimization problem, (ii) controlling local feature sizes, (iii) accelerating the convergence of the geometry in the optimization process, and (iv) the robust and efficient analysis of the geometries described by the LSF. Solutions to (i) and (iii) are drawn from the literature, and utilize the smoothing filter introduced in (2.9). Approaches to handle (ii) and (iv) are developed and studied in this thesis.

In constrast to the smoothing filter in density-based optimization (1.12), LSM-based topology optimization requires regularization to control the size of geometric features. These techniques, such as a minimum feature size measure and shape smoothers, are studied in Section 6.

Chapter 3

eXtended Finite Element Method

The XFEM is an immersed boundary technique that does not require a mesh that conforms to the phase boundary. The method was built upon the concept of partition of unity developed by Babuška and Melenk [1997], and it was originally used to model crack propagation [Belytschko and Black, 1999]. The XFEM augments the standard finite element interpolation space with additional enrichment functions, denoted "enriched degrees-of-freedom", to capture discontinuities in either the state variables or their spatial gradients within an element. The XFEM decomposes the elements cut by the zero level set isosurfaces into subdomains and interfaces to integrate the weak form of the governing equations. This approach avoids the need to interpolate the material properties such as in density methods because each subdomain has a distinct phase. Boundary conditions on the interface are imposed weakly via stabilized Lagrange multipliers [Gerstenberger and Wall, 2008], or via Nitsche's method [Bazilevs and Hughes, 2007]. In the context of fluid flow problems, the enforcement of no-slip boundary conditions along the phase boundaries via the XFEM and a stabilized Lagrange multiplier method was adopted by Kreissl and Maute [2012]; the Lattice Boltzmann Method was employed in combination with a level set-based geometric interface representation for generalized topology optimization of fluids by Makhija and Maute [2014a], among others. For a general overview of the method, refer to Fries and Belytschko [2010].

3.1 Mechanical Model

The XFEM decomposes the elements cut by the zero level set isocontours (in 2D) or isosurfaces (in 3D) into subdomains and interfaces to integrate the weak form of the governing equations. This approach avoids the need to approximate material properties such as in density methods because each subdomain has a distinct phase. For 2D problems and using a bilinear interpolation of the level set field within an element, there are only 8 intersection configurations which can be tabulated (see Figure 3.1). In 3D, there are 127 intersection configurations. To handle this complexity, we compute the intersection points of the zero level set with the element edges and use a Delaunay triangulation to subdivide the element into triangles in 2D and tetrahedra in 3D. These intersection points are denoted as $\boldsymbol{x}_i^\Gamma$. In numerical experiments, this approach has proven robust and computationally inexpensive. Figure 3.2 shows the decomposition of a 2D short cantilever structural beam using our triangulation approach. For more details on the implementation, the reader is referred to Appendix A.

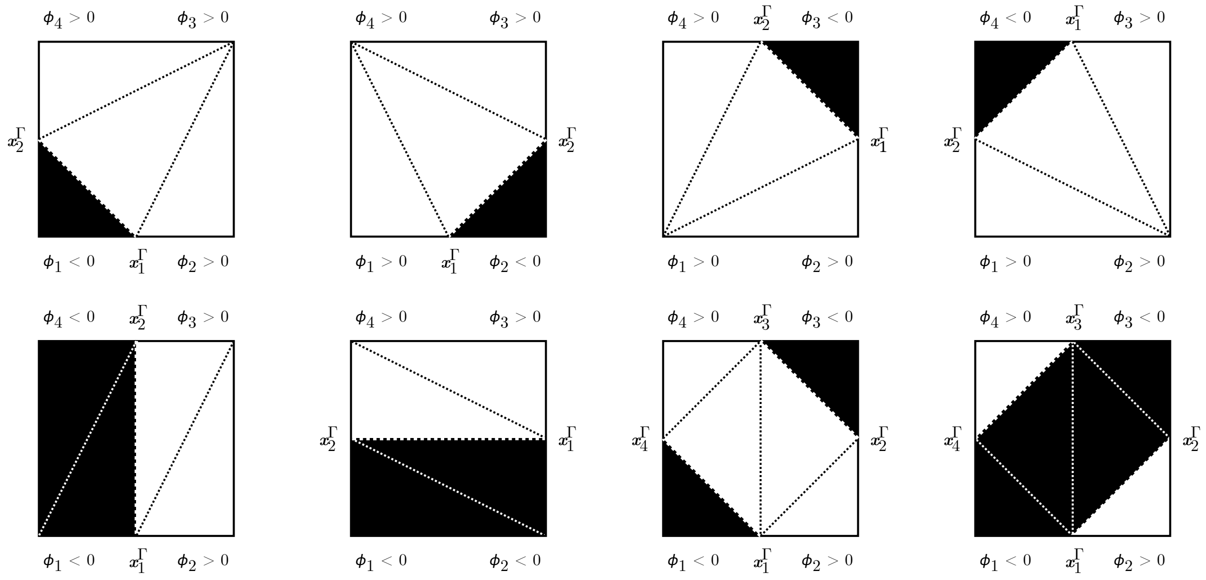

$\phi_4 > 0$ $\phi_3 > 0$
x_2^Γ
$\phi_1 < 0$ x_1^Γ $\phi_2 > 0$
$\phi_4 > 0$ $\phi_3 > 0$
x_2^Γ
$\phi_1 > 0$ x_1^Γ $\phi_2 < 0$
$\phi_4 > 0$ x_2^Γ $\phi_3 < 0$
x_1^Γ
$\phi_1 > 0$ $\phi_2 > 0$
$\phi_4 < 0$ x_1^Γ $\phi_3 > 0$
x_2^Γ
$\phi_1 > 0$ $\phi_2 > 0$
$\phi_4 < 0$ x_2^Γ $\phi_3 > 0$
$\phi_1 < 0$ x_1^Γ $\phi_2 > 0$
$\phi_4 > 0$ $\phi_3 > 0$
x_2^Γ x_1^Γ
$\phi_1 < 0$ $\phi_2 < 0$
$\phi_4 > 0$ x_3^Γ $\phi_3 < 0$
x_4^Γ x_2^Γ
$\phi_1 < 0$ x_1^Γ $\phi_2 > 0$
$\phi_4 > 0$ x_3^Γ $\phi_3 < 0$
x_4^Γ x_2^Γ
$\phi_1 < 0$ x_1^Γ $\phi_2 > 0$

Figure 3.1: A 2D finite element has eight possible decompositions based on the nodal level set values.

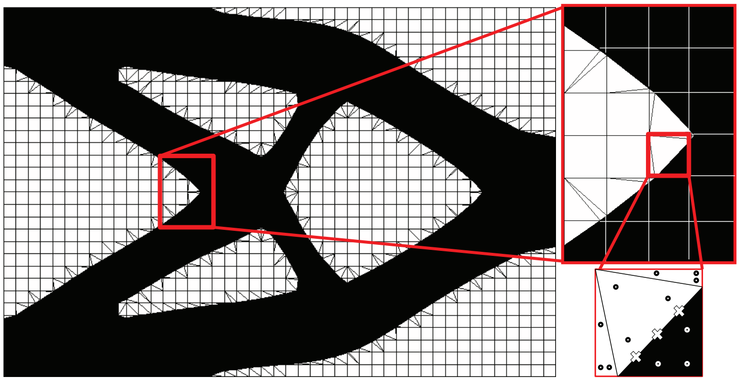


Figure 3.2: In the XFEM, the elements cut by the zero level set are divided into subdomains and interfaces for integration. Circles represent the integration points for the subdomains, while crosses represent the integration points for the interface.

3.2 Solution Enrichment

The governing equations are discretized in space by the XFEM. Different enrichment strategies are available in the literature depending on the type of discontinuity [Fries and Belytschko, 2010]. This study adopts a generalized enrichment strategy based on the Heavisde-step enrichment of Hansbo and Hansbo [2004], which interpolates consistently the solution fields in the presence of small features, and does not suffer from the artificial coupling of disconnected phases. This particular approach has been used by Makhija and Maute [2014b] and Villanueva and Maute [2014], Kreissl and Maute [2012], Lang et al [2014], and Makhija and Maute [2014a], who considered linear elasticity, incompressible Navier-Stokes, linear diffusion, and advection-diffusion problems, respectively.

The XFEM is used to approximate a state variable field in only a single phase or in both phases. Here, we present the most general case where the state variables are modeled in both phases. The approximation for a solution field u within an element is denoted as $\tilde{u}$, and is discretized by a Heaviside-step enrichment strategy:

$$u\left(\boldsymbol{x}\right) \approx \tilde{u}\left(\boldsymbol{x}\right) = \sum_{l=1}^{N_l} \left(H(-\phi(\boldsymbol{x})) \sum_{i=1}^{N_n} v_i(\boldsymbol{x}) \delta_{lk}^{i,f} u_{i,l}^{f} + H(+\phi(\boldsymbol{x})) \sum_{i=1}^{N_n} v_i(\boldsymbol{x}) \delta_{ln}^{i,s} u_{i,l}^{s} \right) \,, \tag{3.1}$$

where l is the enrichment level, N_l is the maximum number of enrichment levels used for each phase, $v_i(\boldsymbol{x})$ are the nodal basis functions, $u_{i,l}^{f}$ and $u_{i,l}^{s}$ are the degrees-of-freedom of enrichment level l at node i in the fluid and solid phases, respectively, ϕ is the level set value, and H denotes the Heaviside function, which turns on and off the interpolation for the phase m, and is defined as:

$$H(\zeta) = \begin{cases} 1 & \zeta > 0 \,, \\ 0 & \zeta < 0 \,. \end{cases} \tag{3.2}$$

The Kronecker delta, $\delta_{ab}^{i,m}$, selects the degrees-of-freedom for the material phase m. The indices k and n denote the active degrees-of-freedom at node i in the fluid and solid phases, respectively. At any given point, only one degree-of-freedom per node is used to interpolate the solution, ensuring that the partition of unity is satisfied.

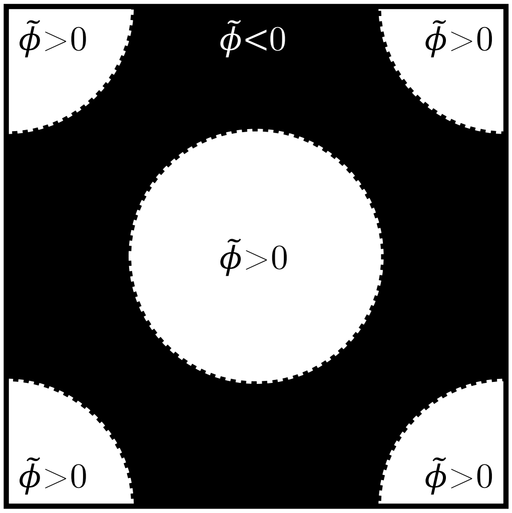


Figure 3.3: Physical model for the enrichment example. The analysis domain contains multiple level set inclusions.

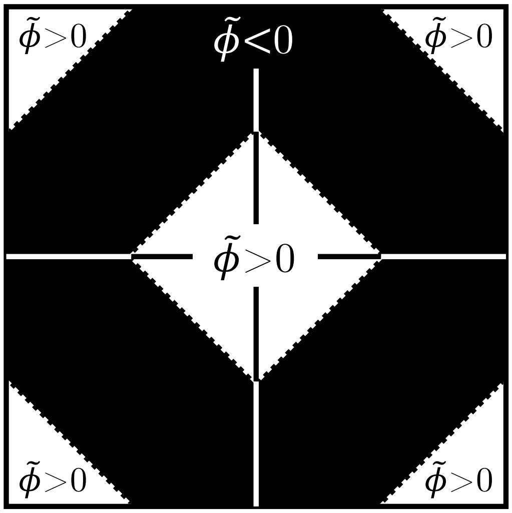


Figure 3.4: Discretized model for the enrichment example. 2D mesh with 4 bilinear elements. Black areas: material phase 1, negative level set values at the nodes; white areas: material phase 2, positive level set values at the nodes.

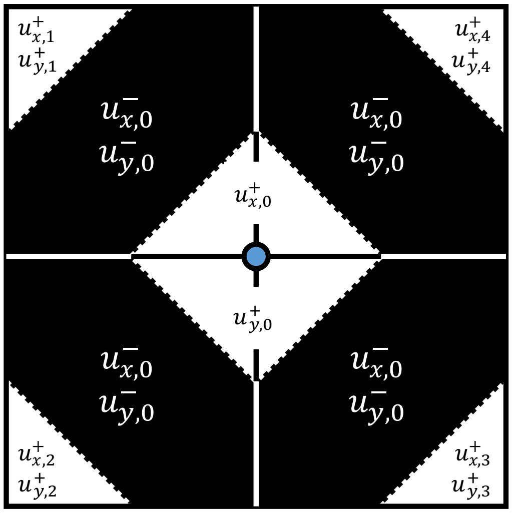


Figure 3.5: The center node, denoted by the color blue, uses different degrees-of-freedom to describe the disconnected phase regions. The subscripts denote the l enrichment level. The maximum number of enrichment levels used for each phase is $N_l = 5$. A value of 0 denotes the original finite element degrees-of-freedom, while other numbers indicate additional "enriched degrees-of-freedom".

For each phase, multiple enrichment levels, i.e. sets of shape functions, may be necessary to interpolate the state variables in multiple, physically disconnected regions of the same phase, c.f. [Terada et al, 2003], [Tran et al, 2011], and [Makhija and Maute, 2014b]. When interpolating the level set field by element-wise linear functions in a structured grid, a maximum of 9 enrichment levels is needed in 2D problems, and 14 enrichment levels in 3D [Villanueva and Maute, 2014]. To illustrate this process with a quick example, consider a physical model that contains multiple inclusions described by a level set field, as shown in Figure 3.3. The corresponding discretized model consists of a 2D mesh with 4 bilinear intersected elements, as shown in Figure 3.4. The boundary conditions are not relevant at this stage. The node at the center of the mesh in Figure 3.4 will use different degrees-of-freedom to interpolate the different subdomains and avoid artificially coupling the disconnected phase regions; this is shown in Figure 3.5.

The Heaviside-step enrichment formulation (3.2) has a singularity for cases in which the material interface lies exactly on a node, i.e. the level set value ϕ_i at node i equals 0. To avoid this issue, we adopt the level set perturbation approach outlined in Choi et al [2012] and Lang et al [2014]. If the magnitude of the level set value at a node is smaller than some critical value, ϕ_c^Γ, the level set value is modified to a shift value, ϕ_s^Γ. The different options for shifting the material interface include: (i) a $-$ve shift, such that upon shifting the interface, the previously intersected node lies in the negative phase, (ii) a +ve shift, such that upon shifting the interface, the previously intersected node lies in the positive phase, and (iii) a signed shift, wherein upon shifting the interface, the node retains its phase. In the scenario where the material interface lies exactly on a node, we resort to a $-$ve shift. This perturbation results in the material interface moving away from the node, solving the singularity issue. Unless otherwise stated, we adopt the $-$ve shift and the values of $\phi_c^\Gamma = \phi_s^\Gamma = 10^{-06} \times h$.

Numerical studies have shown that the influence of this perturbation is negligible for the problems considered here [Coffin and Maute, 2015a]. The effect on the sensitivities of the objectives and constraints with respect to the level set field is addressed in Section 4.6.

For material-void problems, elements entirely in the void phase are omitted in the element assembly process and the degrees-of-freedom that interpolate the void phase are eliminated from the system of equations. These techniques reduce the computational cost of solving the XFEM problem.

The reader is referred to Makhija and Maute [2014b] and to Appendix C for more details on the particular XFEM implementation used in this work.

3.3 Implementation Details

For more details on the enrichment strategy and the algorithmic implementation of Figure 3.5, the reader is referred to Appendix A. For convenience, a summary is provided here. The reader may skip this section, as it only focuses on the algorithmic implementation of the framework and most of its content is intended for software developers.

The internal document in Appendix A was written as both a user and a theory manuals for future software developers who wish to study and implement the LSM-XFEM framework. The report offers all the computational algorithms involved in building a minimum viable product that utilitizes both methods to perform topology optimization.

The first part of the report focuses on the algorithms used to describe the geometry of the design in the analysis mesh. The intersection points of the zero level set are computed on the elemental edges by interpolating the nodal level set values at the nodes of the edge. Then, we use this information to perform a Delaunay triangulation [Lee and Schachter, 1980] and divide the analysis domain into subdomains to perform numerical integration. This process was illustrated in Figures 3.1 and 3.2.

The second part of the report focuses on the approximation of the solution. Previous studies on topology optimization for 3D problems with the XFEM [Li et al, 2012] have employed a simplified enrichment scheme which is limited to “material-void” problems, and may suffer from the artificial coupling of disconnected material. Our work overcomes these issues by adopting a generalized enrichment scheme (3.1). The key challenge of this scheme is to identify the enrichment levels needed to consistently interpolate the solution field 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, loops over all elements connected to the current node. As this approach processes an element repeatedly, the following simple and efficient two-step novel scheme is introduced:

(1) A temporary, local 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 solution field in an elemental subdomain. Because this assignment is done individually for each element, the continuity of the interpolation across elements is not guaranteed.

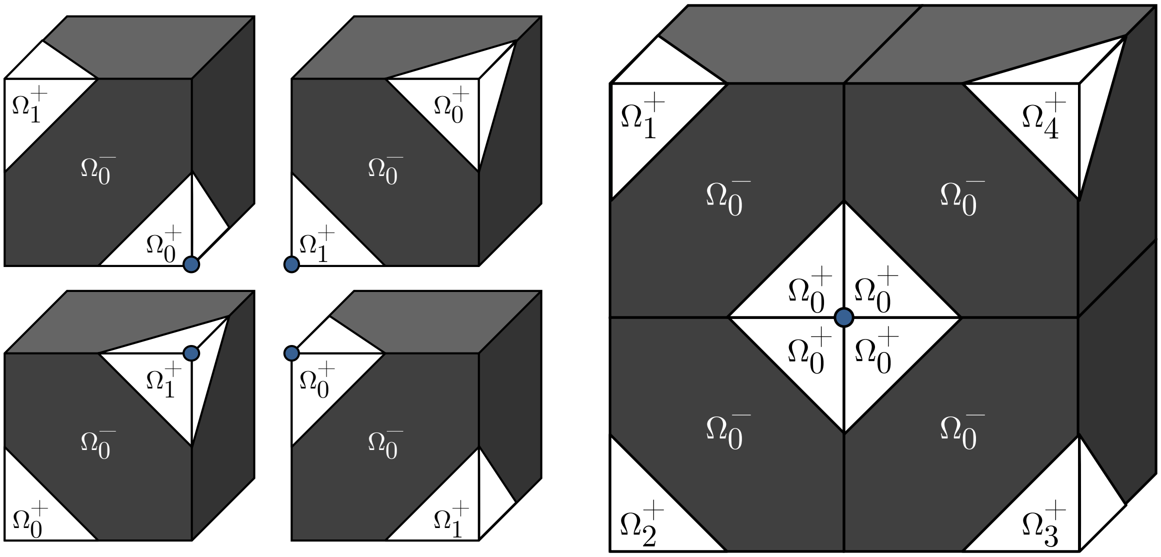
(2) The nodal enrichment levels are constructed using a topology-flipping algorithm. This process ensures that the solution field is interpolated continuously across elements, and by a different set of shape functions for each disconnected elemental subdomain of the same phase. To this end, the cluster 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 algorithm is illustrated in Figure 3.6. Figure 3.6a shows the triangulation and local enrichment levels for a 4-element nodal cluster in 3D. The node of interest is the one located at the center of the element cluster. In Step 1, each subdomain within each phase is assigned a local enrichment level. Applying these local enrichment levels to the degrees-of-freedom would incorrectly approximate the solution field. Analyzing the element cluster around the center nodes shows that these subdomains are disconnected and individual enrichment levels are assigned, as shown in Figure 3.6b.

For more details on the algorithmic implementation, the reader is referred to Appendix A.

3.4 Boundary Conditions

In the context of topology optimization, an immersed boundary technique such as the XFEM is attractive because the method does not require a fixed domain of integration. This characteristic



(a) Local enrichment levels at the element level.

(b) Global enrichment levels at the nodal cluster level.

Figure 3.6: Enrichment levels for a cluster of elements around a center node (blue). Superscripts denote the material phase, and subscripts denote the enrichment level. The node must recognize the enrichment levels needed to consistently interpolate the solutions in the subdomains, and to avoid the artificial coupling of disconnected material.

presents the capability to vary the shape and/or position of the surfaces on which the boundary conditions are applied during the course of the optimization process. The Heaviside enrichment in (3.1) bypasses the issues of kink enrichments, and can represent more general jump discontinuities that are common in, for example, displacements across cracks and temperature fields at small scales across material interfaces [Makhija and Maute, 2014b]. However, the approximation allows for discontinuities of the solution fields along the phase boundaries. Therefore, the continuity is enforced weakly either by the stabilized Lagrange multiplier method [Burman and Hansbo, 2010] or by Nitsche's method [Burman and Hansbo, 2012].

We study stabilized Lagrange multipliers in the context of linear elastic structures, and Nitsche's method for all the other governing equations in Section 4.2.

3.5 Stability

One challenge of the XFEM is that an ill-conditioned system of equations results when the ratio of the phases volumes in an element cut by the zero level set is very small or very large. Such interface configurations are often unavoidable when using fixed meshes in topology optimization. In general, the ill-conditioning impedes the convergence of solvers for nonlinear problems and reduces the performance of iterative linear solvers. Several approaches have been proposed to avoid this ill-conditioning issue, such as the geometric preconditioner of Lang et al [2014], the Jacobi preconditioner of Sauerland and Fries [2013], the preconditioners of Béchet et al [2005] and Menk and Bordas [2011] based on a Cholesky decomposition, and face-oriented ghost-penalty methods [Burman et al, 2006]. The geometric preconditioner of Lang et al [2014] has been studied in the context of linear diffusion [Lang et al, 2014]. Face-oriented ghost-penalty methods have been studied in the context of fluid flow problems, where discontinuities in the spatial gradients of the velocities and the pressure are penalized across the common facets of intersected elements [Schott et al, 2014]. Here, we explore the influence and the performance of the ghost-penalty methods in the context of flow topology optimization (Section 5.2), and the characteristics of the geometric preconditioner in linear diffusion and linear elasticity problems (Section 5.1).

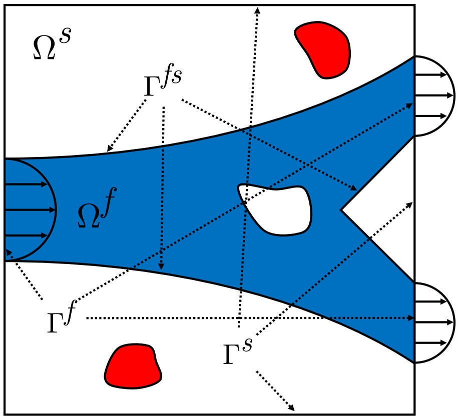


Figure 3.7: Geometry description of the fluid and solid domains with a LSF. Blue region represents the fluid channels. Isolated fluid regions surrounded by solid are denoted in red. White regions represent the solid domain.

A second challenge of the XFEM is that isolated regions may emerge during the optimization process. In the context of flow topology optimization, isolated fluid regions surrounded by solid, as shown in Figure 3.7, produce a singular analysis system because the absolute value of the pressure field is not governed. In the context of linear elasticity topology optimization, isolated free-floating solid particles may emerge and produce rigid body motion. This issue does not exist in an Ersatz material approach because the phase surrounding the isolated regions is modeled via a soft material. A similar approach can be applied to the XFEM to suppress the singularities, and the phase can be modeled via a soft material [Wei et al, 2010]. However, this approach requires accounting for the interface contributions and integrating the governing equations over an additional domain, which increases the computational cost. Here, we augment our fluid model with a penalty formulation to enforce a constraint on the average pressure, and we augment our linear elasticity model with a system of soft springs. However, unlike Villanueva and Maute [2014], where the penalty was applied to the entire domain, we model an auxiliary indicator field to detect these isolated regions and only apply the penalty there (Section 4.2.4).

Chapter 4

Analysis

The main challenge in optimizing the topology of linear elastic structures, incompressible fluid flow, heat transfer, or species transport problems is the modeling and numerical prediction of the displacement, flow, temperature, and species fields. This section introduces the weak form of the governing equations, outlines the temporal discretization schemes, the linear and nonlinear solvers, the sensitivity analysis, and the design criteria.

4.1 Analysis Domain

As described in Section 2, the analysis domain, Ω, is subdivided into a negative and a positive phases, Ω^+ and Ω^-, respectively, as shown in Figure 4.1b. The set of all elements in the domain Ω is denoted as Ω_e. In order to define the boundary conditions, as well as to enforce the continuity of the solution field at the material interface, we define the surfaces $\partial\Omega^-$ and $\partial\Omega^+$ as the whole domain boundaries of the negative and positive phases, respectively (see Figures 4.1d and 4.1e). Furthermore, each of these domain boundaries are subdivided into external and internal surfaces, $\partial\Omega^m_{\text{ext}}$ and $\partial\Omega^m_{\text{int}}$, respectively, where $m = \{-, +\}$ represents the negative or positive domains (see Figures 4.1f and 4.1g). Notice that the internal surfaces are equivalent to the material interface, and therefore, we will favor the use of the term Γ^0 to refer to them from here on out. Similarly, for simplicity, we relabel the external boundaries as simply Γ^m. The Dirichlet boundary conditions on the external surfaces and the material interface are then defined as Γ^m_{D} and Γ^0_{D}, respectively, and the Neumann boundary conditions are defined as Γ^m_{N} and Γ^0_{N}. These boundaries will be used to define the governing equations in the following sections.

4.2 Governing Equations

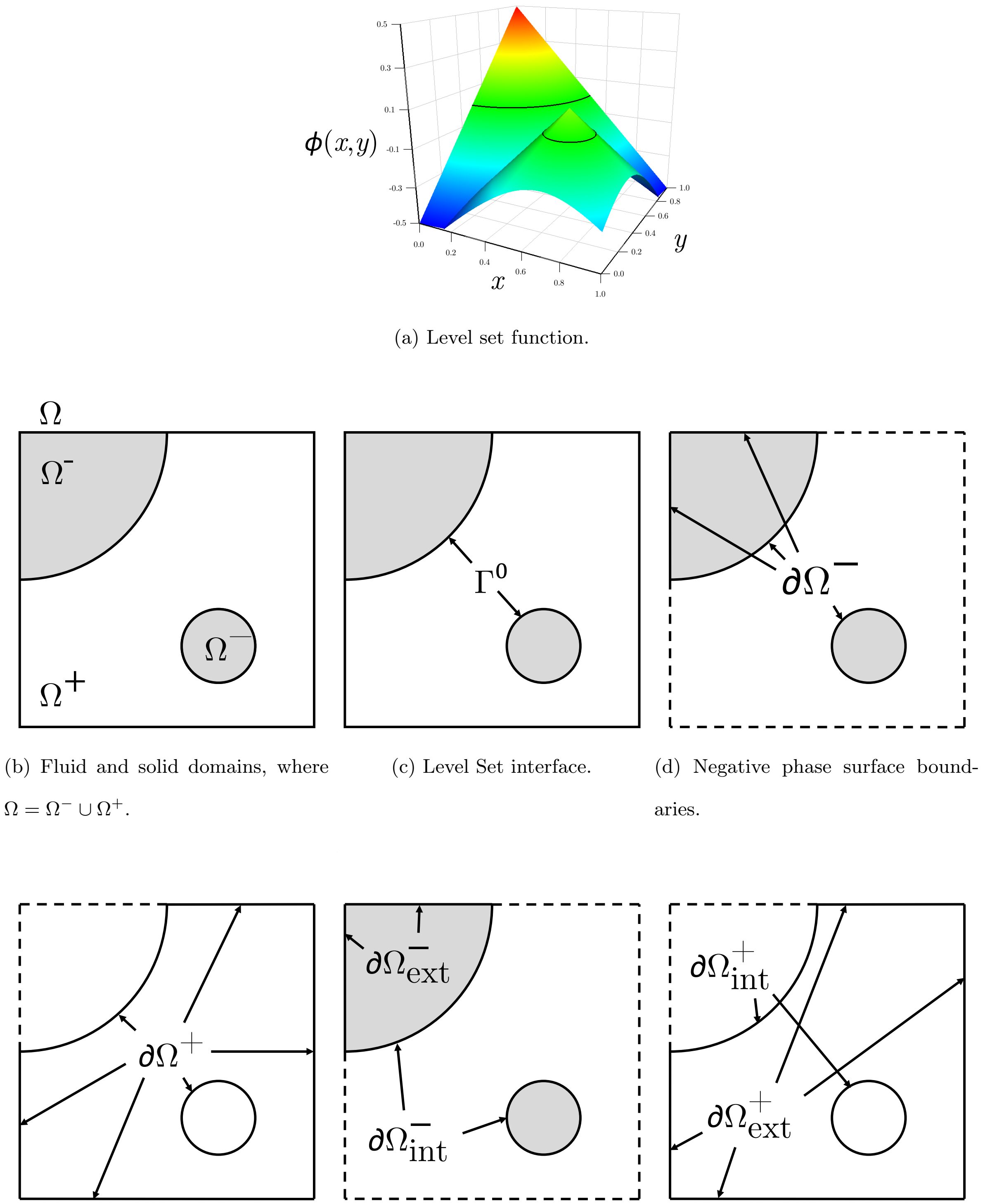
In this work, we consider the analysis of linear elastic structures, laminar incompressible flow, and energy and species transport problems. The structural problems are modeled assuming assuming infinitesimal strains, a linear elastic material behavior, and static conditions.

We model the flow by the incompressible Navier-Stokes equations, which describe the transport of momentum and the conservation of mass. To model heat transfer dominated by natural convection, the Navier-Stokes equations are augmented by a Boussinesq approximation of the buoyancy forces. Energy and species transport is described by coupling an advection-diffusion equation to our flow model. For convenience, we relabel the analysis domain such that the fluid domain, Ω^f, is defined by the negative phase, Ω^-; the solid domain, Ω^s, is defined by the positive phase; and the fluid-solid interface, Γ^{fs} is defined by the material interface, Γ^0. Note that we enforce no-slip boundary conditions at the fluid-solid interface; therefore, the fluid behavior is only modeled in the Ω^f domain. A schematic of the setup for the flow problems is shown in Figure 3.7.

For the flow problems, an indicator field is introduced in the fluid domain to identify isolated fluid regions where a constraint on the average fluid pressure is enforced. For the structural problems, the indicator field identifies free-floating solid particles where a system of soft springs is placed. The governing equations in the fluid and solid phases are summarized subsequently.

4.2.1 Linear Elastic Structural Mechanics

In this work, we consider the topology optimization of structures using the LSM and the XFEM to predict the structural response, assuming infinitesimal strains, a linear elastic material behavior, and static conditions. We model solid-void and solid-solid problems. From Figure 4.1, Γ^m_{N} denotes the surface where traction forces are applied, Γ^m_{D} is the surface with prescribed displacements, and Γ^0_{D} constitutes the material interface where the continuity of the solution is applied.



(a) Level set function.

(b) Fluid and solid domains, where $\Omega = \Omega^- \cup \Omega^+$.

(c) Level Set interface.

(d) Negative phase surface boundaries.

(e) Positive phase surface boundaries.

(f) Negative phase internal and external surface boundaries.

(g) Positive phase internal and external surface boundaries.

Figure 4.1: Geometry description of the negative and positive domains with a level set function.

The strong form of the governing equations is defined as follows:

$$-\nabla \cdot \sigma_{ij}\left(\boldsymbol{u}^m\right) = \hat{f}_i^m \ , \quad \forall \boldsymbol{x} \in \Omega^m \ , \tag{4.1}$$

where $\boldsymbol{u}^m$ is the displacement vector, $\hat{f}_i^m$ is the prescribed body force, and $\sigma_{ij}\left(\boldsymbol{u}^m\right)$ is the stress tensor defined using the following constitutive model:

$$\sigma_{ij}\left(\boldsymbol{u}^m\right) = D_{ijnp}^m \epsilon_{np}\left(\boldsymbol{u}^m\right) \ , \tag{4.2}$$

where D_{ijnp}^m is the fourth order constitutive tensor for the isotropic material of phase m, and $\epsilon_{np}\left(\boldsymbol{u}^m\right)$ is the infinitesimal strain tensor defined as:

$$\epsilon_{np}\left(\boldsymbol{u}^m\right) = \frac{1}{2}\left(\frac{\partial u_n^m}{\partial x_p} + \frac{\partial u_p^m}{\partial x_n}\right) \ . \tag{4.3}$$

Dirichlet and Neumann boundary conditions are imposed on the material interface, Γ^0, and on the external boundaries, Γ^m, as:

$$u_i^m = \hat{u}^m \ , \qquad \forall \boldsymbol{x} \in \Gamma_{\mathrm{D}}^m \ , \tag{4.4}$$

$$\sigma_{ij}\left(\boldsymbol{u}^m\right) \cdot n_i^m = \hat{t}_i^m \ , \qquad \forall \boldsymbol{x} \in \Gamma_{\mathrm{N}}^m \ , \tag{4.5}$$

$$u_i^- = u_i^+ \ , \qquad \forall \boldsymbol{x} \in \Gamma_{\mathrm{D}}^0 \ , \tag{4.6}$$

$$\sigma_{ij}\left(\boldsymbol{u}^-\right) \cdot n_i^\Gamma = \sigma_{ij}\left(\boldsymbol{u}^+\right) \cdot n_i^\Gamma \ , \qquad \forall \boldsymbol{x} \in \Gamma_{\mathrm{D}}^0 \ , \tag{4.7}$$

where n_i^Γ is the normal at the material interface pointing from the negative phase into the positive phase.

After the integration by parts of (4.1), the residual of the weak form of the linear elastic constitutive equations, denoted as $r_{\boldsymbol{u}}$, is decomposed into volumetric and surface contributions:

$$r_{\boldsymbol{u}} = r_{\boldsymbol{u}}^\Omega + r_{\boldsymbol{u},\psi}^\Omega + r_{\boldsymbol{u}}^{\mathrm{D}} + r_{\boldsymbol{u}}^0 + r_{\boldsymbol{u}}^{\mathrm{N}} + r_{\boldsymbol{u}}^{\mathrm{GP}} \ , \tag{4.8}$$

where $r_{\boldsymbol{u}}^\Omega$ is the residual of the volumetric contributions. The term $r_{\boldsymbol{u},\psi}^\Omega$ is used to model a system of soft springs in free-floating solid particles surrounded by void. The terms $r_{\boldsymbol{u}}^{\mathrm{D}}$ and $r_{\boldsymbol{u}}^0$ represent the residuals of the Dirichlet boundary conditions on the external surfaces, and at the material interface, respectively, and $r_{\boldsymbol{u}}^{\mathrm{N}}$ is the residual of the Neumann conditions on the external boundaries. The ghost-penalty term, $r_{\boldsymbol{u}}^{\mathrm{GP}}$, depends on the face-oriented ghost-penalty formulation, and is defined in Section 5.2.

The non-stabilized volumetric contribution is defined as:

$$r_{\boldsymbol{u}}^{\Omega} = \sum_{m \in \{-,+\}} \int_{\Omega^m} \epsilon_{ij}\left(\boldsymbol{v}^m\right) \sigma_{ij}\left(\boldsymbol{u}^m\right) + \boldsymbol{v}_i^m \hat{f}_i^m \, \mathrm{d}\Omega \ , \tag{4.9}$$

where $\boldsymbol{v}^m$ is an admissible test function.

The displacement continuity along phase boundaries for solid-solid problems is imposed via the stabilized Lagrange multiplier method from Makhija and Maute [2014b]:

$$r_{\boldsymbol{u}}^{\mathrm{D},0} = r_{\boldsymbol{u}}^{\mathrm{D},0,\boldsymbol{\lambda}} + r_{\boldsymbol{u}}^{\mathrm{D},0,\boldsymbol{u}} \ , \tag{4.10}$$

where:

$$r_{\boldsymbol{u}}^{\mathrm{D},0,\boldsymbol{\lambda}} = \int_{\Gamma^0} \gamma_{\mathrm{L},\boldsymbol{u}} \delta\lambda_i [\![\boldsymbol{u}]\!] + \delta\lambda_i \left(\lambda_i - \left\{\sigma_{ij}\left(\boldsymbol{u}\right) \cdot n_j^{\Gamma}\right\}\right) \mathrm{d}\Gamma \ , \tag{4.11}$$

where $\left\{\sigma_{ij}\left(\boldsymbol{u}\right) \cdot n_j^{\Gamma}\right\}$ is computed using:

$$\left\{J_i\left(\zeta^-, \zeta^+\right)\right\} = \gamma_J^- J_i\left(\zeta^-\right) + \gamma_J^+ J_i\left(\zeta^+\right) \ , \tag{4.12}$$

with $\gamma_J^- = \gamma_J^+ = 0.5$, $\gamma_{\mathrm{L},\boldsymbol{u}}$ is a constant scaling factor, and:

$$r_{\boldsymbol{u}}^{\mathrm{D},0,\boldsymbol{u}} = -\int_{\Gamma^0} [\![\boldsymbol{v}]\!] \lambda_i \, \mathrm{d}\Gamma \ . \tag{4.13}$$

The unknown field $\boldsymbol{\lambda}$ operates on the traction. In this work, this field is discretized locally, that is, the field is not continuous across elements. We can either assume a constant value for the field in the element, or a bilinear or trilinear field to interpolate it using the nodal shape functions. The higher the factor $\gamma_{\mathrm{L},\boldsymbol{u}}$, the better the interface condition is satisfied, at the cost of numerical stability.

Alternatively, the boundary conditions can be imposed using Nitsche's method as:

$$r_{\boldsymbol{u}}^{\mathrm{D},0} = \int_{\Gamma^0} -[\![\boldsymbol{v}]\!] \left\{\sigma_{ij}\left(\boldsymbol{u}\right) \cdot n_j^{\Gamma}\right\} + \left\{\sigma_{ij}\left(\boldsymbol{v}\right) \cdot n_j^{\Gamma}\right\} [\![\boldsymbol{u}]\!] + \gamma_{\mathrm{N},\boldsymbol{u}} [\![\boldsymbol{v}]\!] [\![\boldsymbol{u}]\!] \, \mathrm{d}\Gamma \ , \tag{4.14}$$

where $\gamma_{\mathrm{N},\boldsymbol{u}}$ is a constant scaling factor. The first, second, and third terms correspond to the consistency, adjoint consistency, and penalty terms of the Nitsche formulation. In this thesis, we use the symmetric formulation for the adjoint consistency term, and its sign is opposite to the first term.

The formulations to impose Dirichlet conditions on external surfaces can be derived from (4.10) and (4.14) by neglecting the second material phase.

The residual contribution from external Neumann boundary conditions is defined as:

$$r_{\boldsymbol{u}}^{\mathrm{N}} = \sum_{m \in \{-,+\}} \sum_{\Omega_e \in \Omega} \int_{\Omega_e \cap \Gamma_{\mathrm{N}}^m} v_i^m \hat{t}_i^m \, \mathrm{d}\Gamma \ . \tag{4.15}$$

Imposing boundary conditions weakly via the stabilized Lagrange multiplier method requires a linear solve per finite element for each Newton iteration. Conversely, the additional terms in Nitsche's method are incorporated naturally into the linear solve of the global system. However, Nitsche's method may require additional stability terms to ensure the convergence of the solver. We will study this in more detail in Section 5.2.

In solid-void topology optimization problems, free-floating solid particles surrounded by void material may emerge, leading to a rigid body mode. We extend the approach of Makhija and Maute [2014b] onto 3D problems and assume that the solid phase is supported by weak fictitious springs. The residual, $r_{\boldsymbol{u},\psi}^{\Omega}$, is defined as:

$$r_{\boldsymbol{u},\psi}^{\Omega} = \int_{\Omega^s} v_i^s k_{\boldsymbol{u}}^s \bar{\psi}^s u_i^s \, \mathrm{d}\Omega \ , \tag{4.16}$$

where $k_{\boldsymbol{u}}^s$ denotes the stiffness of the distributed system of springs, and $\bar{\psi}^f$ is the indicator field. However, unlike Makhija and Maute [2014b], where the penalty was applied to the entire domain, the auxiliary indicator field has a value of 1 in the free-floating regions and zero everywhere else; the equations to model the field will be defined in Section 4.2.4.

4.2.2 Incompressible Navier-Stokes Equations

The flow behavior in the fluid phase is modeled by the transient incompressible Navier-Stokes equations. The heat transfer is dominated by natural convection, and the buoyancy forces are modeled by the Boussinesq approximation. The strong form of the equations is described as follows:

$$
\begin{aligned}
\rho^f \left(\frac{\partial u_i^f}{\partial t} + u_j^f \frac{\partial u_i^f}{\partial x_j} \right) + \frac{\partial p^f}{\partial x_j} \delta_{ij} - 2\mu^f \frac{\partial}{\partial x_j} \Big(\epsilon_{ij} \left(\boldsymbol{u}^f \right) \Big) & \\
+ \rho^f g_i \Big(1 - h_T^f \left(T^f - T_\infty^f \right) \Big) &= 0 \ , \quad \forall \boldsymbol{x} \in \Omega^f \ , \\
\frac{\partial u_i^f}{\partial x_i} &= 0 \ , \quad \forall \boldsymbol{x} \in \Omega^f \ ,
\end{aligned}
\tag{4.17}
$$

where u_i^f is the velocity vector, p^f is the pressure, T^f is the temperature field, ρ^f is the density, μ^f is the dynamic viscosity, g_i denotes the gravity acceleration vector, h_T^f is the fluid thermal expansion coefficient, T_∞^f is the reference temperature value, and $\epsilon_{ij} \left(\boldsymbol{u}^f \right)$ is the strain rate tensor given by:

$$
\epsilon_{ij} \left(\boldsymbol{u}^f \right) = \frac{1}{2} \left(\frac{\partial u_i^f}{\partial x_j} + \frac{\partial u_j^f}{\partial x_i} \right) \ . \tag{4.18}
$$

In this form, the reference temperature drives the magnitude of the buoyancy forces through the fluid domain.

The fluid domain is decomposed into the fluid-solid interface, and the Dirichlet and Neumann external boundaries, Γ_{D}^f and Γ_{N}^f, respectively. The boundary conditions are then given as:

$$
u_i^f = \hat{u}_i^{fs} \ , \qquad \forall \boldsymbol{x} \in \Gamma^{fs} \ , \tag{4.19}
$$

$$
u_i^f = \hat{u}_i^f \ , \qquad \forall \boldsymbol{x} \in \Gamma_{\mathrm{D}}^f \ , \tag{4.20}
$$

$$
\sigma_{ij} \left(\boldsymbol{u}^f, p^f \right) n_j^f = \hat{t}_i^f \ , \qquad \forall \boldsymbol{x} \in \Gamma_{\mathrm{N}}^f \ , \tag{4.21}
$$

where $\hat{u}_i^{fs}$ and $\hat{u}_i^f$ are the prescribed velocities, $\hat{t}_i^f$ is the traction, n_j^f is the normal on the surface pointing outwards, and $\sigma_{ij} \left(\boldsymbol{u}^f, p^f \right)$ is the Cauchy stress tensor for Newtonian fluids:

$$
\sigma_{ij} \left(\boldsymbol{u}^f, p^f \right) = -p^f \delta_{ij} + 2\mu^f \epsilon_{ij} \left(\boldsymbol{u}^f \right) \ . \tag{4.22}
$$

After the integration by parts of (4.17), the residual of the weak form of the incompressible Navier-Stokes equations, denoted as $r_{\boldsymbol{u},p}$, is decomposed into volumetric and surface contributions:

$$
r_{\boldsymbol{u},p} = r_{\boldsymbol{u},p}^{\Omega} + r_{\boldsymbol{u},p}^{\hat{\Omega}} + r_{p,\psi}^{\Omega} + r_{\boldsymbol{u},p}^{\mathrm{D}} + r_{\boldsymbol{u},p}^{fs} + r_{\boldsymbol{u},p}^{\mathrm{N}} + r_{\boldsymbol{u},p}^{\mathrm{GP}} \ , \tag{4.23}
$$

where $r^{\Omega}_{\boldsymbol{u},p}$ and $r^{\hat{\Omega}}_{\boldsymbol{u},p}$ are the residuals of the volumetric contributions, non-stabilized and stabilized, respectively. The term $r^{\Omega}_{p,\psi}$ is used to enforce a constraint on the average pressure in isolated fluid regions surrounded by solid. The terms $r^{\mathrm{D}}_{\boldsymbol{u},p}$ and $r^{fs}_{\boldsymbol{u},p}$ enforce the Dirichlet boundary conditions on the external surfaces, and at the fluid-solid interface, respectively, and $r^{\mathrm{N}}_{\boldsymbol{u},p}$ is the residual of the Neumann conditions on the external boundaries. The stabilization term, $r^{\hat{\Omega}}_{\boldsymbol{u},p}$, depends on the discretization scheme, and the ghost-penalty term, $r^{\mathrm{GP}}_{\boldsymbol{u},p}$, depends on the face-oriented ghost-penalty formulation. Both are defined in Sections 4.3 and 5.2, respectively.

The non-stabilized volumetric contribution is formulated as:

$$\begin{aligned} r^{\Omega}_{\boldsymbol{u},p} = \int_{\Omega^f} \left(v^f_i \rho^f \left(\frac{\partial u^f_i}{\partial t} + u^f_j \frac{\partial u^f_i}{\partial x_j} \right) + \epsilon_{ij} \left(\boldsymbol{v}^f \right) \sigma_{ij} \left(\boldsymbol{u}^f, p^f \right) \right) \mathrm{d}\Omega \\ + \int_{\Omega^f} \left(v^f_i \rho^f g_i - v^f_i \rho^f g_i h^f_T \left(T^f - T^f_{\infty} \right) \right) \mathrm{d}\Omega \\ + \int_{\Omega^f} \left(q^f \frac{\partial u^f_i}{\partial x_i} \right) \mathrm{d}\Omega \ . \end{aligned} \tag{4.24}$$

The first integral describes the momentum equations, with admissible test functions v^f_i; the second integral models the buoyancy forces; and the third integral describes the incompressibility condition, with admissible test function q^f.

Dirichlet boundary conditions are enforced weakly on the fluid-solid interface, and on the external surfaces via Nitsche's method [Nitsche, 1975]. The formulation adopted here is the one described in Schott et al [2014]. The surface residual of the external Dirichlet boundaries is defined as:

$$\begin{aligned} r^{\mathrm{D}}_{\boldsymbol{u},p} = \int_{\Gamma^f_{\mathrm{D}}} \left(v^f_i p^f \delta_{ij} n^f_j - v^f_i 2\mu^f \epsilon_{ij} \left(\boldsymbol{u}^f \right) n^f_j \right) \mathrm{d}\Gamma \\ + \int_{\Gamma^f_{\mathrm{D}}} \left(\beta_p q^f \delta_{ij} n^f_j u^f_i - \beta_\mu 2\mu^f \epsilon_{ij} \left(\boldsymbol{v}^f \right) n^f_j u^f_i \right) \mathrm{d}\Gamma \\ + \int_{\Gamma^f_{\mathrm{D}}} \left(\gamma_{\mathrm{N},\boldsymbol{u}} v^f_i \left(u^f_i - \hat{u}^f_i \right) \right) \mathrm{d}\Gamma \ , \end{aligned} \tag{4.25}$$

where $\gamma_{\mathrm{N},\boldsymbol{u}}$ is a penalty parameter. The first integral of (4.25) is incorporated naturally through the integration by parts of the momentum equations; its two terms are denoted as the pressure and viscous **standard** consistency terms, respectively. The second integral is the addition of pressure and viscous **adjoint** consistency terms, analogous to the consistency terms of the first integral. The third integral introduces an additional penalty term that ensures coercivity of the viscous part of the formulation, and balances the lack of coercivity that is introduced by the viscous standard and adjoint consistency terms [Schott et al, 2014]. The terms β_p and β_μ determine whether the adjoint consistency terms use a symmetric formulation ($\beta_p = +1$, $\beta_\mu = +1$), or a skew-symmetric formulation ($\beta_p = -1$, $\beta_\mu = -1$). In this work, we use the symmetric variant for the viscous adjoint consistency term because it leads to smaller errors compared to the skew-symmetric variant, as reported by Burman [2012]. For the pressure adjoint consistency term, we use the skew-symmetric variation because it consistently controls the mass conservation, $u_i^f n_i^f = 0$ [Bazilevs et al, 2010; Schott et al, 2014]. A similar treatment is applied on the fluid-solid interface for the term $r^{fs}_{\boldsymbol{u},p}$ by using $\hat{u}_i^{fs}$ instead of $\hat{u}_i^f$ in (4.25).

The penalty parameter $\gamma_{\mathrm{N},\boldsymbol{u}}$ is taken from Schott et al [2014] and defined as:

$$\gamma_{\mathrm{N},\boldsymbol{u}} = \alpha_{\mathrm{N},\boldsymbol{u}} \left(\frac{\mu^f}{h} + \frac{\rho^f \left\| \boldsymbol{u}^f \right\|_\infty}{6} \right) , \tag{4.26}$$

and its terms account for viscous-dominated and convective-dominated flows, respectively. The term $\alpha_{\mathrm{N},\boldsymbol{u}}$ is a constant problem-dependent penalty term, and the term $\left\| \boldsymbol{u}^f \right\|_\infty$ is the infinity norm evaluated at each integration point and differentiated at its maximum value. The influence of this penalty term will be studied later in this paper.

The residual contribution from external Neumann boundary conditions is defined as:

$$r^{\mathrm{N}}_{\boldsymbol{u},p} = \int_{\Gamma^f_{\mathrm{N}}} v_i^f \hat{t}_i^f \, \mathrm{d}\Gamma \ . \tag{4.27}$$

An auxiliary indicator field is developed in order to identify isolated fluid regions surrounded by the solid domain. These “puddles” lead to an ill-conditioned system of equations because the absolute value of the pressure is not governed. In order to stabilize the system, we add a penalty formulation to enforce an average pressure. This penalty is defined as:

$$r^{\Omega}_{p,\psi} = \int_{\Omega^f} q^f k_p^f \bar{\psi}^f p^f \, \mathrm{d}\Omega \ , \tag{4.28}$$

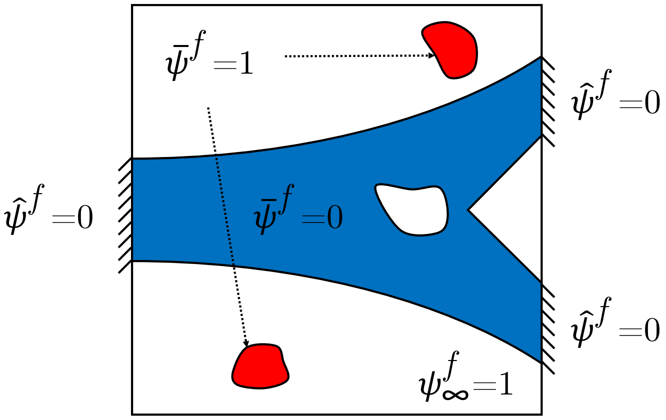


Figure 4.2: Modeling of the auxiliary indicator field. Isolated fluid regions surrounded by solid are denoted in red.

where k_p^f is a scaling factor. The field $\bar{\psi}^f$ serves as a binary indicator with a value of 1 in isolated fluid regions and zero everywhere else, as shown in Figure 4.2; the equations to model the field will be defined in Section 4.2.4. As the term (4.28) is introduced in the conservation of mass equation, we will study its effects on the conservation of mass later in this paper.

4.2.3 Advection-Diffusion Equation

The energy and species transport are modeled by an advection-diffusion equation:

$$\rho^m c_p^m \left(\frac{\partial T^m}{\partial t} + u_i^m \frac{\partial T^m}{\partial x_i} \right) - \frac{\partial}{\partial x_i} \left(J_i \left(T^m \right) \right) - \hat{q}_\Omega^m = 0 \ , \quad \forall \boldsymbol{x} \in \Omega^l \ , \tag{4.29}$$

where ρ^m and c_p^m denote the density and the specific heat capacity, respectively, T^m is the temperature field, u_i^m is the vector of fluid velocities, $\hat{q}_\Omega$ is the volumetric heat source, and $J_i \left(T^m \right)$ is the diffusive heat flux defined as:

$$J_i \left(T^m \right) = k^m \delta_{ij} \frac{\partial T^m}{\partial x_j} \ , \tag{4.30}$$

where k^m is the isotropic thermal conductivity. In contrast to our flow formulation in (4.24), the subscript m (4.29) indicates that we model energy and species transport in both the fluid and solid phases; however, because we apply no-slip boundary conditions at the fluid-solid interface, the residual in (4.36) reduces to a linear diffusion model in the solid region. The residual (4.36) can be used to model species transport by treating the temperature field as a species concentration.

Dirichlet and Neumann boundary conditions are imposed on the fluid-solid interface, Γ^{fs}, and on the external boundaries, Γ^{l}, as:

$$T^m = \hat{T}^m \ , \qquad \forall \boldsymbol{x} \in \Gamma^m_{\mathrm{D}} \ , \tag{4.31}$$

$$T^m = \hat{T}^{fs} \ , \qquad \forall \boldsymbol{x} \in \Gamma^{fs}_{\mathrm{D}} \ , \tag{4.32}$$

$$J_i\left(T^m\right) n^m_i = \hat{q}^m_\Gamma \ , \qquad \forall \boldsymbol{x} \in \Gamma^m_{\mathrm{N}} \ , \tag{4.33}$$

$$J_i\left(T^m\right) n^{fs}_i = \hat{q}^{fs}_\Gamma \ , \qquad \forall \boldsymbol{x} \in \Gamma^{fs}_{\mathrm{N}} \ , \tag{4.34}$$

where n^{fs}_i is the normal vector on the fluid-solid interface pointing towards the solid phase, $\hat{c}^m$ and $\hat{c}^{fs}$ are prescribed temperature concentrations, and $\hat{q}^m_\Gamma$ and $\hat{q}^{fs}_\Gamma$ are prescribed flux values.

Similar to the Navier-Stokes equations (4.24), we denote the weak form as $r_{\boldsymbol{u},T}$, and decompose it into volumetric and surface parts:

$$r_{\boldsymbol{u},T} = r^{\Omega}_{\boldsymbol{u},T} + r^{\hat{\Omega}}_{\boldsymbol{u},T} + r^{\mathrm{D}}_{\boldsymbol{u},T} + r^{\mathrm{D},fs}_{\boldsymbol{u},T} + r^{\mathrm{N}}_{\boldsymbol{u},T} + r^{\mathrm{N},fs}_{\boldsymbol{u},T} + r^{\mathrm{GP}}_{\boldsymbol{u},T} \ , \tag{4.35}$$

where $r^{\Omega}_{\boldsymbol{u},T}$ and $r^{\hat{\Omega}}_{\boldsymbol{u},T}$ are the residuals of the volumetric contributions, non-stabilized and stabilized, respectively; the terms $r^{\mathrm{D}}_{\boldsymbol{u},T}$ and $r^{\mathrm{D},fs}_{\boldsymbol{u},T}$ represent the residual of the Dirichlet boundary conditions at the external boundaries and at the fluid-solid interface, respectively; the terms $r^{\mathrm{N}}_{\boldsymbol{u},T}$ and $r^{\mathrm{N},fs}_{\boldsymbol{u},T}$ describe the residual of the Neumann boundary conditions; and the term $r^{\mathrm{GP}}_{\boldsymbol{u},T}$ models the ghost-penalty formulation. Similar to (4.23), the residual $r^{\hat{\Omega}}_{\boldsymbol{u},T}$ is defined in Section 4.3, and the residual $r^{\mathrm{GP}}_{\boldsymbol{u},T}$ is defined in Section 5.2.

The non-stabilized volumetric residual contribution, $r^{\Omega}_{\boldsymbol{u},T}$, is defined as:

$$r^{\Omega}_{\boldsymbol{u},T} = \sum_{m \in \{f,s\}} \int_{\Omega^m} \left(d^m \rho^m c^m_p \left(\frac{\partial T^m}{\partial t} + u^m_i \frac{\partial T^m}{\partial x_i} \right) + \frac{\partial d^m}{\partial x_i} \Big(J_i\left(T^m\right) \Big) - d^m \hat{q}^m_\Omega \right) \mathrm{d}\Omega \ , \tag{4.36}$$

where d^m is an admissible test function.

The weak enforcement of Dirichlet boundary conditions is modeled using Nitsche's method [Nitsche, 1975]. The residual contribution of the Dirichlet conditions at the fluid-solid interface follows the formulation of Dolbow and Harari [2009], and is defined as:

$$r^{\mathrm{D},fs}_{\boldsymbol{u},T} = \int_{\Gamma^{fs}} \left(- [\![d]\!] \left\{ J_i \left(T^f, T^s \right) \right\} n_i^{fs} + \left\{ J_i \left(d^f, d^s \right) \right\} n_i^{fs} [\![T]\!] + \gamma_{\mathrm{N},T} [\![d]\!] [\![T]\!] \right) \mathrm{d}\Gamma \ , \tag{4.37}$$

where the jump operators are defined as:

$$[\![\zeta]\!] = \zeta^f - \zeta^s \ , \tag{4.38}$$

and:

$$\left\{ J_i \left(\zeta^f, \zeta^s \right) \right\} = \gamma_J^f J_i \left(\zeta^f \right) + \gamma_J^s J_i \left(\zeta^s \right) \ , \tag{4.39}$$

where γ_J^m is a weighting factor defined as:

$$\gamma_J^m = \frac{\left(\int_{\Omega^m} \mathrm{d}\Omega \right) / k^m}{\left(\int_{\Omega^f} \mathrm{d}\Omega \right) / k^f + \left(\int_{\Omega^s} \mathrm{d}\Omega \right) / k^s} \ , \tag{4.40}$$

and $\gamma_{\mathrm{N},T}$ is a penalty term defined as:

$$\gamma_{\mathrm{N},T} = \frac{2 \alpha_{\mathrm{N},c} \int_{\Gamma^{fs}} \mathrm{d}\Gamma}{\left(\int_{\Omega^f} \mathrm{d}\Omega \right) / k^f + \left(\int_{\Omega^s} \mathrm{d}\Omega \right) / k^s} \ , \tag{4.41}$$

with $\alpha_{\mathrm{N},T}$ being a constant. The first, second, and third terms of (4.37) correspond to the standard consistency term, the adjoint consistency term, and the penalty term of the Nitsche formulation, respectively.

The residual contribution from the Dirichlet boundary conditions on the external surfaces is derived from (4.37) by enforcing the condition on each phase individually, and is defined as:

$$r^{\mathrm{D}}_{\boldsymbol{u},T} = \sum_{m \in \{f,s\}} \int_{\Gamma^m_{\mathrm{D}}} \left(- d^m J_i \left(T^m \right) n_i^m + J_i \left(d^m \right) n_i^m T^m + \alpha_{\mathrm{N},c} h^{-1} d^m \left(T^m - \hat{T}^m \right) \right) \mathrm{d}\Gamma \ . \tag{4.42}$$

The Neumann contribution at the fluid-solid interface is defined as:

$$r^{\mathrm{N},fs}_{\boldsymbol{u},T} = \int_{\Gamma^{fs}_{\mathrm{N}}} d^f \hat{q}^{fs}_{\Gamma} \, \mathrm{d}\Gamma \ , \tag{4.43}$$

while the contribution at the external boundaries is defined as:

$$r^{\mathrm{N}}_{\boldsymbol{u},T} = \sum_{m \in \{f,s\}} \int_{\Gamma^m_{\mathrm{N}}} d^m \hat{q}^m_{\Gamma} \, \mathrm{d}\Gamma \ . \tag{4.44}$$

If the fluid-solid interface is adiabatic, the prescribed surface flux, $\hat{q}^{fs}_{\Gamma}$, is set to 0.

4.2.4 Auxiliary Indicator Field

A drawback of the XFEM is that, as an immersed boundary technique, isolated regions may emerge during the optimization process. Fluid particles immersed in a solid domain, or structural particles immersed in a void domain cause the system of equations to become ill-conditioned due to a singular analysis problem. Here, we introduce an auxiliary indicator field to identify these isolated regions. The auxiliary field is modeled as a linear diffusion problem in the material phase of interest, m:

$$\frac{\partial}{\partial x_i}\bigg(J_i\left(\psi^m\right) \bigg) = h_\psi^m \left(\psi^m - \psi_\infty^m\right) \ , \tag{4.45}$$

where ψ^m is the indicator field, h_ψ^m is the heat transfer coefficient, and ψ_∞^m is the reference indicator value. The field is not modeled in the other phase domain. The residual of the weak form is defined as:

$$r_\psi = r_\psi^\Omega + r_\psi^{\mathrm{D}} + r_\psi^{\mathrm{GP}} \ , \tag{4.46}$$

where the residual of the Dirichlet boundary conditions, r_ψ^{D}, is formulated in the same way as in (4.42). Dirichlet boundary conditions are imposed on all inlets and outlets, by setting $\hat{\psi}^f$ to 0 and the Nitsche penalty parameter, $\alpha_{\mathrm{N},\psi}$, to 1. Adiabatic boundary conditions are imposed on the material interface.

The volumetric residual contribution, r_ψ^Ω is defined as:

$$r_\psi^\Omega = \int_{\Omega^m} \left(\frac{\partial \delta\psi^m}{\partial x_i} J_i\left(\psi^m\right) - \delta\psi^m h_\psi^m \left(\psi^m - \psi_\infty^m\right) \right) \mathrm{d}\Omega \ , \tag{4.47}$$

where $\delta\psi^m$ is an admissible test function. The parameters h_ψ^m and ψ_∞^m are set to 0.01 and 1, respectively, so that regions connected to the main domain will have an indicator field value close to 0, while the isolated areas will have a value close to 1.

A smooth-Heaviside projection scheme is then applied to the indicator field to map the values of the solution either to 0 or to 1, and is defined as:

$$\bar{\psi}^m = \frac{1}{2} + \frac{1}{2}\tanh\left(k_w^m\left(\psi^m - k_t^m \psi_\infty^m\right)\right) \ , \tag{4.48}$$

where a larger k_w^m corresponds to a sharper transition at $\psi^m = k_t^m \psi_\infty^m$.

We adopt the values of $k_w^m = 1000$ and $k_t^m = 0.99$ to effectively turn the $\bar{\psi}^f$ term into a binary switch, where a value of 0 corresponds to a regions connected to the main domain, and a value of 1 corresponds to the isolated regions.

4.3 Subgrid Stabilization

The convective terms in the incompressible Navier-Stokes and advection-diffusion equations may cause spurious node-to-node velocity oscillations. Furthermore, the equal-order approximations used for u_i^f and p^f may cause spurious pressure oscillations. To prevent these numerical instabilities, we augment the incompressible Navier-Stokes equations with the Streamline Upwind Petrov-Galerkin (SUPG) and the Pressure Stabilized Petrov-Galerkin (PSPG) stabilization formulations introduced by Tezduyar et al [1992]. The stabilized volumetric residual contribution of (4.23), $r_{\boldsymbol{u},p}^{\hat{\Omega}}$, is defined as:

$$
\begin{aligned}
r_{\boldsymbol{u},p}^{\hat{\Omega}} = \sum_{\Omega_e \in \Omega} \int_{\Omega_e \cap \Omega^f} \Bigg(\Bigg(& \tau_{\mathrm{SUPG},\boldsymbol{u}} \left(u_j^f \frac{\partial v_i^f}{\partial x_j} \right) + \tau_{\mathrm{PSPG}} \left(\frac{1}{\rho^f} \frac{\partial q^f}{\partial x_i} \right) \Bigg) \cdot \\
& \left(\rho^f \left(\frac{\partial u_i^f}{\partial t} + u_j^f \frac{\partial u_i^f}{\partial x_j} \right) + \frac{\partial p^f}{\partial x_j} \delta_{ij} - 2\mu^f \frac{\partial}{\partial x_j} \left(\epsilon_{ij} \left(\boldsymbol{u}^f \right) \right) + \rho^f g_i \left(1 - h_T \left(T^f - T_\infty^f \right) \right) \right) \Bigg) \mathrm{d}\Omega \ ,
\end{aligned}
\tag{4.49}
$$

where Ω_e denotes the set of all elements in the domain Ω, and the stabilization terms $\tau_{\mathrm{SUPG},\boldsymbol{u}}$ and τ_{PSPG} are taken from Tezduyar et al [1992] and defined as:

$$
\tau_{\mathrm{SUPG},\boldsymbol{u}} = \frac{h_\tau}{2 \left\| \boldsymbol{u}_{\mathrm{gp}}^f \right\|} \zeta \left(Re_{\boldsymbol{u}} \right) \ , \tag{4.50}
$$

$$
\tau_{\mathrm{PSPG}} = \frac{h_\tau^\#}{2 \left\| \boldsymbol{U}^f \right\|} \zeta \left(Re_{\boldsymbol{U}}^\# \right) \ , \tag{4.51}
$$

where $\boldsymbol{u}_{\mathrm{gp}}^f$ is the local velocity evaluated at every integration point, and $\left\| \boldsymbol{U}^f \right\|$ is the norm of the "global scaling velocity" [Tezduyar et al, 1992], which is set to the value of the characteristic velocity of the fluid, u_c^f. Based on these velocities, the elemental Reynolds numbers $Re_{\boldsymbol{u}}$ and $Re_{\boldsymbol{U}}^\#$

are defined as:

$$Re_{\boldsymbol{u}} = \frac{\left\| \boldsymbol{u}^f_{\text{gp}} \right\| h_\tau}{2\nu^f} \ , \tag{4.52}$$

$$Re^{\#}_{\boldsymbol{U}} = \frac{\left\| \boldsymbol{U}^f \right\| h^{\#}_\tau}{2\nu^f} \ , \tag{4.53}$$

where ν^f describes the kinematic viscosity of the fluid, and the "element lengths", h_τ and $h^{\#}_\tau$, are computed as:

$$h_\tau = 2 \left(\sum_{i=1}^{N^n_e} \left| \frac{\boldsymbol{u}^f_i}{\left\| \boldsymbol{u}^f_i \right\|} \cdot \nabla \boldsymbol{v}_i \right| \right)^{-1} \ , \tag{4.54}$$

$$h^{\#}_\tau = 2 \sqrt{\frac{\mathcal{V}_e}{\pi}} \ , \tag{4.55}$$

where $\boldsymbol{v}_i$ is the shape function associated with node i, N^n_e is the number of nodes in the element, and $\mathcal{V}_e$ is the volume of the element. The function $\zeta\,(Re)$ is computed as:

$$\zeta\,(Re) = \begin{cases} Re/3 \quad \texttt{if} \ 0 \leq Re \leq 3 \ . \\ 1 \quad \texttt{if} \ Re > 3 \ . \end{cases} \tag{4.56}$$

For more details on our formulation, the reader is referred to Kreissl and Maute [2011].

The stabilized volumetric residual contribution of (4.35), $r^{\hat{\Omega}}_{\boldsymbol{u},T}$, uses the SUPG method, and is defined as:

$$r^{\hat{\Omega}}_{\boldsymbol{u},T} = \sum_{\Omega_e \in \Omega} \int_{\Omega_e \cap \Omega^f} \left(\tau_{\text{SUPG},T} \cdot \left(\frac{1}{\rho^f c^f_p} u^f_i \frac{\partial d^f}{\partial x_i} \right) \cdot \left(\rho^f c^f_p \left(\frac{\partial c^f}{\partial t} + u^f_i \frac{\partial c^f}{\partial x_i} \right) - \frac{\partial}{\partial x_i} \left(J_i \left(c^f \right) \right) \right) \right) \mathrm{d}\Omega \ , \tag{4.57}$$

where the stabilization terms $\tau_{\text{SUPG},T}$ is defined in Franca et al [1992].

4.4 Temporal Discretization

In this thesis, the time integration is performed by a 2-step backward differentiation scheme. The spatial discretization yields a semi-discrete form of the residual equations:

$$\boldsymbol{R}^{(n)} \left(\boldsymbol{u}^{(n)}, \dot{\boldsymbol{u}}^{(n)} \right) = 0 \ , \tag{4.58}$$

where $\boldsymbol{u}$ is the vector of state variables at time step n, and the vector $\dot{\boldsymbol{u}}^{(n)}$ denotes the derivative of $\boldsymbol{u}^{(n)}$ with respect to time.

At time iteration $n = 0$, the initial conditions, $\boldsymbol{u}_0$, are satified for all the state variables such that:

$$\boldsymbol{R}^{(0)} = \boldsymbol{u}^{(0)} - \boldsymbol{u}_0 \ . \tag{4.59}$$

At time iteration $n = 1$, the derivative is computed using a 1-step backward differentiation scheme:

$$\dot{\boldsymbol{u}}^{(1)} = \frac{\boldsymbol{u}^{(1)} - \boldsymbol{u}^{(0)}}{\Delta t^{(1)}} \ , \tag{4.60}$$

where $\Delta t^{(n)}$ is the time step.

For all time steps $n > 1$, the derivative is computed using a 2-step backward differentiation scheme:

$$\dot{\boldsymbol{u}}^{(n)} = \frac{\frac{3}{2}\boldsymbol{u}^{(n)} - 2\boldsymbol{u}^{(n-1)} + \frac{1}{2}\boldsymbol{u}^{(n-2)}}{\Delta t^{(n)}} \ , \quad n = 2 \dots N_t \ , \tag{4.61}$$

where n is the time iteration, and N_t is the number of time iterations.

We utilize a homotopy approach for the steady-state flow problems in the numerical examples. That is, we initially choose a time step Δt that is sufficiently small to achieve stability of the solution, and then we gradually increase it until we achieve a steady-state flow. The final, large time step, and its corresponding solution are used in the subsequent optimization iteration as the initial conditions of the forward analysis. If the solution fails to converge, we repeat the process and we gradually increase the time step again.

4.5 Nonlinear and Linear Solvers

For all time steps $n > 0$, the equilibrium at the time step (n) is satisfied by solving the nonlinear system $\boldsymbol{R}^{(n)}$ via Newton's Method. The system is linearized at $\boldsymbol{u}^{(n)}$ using (4.58) as:

$$\frac{\mathrm{d}\boldsymbol{R}^{(n)}}{\mathrm{d}\boldsymbol{u}^{(n)}} = \left.\frac{\partial \boldsymbol{R}^{(n)}}{\partial \boldsymbol{u}^{(n)}}\right|_{\boldsymbol{u}^{(n)}} + \left.\frac{\partial \boldsymbol{R}^{(n)}}{\partial \dot{\boldsymbol{u}}^{(n)}}\frac{\partial \dot{\boldsymbol{u}}^{(n)}}{\partial \boldsymbol{u}^{(n)}}\right|_{\boldsymbol{u}^{(n)}} \ , \tag{4.62}$$

where the term $\partial\dot{\boldsymbol{u}}^{(n)}/\partial\boldsymbol{u}^{(n)}$ is computed analytically using (4.60) and (4.61).

The linear problem is solved with either a direct or an iterative linear solver. The linear solvers are used as provided by the Trilinos software package [Heroux et al, 2003]. Among the direct solvers provided are: (i) Amesos UMFPACK, a direct linear solver for sequential computations; (ii) Amesos MUMPS, a direct linear solver for parallel computations and (iii) Aztec GMRES, an iterative solver that implements the Generalized Minimal RESidual (GMRES) iterative method [Saad and Schultz, 1986]. Iterative linear solvers can be used in conjunction with preconditioners, such as the Incomplete LU factorization with dual Threshold (ILUT) method [Saad, 1994]. The reader is referred to Heroux et al [2003] for a complete list of solvers available in the Trilinos software package.

4.6 Sensitivity Analysis

The optimization functions considered in this work can be written in discretized form as:

$$\mathcal{Z} = \sum_{n=1}^{N_t} \left(z^{(n)} \left(\boldsymbol{s}, \boldsymbol{u}^{(n)} \right) \right) \, , \tag{4.63}$$

where $\mathcal{Z}$ represents either the objective or the constraints. The optimization problem (1.1) uses a gradient-based algorithm, and is solved by nonlinear programming methods. The gradients of the objective and constraint functions with respect to the design variables, $\boldsymbol{s}$, are computed via the adjoint method. In this work, we adopt the discrete adjoint formulation for nonlinear fluid and coupled systems of Kreissl and Maute [2011] and Golmon et al [2012].

The derivative of the objective function with respect to the optimization variable s_i is decomposed into an explicit and an implicit terms such that:

$$\frac{\mathrm{d}\mathcal{Z}}{\mathrm{d}s_i} = \frac{\partial \mathcal{Z}}{\partial s_i} + \sum_{n=1}^{N_t} \frac{\partial z^{(n)}}{\partial \boldsymbol{u}^{(n)}}^{\mathsf{T}} \frac{\partial \boldsymbol{u}^{(n)}}{\partial s_i} \, . \tag{4.64}$$

From (2.9), we can expand the explicit term as:

$$\frac{\partial \mathcal{Z}}{\partial s_i} = \frac{\partial \mathcal{Z}}{\partial \boldsymbol{x}_k^{\Gamma}} \frac{\partial \boldsymbol{x}_k^{\Gamma}}{\partial \phi_j} \frac{\partial \phi_j}{\partial s_i} \, . \tag{4.65}$$

The first term describes the change of the optimization function with respect to the position of the interface, defined by the intersection point coordinates, $\boldsymbol{x}_k^{\Gamma}$, on the element edges (c.f. Section

3.3). The second term represents the dependence of $\boldsymbol{x}_k^\Gamma$ with respect to the nodal level value, ϕ_j. The last term captures the explicit dependence of ϕ_j on the design variables, which is computed analytically by using (2.9).

The implicit term in (4.64) is computed by the adjoint method (1.10). In this work, the derivative of the optimization function with respect to the state variables, $\partial z^{(n)}/\partial \boldsymbol{u}^{(n)}$, is computed analytically.

As stated in Section 3.2, the derivative of the residual function with respect to the design variables deserves particular attention. In this work, we decompose the derivative of an elemental residual, $\boldsymbol{R}_e^{(n)}$, at time step n, as follows:

$$\frac{\partial \boldsymbol{R}_e^{(n)}}{\partial s_i} = \sum_{j=1}^{N_n^e} \sum_{k=1}^{N_n^\Gamma} \frac{\partial \boldsymbol{R}_e^{(n)}}{\partial \boldsymbol{x}_k^\Gamma} \frac{\partial \boldsymbol{x}_k^\Gamma}{\partial \phi_j} \frac{\partial \phi_j}{\partial s_i} \;, \tag{4.66}$$

where N_n^e is the number of nodes per element, and N_n^Γ is the number of intersection points per element. The first term in the double sum of (4.66) describes the change of the elemental residual with respect to the position of the interface, and the second and third terms are equivalent to their counterparts in (4.65). The decomposition in (4.66) illustrates that the proposed topology optimization framework utilizes shape derivatives to update the design in the optimization process.

The first two terms of (4.65) and (4.66) are computed by a central finite difference scheme. The perturbation size, $\Delta\phi_{\text{FD}}$, is set to $\phi_c^\Gamma/2$, in order for the lower and upper perturbed level set values to always be larger than 0. Note that the level set perturbation scheme introduced in Section 3.2 is not applied during the finite difference. This finite difference approach does not change the sign of the level set value ϕ_j. Because of this scheme, a non-cut element will not be suddenly cut by the perturbed level set functions during the sensitivity analysis. Therefore, for efficiency purposes, only the derivatives of elements cut by the level set function need to be computed. The derivatives of non-cut elements vanish. In the absence of a smoothing filter (2.9), the derivative $\partial\phi_j/\partial s_i = 1$; and for a level set field defined using a perfect signed distance function, the movement of the interface represents exactly the change in the level set field, i.e. $\partial \boldsymbol{x}_k^\Gamma/\partial\phi_j = 1$. Numerical tests in Section 7 rendered this finite differencing approach as being accurate and computationally efficient.

4.7 Measures and Design Criteria

Here, we detail the common measures and design criteria used to study the analysis capabilities of our framework, and to formulate the objectives and constraints in the optimization problems of Section 7. These include:

4.7.1 Strain Energy

This criteria measures the compliance of a mechanism.

$$\mathcal{U} = \sum_{m \in \{-,+\}} \int_{\Omega^m} \Big(\sigma_{ij}\left(\boldsymbol{u}^m\right) \epsilon_{ij}\left(\boldsymbol{u}^m\right) \Big) \, \mathrm{d}\Omega \ , \tag{4.67}$$

where $\sigma_{ij}\left(\boldsymbol{u}^m\right)$ is the elastic stress, and $\epsilon_{ij}\left(\boldsymbol{u}^m\right)$ is the elastic strain.

4.7.2 Drag Coefficient

The drag coefficient is used to qualify the forces of laminar flow on a surface, and is defined as:

$$c_D = -2e \Big(\rho^f |\boldsymbol{u}_c^f|^2 L_c \Big)^{-1} \left(\int_{\Gamma^f} \sigma_{ij}\left(\boldsymbol{u}^f, p^f\right) n_j^f \, \mathrm{d}\Gamma \right) \ , \tag{4.68}$$

where e is a unit vector pointing in the direction of the inflow velocity, and L_c is the characteristic length.

4.7.3 Mass Flow Rate

The mass flow rate criterion computes the mass of the fluid that passes through a surface per unit of time, and is defined as:

$$\dot{m} = \int_{\Gamma^f} \left(\rho^f u_i^f n_i^f \right) \mathrm{d}\Gamma \ . \tag{4.69}$$

4.7.4 Total Pressure

The total pressure criterion measures the sum of the static and dynamic pressures over a surface:

$$\mathcal{T} = \int_{\Gamma^f} \left(p^f + \frac{\rho^f |\boldsymbol{u}^f|^2}{2} \right) \mathrm{d}\Gamma \; . \tag{4.70}$$

4.7.5 Volume

The volumes of the negative and positive domains are computed as:

$$\mathcal{V}^- = \int_{\Omega^-} \mathrm{d}\Omega \; , \tag{4.71}$$

$$\mathcal{V}^+ = \int_{\Omega^+} \mathrm{d}\Omega \; . \tag{4.72}$$

4.7.6 Surface Area

The surface area criterion is computed at the material interface, and is defined as:

$$\mathcal{S} = \int_{\Gamma^0} \mathrm{d}\Gamma \; . \tag{4.73}$$

Reducing the value of this measure, either with a penalty in the objective functional or with a constraint, discourages the emergence of small geometric features and oscillatory shapes in the optimization problem. While a surface area penalty does not allow explicitly controlling the local shape and the feature size [Villanueva and Maute, 2014], it has been reported effective in regularizing flow optimization problems [van Dijk et al, 2013; Makhija and Maute, 2014a].

4.7.7 Target Scalar Value

To measure the maximum difference between a current species concentration and a target species concentration, T^f_{ref}, over the fluid phase, we use the Kreisselmeier-Steinhauser function [Kreisselmeier and Steinhauser, 1979]:

$$\mathcal{K} = \frac{1}{\beta_K} \ln \int_{\Gamma^f} \left(e^{\beta_K \left(T^f - T^f_{\text{ref}} \right)^2} \right) \mathrm{d}\Gamma \; , \tag{4.74}$$

where T^f is a species concentration modeled by (4.36). A larger value of β_K increases the enforcement of the function, but may result in large sensitivities that affect the convergence of the optimization problem.

4.7.8 Thermal Compliance

The thermal compliance criterion measures the rate of heat transfer, and is defined as:

$$\mathcal{Q}^m = \int_{\Omega^m} \left(\hat{q}_\Omega^m T^m \right) \mathrm{d}\Omega \ , \tag{4.75}$$

where T^m and $\hat{q}_\Omega^m$ are a temperature field and the volumetric heat flux modeled by (4.36).

4.7.9 Stresses

In traditional finite elements approaches, the gradients of the state variables are often extrapolated from the integration points to the nodes. In immersed boundary methods, this is no longer feasible, as the finite element has been divided into multiple integration domains. Here, we develop an averaging approach to measure the gradients across all integration domains within a finite element, and then project that information to the nodes. Our average gradient, denoted as $\tilde{\sigma}_{ij}\left(\boldsymbol{u}_l^m\right)$, is defined as follows:

$$\tilde{\sigma}_{ij}\left(\boldsymbol{u}^m\right) = \left(\int_{\Omega^m} (v_i^m)^\intercal v_i^m \, \mathrm{d}\Omega \right)^{-1} \int_{\Omega^m} v_i^\intercal \sigma_{ij}\left(\boldsymbol{u}^m\right) \mathrm{d}\Omega \ , \tag{4.76}$$

where v_i^m is an admissible test function, and $\sigma_{ij}\left(\boldsymbol{u}^m\right)$ is a gradient formulation, such as the stress tensor defined in (4.22).

Chapter 5

Stability

One challenge of the XFEM is that an ill-conditioned system of equations results when an element cut by the zero level set has a small ratio of volumes bisected by the interface; this process is illustrated in Figure 5.1. Such interface configurations are often unavoidable when using fixed meshes in topology optimization and lead to an ill-conditioning of the system, which manifests itself through an increase in the condition number of the linearized system, and may slow down or prevent the convergence of the nonlinear problems. Several approaches have been proposed to avoid this ill-conditioning issue, such as the geometric preconditioner of Lang et al [2014], and face-oriented ghost-penalty methods [Burman et al, 2006], among others. The former option was previously applied to linear diffusion problems in Lang et al [2014], and the latter was studied in linear diffusion problems [Burman and Hansbo, 2012], Stokes flows [Burman and Hansbo, 2014], and incompressible Navier-Stokes flows [Schott and Wall, 2014]. Prior to this dissertation, neither of these options had been applied and studied in the context of 3D topology optimization. We explore both options in Section 7. The preconditioner is studied in the context of linear elasticity, while face-oriented ghost-penalty methods are used for laminar incompressible flow problems.

5.1 Geometric Preconditioner

As described in Section 3.2, the degrees-of-freedom $\boldsymbol{u}_{i,l}^{m}$ interpolate the solution field in topologically connected subdomains, where $\boldsymbol{u}$ is the state variable, and m is the material phase of the elements connected to node i. As the total volume of these subdomains vanishes, the discretized

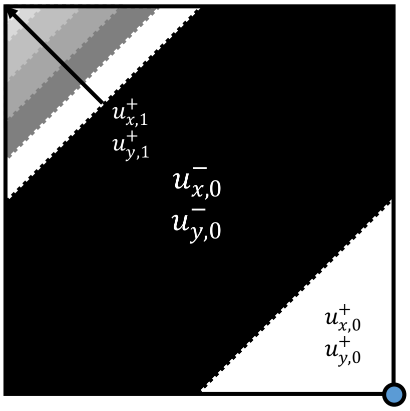


Figure 5.1: Example of a decreasingly smaller intersection area for degrees-of-freedom $u_{x,1}^{+}$ and $u_{y,1}^{+}$.

model becomes increasingly ill-conditioned; i.e. the condition number of the stiffness matrix rapidly increases. This phenomenon is more pronounced in 3D problems than in 2D ones.

To mitigate this ill-conditioning issue, we expand the geometric preconditioning scheme of Lang et al [2014], which was introduced and studied for 2D heat conduction and flow problems, onto 3D problems for different physical phenomena. 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:

$$\tilde{\boldsymbol{u}} = \boldsymbol{T}\boldsymbol{u} \ , \tag{5.1}$$

where $\boldsymbol{u}$ is the vector of state variables, $\boldsymbol{T}$ is a transformation matrix, and $\tilde{\boldsymbol{u}}$ is the solution vector in the transformed space. The residual, $\tilde{\boldsymbol{R}}$, and the stiffness matrix, $\tilde{\boldsymbol{K}}$, in the transformed space are defined as:

$$\begin{aligned} \tilde{\boldsymbol{R}} &= \boldsymbol{T}^{\intercal}\boldsymbol{R} \ , \\ \tilde{\boldsymbol{K}} &= \boldsymbol{T}^{\intercal}\boldsymbol{K}\boldsymbol{T} \ , \end{aligned} \tag{5.2}$$

where the residual, $\boldsymbol{R}$, and the stiffness matrix, $\boldsymbol{K}$, result from integrating the weak form of the governing equations using the XFEM approximation (3.1).

The preconditioner $\boldsymbol{T}$ 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:

$$\boldsymbol{T}^m_{i,l} = \left(\max_{\Omega_e \in \Omega^c_e} \frac{\int_{\Omega_e \cap \Omega^m_l} \nabla \boldsymbol{v}_i(\boldsymbol{x}) \cdot \nabla \boldsymbol{v}_i(\boldsymbol{x}) \, \mathrm{d}\Omega}{\int_{\Omega_e \cap \Omega^m} \nabla \boldsymbol{v}_i(\boldsymbol{x}) \cdot \nabla \boldsymbol{v}_i(\boldsymbol{x}) \, \mathrm{d}\Omega} \right)^{-1/2} \ , \tag{5.3}$$

where $\boldsymbol{T}^m_{i,l}$ corresponds to the degree-of-freedom $\boldsymbol{u}^m_{i,l}$ of node i, $m = \{-,+\}$ is the material phase, l is the enrichment level, Ω^c_e is the set of elements connected to node i, Ω^m_l is the domain of phase m that is integrated with enrichment level l with respect to node i, $\boldsymbol{v}_i(\boldsymbol{x})$ is the set of admissible test functions. The components of the matrix increase as the region of influence of a degree-of-freedom decreases. The entries $T^m_{i,l}$ of node i that are not connected to at least one intersected element are set to one. The integration domains are illustrated in Figure 5.2.

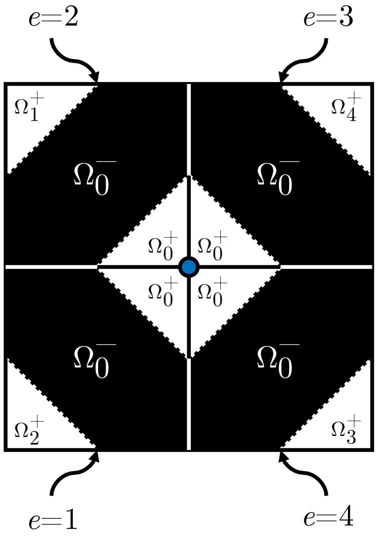


Figure 5.2: Integration subdomains for the spatial derivatives of the shape functions used in the geometric preconditioner. The center node, denoted by the color blue, denotes the node i. The superscript denotes the material phase m, and the subscript denotes the enrichment level l.

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

$$\boldsymbol{T}^m_{i,l} \geq T_{\text{tol}} \ , \tag{5.4}$$

where T_{tol} is a specified tolerance. As studies by Lang et al [2014] have shown, the above preconditioning scheme is rather insensitive to the value of T_{tol}, and the value is typically set larger than 10^{+08}. For more details on this formulation, the reader is referred to the paper by Lang et al [2014].

5.2 Face-oriented Ghost-penalty Methods

Face-oriented ghost-penalty stabilization terms are used in the vicinity of the zero level set interface, c.f. [Burman et al, 2006], to penalize discontinuties in the spatial gradients of the solution fields across the common facets of intersected elements. In this section, we describe the formulations for all the physical phenomena we model in this thesis.

5.2.1 Displacement Field

To maintain stability of the system, and ensure accuracy in the prediction of stresses for linear elastic problems, we utilize face-oriented ghost-penalty methods to penalize the jump of the gradients of the displacement as follows:

$$r_{\boldsymbol{u}}^{\text{GP}} = \sum_{m \in \{-,+\}} \sum_{F \in \Xi^m} \int_F \frac{\alpha_{\text{GP}}}{E} h \left[\!\left[\frac{\partial v_i^m}{\partial x_j} \right]\!\right] n_j^m \left[\!\left[\frac{\partial u_i^m}{\partial x_k} \right]\!\right] n_k^m \, \mathrm{d}\Gamma \ , \tag{5.5}$$

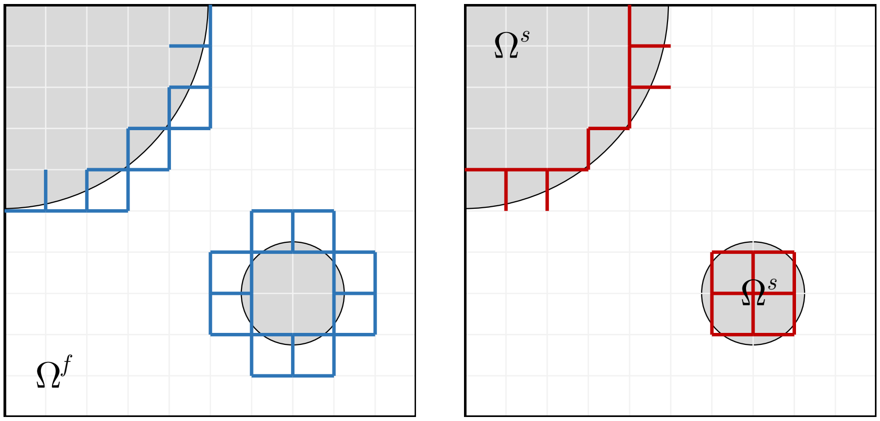
where α_{GP} is a constant scaling factor, and E is the isotropic Young's modulus.

The jump operator is defined as:

$$[\![\zeta]\!] = \zeta|_{\Omega_e^1} - \zeta|_{\Omega_e^2} \ , \tag{5.6}$$

and is evaluated at the facet between two adjacent elements, Ω_e^1 and Ω_e^2. This formulation overcomes the issue of having a small ratio of volumes on elements bisected by the interface because the domain of integration is the entire edge, regardless of the intersection configuration. As illustrated in Figure

5.3, the set Ξ^- belonging to the domain Ω^- contains all facets F in the immediate vicinity of the material interface, for which at least one of the two adjacent elements is cut by the interface. The set Ξ^+ for the domain Ω^+ is defined analogously.



(a) Faces Ξ^- in domain Ω^-.

(b) Faces Ξ^+ in domain Ω^+.

Figure 5.3: Integration domains for the face-oriented ghost-penalty method.

5.2.2 Velocity and Pressure Fields

The ghost-penalty terms for the residual contribution of the incompressible Navier-Stokes equations are defined as:

$$r_{\boldsymbol{u},p}^{\mathrm{GP}} = r_{\boldsymbol{u},p}^{\mathrm{GP},\mu} + r_{\boldsymbol{u},p}^{\mathrm{GP},p} + r_{\boldsymbol{u},p}^{\mathrm{GP},\boldsymbol{u}} \ , \tag{5.7}$$

where $r_{\boldsymbol{u},p}^{\mathrm{GP},\mu}$, $r_{\boldsymbol{u},p}^{\mathrm{GP},p}$, and $r_{\boldsymbol{u},p}^{\mathrm{GP},\boldsymbol{u}}$ are the viscous, pressure, and convective ghost-penalty formulations, respectively.

To overcome stability issues related to the weak enforcement of boundary conditions via Nitsche's method, we apply the viscous face-oriented ghost-penalty formulation as proposed by Burman and Hansbo [2014]:

$$r_{\boldsymbol{u},p}^{\mathrm{GP},\mu} = \sum_{F \in \Xi^f} \int_F \left(\gamma_{\mathrm{GP},\mu} \left[\!\!\left[\frac{\partial v_i^f}{\partial x_j} \right]\!\!\right] n_j^f \left[\!\!\left[\frac{\partial u_i^f}{\partial x_k} \right]\!\!\right] n_k^f \right) \mathrm{d}\Gamma \ , \tag{5.8}$$

where $\gamma_{\mathrm{GP},\mu}$ is a penalty parameter defined as:

$$\gamma_{\mathrm{GP},\mu} = \alpha_{\mathrm{GP},\mu} \mu^f h \ , \tag{5.9}$$

and $\alpha_{\mathrm{GP},\mu}$ is a constant scaling factor.

To control pressure instabilities due to a violated inf-sup condition for equal-order approximations used for u_i^f and p^f [Schott et al, 2014], a pressure ghost-penalty stabilization term is applied:

$$r_{\boldsymbol{u},p}^{\mathrm{GP},p} = \sum_{F \in \Xi^f} \int_F \left(\gamma_{\mathrm{GP},p} \left[\!\left[\frac{\partial q^f}{\partial x_j} \right]\!\right] n_j^f \left[\!\left[\frac{\partial p^f}{\partial x_k} \right]\!\right] n_k^f \right) \mathrm{d}\Gamma \ , \tag{5.10}$$

where $\gamma_{\mathrm{GP},p}$ is a penalty parameter defined as:

$$\gamma_{\mathrm{GP},p} = \alpha_{\mathrm{GP},p} \left(\frac{\mu^f}{h} + \frac{\rho^f \|\boldsymbol{u}^f\|_\infty}{6} \right)^{-1} h^2 \ , \tag{5.11}$$

and accounts for the viscous and convective flow regimes, c.f. [Burman et al, 2006]. The term $\alpha_{\mathrm{GP},p}$ is a constant scaling parameter.

For high Reynolds number flows, [Schott and Wall, 2014] proposed a convective ghost-penalty formulation to have sufficient control over the convective derivative, $u_i^f \nabla u_i^f$, of the incompressible Navier-Stokes equations. This formulation is defined as:

$$r_{\boldsymbol{u},p}^{\mathrm{GP},\boldsymbol{u}} = \sum_{F \in \Xi^f} \int_F \left(\gamma_{\mathrm{GP},\boldsymbol{u}} \left[\!\left[\frac{\partial v_i^f}{\partial x_j} \right]\!\right] n_j^f \left[\!\left[\frac{\partial u_i^f}{\partial x_k} \right]\!\right] n_k^f \right) \mathrm{d}\Gamma \ , \tag{5.12}$$

where the parameter $\gamma_{\mathrm{GP},\boldsymbol{u}}$ is a penalty factor defined as:

$$\gamma_{\mathrm{GP},\boldsymbol{u}} = \alpha_{\mathrm{GP},\boldsymbol{u}} \rho^f \left\| u_i^f n_i^f \right\| h^2 \ , \tag{5.13}$$

and $\alpha_{\mathrm{GP},\boldsymbol{u}}$ is a constant scaling parameter.

Additional ghost-penalty measures have been proposed in the literature, for example, to control instabilities arising from the incompressibility constraint. However, these additional formulations are not considered in our numerical examples because previous studies have not revealed any further improvement for the laminar flow situations analyzed here, as stated by Schott et al [2014].

5.2.3 Temperature Field

To stabilize the temperature field (4.36), we use the formulation from Burman and Hansbo [2012]:

$$r_{\boldsymbol{u},T}^{\mathrm{GP}} = \sum_{m \in \{f,s\}} \sum_{F \in \Xi^m} \int_F \left(\gamma_{\mathrm{GP},T} \left[\!\!\left[\frac{\partial d^m}{\partial x_i} \right]\!\!\right] n_i^f \, [\![J_j \left(T^m \right)]\!] \, n_j^f \right) \mathrm{d}\Gamma \ , \tag{5.14}$$

where $\gamma_{\mathrm{GP},T}$ is a penalty parameter defined as:

$$\gamma_{\mathrm{GP},T} = \alpha_{\mathrm{GP},T} h \ , \tag{5.15}$$

and $\alpha_{\mathrm{GP},T}$ is a scaling constant.

The ghost-penalty formulation for the auxiliary indicator field is identical to (5.14), except that it operates on a different admissible test function, $\delta\psi^m$, a different set of degrees-of-freedom, ψ^m, and a different scaling factor, $\alpha_{\mathrm{GP},\psi}$.

5.2.4 Parameters

The values for α_{GP}, $\alpha_{\mathrm{GP},\mu}$, $\alpha_{\mathrm{GP},p}$, $\alpha_{\mathrm{GP},\boldsymbol{u}}$, and $\alpha_{\mathrm{GP},T}$ are set on a per-problem basis. The value of $\alpha_{\mathrm{GP},\psi}$ is set to 0.05 for all numerical examples, in accordance to the parameter used by Burman and Hansbo [2012] for a linear diffusion field.

Chapter 6

Regularization

In recent decades, topology optimization has become an important tool in the design of engineering parts. Industrial manufacturing processes often impose constraints on the design of the part [Schmitt and Steinmann, 2015] that must be considered during the optimization process to ensure that the part can be built. Using postprocessing techniques to alter the optimized geometry and satisfy the side conditions given by the manufacturing process might lead to a loss of quality of the part. Therefore, manufacturing constraints have to be incorporated directly into the optimization problem in order to ensure that the optimized design is at the same time manufacturable. Furthermore, controlling the shape of the geometry ensures that we can accurately predict the physical response of its features. In LSM-based topology optimization, the formulation of the optimization problem (1.1) is often augmented by additional regularization measures to control the local or the global shape of the design geometry. In this section, we introduce several novel regularization techniques that operate on the discretized material interface as provided by the XFEM (2.1).

6.1 Shape Smoothers

Manufacturing constraints considered in shape and topology optimization are often expressed in terms of the curvature value on the surface of the optimized material layout. This work introduces shape smoothers to increase the smoothness, and therefore, decrease the curvature of discretized surfaces. This is of interest when the optimized part is fabricated using 3D printing technology, as the maximum resolution size is limited and the printing technique may not be able to properly represent sharp corners. These shape smoothers will be applied on boundaries defined by an immersed boundary technique; here, we utilize the XFEM (see Section 3). Rather than maximizing the smoothness of the surface using postprocessing methods, the curvature minimization is inserted directly into the optimization problem.

6.1.1 Curvature and Mean Curvature

Curvature is a mathematical concept that describes how sharply a curve bends; that is, curvature measures how much a geometric object deviates from being flat. For example, the curvature of a straight line is identically zero, 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 [Newton and Colson, 1736]. 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 [Coolidge, 1952].

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, and not its magnitude, we will consider the unit tangent vector.

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

$$\kappa = \kappa_{\boldsymbol{t}} = \left\| \frac{\mathrm{d}\boldsymbol{t}}{\mathrm{d}s} \right\| \ , \tag{6.1}$$

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

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

$$\kappa = \kappa_{\boldsymbol{n}} = \left\| \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \right\| \ . \tag{6.2}$$

Other definitions of curvature as a function of the normal unit vector can be found in Wang and Wang [2005], Wang and Wang [2004], and Luo et al [2008a], where the mean curvature measure was used:

$$H = \| \nabla \cdot \boldsymbol{n} \| \ . \tag{6.3}$$

6.1.2 Comparison of Curvature and Mean Curvature in $\mathbb{R}^2$

Mathematically, (6.2) and (6.3) 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.

6.1.2.1 Curvature of a Circle

In this example, we will compute the curvature of a circle with radius a to use it as an analytical solution to our curvature measures of Section 7.10.1. We can parametrize the curve of the circle as a function of the arclength, s, as:

$$\boldsymbol{r}\left(s\right) = \left\{ a \cos\frac{s}{a}, a \sin\frac{s}{a} \right\} \ . \tag{6.4}$$

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

$$\boldsymbol{n}\left(s\right) = \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}s} \left(\left\| \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}s} \right\| \right)^{-1} \ , \tag{6.5}$$

where $\mathrm{d}\boldsymbol{r}/\mathrm{d}s$ is computed as:

$$\begin{aligned} \frac{\mathrm{d}\boldsymbol{r}}{\mathrm{d}s} &= \left\{ a \left(-\frac{1}{a} \sin\frac{s}{a} \right), a \left(\frac{1}{a} \cos\frac{s}{a} \right) \right\} \ , \\ &= \left\{ -\sin\frac{s}{a}, \cos\frac{s}{a} \right\} \ . \end{aligned} \tag{6.6}$$

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

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

and its derivative with respect to the arclength is:

$$\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s}=\left\{-\frac{1}{a}\cos\frac{s}{a},-\frac{1}{a}\sin\frac{s}{a}\right\}\ . \tag{6.8}$$

Finally, the curvature $\kappa_{\boldsymbol{n}}$ is:

$$\kappa_{\boldsymbol{n}}=\left\|\frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s}\right\|=\frac{1}{a}\ . \tag{6.9}$$

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

6.1.2.2 Mean Curvature of a Circle

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

$$\begin{aligned} H &= \left\|\frac{\partial n_x}{\partial x}+\frac{\partial n_y}{\partial y}\right\|\ , \\ &= \left\|\frac{\partial n_x}{\partial s}\frac{\partial s}{\partial x}+\frac{\partial n_y}{\partial s}\frac{\partial s}{\partial y}\right\|\ . \end{aligned} \tag{6.10}$$

From (6.4), we parametrize x as $a\cos\left(s/a\right)$ and y as $a\sin\left(s/a\right)$. Therefore:

$$\begin{aligned} \frac{\partial s}{\partial x} &= \frac{x}{\sqrt{x^2+y^2}}=a\frac{\cos\left(s/a\right)}{a}\ , \\ \frac{\partial s}{\partial y} &= \frac{y}{\sqrt{x^2+y^2}}=a\frac{\sin\left(s/a\right)}{a}\ . \end{aligned} \tag{6.11}$$

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

$$\begin{aligned} \frac{\partial n_x}{\partial s} &= -\frac{1}{a}\cos\frac{s}{a}\ , \\ \frac{\partial n_y}{\partial s} &= -\frac{1}{a}\sin\frac{s}{a}\ . \end{aligned} \tag{6.12}$$

Finally, the mean curvature is

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

We can conclude that the curvature in (6.2) and the mean curvature in (6.3) formulations are the same for a geometrical object in $\mathbb{R}^2$. Note, however, that this assertion is not applicable to 3D objects.

6.1.2.3 Curvature of a Sinusoidal Wave

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

$$\boldsymbol{r}\left(s\right) = \left\{s, A\sin\pi s\right\} \ , \tag{6.14}$$

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

$$\kappa_{\boldsymbol{n}} = \pi^2 \left(\frac{A^2(1+\cos\left(\pi s\right)^2)\sin\left(\pi s\right)^2}{(1+A\pi\cos\left(\pi s\right)^2)^3} \right)^2 \ . \tag{6.15}$$

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

6.1.3 Curvature Squared

Dyn et al [2001] and Olsson and Boykov [2012] used the square power of the curvature $\kappa_{\boldsymbol{n}}$ as their measurement formulation. Because the shape of interest is the zero level set isocontour, we will measure the curvature in our design domain by integrating $\kappa_{\boldsymbol{n}}^2$ over the material interface (2.1).

$$\kappa^{\Gamma} = \int_{\Gamma^0} \kappa_{\boldsymbol{n}}^2 \, \mathrm{d}\Gamma = \int_{\Gamma^0} \left\| \frac{\mathrm{d}\boldsymbol{n}}{\mathrm{d}s} \right\|^2 \mathrm{d}\Gamma \ . \tag{6.16}$$

Given that curvature and mean curvature are equivalent in $\mathbb{R}^2$, we will focus on the curvature formulation of (6.2) to measure the smoothness of our designs.

6.1.4 Curvature of the Level Set Interface

As shown in Section 6.1.1, we need to define the normal vector of the level set field in order to measure the curvature along the Γ^0 isocontour. By definition of (2.2), we define the normal of the level set field as:

$$\boldsymbol{n}_{\phi} = \frac{\nabla\phi}{\|\nabla\phi\|} \ . \tag{6.17}$$

The normal will be interpolated at the interface Γ^0 from the nodally defined level set field, using bilinear or trilinear shape functions. Note that this formulation does not guarantee continuity of the level set normals at the nodes. For example, Figure 6.1 shows an inclusion embedded within 4

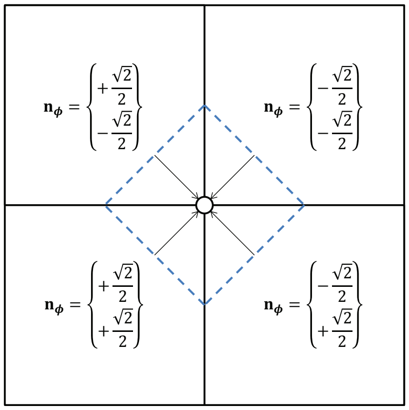

$\mathbf{n}_{\phi} = \begin{Bmatrix} +\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{Bmatrix}$
$\mathbf{n}_{\phi} = \begin{Bmatrix} -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} \end{Bmatrix}$
$\mathbf{n}_{\phi} = \begin{Bmatrix} +\frac{\sqrt{2}}{2} \\ +\frac{\sqrt{2}}{2} \end{Bmatrix}$
$\mathbf{n}_{\phi} = \begin{Bmatrix} -\frac{\sqrt{2}}{2} \\ +\frac{\sqrt{2}}{2} \end{Bmatrix}$

Figure 6.1: Discontinuities for a cuboid-like inclusion in the $\boldsymbol{n}_{\phi}$ field at a node, denoted by the circle at the center.

finite elements. The value of $\boldsymbol{n}_{\phi}$ at a node is interpolated differently from each one of its 4 adjacent elements. We will denote this formulation as κ_{ϕ}^{Γ}.

In our second formulation, we denote the level set normal as $\boldsymbol{n}_u$, and enforce (6.17) weakly as:

$$\int_{\Omega} \delta\boldsymbol{n}_u \left(\boldsymbol{n}_u \left\|\nabla\phi\right\| - \nabla\phi\right) \, \mathrm{d}\Omega = 0 \ , \tag{6.18}$$

where $\delta\boldsymbol{n}_u$ is an admissible test function. Our solution is computed using the XFEM to allow us to interpolate the state variables on the zero level set interface. Therefore, we need to ensure continuity of the $\boldsymbol{n}_u$ degrees-of-freedom at the interface. We will impose a penalty formulation for the normal unit vector such that $\boldsymbol{n}_u^+ = \boldsymbol{n}_u^-$ at Γ^0:

$$\begin{aligned} \gamma_\kappa \int_{\Gamma^+} \delta\boldsymbol{n}_u^+ \left(+\boldsymbol{n}_u^+ - \boldsymbol{n}_u^-\right) \, \mathrm{d}\Gamma^+ = 0 \ , \\ \gamma_\kappa \int_{\Gamma^-} \delta\boldsymbol{n}_u^- \left(-\boldsymbol{n}_u^+ + \boldsymbol{n}_u^-\right) \, \mathrm{d}\Gamma^- = 0 \ , \end{aligned} \tag{6.19}$$

where Γ^+ is the material interface, Γ^0, facing the positive phase, Γ^- is the one facing the negative phase, and γ_κ is a scaling factor. (6.17) 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 alternate formulation, 6.18, seeks to study the effect of using a continuous field on the curvature measure. We will denote this formulation as $\kappa_{\boldsymbol{n}}^{\Gamma}$.

6.1.5 Spline Curvature

This formulation will consider the interface of the cut elements as geometrical objects, and compute the normal unit vector from the coordinates of these objects. This approach will not be used with (6.1.3), but rather with an equivalent formulation by measuring strain energy.

The square power of the curvature is an equivalent measure to strain energy. This concept is derived from the theory of spline curves [Zhang et al, 2001]. This allows us to model the discretized interface, Γ^0, as a set of linear elastic structural beam elements. Then, we can solve a finite element problem and compute the strain energy of the interface. We denote this formulation as the spline curvature, and define it as:

$$\kappa_{\boldsymbol{u}}^{\Gamma} = \frac{1}{2} \int_{\Gamma^0} \boldsymbol{\sigma} : \boldsymbol{\epsilon} \, \mathrm{d}\Gamma \ . \tag{6.20}$$

To illustrate the application of this approach, consider a level set function that describes a sinusoidal wave (7.41) with an amplitude A_s of 0.50, as shown in Figure 6.2. The red dot in the figure indicates

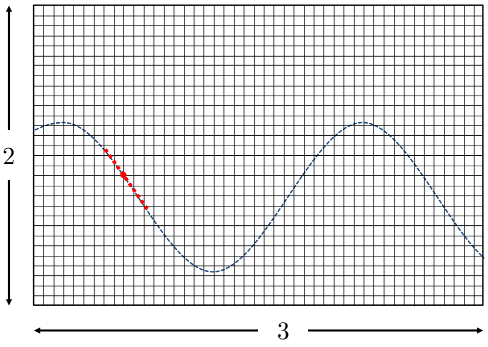


Figure 6.2: Sinusoidal wave with an amplitude of 0.50. The mesh discretization is 45×30 elements. The red dot indicates one of the points, $\boldsymbol{x}_i^{\Gamma}$, at which the level set function intersects an element. The red dotted line indicates all cut elements that lie on the same interface as the red dot within a search radius $r_\kappa = 0.4$.

one of the many points, $\boldsymbol{x}_i^\Gamma$, at which the level set function intersects an element. To compute the strain energy at this location, we define a search radius r_κ, and look for all the cut elements that share the same interface and that are within distance of this radius, as shown in Figure 6.3. From these elements, we extract the interfaces and build a set of structural linear beam elements, as shown in Figure 6.4. 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 have a single rotational degree-of-freedom, as shown in Figure 6.4. We assume the rotations at the internal nodes are continuous, such that:

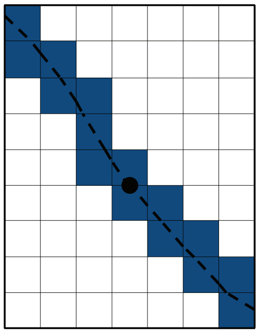


Figure 6.3: Zoomed image for the cut elements within the search radius $r_\kappa = 0.4$. Blue elements represent the cut elements that are on the same interface.

$$\theta_z^L - \theta_z^R = \alpha_{CR} \ , \tag{6.21}$$

where θ_z^L is the rotation of the left adjacent element around the z axis, θ_z^R is the rotation of the right one, and α_{CR} is computed for every node as a function of the geometrical normal, $\boldsymbol{n}_g$, of its adjacent elements as:

$$\begin{aligned} \boldsymbol{n}_g^C &= \frac{\boldsymbol{n}_g^L + \boldsymbol{n}_g^R}{2} \ , \\ \boldsymbol{n}_g^{LC} &= \boldsymbol{n}_g^L \times \boldsymbol{n}_g^C \ , \\ \boldsymbol{n}_g^{CR} &= \boldsymbol{n}_g^C \times \boldsymbol{n}_g^R \ , \\ \alpha_{CR} = \frac{{\boldsymbol{n}_g^{LC}}_z}{\sqrt{1 - {\boldsymbol{n}_g^{LC}}_z^{\,2}}} &+ \frac{{\boldsymbol{n}_g^{CR}}_z}{\sqrt{1 - {\boldsymbol{n}_g^{CR}}_z^{\,2}}} \ . \end{aligned} \tag{6.22}$$

This computation is illustrated in Figure 6.5. The condition in (6.21) is imposed by Lagrange multipliers.

Initial computations showed that the strain energy of the system is highly sensitive to the discretization of the boundary. If the projection of an intersection point from the level set 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_κ to allow the points to move to what their correct location should be given the enforcement of the rotations, as shown in Figure 6.4. Unless otherwise stated, the value for k_κ is set to 10^{-02}, and the value for r_κ is set set 2.4 h, where h is the element length scale of the background mesh.

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

$$\mathcal{U}^{\mathrm{C}} = \frac{1}{2}\left(\mathcal{U}^{\mathrm{L}} + \mathcal{U}^{\mathrm{R}}\right) \ , \tag{6.23}$$

where $\mathcal{U}$ denotes the strain energy, and $\mathcal{U}^{\mathrm{C}}$ 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.

Additionally, if the length of an element is small, i.e. $\left\|\Omega_e^\Gamma\right\| << 10^{-06}$, where Ω_e^Γ is an element on the discretized boundary, the element is merged with one of its adjacent elements to avoid the ill-conditioning of the system. This merging requires special attention such that the current node over which we are computing the curvature does not disappear. This process is illustrated in Figure 6.6. Furthermore, (6.23) is modified such that the strain energy is only added over the elements that were not merged.

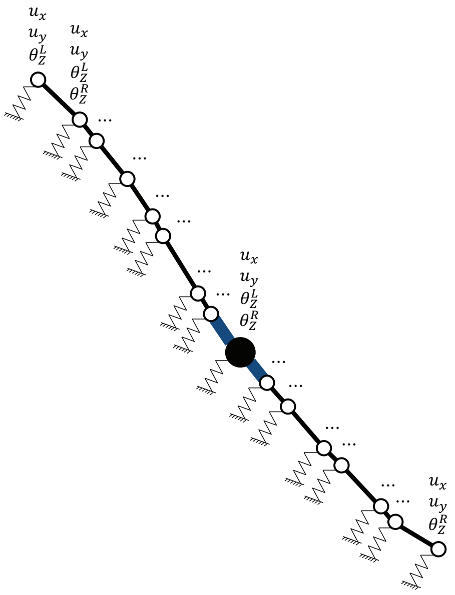


Figure 6.4: Discretization of the structural linear beam elements. Inner nodes have two separate rotational degrees-of-freedom, left and right, one for each adjacent element. Nodes at the end of the boundary have a single rotational degree-of-freedom. All nodes rest on a field of artificial springs. Strain energy is computed over the elements displayed in blue.

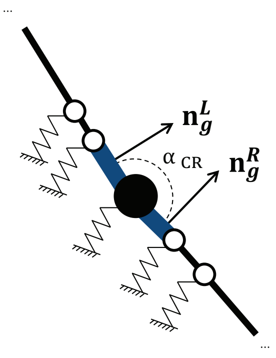
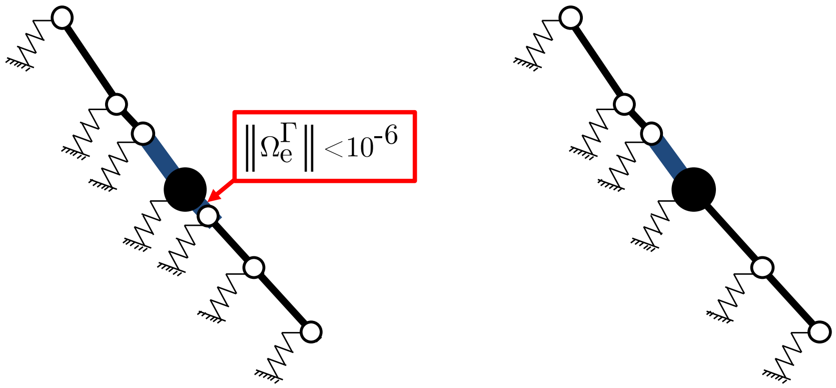


Figure 6.5: Computation of α_{CR}. α_{CR} is a function of the geometrical normal unit vectors of the adjacent elements.



(a) If the length of an element is small, the element is merged with one of its adjacent elements.

(b) The node of interest is not merged. The strain energy is only computed over the elements that were not merged, denoted in blue.

Figure 6.6: Merging process when an element on the boundary is small ($<< 10^{-06}$).

We will revisit these formulations in the numerical examples of Section 7.

6.2 Level Set Gradient Measure

LSM-based topology optimization requires regularization to control the size of geometric features. The smallest feature possible is limited to the size of an element if the magnitude of the gradient of the level set field at the interface is 1 and if the range of level set values in the design domain equals the size of the element. In this work, we utilize the measure of the LSF spatial gradients introduced in Coffin and Maute [2015b] to penalize the occurrence of features with a size smaller than the element size h, which is defined as:

$$\mathcal{G} = \int \mathrm{e}^{-{\alpha_\phi}^2} (|\nabla\phi| - 1)^2 \mathrm{d}\Omega \ , \tag{6.24}$$

with:

$$\alpha_\phi = e_p \frac{\phi}{\Delta\phi} \ , \tag{6.25}$$

where e_p is the penalization parameter, and $\Delta\phi$ is the range of level set values in the design domain defined as:

$$\Delta\phi = \phi_{\text{max}} - \phi_{\text{min}} \ . \tag{6.26}$$

The first term vanishes away from the zero level set, but conversely, becomes unity in its vicinity. The second term promotes a signed distance-like level set field (i.e. $|\nabla\phi| = 1$). The combination of these two terms penalizes the level set gradients that do not match the desired value of 1, but only in the locality of the fluid-solid interface. The measure is used in combination with a restriction on the lower and upper bounds of the optimization variables to half the element size, $s_i^L = -h/2$ and $s_i^U = h/2$, which by (6.26) defines $\Delta\phi$ as h. Using our example from Figure 4.1, we illustrate this process in Figure 6.7. Features smaller than the element size violate the prescribed gradient and are penalized by the measure. In accordance to the parameters used by Coffin and Maute [2015a], the value of e_p is set to 10. Note that the feature size control introduced in this section is bounded to the interpolation of the LSF within neighboring elements; therefore, it cannot be applied to control arbitrary feature sizes. For more details on the formulation, refer to Coffin and Maute [2015b].

6.3 Minimum Feature Size

In this thesis, we introduce a novel measure to control the minimum feature size of a design geometry in topology optimization problems. Controlling the minimum feature size of an optimized material layout is paramount to ensure that the governing equations can accurately model the physical response of the feature, and that the design can be manufactured. Our feature size control method relies on the explicit LSM (2.9) and on the XFEM (3.1), which preserves the crisp geometry definition of the LSM (including sharp corners). Several feature size control techniques for LSM-based topology optimization rely on preserving a signed distance field in the LSF; however, this requires a refined mesh to accurately represent the geometry. In constrast with an implicit LSM, an explicit LSM does not retain the signed distance behavior of the LSF, and would require a reconstruction of the field to recover it, which is computationally expensive. An example of a feature size control technique which does not depend on a signed distance field is the quadratic energy method of Chen et al [2008], which compares the tangent vectors on the interface at different points. Nevertheless, this approach is not effective when computing the measure on discretized surfaces. The feature size control measure developed here identifies violations of a minimum feature size, and does not require a signed distance field because it operates on the discretized polygon mesh of the level set interface, as provided by the XFEM. Single-material and two-material problems are considered, and the minimum feature size is prescribed either in a single phase or in both phases. The measure is incorporated either as a penalty on the objective or as a constraint in the optimization problem, and is able to deter the formation of features smaller than the prescribed size. The measure is demonstrated on structural and convective heat transfer topology optimization problems in Section 7. The influence of the parameters in our feature size formulation is also studied. A comparison of our formulation against the one from Chen et al [2008] is also provided.

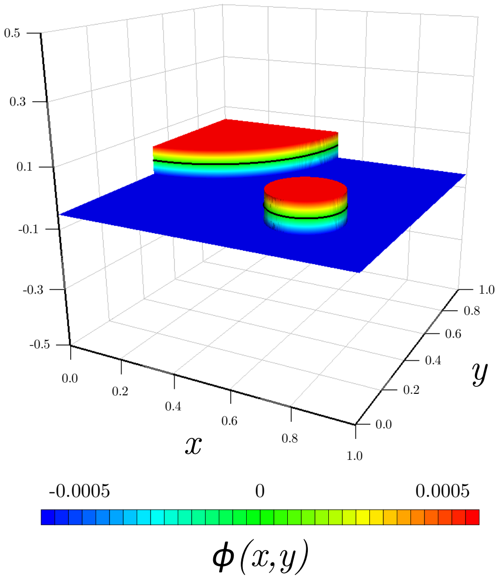


Figure 6.7: Level set field for the LSF of Figure 4.1. The field was plotted in a mesh with an h value of 0.001. The lower and upper limits are then set to ∓0.0005, respectively. Any level set value smaller or larger than the limits is cut from the function. Note that the values in the vicinity of the material interface (solid black line) are not modified.

6.3.1 Background

In recent decades, topology optimization has gained traction as a practical computational design technique. The method is appealing because, unlike shape optimization, it minimizes the influence of the initial design with its ability to produce both shape and topological changes in the optimization process. The control of the “feature size” or the “length scale” of the design is important to ensure its manufacturability and the accurate analysis of its functionality. Manufacturing methods have limitations on the length scales that they can accurately construct. Given these limitations, it may be necessary to incorporate controls on the feature size to ensure that designs are not overly difficult or costly to build. Furthermore, given a particular mesh, it may be necessary to control the feature size to ensure that the physical response can still be properly modeled. That is, the mesh used for the discretization must be sufficiently fine to accurately represent the physical response of the feature size in a given geometry. Adaptive mesh refinement methods are an alternative approach; however, as the feature size approaches zero, these techniques becomes impractical as well.

Feature size control techniques have been studied in both density-based and LSM-based topology optimization methods. Density-based topology optimization studied the limitations of manufacturing processes [Zhou et al, 2014], and used, for example, projection schemes [Guest et al, 2004], local density variations [Poulsen, 2003], robust design formulations [Schevenels et al, 2011], medial surface reconstruction [Zhang et al, 2014], and three-field-schemes [Zhou et al, 2015]. Multiple approaches have been proposed for feature size identification and control in LSM-based topology optimization. The approaches vary in the way the information is used, and may take advantage of the existing signed distance or near-signed distance behavior of the LSF to identify geometrical features. For example, Allaire et al [2014], Guo et al [2014] and Xia and Shi [2015] utilized the signed distance field to identify the skeleton of the geometry by computing the curvature of the function. The feature size measures are then incorporated into the optimization problem as either contributions in the objective function or as constraints. It is worth noting, however, that the maintenance and/or construction of a signed distance field increases the computational cost of these methods. Alternatively, the discretized phase interface itself can also be used to identify features. A quadratic energy function, which was defined as a double integral over the phase interface, was computed by Chen et al [2008] and Luo et al [2008b]. A fictitious interface energy derived from the phase field method was studied by Yamada et al [2010] We will study the formulation from Chen et al [2008] in more detail, and compare it against our own feature size measure in Section 7.

Here, we utilize the XFEM to represent the material distribution, and to discretize the governing equations in space (see Section 3). The XFEM has been shown to preserve the crispness of the phase interface as described by the LSF (2.2). The XFEM decomposes the elements cut by the zero level set into subdomains and interfaces that it uses to integrate the weak form of the governing equations. The decomposition of the element into triangular (in 2D) or tetrahedral (in 3D) subdomains yields a polygon mesh that describes the phase interface via triangle edges (in 2D) or tetrahedron faces (in 3D). The polygon mesh allows us to compute geometrical quantities along the phase interface, such as the feature size measure introduced in this document.

In this thesis, we develop a minimum feature size measure that directly utilizes the XFEM polygon mesh of the level set interface and its geometrical data. This approach bypasses the need for a signed distance field of the LSF to compute the geometrical quantities, or to enforce a minimum feature size. Additionally, the XFEM polygon mesh allows the measure to directly identify the geometry of the phase interface as it is incorporated into the analysis of the governing equations. The measure is constructed to identify violations of a prescribed minimum feature size, and is incorporated in the optimization problem as an inequality constraint. The use of the feature size measure as part of the optimization formulation gives the optimization algorithm the most freedom to find the optimal material layout, as compared against methods that incorporate the measure as part of the parameterization of the geometry [Coffin and Maute, 2015b].

6.3.2 Feature Size Measure

In this work, we develop a novel minimum feature size measure for topology optimization problems. The measure is computed over the polygon mesh generated by the XFEM discretization of the level set interface. We seek to compute a scalar value that identifies whether or not a minimum feature size is violated. Using the geometry of our polygon mesh will ensure consistency between the XFEM analysis of the governing equations and the computation of the feature size measure. Our goal is to identify small features with a simple and clear formulation. The approach detailed here is closely related to the work of Chen et al [2008], who used a quadratic energy approach to identify small features in the material layout. We will provide a detailed comparison of both methods in Section 7.

Similar to Chen et al [2008], the feature size measure, denoted as $\mathcal{M}^0$, is a double integral over the level set interface, and the integrand is a product of two Heaviside functions, written as:

$$\mathcal{M}^0 = \int_{\Gamma_2^0} \int_{\Gamma_1^0} \hat{H}_{tx} \left(\phi_{tx} \left(\boldsymbol{s} \right) \right) \cdot \hat{H}_x \left(\phi_x \left(\boldsymbol{s} \right) \right) \, \mathrm{d}\Gamma \mathrm{d}\Gamma \ , \tag{6.27}$$

where:

$$\phi_x \left(\boldsymbol{s} \right) = r_x - \left| \boldsymbol{x}_{12}^{\Gamma} \left(\boldsymbol{s} \right) \right| \ , \tag{6.28}$$

$$\phi_{tx} \left(\boldsymbol{s} \right) = \left| \boldsymbol{t}_{12}^{\Gamma} \left(\boldsymbol{s} \right) \right| \ / \ \left| \boldsymbol{x}_{12}^{\Gamma} \left(\boldsymbol{s} \right) \right| - r_{tx} \ , \tag{6.29}$$

and where Γ_1^0 and Γ_2^0 are the first and second integrals over the level set interface, respectively, $\boldsymbol{x}_{12}^{\Gamma} \left(\boldsymbol{s} \right)$ represents the Euclidean path between intersection points $\boldsymbol{x}_1^{\Gamma}$ and $\boldsymbol{x}_2^{\Gamma}$, and is defined as:

$$\boldsymbol{x}_{12}^{\Gamma} \left(\boldsymbol{s} \right) = \boldsymbol{x}_1^{\Gamma} \left(\boldsymbol{s} \right) - \boldsymbol{x}_2^{\Gamma} \left(\boldsymbol{s} \right) \ , \tag{6.30}$$

and $\boldsymbol{t}_{12}^{\Gamma} \left(\boldsymbol{s} \right)$ is the topological or geodesic path between points $\boldsymbol{x}_1^{\Gamma}$ and $\boldsymbol{x}_2^{\Gamma}$ along the level set interface. Both distances are functions of the vector of design variables, $\boldsymbol{s}$. A graphical representation of these quantities is illustrated in Figure 6.8.

The Heaviside function, $\hat{H}_{\zeta} \left(a \right)$, is modeled as a smooth Heaviside to ensure the differentiability of the measure with respect to the position of the interface points, and is defined as:

$$\hat{H}_\zeta(a) = \begin{cases} a \leq -w_\zeta & 0 \ , \\ -w_\zeta < a < +w_\zeta & \frac{1}{2} + \frac{a}{w_\zeta}\left(\frac{15}{16} - \frac{a^2}{w_\zeta^2}\left(\frac{5}{8} - \frac{3}{16}\frac{a^2}{w_\zeta^2}\right)\right) \ , \\ +w_\zeta \leq a & 1 \ . \end{cases} \tag{6.31}$$

The w_ζ parameters are chosen as functions of the feature size, and the value for w_x is set to $r_x/2$.

The optimization problem requires that we compute the sensitivities of the feature size measure with respect to the coordinates of the points at which the level set interface cuts the design domain (see Coffin and Maute [2015a]). Computing the sensitivities of the first Heaviside function is rather complex and involved due to the formulation of the geodesic distance, which is dependent on different coordinates across the design domain. To circumvent this issue, we choose a small value for w_{tx}, and set the parameter equal to $r_x/10$. This approach effectively turns the derivative into a binary on-off switch due to the nature of the Heaviside projection, and eases the computational complexity.

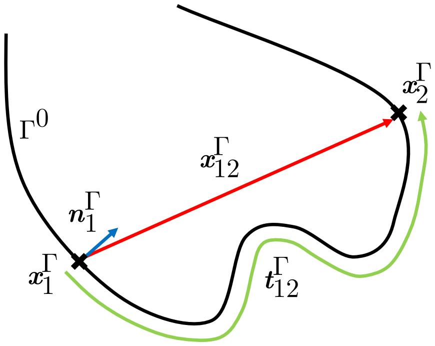

Γ^0
$\boldsymbol{x}_2^\Gamma$
$\boldsymbol{x}_{12}^\Gamma$
$\boldsymbol{n}_1^\Gamma$
$\boldsymbol{x}_1^\Gamma$
$\boldsymbol{t}_{12}^\Gamma$

Figure 6.8: Description of the Euclidean and geodesic paths used in the feature size measure.

The first Heaviside identifies points that are far in geodesic distance. The parameter r_{tx} is initially chosen to be:

$$r_{tx} = \sqrt{2} \ . \tag{6.32}$$

This value is the relative distance of opposite equidistant points around a 90° corner, as shown in Figure 6.9. The impact of this parameter will be discussed and studied later in the numerical examples. The second Heaviside function identifies points that are close in Euclidean distance, violating the minimum feature size r_x.

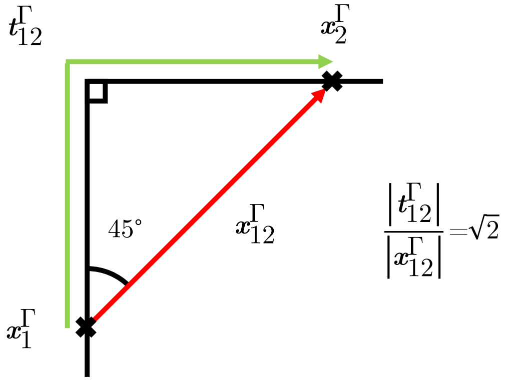

$\boldsymbol{t}_{12}^{\Gamma}$
$\boldsymbol{x}_{2}^{\Gamma}$
45°
$\boldsymbol{x}_{12}^{\Gamma}$
$\frac{\left|\boldsymbol{t}_{12}^{\Gamma}\right|}{\left|\boldsymbol{x}_{12}^{\Gamma}\right|}=\sqrt{2}$
$\boldsymbol{x}_{1}^{\Gamma}$

Figure 6.9: Initial estimate for an appropriate r_{tx} value with points equidistant about a right-angle corner.

The feature size measure in (6.27) does not distinguish between features of different phases. To allow the measure to identify features formed by a particular phase, we extend the measure with a third Heaviside function and denote it as $\mathcal{M}^m$, where $m = \{-, +\}$ and represents the material phase. The new measure is then defined as:

$$\mathcal{M}^m = \int_{\Gamma_2^0} \int_{\Gamma_1^0} \hat{H}_{tx}\left(\phi_{tx}\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_{x}\left(\phi_{x}\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_{n}\left(\phi_{n}\left(\boldsymbol{s}\right)\right) \, \mathrm{d}\Gamma \mathrm{d}\Gamma \ , \tag{6.33}$$

where:

$$\phi_n\left(\boldsymbol{s}\right) = \operatorname{sign}(m) \cdot \left(\boldsymbol{x}_{12}^{\Gamma}\left(\boldsymbol{s}\right) \cdot \boldsymbol{n}_{1}^{\Gamma}\left(\boldsymbol{s}\right)\right) \ . \tag{6.34}$$

The additional Heaviside function, $\hat{H}_n$, allows the measure to identify features that are formed by a phase. This identification is performed by comparing the Euclidean path between the two points to the interface normal at the first point. The normal is constructed such that it points from the negative phase into the positive one. If the dot product of the normal $\boldsymbol{n}_1^{\Gamma}$ and the vector $\boldsymbol{x}_{12}^{\Gamma}$ is positive, then the feature is identified as being formed by the positive phase. If the value is negative, the feature is formed by the negative phase. The parameter w_s, as described in (6.31), is chosen in the same manner as w_{tx}, and the value is set to $r_x/10$.

Additional Heaviside functions are introduced to study the relationship and influence of the different parameters. Here, an additional Heaviside is introduced to study the relationship between the geodesic path (6.29) and the r_x parameter, and to smooth the response of the minimum feature size measure. This formulation, $\mathcal{M}_a^m$, extends (6.33) as:

$$\mathcal{M}_a^m = \int_{\Gamma_2^0} \int_{\Gamma_1^0} \hat{H}_{tx}\left(\phi_{tx}\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_x\left(\phi_x\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_a\left(\phi_a\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_n\left(\phi_n\left(\boldsymbol{s}\right)\right) \left|\phi_x\left(\boldsymbol{s}\right)\right|^2 \mathrm{d}\Gamma \mathrm{d}\Gamma \ , \tag{6.35}$$

where:

$$\phi_a\left(\boldsymbol{s}\right) = \left|\boldsymbol{t}_{12}^{\Gamma}\left(\boldsymbol{s}\right)\right| - r_a \cdot r_x \ , \tag{6.36}$$

where r_a is a scaling factor.

The measure in (6.33) is modified to study the effect of the Heaviside in (6.28), and is defined as:

$$\mathcal{M}_x^m = \int_{\Gamma_2^0} \int_{\Gamma_1^0} \hat{H}_{tx}\left(\phi_{tx}\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_x\left(\phi_x\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_n\left(\phi_n\left(\boldsymbol{s}\right)\right) \left|\phi_x\left(\boldsymbol{s}\right)\right|^2 \mathrm{d}\Gamma \mathrm{d}\Gamma \ , \tag{6.37}$$

where the values for w_x, w_{tx}, and w_n are chosen to be 0.001.

The term ϕ_{tx} in (6.37) is modified to study the influence of this parameter, and the new measure is defined as:

$$\mathcal{M}_{tx}^m = \int_{\Gamma_2^0} \int_{\Gamma_1^0} \hat{H}_{tx}\left(\phi_{txx}\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_x\left(\phi_x\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_n\left(\phi_n\left(\boldsymbol{s}\right)\right) \left|\phi_x\left(\boldsymbol{s}\right)\right|^2 \mathrm{d}\Gamma \mathrm{d}\Gamma \ , \tag{6.38}$$

where ϕ_{txx} is defined as:

$$\phi_{txx}\left(\boldsymbol{s}\right) = \left|\boldsymbol{t}_{12}^{\Gamma}\left(\boldsymbol{s}\right)\right| \ / \ r_x - r_{tx} \ . \tag{6.39}$$

In order to identify points that are in separate surfaces, but not penalize curvature, (6.37) is modified as:

$$\mathcal{M}_w^m = \int_{\Gamma_2^0} \int_{\Gamma_1^0} \hat{H}_{tx}\left(\phi_w\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_x\left(\phi_x\left(\boldsymbol{s}\right)\right) \cdot \hat{H}_n\left(\phi_n\left(\boldsymbol{s}\right)\right) \left|\phi_x\left(\boldsymbol{s}\right)\right|^2 \mathrm{d}\Gamma \mathrm{d}\Gamma \ , \tag{6.40}$$

where ϕ_w is defined as:

$$\phi_w\left(\boldsymbol{s}\right) = \left|\boldsymbol{t}_{12}^{\Gamma}\left(\boldsymbol{s}\right)\right| - r_x \cdot r_{tx} + w_{tx} \ . \tag{6.41}$$

The measures in (6.27) and (6.33) will be incorporated into the optimization problem as an inequality constraint. We will study two variations of the constraint, one where the feature size measure is normalized by the perimeter and one where the measure is included directly. The feature size measure, $\mathcal{M}$, will either be computed over both phases (6.27), or over the negative phase (6.33). The normalized constraint, $g^m_{\bar{\mathcal{M}}}$, is defined as:

$$g^m_{\bar{\mathcal{M}}} = \frac{\mathcal{M}^m}{\mathcal{S}^2} - c_{\bar{\mathcal{M}}} \leq 0 \; , \tag{6.42}$$

where the feature size measure, $\mathcal{M}$, is normalized with the square of the perimeter (in 2D) or surface area (in 3D) of the level set interface, denoted as $\mathcal{S}$, and bound by some small value, $c_{\bar{\mathcal{M}}}$. The smaller $c_{\bar{\mathcal{M}}}$, the more strongly the constraint is enforced. This form is most useful in problems where the perimeter is constrained. However, this approach may lead to geometric features that only act to increase the design's perimeter. The increase in perimeter allows for larger feature size violations in the normalized form of the constraint (6.42). In such cases, the unnormalized form of the constraint might be more benefitial in order to bypass this difficulty. The constraint is written as:

$$g^m_{\mathcal{M}} = \mathcal{M}^m / c_{\mathcal{M}} - 1 \leq 0 \; , \tag{6.43}$$

where $c_{\mathcal{M}}$ is an allowable violation. We will study both constraints in the numerical examples of Section 7.

6.3.3 Implementation Details

In a parallel computation, each processor possesses a partition of the computational mesh, i.e. the analysis domain. To compute certain measures, such as the curvature of the level set field or the integral introduced in (6.27), we require information about the level set interface across the entire computational domain, not the partition alone. This information is mainly comprised of the coordinates of the points that form the level set interface, and their connectivity. The following steps are performed in the `femdoc` code to gather this information:

(1) Extract the level set interface topology and intersection points. Each processor performs this task locally. The interface topology is computed by looping over all elements cut by the zero level set isosurface. The intersection points are copied from the `Mesh` class. We will refer to this set of data as the XFEM surface mesh.

(2) Communicate the interface topology and intersection points to the root processor. In this case, the root processor is the processor with rank 0. All other (lower-ranked) processors send their local surface mesh information to processor 0.

(3) The root processor sorts over the information and removes duplicate intersection points. Given that the partitioning of the original computational domain is element-based, there should not be duplicate interface topologies. The interface topologies of each processor are assembled one after the other into a single matrix.

(4) The root processor returns the sorted and assembled information to all processors. At this point, each processor has a copy of the entire surface mesh.

(5) All processors proceed to compute the measure of interest and its partial derivative with respect to the intersection points. However, if there is a loop over the surface elements, the processors only compute the information over the elements that they originally owned.

(6) All processors compute the derivative of the intersection points with respect to the design variables, and these values get post-multiplied by the sensitivities of the measure of interest.

In a previous implementation, all processors would send the surface mesh information to the root processor, and the root processor would proceed to compute the measure by itself. This proved to be extremely computationally expensive in 3D. By having each processor possess a copy of the surface mesh, the computation time of the measure scaled by the number of processors. However, note that this is a trade-off between memory and speed. In this situation, we chose to sacrifice memory by creating multiple copies of the same surface mesh across the processors, in order to gain speed and decrease the computational time.

Also note that the processors only communicate with the root processor, and viceversa. The processors do not communicate with one another. This reduces the cost of the communications from $O\left(n^2\right)$ to $O\left(n\right)$, where n is the number of processors.

The geodesic distance is computed using the Surface Mesh Shortest Path module introduced in CGAL 4.7 [CGAL, 2009]. For a tutorial on how to use the module, see Appendix B.

The XFEM surface mesh is represented in CGAL through the `Polyhedron_3` class. This class has exhibited a lack of robustness when the level set value at a node is small and leads to a small intersection area. The class fails to recognize the surface mesh as a valid polyhedron, which means we cannot proceed to compute the geodesic distances. Using larger level set critical and shift values, ϕ_c^Γ and ϕ_s^Γ, respectively, has shown to ameliorate this issue (see Section 4.6). However, the effect that using larger values for ϕ_c^Γ and ϕ_s^Γ has on the sensitivities needs to be studied further. The issue was lastly resolved by removing very small elements from the CGAL surface mesh with a clause that was added to the class from Appendix B. If CGAL fails to add a surface element to the surface mesh, we assume it is because the element is poorly scaled due to the level set value at a node being near zero. In such a case, we ignore the element and it is not added to the CGAL polyhedron mesh.

Chapter 7

Numerical Examples

In the following, we study the characteristics of the proposed LSM-XFEM topology optimization framework for 2D and 3D, linear diffusion, linear elasticity, steady-state and transient laminar incompressible flow, and energy and species transport problems. We apply topology optimization to several problems in order to study the characteristics of the framework with respect to different physical phenomena. The 2D and 3D analysis domains are discretized in space by bilinear quadrilateral elements and trilinear hexahedral elements, respectively. Unless otherwise stated, geometric and material parameters are given in non-dimensional and self-consistent units.

The optimization problems are solved via the Globally Convergent Method of Moving Asymptotes (GCMMA) of Svanberg [1995]. The GCMMA parameters are given in Table 7.1. The optimization problem is considered converged if: (i) the change of the objective function relative to the initial objective value is less than 10^{-06}, and (ii) all constraints are satisfied.

7.1 Sensitivity Analysis

In this section, we present 4 numerical examples to study the characteristics and accuracy of the shape sensitivities of the Heaviside-enriched XFEM framework. Examples 7.1.1, 7.1.2, and 7.1.3 use a 2D steady-state linear diffusion model (4.29) to study the behavior of the shape sensitivities for various types of evolving shapes. Example 7.1.4 studies an incompressible fluid flow (4.17), and showcases the applicability of the proposed framework to a nonlinear system in 3D. We show that the smoothness of the shape sensitivities is primarily driven by the error in the discretization of

	Value
Relative step size	0.04
Minimum asymptote adaptivity	0.5
Initial asymptote adaptivity	0.7
Maximum adaptivity	1.43
Constraint penalty	100

Table 7.1: GCMMA parameters for the topology optimization problems.

the interface geometry and not by the approach in computing the shape sensitivities. As a result, smoother shape sensitivities are obtained with mesh refinement.

Boundary conditions are imposed weakly using Nitsche's method. Unless mentioned otherwise, an interface constraint penalty of $\alpha_{\mathrm{N},T} = 1.0$ is used to enforce continuity in the temperature field across the material interface in all the steady-state linear diffusion examples. For the incompressible fluid flow problem, we set the interface constraint penalty to $\alpha_{\mathrm{N},\boldsymbol{u}} = 1000$ to enforce a no-slip condition at the fluid-solid interface. To mitigate issues with ill-conditioning caused by an extremely small (or large) ratio of phase volumes within an element cut by the zero level set, we use the geometric preconditioning scheme of Lang et al [2014]. The linear problems of both the forward and the sensitivity analyses are solved using a direct solver.

7.1.1 Bar with a Strip Inclusion

This example is divided into three subsections. The first subsection establishes the accuracy of the proposed semi-analytical approach (4.66) by comparing it against finite differenced shape sensitivities computed using a body-fitted mesh. The second subsection discusses the influence of shifting the interface (c.f. Section 3.2) on the semi-analytically computed shape sensitivities (4.66), and a comparison is provided against finite differencing $\mathrm{d}\mathcal{Z}/\mathrm{d}s_i$ with a perturbation size of $\Delta\phi_{FD}$. The third subsection presents the shape sensitivities corresponding to a response function evaluated

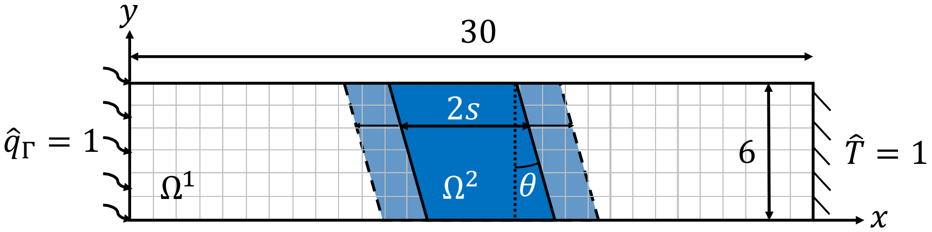


Figure 7.1: Problem setup for the linear diffusion example with a strip inclusion.

at a region that changes its material phase.

We consider the two-phase problem setup shown in Figure 7.1. A strip inclusion of conductivity $k_2 = 10$, is inclined at an angle θ to the vertical axis. This inclusion is embedded into a rectangular matrix of conductivity $k_1 = 1$. The strip is centered along $x = 15$. A Neumann boundary condition of $\hat{q}_\Gamma = 1$ is applied along the left edge. A Dirichlet boundary condition of $\hat{T} = 1$ is applied to the right edge. The top, right, and bottom edges are assumed to be adiabatic. The design variable, s, is half the width of the strip inclusion, measured parallel to the horizontal axis.

7.1.1.1 Accuracy comparison of shape sensitivities

Here we study the accuracy of the shape sensitivities using the proposed semi-analytical approach with respect to the shape sensitivities obtained using a body-fitted mesh. We use the problem setup of Figure 7.1 with $\theta = 30°$. For the body-fitted approach, a standard finite element mesh, of 120 by 24 elements, is body-fitted with respect to the strip inclusion for every s and the corresponding design geometry perturbations due to $\Delta\phi_{\text{FD}}$. The former is used to compute the response function while the latter is used to compute the finite differenced body-fitted shape sensitivities. Figure 7.2 presents the response function and the shape sensitivities as a function of the design variable. Even though the response plots match closely, the shape sensitivities using the XFEM are not as smooth as those obtained using the body-fitted mesh. Mesh refinement leads to convergence of the sensitivities computed using the XFEM. It is also noted that the interface constraint penalty (4.37) influences the accuracy of the sensitivities. Table 7.2 presents the

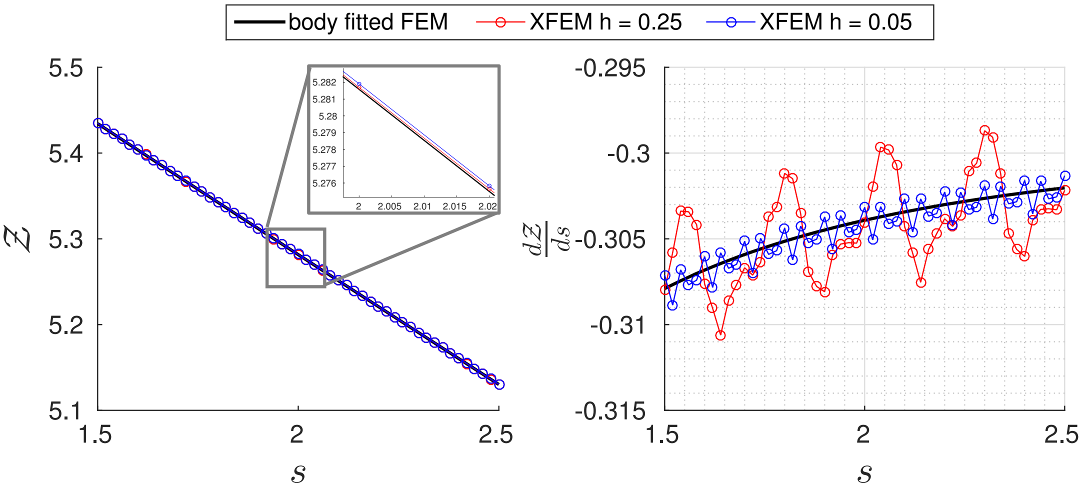


Figure 7.2: Comparison with body-fitted mesh, $\theta = 30^0$.

difference in shape sensitivities using the XFEM with respect to the body-fitted shape sensitivities. This relative difference is averaged over the number of designs corresponding to the different values of the design variable s. Also presented in Table 7.2 is the interface constraint error averaged over the number of different designs. Note that a higher interface constraint penalty does not guarantee a more accurate behavior of shape sensitivities.

7.1.1.2 Influence of interface shift on semi-analytical shape sensitivites

In this subsection, we discuss the influence of the material interface shift described on the computation of the semi-analytical shape sensitivities. As described in Section 3.2, the material interface is not allowed to intersect the node, and we shift the interface if it comes within a critical distance of a node. Assuming a signed distance field (i.e. $\partial \boldsymbol{x}_k^\Gamma / \partial \phi_j = 1$), the critical shift region is defined as $x_c^\Gamma = \phi_c^\Gamma = 10^{-06} h$, where a change in the level set field results in an identical change in the coordinates of the intersection points. Using a mesh of size $h = 1$, and setting $\theta = 0°$, the design variable is varied in increments of $\Delta s = 0.15 x_c^\Gamma$. The response function, $\mathcal{Z}$, is the temperature measured at the node located at $\boldsymbol{x} = (0, 0)$. This setup allows us to study the behavior of the shape sensitivities within the critical region as the material interface crosses a node. Figure 7.3 plots

Table 7.2: Influence of interface constraint penalty.

h	$\alpha_{\mathrm{N},T}$	Avg. difference w.r.t. FEM	Avg. interface constraint error
0.25	1.0	0.69%	1.00×10^{-02}
0.25	10.0	0.40%	4.26×10^{-03}
0.25	100.0	0.65%	2.56×10^{-03}
0.05	1.0	0.23%	3.57×10^{-03}

the response function for each of the shift options listed in Section 3.2. Note that the interface intersects a node at $s = 2$. The interface shift is performed such that the new position of the material interface is assumed at a distance x_c^Γ from the node in concern. This results in a distinct discontinuity in the response function within the critical shift region.

Figure 7.4 presents a comparison of shape sensitivities computed using the proposed semi-analytical approach against central, forward, and backward finite differenced shape sensitivities with $\theta = 0°$. A perturbation size of $\Delta\phi_{FD} = 2x_c^\Gamma$ was used. The analytical solution establishes that the shape sensitivities must remain constant with the varying thickness of a vertical strip inclusion because the temperature response is linear. However, as can be noticed in Figure 7.4, inaccurate shape sensitivities are obtained for certain values of the design variable s. The inaccuracy in the semi-analytical and finite difference plots is a result of performing an interface shift during the finite differencing of $\partial \boldsymbol{R}_e / \partial \boldsymbol{x}_k^\Gamma$ and $\partial \mathcal{Z} / \partial s$ respectively. In contrast, when the interface position is not shifted during the sensitivity computation, accurate sensitivities are obtained using the semi-analytical approach as shown in Figure 7.5. The finite differenced sensitivities are still inaccurate in the vicinity of the critical shift region. This is because they are obtained by finite differencing the response function, and the interface shift is always performed within the critical shift region during its computation. When computing shape sensitivities via the semi-analytical approach, design perturbations were performed about the interface configuration used to compute the response.

While finite differencing $\partial \boldsymbol{R}_e / \partial \boldsymbol{x}_k^\Gamma$, design perturbations may result in the material interface

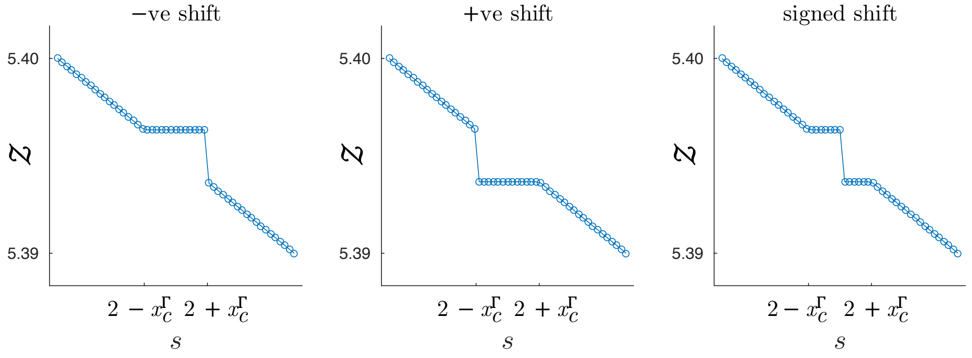


Figure 7.3: Influence of shifting the interface on the response function.

crossing over to a neighboring element. In such a scenario, a forward or backward finite differencing of $\partial \boldsymbol{R}_e/\partial \boldsymbol{x}_k^\Gamma$ is performed depending on the interface configuration. To avoid running into such issues, the finite difference perturbation size, $\Delta\phi_{\mathrm{FD}}$, is chosen to be less than the size of the critical shift region. Such a choice allows for central finite differencing at all times regardless of the interface position. For all results presented from hereon, the −ve shift approach with a critical shift region of $x_c^\Gamma = 10^{-06}h$ was used alongside a finite difference perturbation size of $\Delta\phi_{\mathrm{FD}} = 5 \times 10^{-07}h$.

7.1.1.3 Heaviside-induced strong discontinuity in shape sensitivities

In this subsection, we present shape sensitivities corresponding to a response function evaluated at a node which changes material phase with change in design. We use the problem setup of Figure 7.1 with $\theta = 30^0$. Shape sensitivities are recorded for every configuration as the design variable is varied in constant increments of $\Delta s = 0.02$. The response function, $\mathcal{Z}$, is the temperature measured at the node located at $\boldsymbol{x} = (14.5, 3)$. Figure 7.6 presents the response function and the corresponding shape sensitivities as a function of the design variable. There is a kink in the variation of the response function leading to a strong discontinuity in the shape sensitivities. This discontinuity corresponds to the the design configuration when the node changes material phases from 1 to 2.

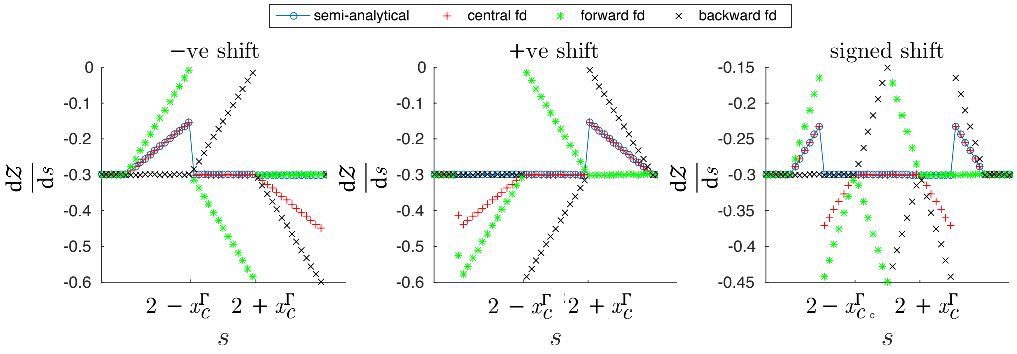


Figure 7.4: Influence of shifting the interface during computation of shape sensitivities.

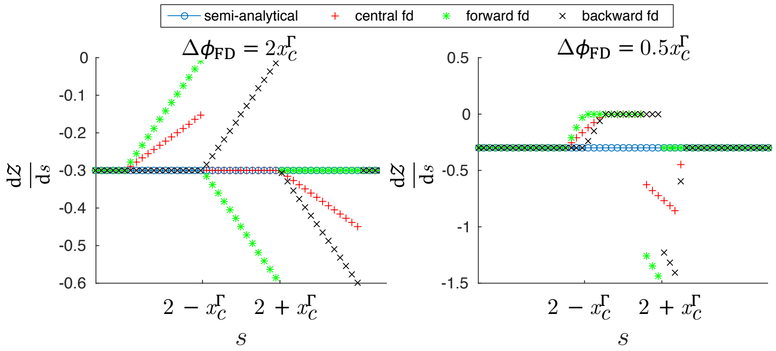


Figure 7.5: Finite differencing $\partial \boldsymbol{R}_e / \partial \boldsymbol{x}_k^\Gamma$ without interface shift, −ve shift.

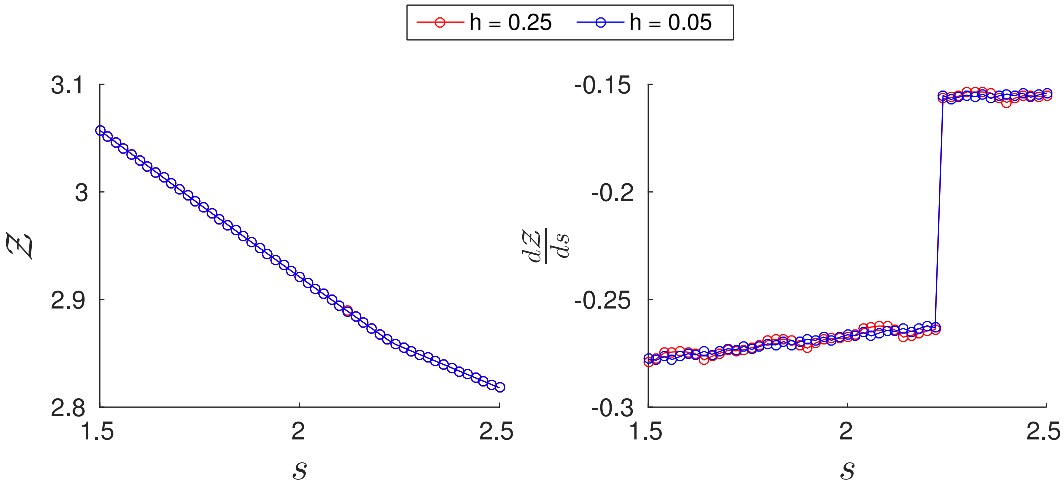


Figure 7.6: Shape sensitivities for response function evaluated at $\boldsymbol{x} = (14.5, 3)$.

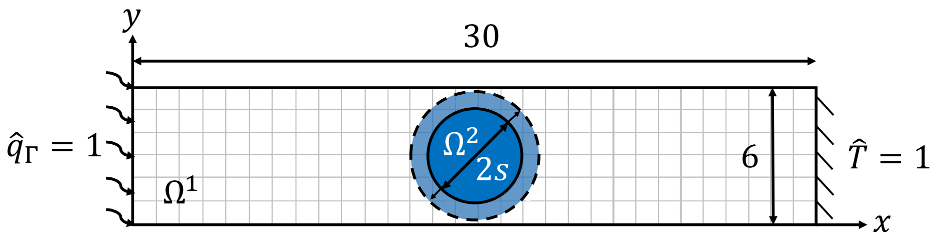


Figure 7.7: Problem setup for the linear diffusion example with a circular inclusion.

7.1.2 Bar with a Circular Inclusion

The goal of this example is to investigate the dependency of the shape sensitivities on the discretization of the design geometry by evaluating the perimeter of the discretized inclusion. We consider the two-phase problem setup shown in Figure 7.7. A circular inclusion of conductivity $k_2 = 10$, is embedded into a rectangular matrix of conductivity $k_1 = 1$. The circular inclusion is centered at $\boldsymbol{x} = (15, 3)$. A Neumann boundary condition of $\hat{q}_\Gamma = 1$ is applied to the left edge. A Dirichlet boundary condition of $\hat{T} = 1$ is applied to the right edge. The top, right, and bottom edges are assumed to be adiabatic. The design variable, s, is the radius of the circular inclusion. To compute the shape sensitivities, the design variable is varied in constant increments of $\Delta s = 0.02$. The response function, $\mathcal{Z}$, is the temperature measured at the node located at $\boldsymbol{x} = (0, 0)$. Alongside the sensitivities of the response function, we also focus on the sensitivities of the perimeter of the inclusion. The circular inclusion has a perimeter of $2\pi s$, and consequently, the sensitivity of the perimeter is expected to be 2π.

The response function and the corresponding shape sensitivities for a mesh size of $h = 0.25$ are plotted in Figure 7.8. A visibly smooth, but not linear, response function is obtained. However, the shape sensitivities obtained are not smooth. Plotted alongside the sensitivities for the response function are the sensitivities of the perimeter of the circular inclusion. A strong correlation can be observed in the behavior of the two plots, implying the influence of the discretization of the design geometry on the computation of the shape sensitivities. Consequently, a comparison with a finer mesh is drawn in Figure 7.9. With spatial refinement, the sensitivities for the inclusion

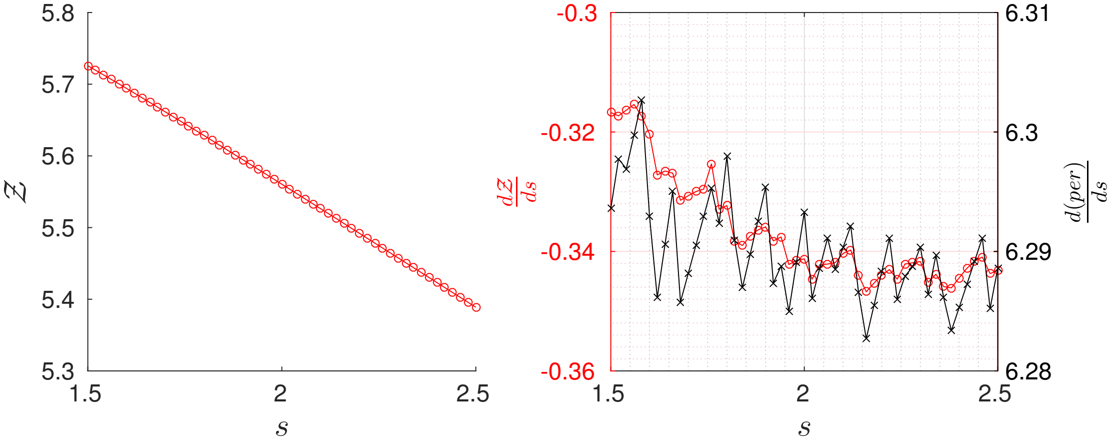


Figure 7.8: Response function and shape sensitivities for the circular inclusion example, $h = 0.25$.

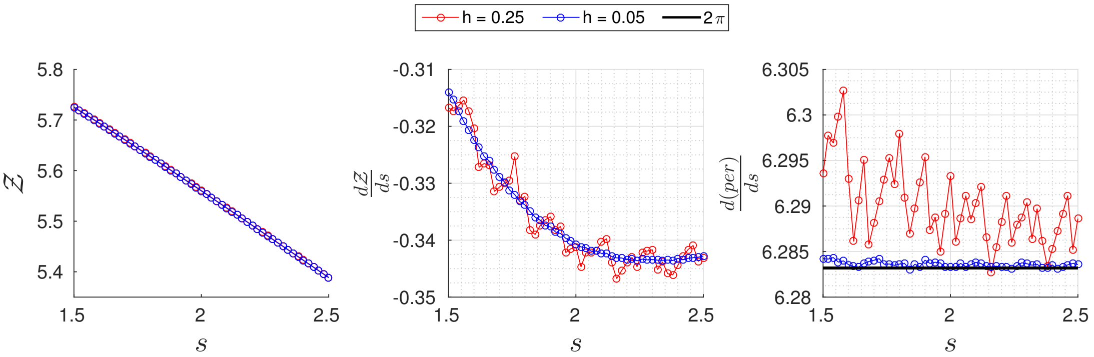


Figure 7.9: Smoother shape sensitivities with mesh refinement.

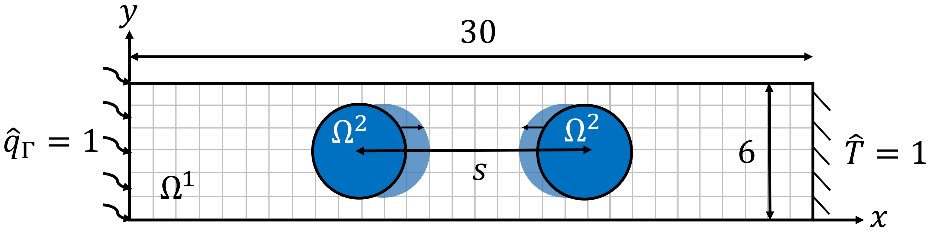


Figure 7.10: Problem setup for the linear diffusion example with two circular inclusions.

interface perimeter approach the constant value of 2π indicating a much smoother discretization of the interface geometry. This smooth discretization of the interface geometry in turn leads to a smoother behavior of the shape sensitivities.

7.1.3 Bar with Two Moving Circular Inclusions

This example investigates the evolution of the response function and the corresponding shape sensitivities for a scenario which simulates the merging of shapes in topology optimization. As a result, we consider a two inclusion problem setup as shown in Figure 7.10. Two circular inclusions of conductivity $k_2 = 10$, are embedded into a rectangular matrix of conductivity $k_1 = 1$. The circular inclusions have a radius of 2. The left and right inclusions are initially centered at $\boldsymbol{x} = (12, 3)$ and $\boldsymbol{x} = (18, 3)$ respectively. A Neumann boundary condition of $\hat{q}_\Gamma = 1$ is applied to the left edge. A Dirichlet boundary condition of $\hat{T} = 1$ is applied to the right edge. The top, right, and bottom edges are assumed to be adiabatic. The design variable, s, is the distance between the centers of the two circular inclusions. To compute the shape sensitivities, the design variable is varied in constant increments of $\Delta s = -0.025$. The response function, $\mathcal{Z}$, is the temperature measured at the node located at $\boldsymbol{x} = (0, 0)$.

The response function and the corresponding shape sensitivities for mesh sizes of $h = 0.25$ and $h = 0.05$ are plotted in Figure 7.11. Because $\Delta\phi_{FD} < \phi_c^\Gamma$, the inclusions never merge during the shape sensitivity analysis. The same occurs during a traditional topology optimization problem. Therefore, from the perspective of shape sensitivity analysis, we actually do not take into account the effect of the topological change. Moreover, the merging of topological features is not a continu-

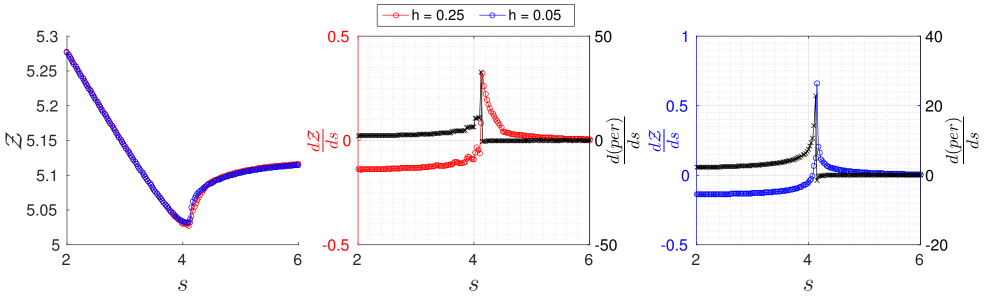


Figure 7.11: Response function and shape sensitivities for merging topologies.

ous phenomenon as can be observed by the evolution of the response function in Figure 7.12. The jump in the response function leads to a spike in the shape sensitivities when the two inclusions merge. Figure 7.12 shows the convergence of the response function and the corresponding shape sensitivities with spatial mesh refinement, in the region where the inclusions merge.

7.1.4 Steady-state Laminar Flow Around a Sphere

In contrast to the previous examples, the current problem setup is governed by a nonlinear set of equations in 3D for modeling incompressible fluid flow (4.17). The goal here is to show the applicability of the numerical framework discussed in the current study to nonlinear problems in 3D. We consider a solid spherical inclusion surrounded by an incompressible fluid in a rectangular box as depicted in Figure 7.13. A flow of with Reynolds number $Re = 10$ is considered. The fluid density and viscosity are assumed to be 1. The inclusion has an initial radius of 0.5, and is centered at $\boldsymbol{x} = (2, 2, 2)$. No-slip conditions are enforced along the fluid-solid interface. A parabolic laminar flow is developed at the inlet. Due to the uniform cross sections of the inlet and outlet, the pressure loss inside the channel depends solely on the geometry of the inclusion. Consequently, shape sensitivities are computed for the response function, $\mathcal{Z}$, defined as the difference in total pressure measured between the inlet and outlet, with the design variable, s, being the radius of the spherical inclusion. The radius is varied in constant increments of $\Delta s = 0.01$. The response function and the corresponding shape sensitivities for $h = 0.1$ are plotted in Figure 7.14. Smooth

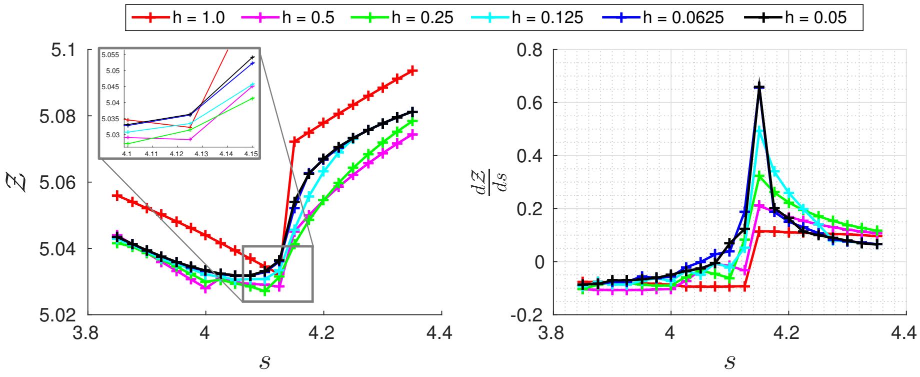


Figure 7.12: Convergence of the solution with spatial mesh refinement.

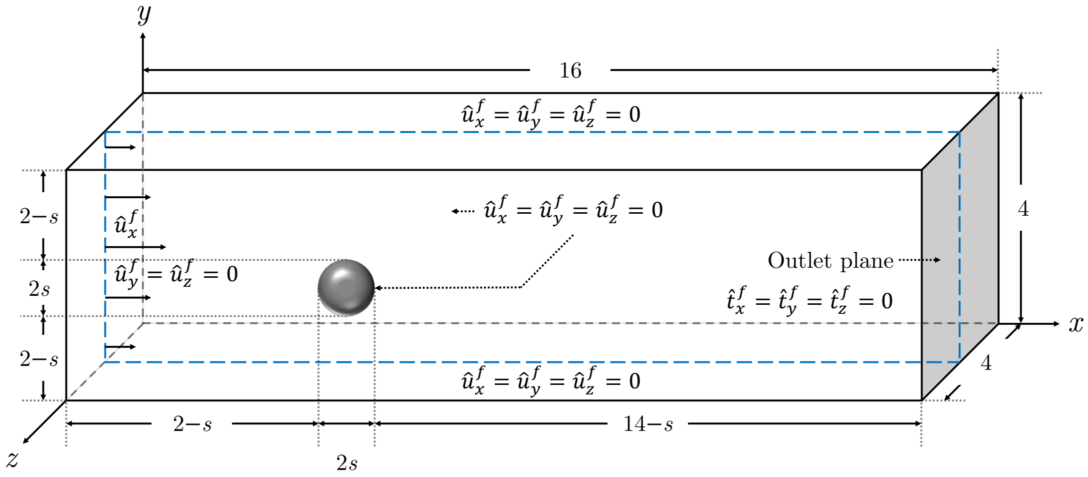

y
16
$\hat{u}_x^f = \hat{u}_y^f = \hat{u}_z^f = 0$
$2-s$
$\hat{u}_x^f$
$\hat{u}_y^f = \hat{u}_z^f = 0$
$2s$
$2-s$
$\hat{u}_x^f = \hat{u}_y^f = \hat{u}_z^f = 0$
Outlet plane
$\hat{t}_x^f = \hat{t}_y^f = \hat{t}_z^f = 0$
4
x
4
$\hat{u}_x^f = \hat{u}_y^f = \hat{u}_z^f = 0$
z
$2-s$
$2s$
$14-s$

Figure 7.13: Incompressible fluid flow with void spherical inclusion.

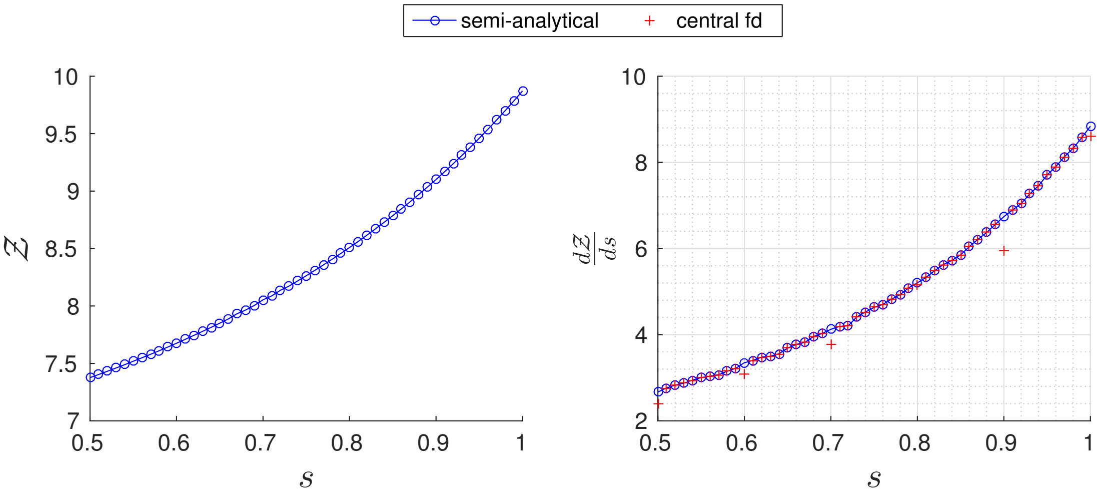


Figure 7.14: Response function and shape sensitivities for problem setup in Figure 7.13.

behavior of semi-analytical shape sensitivities is observed. Plotted alongside the semi-analytical sensitivities are the finite differenced shape sensitivities. As reported in Section 7.1.1.2, the semi-analytical sensitivities are in agreement with the finite differenced sensitivities except for when the material interface lies within the critical shift region.

7.2 Geometric Preconditioner

In this chapter, we study and validate our geometric preconditioner formulation with a 2D linear diffusion problem. The validatity of our approach will be measure in terms of the accuracy of the solution. For such as task, we will compare the solution obtained using the XFEM and the preconditioner against a refined body-fitted solution. Furthermore, we will measure the error in the XFEM solution by measuring the jump of the solution fields across the interface. Finally, we will measure the condition number, κ, and show that the preconditioner reduces the ill-conditioning of the system.

These examples were computed as part of a set of validation tests for the XFEM framework in the Fall of 2012. The results are included in the internal report in Appendix A. This version of the framework did not include Nitsche's method as a way to impose boundary conditions weakly.

Therefore, the problems utilize an early version of our stabilized Lagrange multipliers formulation 4.11. That is, the Lagrange multipliers were solved on an interface-per-interface basis. At the time of the writing of this thesis, this formulation has changed, and we now mostly compute the Lagrange multipliers by grouping interfaces with the same enrichment level into a single problem. Nevertheless, the problems studied here still provide insight into the characteristics of the framework.

7.2.1 Methodology

Two formulations are used to corroborate the results of the geometric preconditioner (5.3). The first measure, denoted as $\mathcal{S}_{\text{error}}$, computes the difference in solutions at the material interface, and is defined as:

$$\mathcal{S}_{\text{error}} = \left(\left(\int_{\Gamma^0} \left\| u^+ - u^- \right\|^2 \mathrm{d}\Gamma \right) \cdot \left(\int_{\Gamma^0} \mathrm{d}\Gamma \right)^{-1} \right)^{\frac{1}{2}} . \tag{7.1}$$

This equation computes the jump in the solution fields across the interface in the analysis domain, and scales the value with the perimeter (in 2D) or the surface area (in 3D) of the interface. Since the model we have implemented is based on inclusions and not on crack propagation, the interface error should approach zero as we refine the mesh.

The second measure, denoted as L^2_{error}, compares the relative difference between the XFEM solution and the refined body-fitted solution, and is defined as:

$$L^2_{\text{error}} = \left(\left(\int \tilde{\boldsymbol{u}} - \hat{\boldsymbol{u}}_{\text{FE}} \, \mathrm{d}\Omega \right) \cdot \left(\int \hat{\boldsymbol{u}}_{\text{FE}} \, \mathrm{d}\Omega \right)^{-1} \right)^{\frac{1}{2}} , \tag{7.2}$$

where $\tilde{\boldsymbol{u}}$ is defined in (3.1), and $\hat{\boldsymbol{u}}_{\text{FE}}$ represents the solution of the body-fitted problem.

The problem setup is shown in Figure 7.15. We model a two-material 2D linear diffusion problem. The analysis domain has a width of 20 units and a height of 20 units. Dirichlet boundary conditions are imposed on the sides. Because the level set interface never touches the boundaries, the Dirichlet conditions are applied in the strong form. The species field is prescribed to 0 on the left side, and 100 on the right side. The circular inclusion at the center of the model represents the positive material phase. We modify the radius of the circular inclusion from 1 units to 3 units in

500 steps. We measure the effects of applying mesh refinement, and using different conductivity ratios for both phases. The body-fitted analysis was computed using the same setup of Figure 7.15, and the mesh was refined until the solution reached convergence.

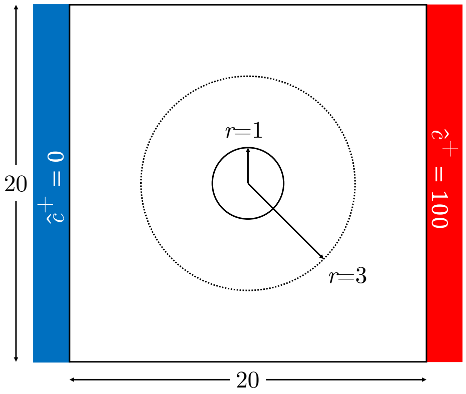


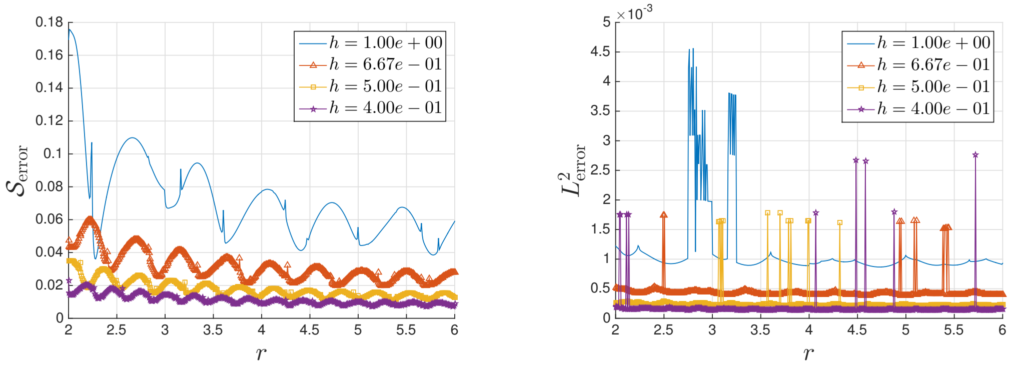
Figure 7.15: Problem setup for the 2D linear diffusion example of Section 7.2.

7.2.2 Mesh Refinement Sweep

In this test, we vary the element length scale, h, while the conductivity ratio between the two materials, k^+/k^-, remains fixed at 10. No preconditioner scaling is applied. The value for the interface penalty parameter, $\alpha_{\mathrm{N},c}$, is 10. The different mesh sizes used are $h = 1.00 \times 10^{+00}$, $h = 6.67 \times 10^{-01}$, $h = 5.00 \times 10^{-01}$, and $h = 4.00 \times 10^{-01}$.

Figure 7.16a shows the convergence of the XFEM solution with mesh refinement. As the mesh is refined, the interface error drops and approaches zero. Additionally, Figure 7.16b shows that as the mesh is refined, the difference of the XFEM solution with respect to the refined body-fitted one decreases. The peaks in the plot can be attributed to the discretization used to compute the L^2 error.

To compute the L^2 error, the XFEM and the body-fitted solutions are mapped to a separate discretized mesh with $h = 2.0 \times 10^{-01}$, where the difference in values between them is measured. This process was performed in ParaView [Squillacote and Ahrens, 2007]. The L^2_{error} measure proved to be sensitivite to the level of refinement of the discretized domain used to compute the differences,



(a) Interface error for the mesh refinement sweep.

(b) L^2 error for the mesh refinement sweep.

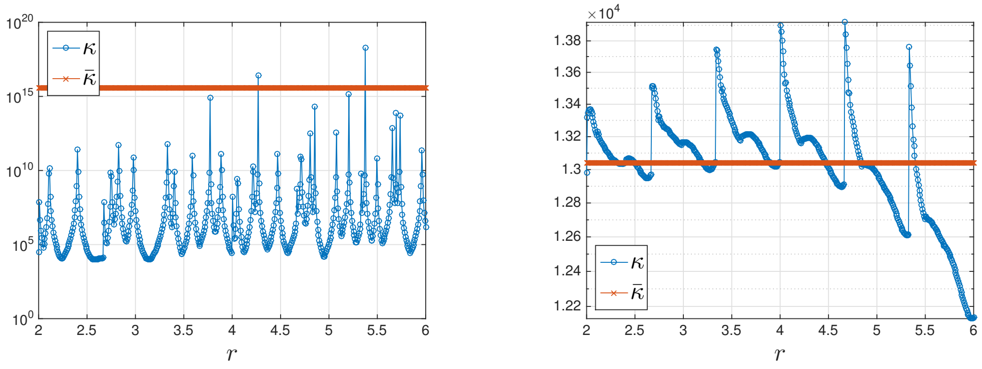
Figure 7.16: Interface and L^2 errors for the mesh refinement sweep. The oscillatory behavior matches the results from Lang et al [2014]. Peaks in the L^2_{error} measure are a product of the discretization used to compute the error.

especifically when the nodes of the original meshes aligned with the nodes of the mapped domain. Further examples utilize an h value of 1.94175×10^{-01} to decrease the likelihood of the nodes aligning with one another.

The average condition number of the matrices was in the order of 10^{+15}, which indicates that the system became ill-conditioned as the radius increased and the intersection patterns changed. However, the high condition number does not seem to affect the convergence of the solution. This behavior can be attributed to the fact that this is not a complex problem in terms of the physics nor in terms of the discretized interface configuration. We will study the effects of the preconditioner on the condition number next.

7.2.3 Condition Number

In this sweep, we measure the condition number of the system for a problem with an h value of 6.67×10^{-01}, a conductivity ratio of 10, and an interface penalty parameter of 10. We compute the solution with and without using the preconditioner scaling formulation (5.3).



(a) Without preconditioner.

(b) With preconditioner scaling.

Figure 7.17: Condition numbers sweep, with and without the preconditioner scaling. $\bar{\kappa}$ represents the average condition number across the sweep.

The plots in Figure 7.17 show that the preconditioner effectively reduces the condition number of the system in the presence of vanishing zones-of-influence. The oscillations in the plots of Figures

7.16 and 7.17 show that as the radius of the inclusion increases, the intersection patterns more or less repeat, leading to similar discretization errors at the interface.

7.2.4 Conductivity Ratio Sweep

In this sweep, we test five different conductivity ratios, $k^+/k^- = \{0.1, 1, 10, 100, 1000\}$. The mesh size, h, has a value of 6.67×10^{-01}, and the interface penalty parameter is 10.

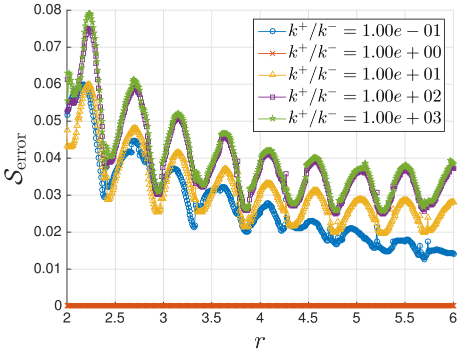


Figure 7.18: Conductivity ratio sweep. The results show that the interface error increases as the ratio of conductivities becomes larger.

Figure 7.18 shows the interface error of the conductivity ratio sweep. The plot displays the same oscillatory behavior seen in the previous examples. The results show that when the material conductivity is the same for both materials (i.e. a "quasi-FEM" problem), the interface error is in the order of $O(\epsilon)$. Contrarily, the interface error increases as the ratio of conductivities becomes larger for the other values.

7.2.5 Discussion

The results presented in these examples showed that the differences in solutions between the XFEM and a classical finite element body-fitted problem for a 2D linear diffusion model are small. The XFEM produced a system of equations with a higher condition number due to the degrees-of-freedom interpolation the solution in small intersection areas; however the geometric preconditioner scheme of Lang et al [2014] was able to resolve this shortcoming of the method. The XFEM showed that it is prone to errors with respect to the mesh refinement and the material properties of the different phase regions. However, we showed convergence of the solution with mesh refinement. The influence of the interface penalty parameter to enforce the continuity of the solution at the interface will be studied in later examples.

7.3 Linear Elasticity

We study the features of the proposed LSM-XFEM topology optimization approach with numerical examples for linear elastic structures. The LSM-XFEM results of solid-void and solid-solid problems are compared against the ones of the SIMP approach outlined in Section 1.2.4. In all examples, we seek to minimize the strain energy (4.67) subject to a constraint on the volume of the stiff phase (4.71). 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 (1.12) and (2.9), and perimeter constraints (4.73).

The linear systems of the forward and adjoint problems are solved by a parallel implementation of the GMRES method [Heroux et al, 2003]. 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 β are kept constant in the optimization process, i.e. no continuation approach is used. In the LSM-XFEM examples, the spring stiffness value, $k^s_{\boldsymbol{u}}$, is 10^{-06}.

While the LSM-XFEM results can be directly used to fabricate the structure, for example by 3D printing, the SIMP results need to be postprocessed. From a practitioner perspective, only the postprocessed SIMP results should be compared against the LSM-XFEM results. To this end, we postprocess the SIMP results with a lumping method that uses the isocontours of the density distribution. To obtain a strict “0-1” density distribution with smooth phase boundaries, we construct isovolumes for different threshold values, ρ_{T}, from the nodal density values, $\hat{\rho}_i$; see Section 1.2.4. The volume enclosed by the isocontour with $\hat{\rho} \geq \rho_{\mathrm{T}}$ 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 ρ_{T} values is analyzed conveniently with the XFEM. We refer to this postprocessing approach as isovolume 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 $0 < \hat{\rho} < 1$ as follows:

$$\bar{\rho} = \int \hat{\rho}\,(1 - \hat{\rho})\,\mathrm{d}\Omega \left(\int \mathrm{d}\Omega \right)^{-1} . \tag{7.3}$$

For more details on the examples studied here, the reader is referred to Villanueva and Maute [2014].

7.3.1 Design of a 2D Short Cantilever Beam

In this example, we study a short cantilever beam in 2D. The optimized material layout will serve as a reference solution, and will be compared against designs in subsequent examples that utilize additional regularization techniques (see Section 6). The problem setup is shown in Figure 7.19. We model a solid-void problem with an auxiliary indicator field over the solid domain to detect free-floating particles. The objective is stated as follows:

$$\mathcal{Z} = \frac{\mathcal{U}^s}{\left\| \mathcal{U}^{s(0)} \right\|} + w_{\mathcal{S}} \frac{\mathcal{S}}{\left\| \mathcal{S}^{(0)} \right\|} \ , \tag{7.4}$$

where $\mathcal{U}^s$ represents the strain energy (4.67) of the solid phase, and the superscript (0) denotes the value of the initial design. We impose a maximum volume fraction of the solid domain of 50%. The remaining problem parameters are shown in Table 7.3. Symmetry of the design variables is imposed along the y-axis. The elements entirely in the “void” phase are omitted in the element assembly

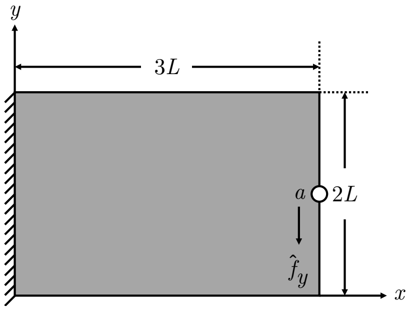


Figure 7.19: Problem setup for the 2D short cantilever beam example.

process and the degrees-of-freedom that interpolate the "void" phase are eliminated from the system of equations; both techniques reduce the computational cost of solving the XFEM problem.

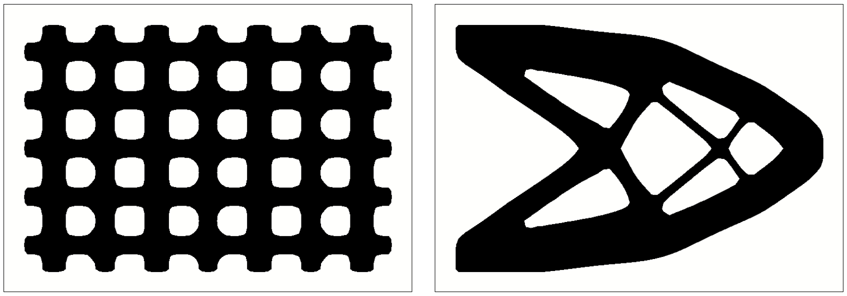
The initial design, shown in Figure 7.20a, consists of a solid box with 8 × 6 circular void inclusions of radius $0.125L$. A void area with a thickness of $0.02L$ is modeled around the design domain to guarantee that all the boundaries of the design are part of the material interface. The converged optimized material layout after 400 iterations is shown in Figure 7.20b. The objective and volume constraints plots are shown in Figure 7.21. The truss-like design matches well the results from the literature. The problem does not suffer from numerical artifacts or "checkerboard"-like patterns as seen by Makhija and Maute [2014b]. The strain energy varies from 2.393×10^{-01} to 1.364×10^{-01}; this occurs because the optimization problem first satisfies the volume constraint, and then minimizes the objective until a feasible minimum is found, as shown in Figure 7.21. We will revisit this example in subsequent sections.

7.3.2 Design of a 2D MBB Beam

In this example, we study the well-known MBB beam design problem. The exact parameter set is taken from Sigmund [2009], and the problem setup is shown in Figure 7.22. The objective is to minimize the strain energy of a beam by arranging a limited amount of solid material in a domain. The solid domain is denoted by the negative phase of the LSF. The lower and upper limits, s_i^L and s_i^U, of the design variables are set to $\mp h/2$, respectively. Similar to the example in Section

	Value
Element length scale	$h = 5.0 \times 10^{-02} L$
Young's modulus	$E^s = 1.0$
Poisson's ratio	$\nu^s = 10^{-01}$
Preconditioner tolerance	$T_{\text{tol}} = 10^{+06}$
Nitsche displacement penalty	$\gamma_{\text{N},\boldsymbol{u}} = 10^{+02}$
Nitsche species penalty	$\alpha_{\text{N},c} = 10^{+01}$
Aux. pressure penalty	$k_p^s = 10^{-06}$
Surface area penalty	$w_{\mathcal{S}} = 10^{-01}$
Maximum volume constraint	50%
Smoothing filter radius	$r_\phi = 2.4\ h$

Table 7.3: Problem parameters for the 2D short cantilever and MBB beams examples.



(a) Initial design.

(b) Optimized design.

Figure 7.20: Material layouts for the 2D short cantilever beam example.

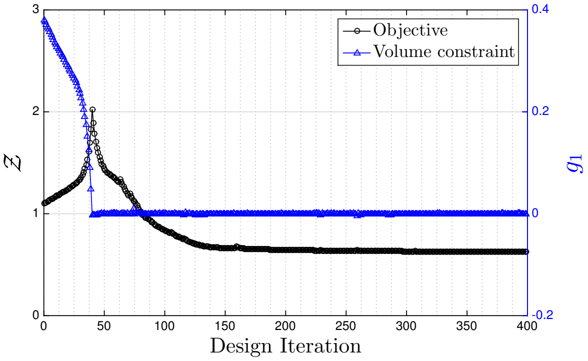


Figure 7.21: Objective and volume constraint plots for the 2D short cantilever beam example.

7.3.1, additional padding regions are included in the mesh around the design domain. In these regions, the design variables are fixed to the upper limit, s_i^U, to guarantee that they remain a void domain. These additional regions are used to ensure that all the boundaries of the design geometry are part of the XFEM interface (rather than part of an external surface). The problem parameters are the same as in Table 7.3, with the exception of the element size, which is set to $h = 6.7 \times 10^{-03}$.

The initial design is shown in Figure 7.23a. The design domain is initialized with 27 × 9 cuboid inclusions of radii $2h$. The design variables contained within a radius of $10h$ of the anchor and load regions are prescribed to the upper limit s_i^U. The converged optimized material layout after 400 iterations is shown in Figure 7.23b. The objective and volume constraint plots are shown in Figure 7.24. Optimal designs typically have many thin truss-like structures (see Michell [1904]), which match well the results computed in this study. The strain energy varies from 8.564×10^{-02} to 5.452×10^{-01}. The objective plot displays oscillatory behavior for certain design geometries; this is attributed to the lack of a measure to control the thickness of these trusses when they become smaller than the element size.

7.3.3 Design of a 3D Cube with a Center Load

We consider the solid-void optimization problem depicted in Fig. 7.25. With this example we will illustrate the basic features of the LSM-XFEM approach for 3D problems and show that the

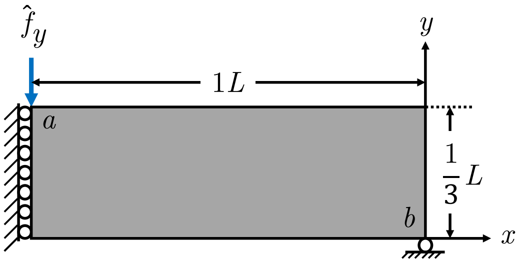
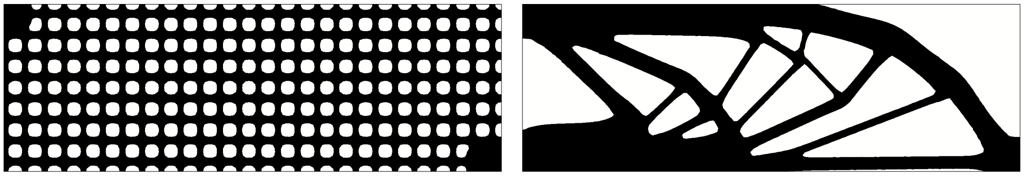


Figure 7.22: Problem setup for the 2D MBB beam example.



(a) Initial design.

(b) Optimized design.

Figure 7.23: Material layouts for the 2D MBB beam example.

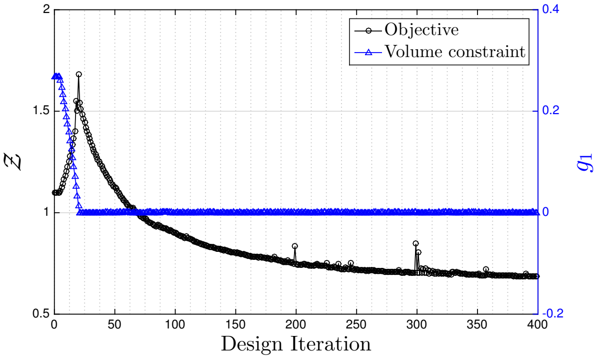


Figure 7.24: Objective and volume constraint plots for the 2D MBB beam example.

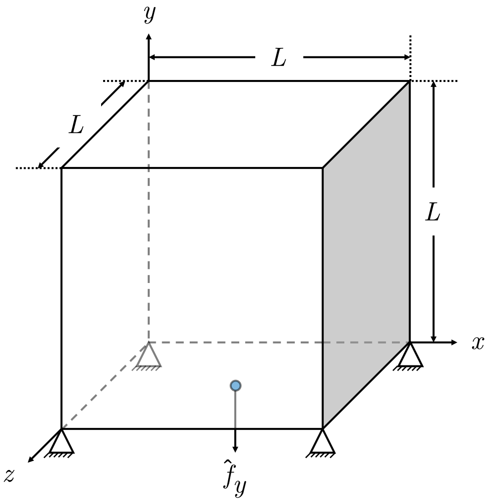


Figure 7.25: Problem setup for the cube with a center load example.

proposed LSM-XFEM approach and the SIMP formulation may exhibit comparable convergence behaviors as the mesh is refined.

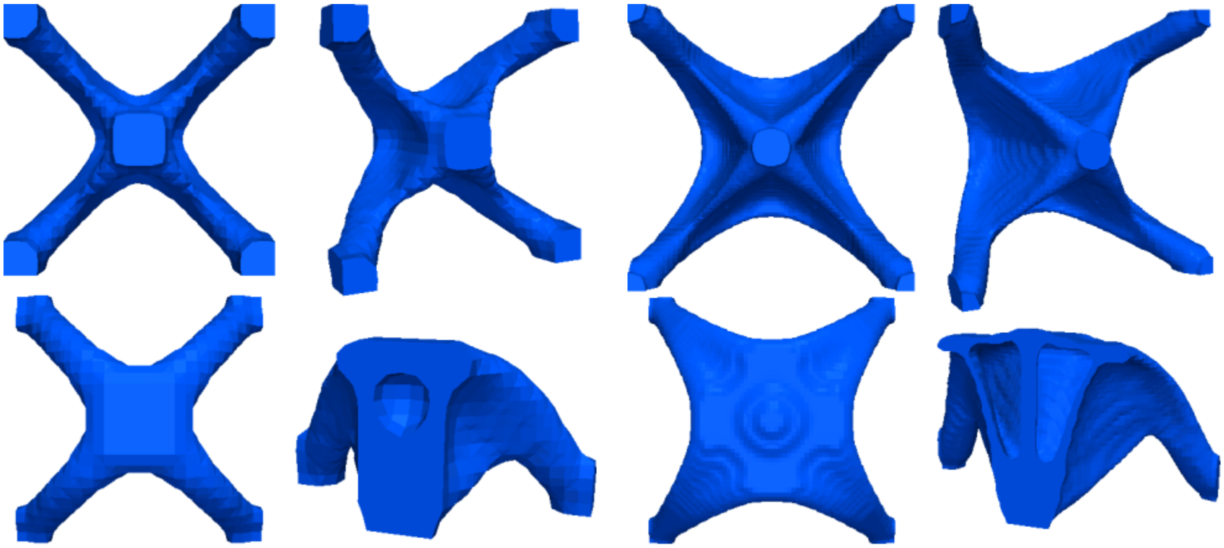
The $L \times L \times L$ cubical design domain is pinned at its four bottom corners in the vertical direction and a unit force is applied at the center of the bottom face. The Young's modulus of the stiff phase is set to 1 and the Poisson's ratio to 0.3. The maximum volume of the stiff phase is 10%. We compare LSM-XFEM and SIMP results for two mesh sizes: $24 \times 24 \times 24$ 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 $E^{+} = 10^{-09}$. 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 h). While this approach does not ensure mesh-independent optimization results, it still prevents the formation of checker-board 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 $\hat{\rho}_i = 0.1$. The optimized material distributions are shown in Figure 7.26 where material with a density lower than $\hat{\rho}_i < 0.75$ is considered void. The strain energies are reported in Table 7.4. 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 postprocessed with the IDL approach described above. The strain energies for varying threshold values, ρ_{T}, are plotted in Figure 7.27. The volume constraint is met for $\rho_{\mathrm{T}} = 0.78$ for the coarse mesh and $\rho_{\mathrm{T}} = 0.44$ for the fine mesh. The value of ρ_{T} is higher for the coarse mesh because it cannot converge to a design with void inclusions. 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 volume fractions of intermediate densities (7.3) are 0.2850 and 0.0189 for the coarse and fine mesh, respectively. The postprocessed designs have lower strain energies because the postprocessing counteracts the effect of the density filter (1.12).

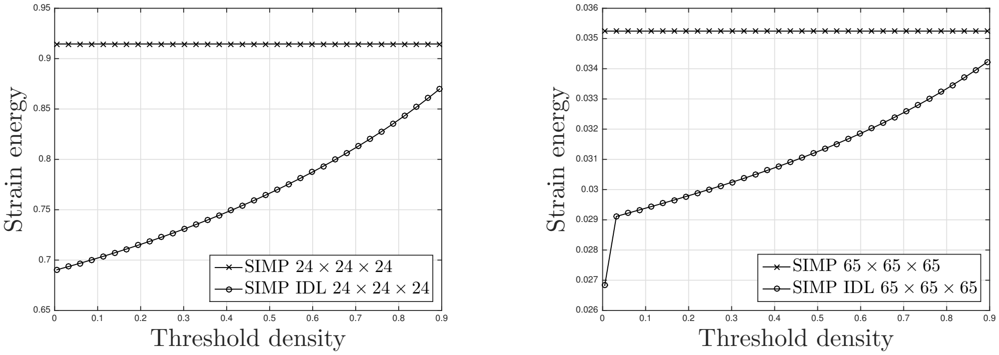
The same optimization problem is solved with the proposed LSM-XFEM approach. The smoothing radius is set to $r_\phi = 3.2$ for the coarse mesh and $r_\phi = 1.182$ for the fine mesh. No perimeter constraint is imposed. We seed the initial design with two different configurations of void inclusions to 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



(a) 24 × 24 × 24 mesh.

(b) 65 × 65 × 65 mesh.

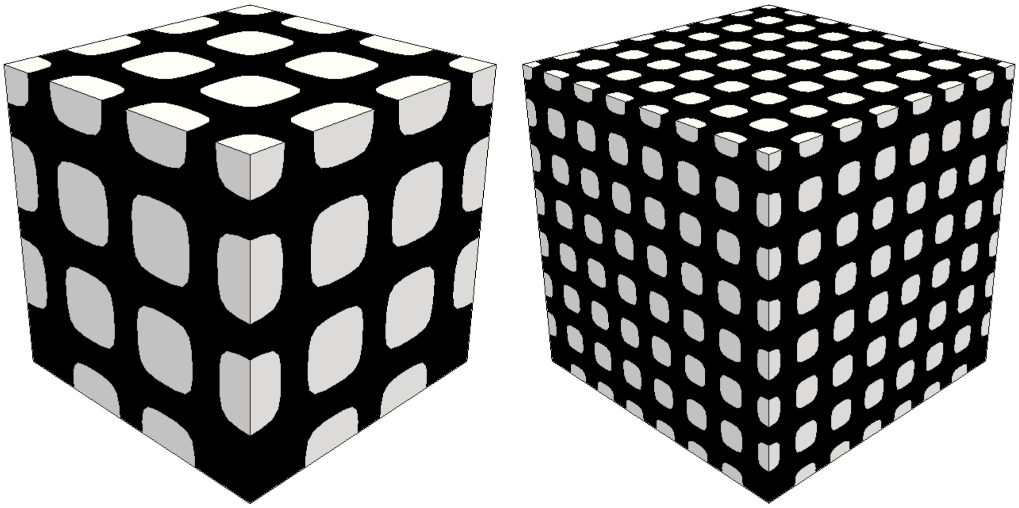
Figure 7.26: SIMP results for the cube with a center load problem; clockwise: bottom, side, top, and clipped views.



(a) 24 × 24 × 24

(b) 65 × 65 × 65

Figure 7.27: IDL postprocessing of SIMP results for the cube with a center load problem.



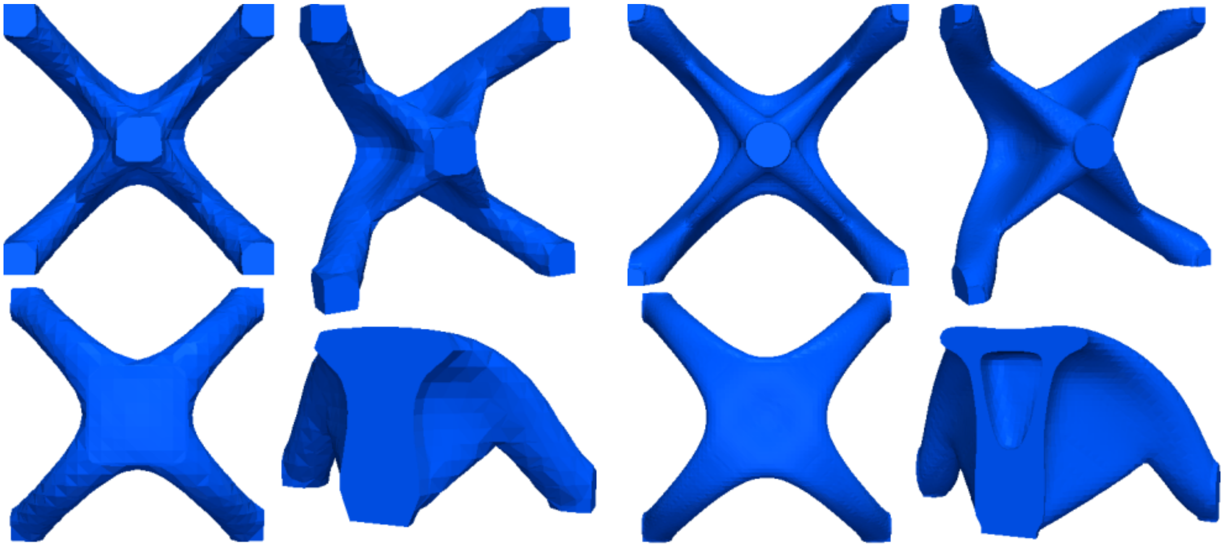
(a) 3 × 3 × 3 holes.

(b) 7 × 7 × 7 holes.

Figure 7.28: Initial level set configurations for the cube with a center load problem.

	Mesh size	Strain energy
SIMP	24 × 24 × 24	9.1456×10^{-01}
	65 × 65 × 65	3.5244×10^{-02}
XFEM	24 × 24 × 24	$1.0082 \times 10^{+00}$
	65 × 65 × 65	3.5519×10^{-02}

Table 7.4: Comparison of strain energies of SIMP and LSM-XFEM results for the cube with a center load problem; the corresponding designs are shown in Figures 7.26 and 7.29.



(a) 24 × 24 × 24 mesh.

(b) 65 × 65 × 65 mesh.

Figure 7.29: LSM-XFEM results for cube with center load problem; clockwise: bottom, side, top, and clipped views.

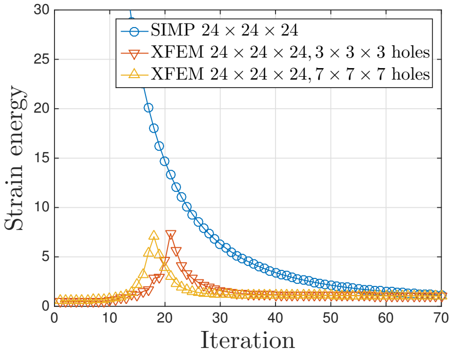


Figure 7.30: Evolution of strain energies in the optimization process for SIMP and LSM-XFEM approaches.

defined as:

$$s_i = (x_i - x_c)^{10} + (y_i - y_c)^{10} + (z_i - z_c)^{10} - r^{10} \ , \tag{7.5}$$

and then compute ϕ_i by using (2.9). One configuration has 3 × 3 × 3 equally spaced holes with radii of 5.5, the other has 7 × 7 × 7 holes with radii of 2.0, as shown in Figure 7.28. 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 Figure 7.29. The strain energies of the optimized designs are given in Table 7.4. The convergence history for the coarse meshes in SIMP and LSM-XFEM is shown in Figure 7.30.

For the example considered here, the SIMP and LSM-XFEM results match well in regards to the geometry. 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. 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.

7.3.4 Design of a 3D Cuboid Under Torsion

The second solid-void example is taken from Nguyen et al [2012] and reveals differences in the SIMP and the 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 $4L \times 1L \times 1L$, as shown in Figure 7.31. 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's 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.

7.3.4.1 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 $E^{+} = 10^{-09}$. The design domain is initialized with a uniform material distribution of $\hat{\rho}_i = 0.1$. The penalization factor is $p = 3$. First, we consider a projection parameter of $\beta = 0$ and scale the smoothing radius with the element size: $r_\rho = 1.6\ h$.

The optimized material distributions are shown in Figure 7.32 where material with a density lower than $\hat{\rho}_i < 0.35$ is considered void. The strain energies are reported in Table 7.5 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 2D problems. The optimized material layout switches abruptly from a grid-type structure, which conceptually agrees with the results of Nguyen et al [2012], to a hollow square prism design. In contrast to 2D 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

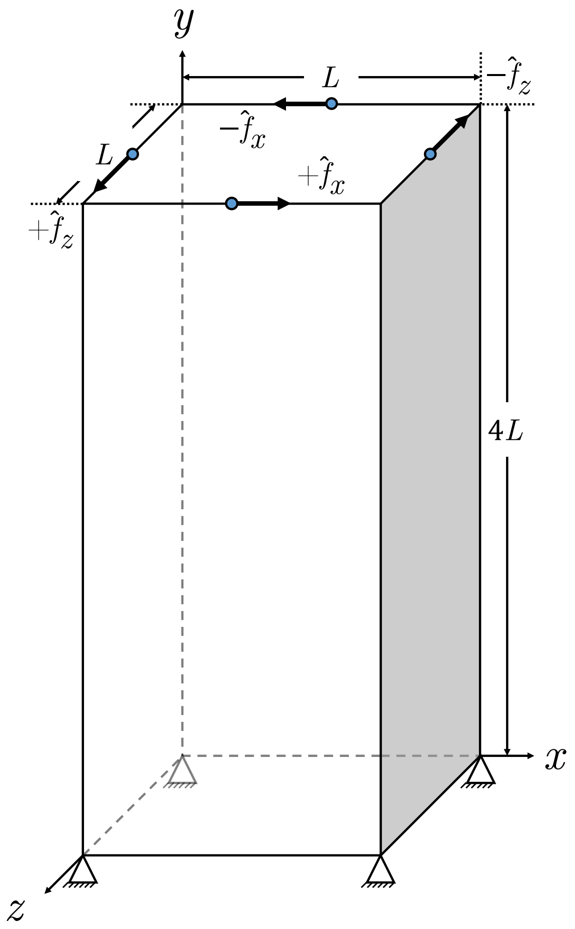
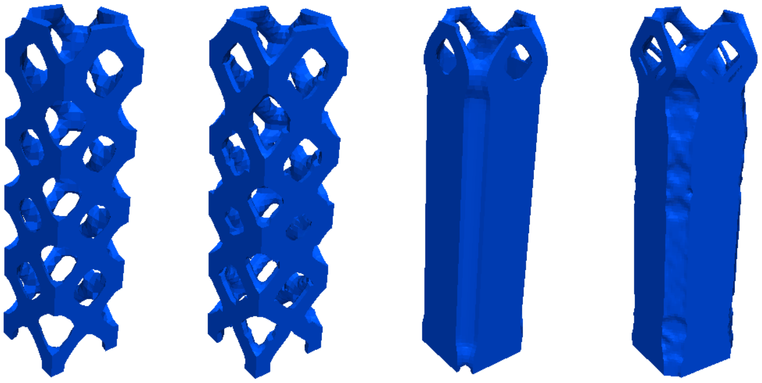


Figure 7.31: Problem setup for the cuboid under torsion example.

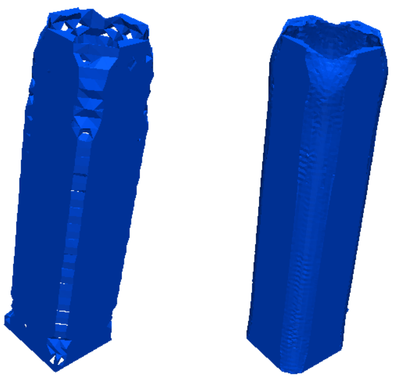
	Mesh size	Strain energy
SIMP	$40 \times 10 \times 10$	$7.5195 \times 10^{+03}$
	$60 \times 15 \times 15$	$4.2076 \times 10^{+03}$
	$80 \times 20 \times 20$	$4.0298 \times 10^{+03}$
	$120 \times 30 \times 30$	$2.6555 \times 10^{+03}$

Table 7.5: Strain energies of SIMP results for cuboid under torsion problem; SIMP parameters: $p = 3$, $r_\rho = 1.6\ h$, and $\beta = 0$; the corresponding designs are shown in Figure 7.32.



(a) 40 × 10 × 10. (b) 60 × 15 × 15. (c) 80 × 20 × 20. (d) 120 × 30 × 30.

Figure 7.32: SIMP results of cuboid under torsion problem for different levels of mesh refinement; SIMP parameters: $p = 3$, $r_\rho = 1.6\ h$, $\beta = 0$.



(a) $40 \times 10 \times 10$ (b) $120 \times 30 \times 30$

Figure 7.33: LSM-XFEM results of cuboid under torsion problem for two levels for mesh refinement; LSM-XFEM parameters: $r_\phi = 1.6\ h$, no perimeter constraint.

combination of SIMP penalization and density smoothing. We will revisit this issue again later.

The LSM-XFEM results for a smoothing radius of $r_\phi = 1.6\ h$ are shown in Figure 7.33. 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 designs 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 Table 7.6. 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 overpredicting 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 results is primarily caused by the density filter,

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

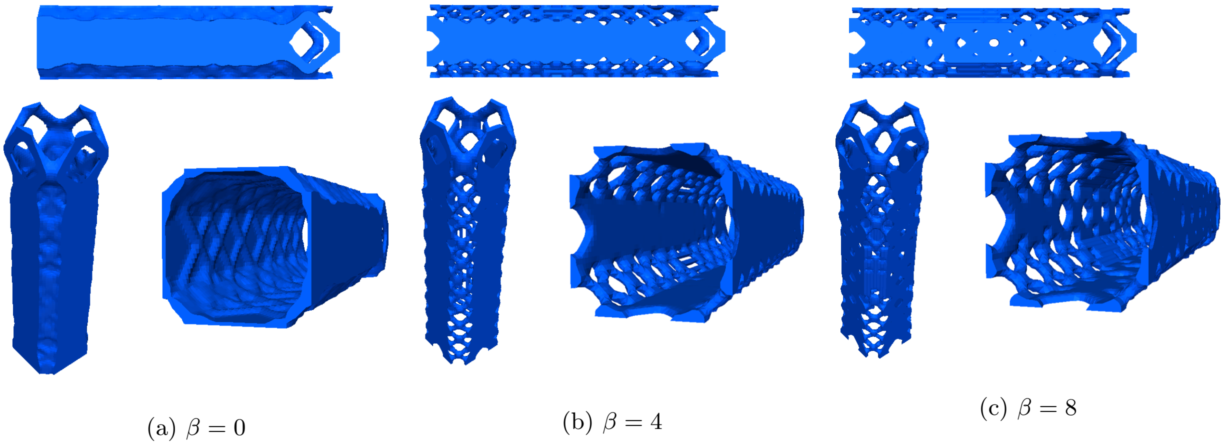
Table 7.6: Strain energies of LSM-XFEM results for cuboid under torsion problem; LSM-XFEM parameters: $r_\phi = 1.6\ h$, no perimeter constraint; the corresponding designs are shown in Figure 7.33.

which prevents the material distribution to converge to a “0-1” result.

First, we study the influence of the projection scheme (1.14) on the SIMP results for the most refined mesh. The optimized material distributions for $\beta = 4.0$ and $\beta = 8.0$ are shown in Figure 7.34, where material with a density lower than $\hat{\rho}_i < 0.35$ is considered void. For convenience the result for $\beta = 0.0$ is shown again. Table 7.7 reports on the strain energies and the volume fractions of intermediate densities, $\bar{\rho}$, as the projection parameter, β, is increased. The higher β, the lower $\bar{\rho}$ and the lower the strain energy, approaching the one of the LSM-XFEM result. Note that as β increases, the more holes emerge. The thickness of the walls for $\beta = 8$ is 0.0434, which is smaller than the value for $\beta = 0.0$, 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 β, we postprocess the SIMP results for $\beta = 0.0$ by the IDL postprocessing method. Figure 7.35 shows the strain energy of the postprocessed design over the threshold density, ρ_{T}, for the coarsest and the finest mesh. The volume constraint is satisfied for a threshold value of $\rho_{\mathrm{T}} = 0.4634$ for the coarse mesh, and $\rho_{\mathrm{T}} = 0.5174$ 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 Table 7.8.

The strain energy of the postprocessed results of the fine mesh is rather similar to the result obtained for SIMP with $\beta = 8.0$ in Table 7.7 and the LSM-XFEM results in Table 7.6. For the



(a) $\beta = 0$

(b) $\beta = 4$

(c) $\beta = 8$

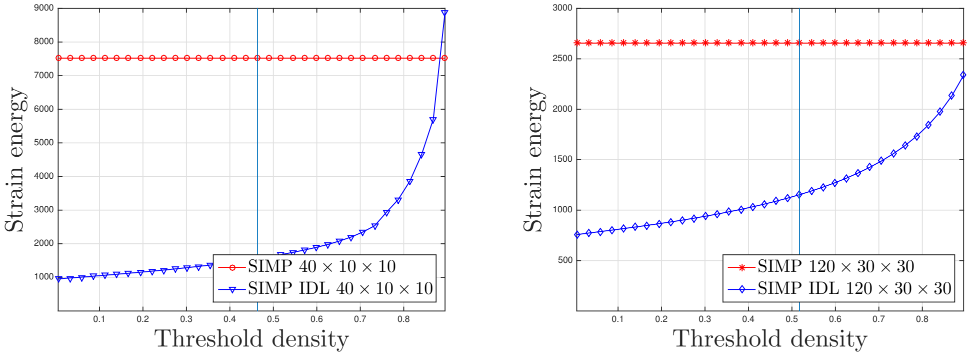
Figure 7.34: SIMP results for different projection parameters $\beta = [0.0, 4.0, 8.0]$; mesh size: $120 \times 30 \times 30$.

	β projection	Strain energy	$\bar{\rho}$ utilization
SIMP	0	$2.6555 \times 10^{+03}$	4.0688×10^{-02}
	4	$2.0264 \times 10^{+03}$	2.5131×10^{-02}
	8	$1.9039 \times 10^{+03}$	1.9509×10^{-02}

Table 7.7: Strain energies of SIMP results for different projection parameters $\beta = [0.0, 4.0, 8.0]$; the corresponding designs are shown in Fig. 7.34.

	Mesh	ρ_{T}	Strain energy
SIMP	$40 \times 10 \times 10$	0.4634	$1.5601 \times 10^{+03}$
	$120 \times 30 \times 30$	0.5174	$1.1530 \times 10^{+03}$

Table 7.8: Strain energies of the SIMP-IDL designs of cuboid under torsion problem; the corresponding designs are shown in Fig. 7.32.



(a) 40 × 40 × 40

(b) 120 × 30 × 30

Figure 7.35: IDL postprocessing of SIMP results for cuboid under torsion problem; the vertical lines mark the threshold values at which the volume constraint is satisfied.

coarse mesh, the strain energy is well below the raw SIMP results from Table 7.5 but still above the results for the LSM-XFEM approach in Table 7.6. As we will see below, this is because of the larger smoothing radius which prevents the formation of smaller features and thinner walls.

7.3.4.2 Feature size control

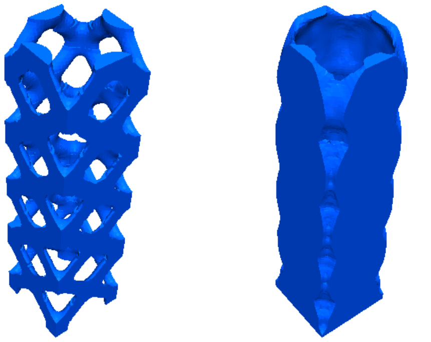
The mesh refinement study above suggests that the results of the LSM-XFEM approach are less sensitive to mesh refinement than the SIMP method without mesh-independent filtering. Geometric features, such as the thin walls, can be represented on coarse and fine meshes, independent of their size. This observation is in agreement with studies for two-dimensional problems, see for example Kreissl and Maute [2012], 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 7.36a 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 Figure 7.32a. Comparing the SIMP results in Figure 7.36a and Figure 7.32a confirms the finding of numerous studies [Bendsøe and Sigmund, 2003] 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 Figure 7.36a is given in Table 7.9.

A similar effect is not observed in the LSM-XFEM approach when we apply the same filter radius, r_ϕ, used earlier for the coarse mesh to the fine mesh. Figure 7.36b 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 Table 7.9.

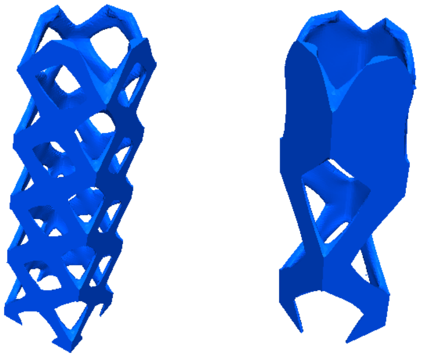
	Mesh size	Strain energy
SIMP	$120 \times 30 \times 30$	$6.5772 \times 10^{+03}$
LSM-XFEM	$120 \times 30 \times 30$	$8.2077 \times 10^{+02}$

Table 7.9: 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. 7.36.



(a) SIMP (b) LSM-XFEM

Figure 7.36: SIMP and LSM-XFEM results for larger smoothing radius; mesh size: $120 \times 30 \times 30$.



(a) Restarted from Figure 7.36a.

(b) Restarted from Figure 7.33b.

Figure 7.37: LSM-XFEM results for cuboid under torsion problem using a perimeter constraint; mesh size: $120 \times 30 \times 30$.

To control the overall structural complexity in the LSM, the formulation of the optimization problem (1.1) is often augmented by a perimeter constraint [van Dijk et al, 2013]. 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 × 30 × 30 mesh using the LSM-XFEM approach. We measure the perimeter of the SIMP result shown in Figure 7.36a and impose this value as an upper bound on the perimeter. One problem uses the SIMP result in Figure 7.36a as the initial design, and the other one uses the LSM-XFEM result in Figure 7.33b. The results are shown in Figure 7.37 and the strain energies are given in Table 7.10.

Depending on the initial designs, the LSM-XFEM problems converge to different designs. While the design in Figure7.37b 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 Figure 7.37a resembles closely the SIMP result from which it was restarted. However, considering the strain energy in Table 7.10, this design has a larger strain energy than the one in Figure 7.37b.

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 Figure 7.37b shows. The design has more structural features than the design without perimeter constraint in Figure 7.33b.

7.3.5 Design of a 3D 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 discussed in Section 3 is not applicable to such problems. The generalized enrichment strategy in (3.1) is required and

	Initial design	Strain energy
LSM-XFEM	SIMP	$1.3321 \times 10^{+03}$
	LSM-XFEM	$1.3185 \times 10^{+03}$

Table 7.10: 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 Figure 7.37.

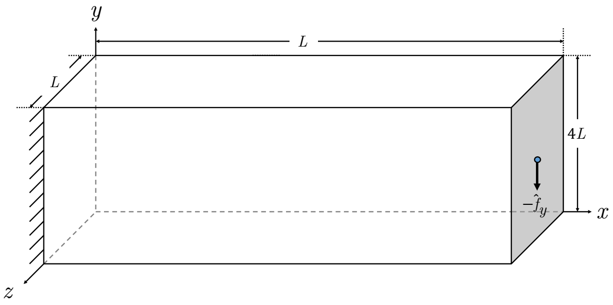


Figure 7.38: Initial setup for two-phase cantilever beam problem.

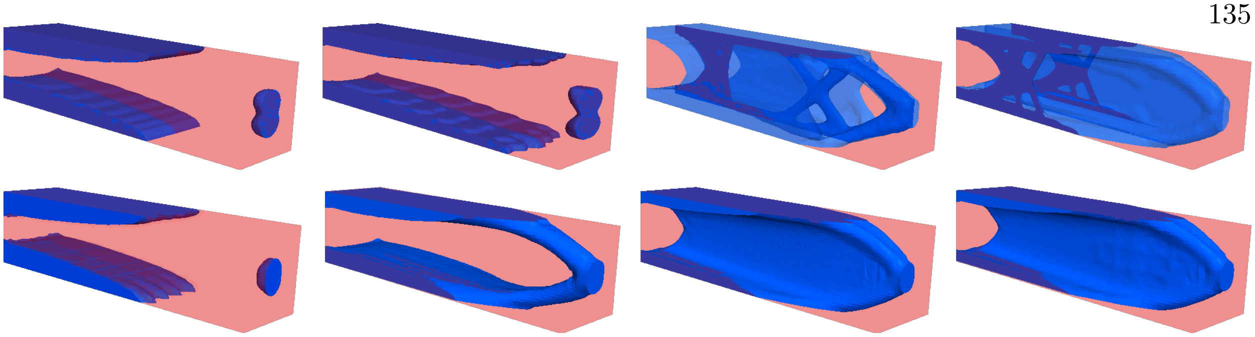


Figure 7.39: SIMP and LSM-XFEM results for different stiffness ratios: (from left to right) $E^+ = 0.5E^-$, $E^+ = 0.1E_-$, $E^+ = 0.01E^-$, E^+ is void; SIMP results (top row); LSM-XFEM results (bottom row).

the interface conditions of (4.10) need to be satisfied.

We study the optimal two-phase layout of a $4L \times 1L \times 1L$ cantilevered beam subject to a tip load; see Figure 7.38. The stiff negative phase has Young's modulus of $E^- = 1.0$; three values of Young's moduli for the soft phase are considered: $E^+ = \{0.5, 0.1, 0.01\}$. Both phases have a Poisson's 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 h) and the projection parameter of $\beta = 0$. The design domain is initialized with a uniform material distribution of $\hat{\rho}_i = 0.3$. The optimized material distributions are shown in Figure 7.39 where material with a density lower than $\hat{\rho}_i < 0.25$ is transparent. The strain energies are reported in Table 7.11.

The LSM-XFEM results for a smoothing radius of $r_\phi = 0.05333$ (1.6 h) are shown in Figure 7.39 and the strain energies are given in Table 7.11. 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 satisfies the volume constraint for the stiff phase. Note that the interface condition is enforced via the stabilized Lagrange multiplier method with an element wise constant Lagrange

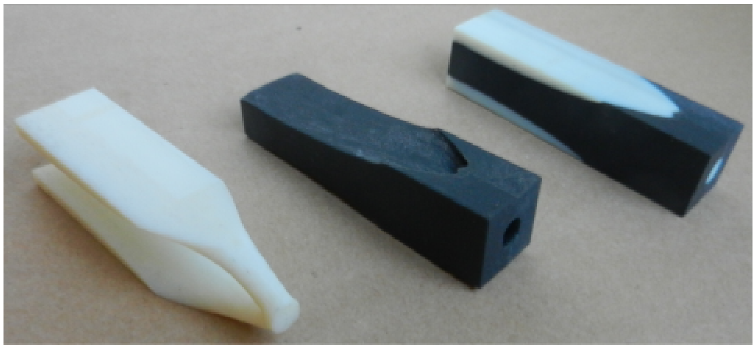


Figure 7.40: LSM-XFEM optimized two-phase design for $E^{+} = 0.1E^{-}$ realized by 3D printing.

multiplier (4.10), and a consistency factor of $\gamma_{\mathrm{L},\boldsymbol{u}} = 10\ \ (E^{-} + E^{+})$.

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 3D 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 $E^{+} = 0.1E^{-}$ in Figure 7.40. The structure was fabricated with a polyjet 3D printing process on a Connex Objet 260 printer. White material represents the negative phase, black represents the positive phase. The left and center pieces show the individual phases printed separately, the printed two-phase design is shown on the right.

7.4 Face-oriented Ghost-penalty Methods

In this section, we study and validate the face-oriented ghost-penalty formulation introduced in Section 5.2. We compare the accuracy and convergence of the proposed framework against a body-fitted problem taken from the literature.

	Stiffness ratio	Strain energy
SIMP	$E^{+} = 0.50E^{-}$	4.4081×10^{-05}
	$E^{+} = 0.10E^{-}$	6.2862×10^{-05}
	$E^{+} = 0.01E^{-}$	7.8627×10^{-05}
	E^{+} is void	7.6721×10^{-05}
LSM-XFEM	$E^{+} = 0.50E^{-}$	4.3221×10^{-05}
	$E^{+} = 0.10E^{-}$	5.9192×10^{-05}
	$E^{+} = 0.01E^{-}$	6.4448×10^{-05}
	E^{+} is void	6.6283×10^{-05}

Table 7.11: Strain energies of SIMP and LSM-XFEM results for different stiffness ratios; the corresponding designs are shown in Figure 7.39.

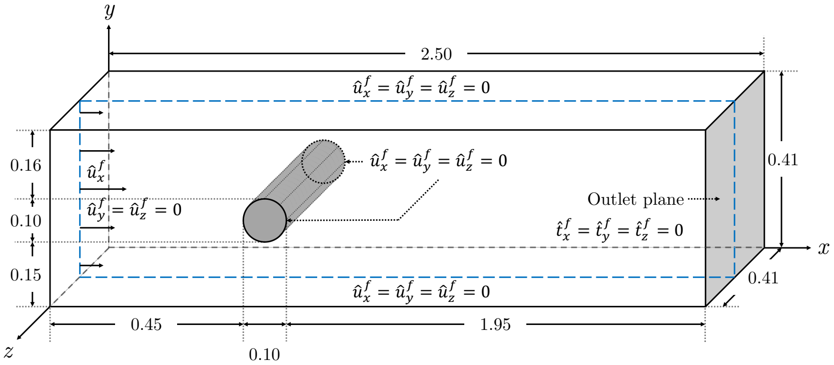

y
2.50
$\hat{u}_x^f = \hat{u}_y^f = \hat{u}_z^f = 0$
0.16
$\hat{u}_x^f$
$\hat{u}_x^f = \hat{u}_y^f = \hat{u}_z^f = 0$
0.41
0.10
$\hat{u}_y^f = \hat{u}_z^f = 0$
Outlet plane
$\hat{t}_x^f = \hat{t}_y^f = \hat{t}_z^f = 0$
x
0.15
0.41
$\hat{u}_x^f = \hat{u}_y^f = \hat{u}_z^f = 0$
0.45
1.95
z
0.10

Figure 7.41: Problem setup for the flow around a cylinder example. Blue dashed lines denote the symmetry plane.

7.4.1 Verification of the LSM-XFEM Analysis for Laminar Flow Problems

In this example, we seek to verify the analysis capabilities of our LSM-XFEM optimization framework. The validity of our approach will be measured in terms of the accuracy of the flow solutions with respect to solutions from the literature, and in terms of the convergence of the flow solutions with respect to different levels of mesh refinement. We use the 3D-1Z problem from Schäfer et al [1996], which considers a 3D laminar steady-state flow around a cylinder. No indicator fields are modeled in this example because the geometry does not change during the analysis. The problem setup is shown in Figure 7.41. The inflow condition is:

$$\hat{u}_x^f\left(0,y,z\right) = 16u_m^f yz \cdot \left(\frac{\left(0.41-y\right)\left(0.41-z\right)}{0.41^4}\right) \ , \qquad \hat{u}_y^f = \hat{u}_z^f = 0 \ , \tag{7.6}$$

where $u_m^f = 0.45$. A traction-free boundary condition is imposed on the outlet. No-slip boundary conditions are imposed on the surface of the cylinder and on all other planes. The characteristic velocity is a function of $\hat{u}_x^f$ in (7.6), and is defined as:

$$u_c^f = \frac{4}{9}\hat{u}_x^f\left(0, \frac{0.41}{2}, \frac{0.41}{2}\right) \ . \tag{7.7}$$

The characteristic length, L_c, is defined as the diameter of the cylinder, 0.1, which yields a Reynolds number of 20. The following quantities are computed: the drag coefficient (4.68) around the cylinder, c_D, and the total pressure difference (4.70) between the inlet and the outlet planes, $\mathcal{T}_{\text{in}} - \mathcal{T}_{\text{out}}$. The numerical solutions provided in the study by Schäfer et al [1996] give the values of 6.05 and 6.25 as the lower and upper bounds for the drag, respectively. These bounds were computed from the numerical results provided by several research groups through different numerical schemes, such as Finite Difference, Finite Volume, and Finite Element Methods, among others, and mesh convergence studies. No reference solution is provided for the total pressure difference; however, given that this measure is used as the objective of the optimization problems in several numerical examples ahead, it is important to study its convergence.

The mesh for the body-fitted problem utilizes a boundary layer around the cylinder, with 128 elements on the surface and 64 layers. The width of the first layer is 1.0832×10^{-09}, and the exponential growth factor of each subsequent layer is 1.2. The number of elements on the inlet and outlet surfaces is 40 × 40 and 20 × 20, respectively. No symmetry boundary conditions are imposed. All boundary conditions are imposed in the strong form, and no face-oriented ghost-penalty formulation is applied. The total number of degrees-of-freedom is $3,337,257$. The remaining parameters used for the problem are shown in Table 7.12.

In our LSM-XFEM approach, we use a fixed background mesh and perform a mesh refinement study to evaluate the accuracy of the framework. The mesh for the problem is constructed using a local hierarchical mesh refinement. We do not consider adaptive mesh refinement nor boundary layer meshes in this example, nor in our topology optimization examples below, as we do not have the means to do so as the design changes during the optimization process. The hierarchical mesh refinement utilized here does not provide the same resolution as the boundary layer meshing scheme above. However, we use it to study the characteristics of our analysis with respect to certain levels of refinement; this information will be useful to select an appropiate element size for the meshes we will use in our optimization problems. The cylinder is modeled using a level set function:

$$\phi_i\left(\boldsymbol{x}\right) = r_c - \left(\left(x_i - x_c\right)^2 + \left(y_i - y_c\right)^2\right)^{\frac{1}{2}} \ , \tag{7.8}$$

where the radius of the cylinder is set to $r_c = 0.05$, and the coordinates at the center of the cylinder are set to $x_c = 0.5$ and $y_c = 0.2$.

Our first level of mesh refinement has an element size of $h = 1.14 \times 10^{-02}$ for elements located at $x < 1$ and $h = 3.41667 \times 10^{-02}$ for all other elements. Subsequently, we only refine elements that are intersected by the zero level set isosurface of (7.8); we perform this process thrice. In consequence, we end up with 4 refinement levels, as shown in Figure 7.42. We also measure the influence of the Nitsche penalty parameter, $\alpha_{\mathrm{N},\boldsymbol{u}}$, for 4 different values: 10, 10^2, 10^3, and 10^4. The list of parameters is shown in Table 7.13.

The hierarchical mesh refinement scheme leads to a larger number of elements compared to the boundary layer approach. To reduce the computational cost, we model half of the domain by setting the $z = 0.205$ plane as the symmetry axis, and impose symmetry boundary conditions

	Value
Characteristic velocity	$u_c^f = 0.2$
Characteristic length	$L_c = 0.1$
Dynamic viscosity	$\mu^f = 0.001$
Density	$\rho^f = 1$
Nitsche velocity penalty	$\alpha_{\mathrm{N},\boldsymbol{u}} = 0$
Viscous ghost-penalty	$\alpha_{\mathrm{GP},\mu} = 0$
Pressure ghost-penalty	$\alpha_{\mathrm{GP},p} = 0$
Convective ghost-penalty	$\alpha_{\mathrm{GP},\boldsymbol{u}} = 0$
Pressure constraint parameter	$k_p^f = 0$

Table 7.12: Problem parameters for the flow around a cylinder example (body-fitted).

by setting $\hat{u}_z^f = 0$ along this plane. The inflow and outflow conditions are the same as in the body-fitted problem setup. All boundary conditions are enforced weakly. The total number of degrees-of-freedom for the finest mesh is $8,019,736$. The face-oriented ghost-penalty methods increase the bandwidth of the sparse matrix of the system, which may increase the linear solve time. This additional computational cost requires special attention when solving these problems.

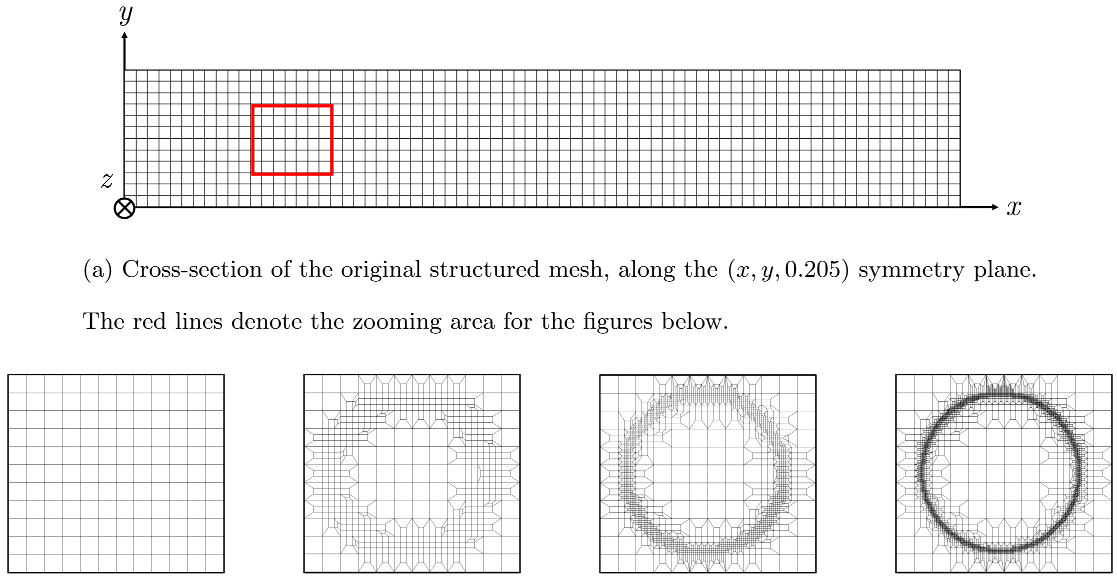
The results for the body-fitted and the LSM-XFEM problems are shown in Figure 7.43. The drag coefficient for the body-fitted problem is 6.169, well within the lower and upper bounds established in the study by Schäfer et al [1996]. The total pressure difference is 0.0213. For our LSM-XFEM analysis, we can observe in Figure 7.43a that the drag coefficient solutions are not fully converged, but that the values for the finest mesh are within the lower and upper bounds. It is not clear if we would still remain within the bounds in a fully converged configuration. This can be attributed to the lack of a boundary layer mesh. In contrast, the total pressure difference in Figure 7.43b displays a higher convergence rate than the drag coefficient because it characterizes the global flow solution, while the drag is a local measure along the cylinder surface that depends on the spatial gradients. The total pressure difference reaches the same solution as its body-fitted counterpart. For both solutions, we observe that as the mesh is refined, the influence of the Nitsche velocity parameter in (4.26) vanishes. The relative maximum difference between the body-fitted problem and the LSM-XFEM solutions at the finest mesh is 0.4% for the drag coefficient and 0.03% for the total pressure drop. The analysis of our LSM-XFEM framework is sufficiently accurate. Our discretization scheme may suffer from inaccuracies only in local quantities that strongly depend on the resolution of the boundary layer phenomena. In this case, the predicted performance of the final optimization result should be verified using a body-fitted mesh with a resolved boundary layer.

7.5 Verification of the Average Pressure Constraint

In this example, we seek to verify the penalty formulation in (4.28) with respect to the accuracy to which the mass conservation is satisfied. We will model steady-state flow through a

	Value
Element size	$h = \{1.14 \times 10^{-02}, 3.79 \times 10^{-03}, 1.27 \times 10^{-03}, 4.22 \times 10^{-04}\}$
Characteristic velocity	$u_c^f = 0.2$
Characteristic length	$L_c = 0.1$
Dynamic viscosity	$\mu^f = 0.001$
Density	$\rho^f = 1$
Nitsche velocity penalty	$\alpha_{\mathrm{N},\boldsymbol{u}} = \{10, 10^2, 10^3, 10^4\}$
Viscous ghost-penalty	$\alpha_{\mathrm{GP},\mu} = 0.05$
Pressure ghost-penalty	$\alpha_{\mathrm{GP},p} = 0.005$
Convective ghost-penalty	$\alpha_{\mathrm{GP},\boldsymbol{u}} = 0.05$
Pressure constraint parameter	$k_p^f = 0$

Table 7.13: Problem parameters for the flow around a cylinder example (LSM-XFEM).



(a) Cross-section of the original structured mesh, along the $(x, y, 0.205)$ symmetry plane. The red lines denote the zooming area for the figures below.

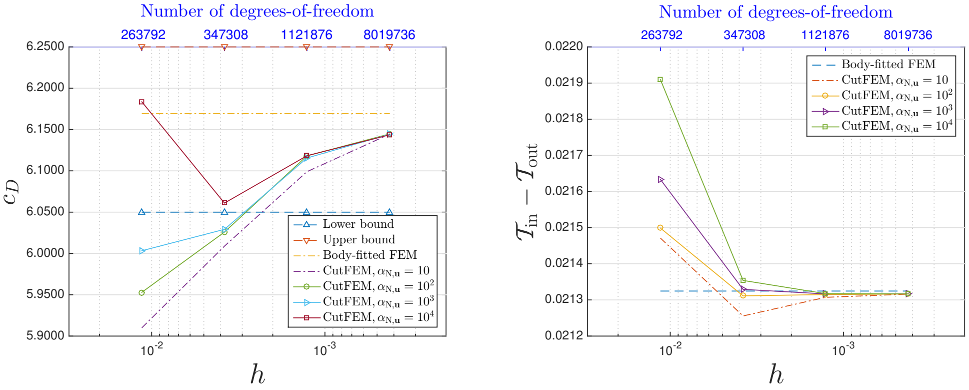
(b) $h = 1.14 \times 10^{-02}$

(c) $h = 3.79 \times 10^{-03}$

(d) $h = 1.27 \times 10^{-03}$

(e) $h = 4.22 \times 10^{-04}$

Figure 7.42: Mesh refinement levels using a local hierarchical mesh refinement for validation of the LSM-XFEM framework. The h values represent the minimum element sizes in the mesh.



(a) Drag coefficient.

(b) Total pressure difference.

Figure 7.43: Comparison of the body-fitted and LSM-XFEM solutions for the flow around a cylinder example.

bent pipe, and measure the relative mass flow rate difference between the inlet and the outlet, for different values of the pressure penalty parameter, k_p^f, and with and without an isolated fluid region.

The problem setup is shown in Figure 7.44. The mesh uses a structured grid, and the level set method is used to describe the pipe and sphere fluid domains. The level set functions to outline these geometries are defined in Burman et al [2014]. The sphere produces a singular analysis problem because the absolute value of the pressure is not governed within it, which results in pressure oscillations. We only model half of the domain, and apply symmetry boundary conditions along the $z = 2.5L$ plane. We study the pressure penalty formulation in (4.28) and, for illustration purposes, a formulation in which the pressure penalty is applied over the entire fluid domain rather than on the isolated "puddles" exclusively. The problem parameters are shown in Table 7.14.

The inflow condition is:

$$\hat{u}_x^f(0, y, z) = u_c^f \cdot \left(\left(-\frac{4}{L_c^2} \right) \cdot \left((y - y_c)^2 + (z - z_c)^2 \right) + 1 \right) \ , \qquad \hat{u}_y^f = \hat{u}_z^f = 0 \ , \tag{7.9}$$

where y_c and z_c are the coordinates at the center of the inflow, as defined in Figure 7.44. The characteristic velocity is $u_c^f = 200$, and the characteristic length is defined as the diameter of the pipe, $l_c = 1$, for a Reynolds number of 200. A traction-free boundary condition is imposed on the outlet. No-slip boundary conditions are imposed on the surfaces of the pipe and of the sphere.

The relative error between the inlet and outlet mass flow rates is shown in Figure 7.45. The flow without the sphere and without the constraint on the pressure results in relative error of 0.03%. Applying the formulation over the entire domain, similar to the approach used by Villanueva and Maute [2014] for linear elasticity problems, can cause a significant error in the mass conservation if a sufficiently large value of k_p^f is chosen. Conversely, if we apply the formulation exclusively to the isolated fluid regions through the use of the indicator field in (4.46), we do not have an accuracy issue, and we eliminate the singular analysis problem. The penalty parameter will be set to $k_p^f = 1$ for all numerical examples below.

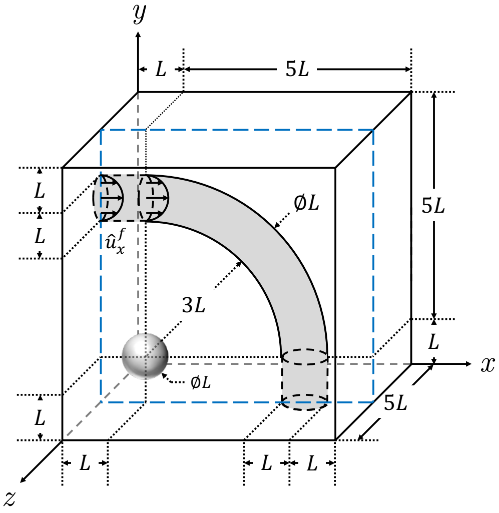


Figure 7.44: Problem setup for the bent pipe example. Blue dashed lines denote the symmetry plane.

	Value
Mesh size	$120 \times 120 \times 50$ (half domain)
Element size	$h = 0.05L$
Characteristic velocity	$u_c^f = 200$
Characteristic length	$L_c = 1$
Dynamic viscosity	$\mu^f = 1$
Density	$\rho^f = 1$
Nitsche velocity penalty	$\alpha_{\mathrm{N},\boldsymbol{u}} = 100$
Viscous ghost-penalty	$\alpha_{\mathrm{GP},\mu} = 0.5$
Pressure ghost-penalty	$\alpha_{\mathrm{GP},p} = 0.05$
Convective ghost-penalty	$\alpha_{\mathrm{GP},\boldsymbol{u}} = 0.5$
Pressure constraint parameter	$k_p^f = \left\{10^{-08}, 10^{-06}, 10^{-04}, 10^{-02}, 1\right\}$

Table 7.14: Problem parameters for the bent pipe example.

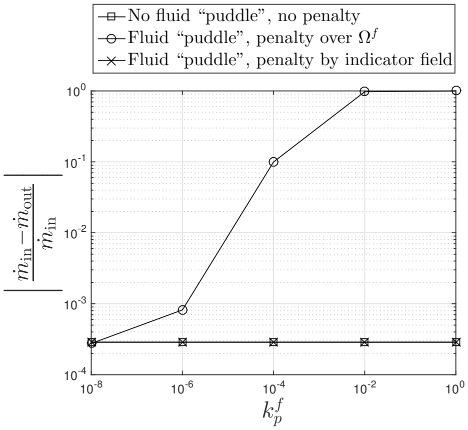


Figure 7.45: Comparison of the influence of the average pressure constraint on the relative mass flow rate difference between the inlet and the outlet for the bent pipe example.

7.6 Incompressible Flow

In the following, we study the characteristics of the proposed LSM-XFEM topology optimization framework for steady-state and transient laminar incompressible flow problems in 3D.

7.6.1 Design of a Manifold with Multiple Outlets

In this example, we consider the design of a steady-state flow bend, with multiple inlets and outlets, where the objective is to minimize the total pressure drop between the inlets and the outlets at steady-state, in order to study the behavior and capabilities of our LSM-XFEM topology optimization framework. The example is the 3D analog to the 2D problem found in Pingen et al [2010]. The problem setup is shown in Figure 7.46. The design domain has 2 inlets (on the left and right), and 4 outlets (on all other planes). The inflow condition is formulated in the same way as (7.9), and the characteristic velocity is $u_c^f = 200$. Traction-free boundary conditions are imposed on the outlets. No-slip boundary conditions are imposed on the fluid-solid interface. We only model an eighth of the domain, and symmetry boundary conditions are imposed on the $x = 3.5L$, $y = 3.5L$, and $z = 3.5L$ planes.

In addition to minimizing the total pressure drop, we minimize the surface area of the fluid-solid interface, and the objective is defined as:

$$\mathcal{Z} = \frac{\sum\limits_{i=1}^{2} \mathcal{T}_{\text{in},i} - \sum\limits_{i=1}^{4} \mathcal{T}_{\text{out},i}}{\left\| \sum\limits_{i=1}^{2} \mathcal{T}^0_{\text{in},i} - \sum\limits_{i=1}^{4} \mathcal{T}^0_{\text{out},i} \right\|} + w_{\mathcal{S}} \frac{\mathcal{S}}{\|\mathcal{S}^0\|} \ , \tag{7.10}$$

where the superscript “0” denotes the values of the initial design, the subscript i denotes the i-th inlet or outlet, and $w_{\mathcal{S}}$ is a constant scaling factor. The use of the surface area as a contribution to the objective function has been applied previously to species transport topology optimization by Makhija and Maute [2014a] to improve the smoothness of the final design, and to regularize the optimization problem.

The design is subject to a 5% volume constraint of the fluid domain to suppress trivial

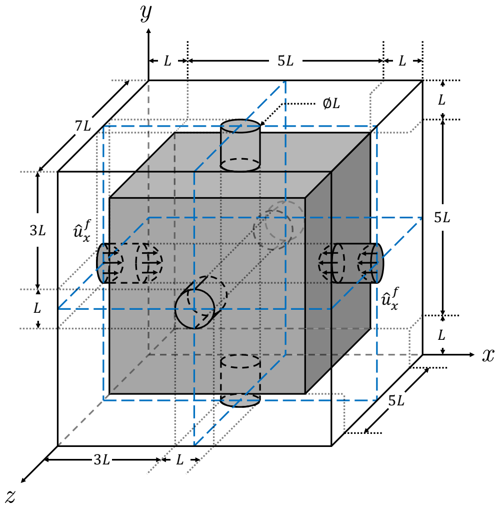
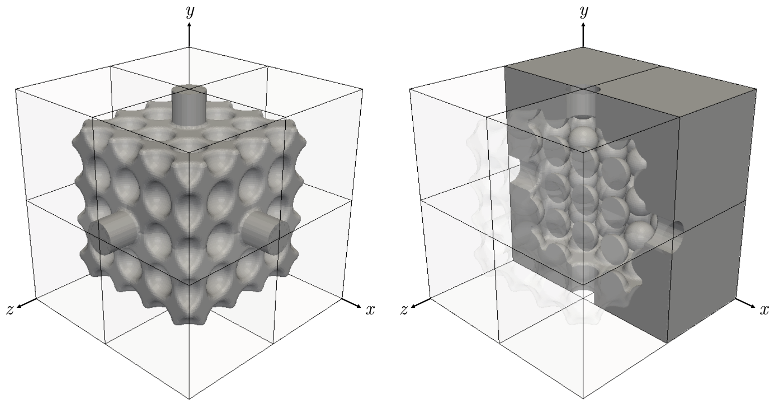


Figure 7.46: Problem setup for the multiple outlets example. Blue dashed lines denote the symmetry planes.



(a) Fluid domain, Ω^f (b) Half of the solid domain, Ω^s

Figure 7.47: Initial design for the multiple outlets example.

solutions, and to promote the formation of distinct fluid channels:

$$g_1 = \frac{\mathcal{V}^f}{0.05\,(\mathcal{V}^f + \mathcal{V}^s)} - 1 \;. \tag{7.11}$$

Further, we wish to impose a constraint such that the amount of mass flow exiting through each outlet is the same. Given that the GCMMA algorithm does not allow equality constraints, we recourse to imposing inequality constraints with lower and upper limits on the mass flow rates. The upper and lower bounds are set to 25% ± 1.25%, respectively, where the tolerance value of ±1.25% was chosen in order to not overconstrain the optimization problem. The constraints are defined as follows:

$$g_{i+1} = 1 - \frac{\dot{m}_{\mathrm{out},i}}{(23.75\%)\,(\dot{m}_{\mathrm{in},1} + \dot{m}_{\mathrm{in},2})} \;, \qquad i = 1 \ldots N_{\mathrm{out}} \;, \tag{7.12}$$

$$g_{i+5} = \frac{\dot{m}_{\mathrm{out},i}}{(26.25\%)\,(\dot{m}_{\mathrm{in},1} + \dot{m}_{\mathrm{in},2})} - 1 \;, \qquad i = 1 \ldots N_{\mathrm{out}} \;, \tag{7.13}$$

where $\dot{m}_{\mathrm{in},i}$ and $\dot{m}_{\mathrm{out},i}$ are the mass flow rates at the i-th inlet and i-th outlet, respectively, and N_{out} is the number of outlets.

The remaining parameters are given in Table 7.15. The design domain is initialized with 5 × 5 × 5 spherical solid inclusions of radii $0.5L$, as shown in Figure 7.47. In our experience, the flow topology optimization problems studied here are rather insensitive to the initial design as long

	Value
Mesh size	$56 \times 56 \times 56$ (quarter domain)
Element size	$h = 0.0625L$
Characteristic velocity	$u_c^f = 200$
Characteristic length	$L_c = 1$
Dynamic viscosity	$\mu^f = 1$
Density	$\rho^f = 1$
Nitsche velocity penalty	$\alpha_{\mathrm{N},\boldsymbol{u}} = 100$
Viscous ghost-penalty	$\alpha_{\mathrm{GP},\mu} = 0.5$
Pressure ghost-penalty	$\alpha_{\mathrm{GP},p} = 0.05$
Convective ghost-penalty	$\alpha_{\mathrm{GP},\boldsymbol{u}} = 0.5$
Pressure constraint parameter	$k_p^f = 1$
Surface area scaling weight	$w_{\mathcal{S}} = 0.01$
Volume constraint	5%
Number of design variables	$132,651$ (quarter domain)
Design variables bounds	$s_i^L = -0.03125L$, $s_i^U = +0.03125L$
Smoothing filter radius	$r_\phi = 2.4h$

Table 7.15: Problem parameters for the multiple outlets example.

as the number of inclusions is sufficiently large. Additional mechanisms for seeding solid inclusions could be added to the proposed LSM-XFEM framework, such as topological derivatives [Sá et al, 2016]; however, they are outside the scope of this study.

The converged optimized design after 100 iterations is shown in Figure 7.48, which we can observe resembles well the results from Pingen et al [2010]. Figure 7.49a shows the convergence plots of the objective and the volume constraint. The initialization process described above leads to an initial design that violates the volume constraint. Initially, the objective is increased while lowering the volume constraint value. Once the constraint is satisfied, the objective is reduced until a feasible minimum is found. The objective changes from a normalized value of 1.1, where the mass constraint was violated, to 1.56. Figures 7.49b and 7.49c show the convergence plots for the lower and upper bounds of the mass inequality constraints. We can observe that the mass flow rate constraints are also satisfied, and the amount of fluid flow exiting through each outlet is virtually the same. The mass flow rate at the inlets is 6.24, while the rates at each of the outlets is 1.56.

7.6.2 Design of a Manifold with Variable Outlets

In the context of flow topology optimization, our LSM-XFEM framework allows us to impose boundary conditions on inflow and outflow surfaces that may vary in position, shape, and/or size during the optimization process. Because we solve our optimization problem via nonlinear programming methods, we can introduce these variables as parameters in our optimization problem in addition to the level set parameters. In this example, we study a fluid flow problem with one inlet and multiple outlets, similar to the one from Example 7.6.1, with the caveat that the outlets are allowed to vary in position and size by using the level set parametrization of (2.11). The outlets are described as cylinders.

The problem setup is shown in Figure 7.50. Fluid flows into the domain through the top inlet, and exits through the 2 outlets on the left and right, and the 2 outlets on the bottom. The problem parameters, the inflow, outflow, and interface boundary conditions, and the objective are the same as in Example 7.6.1. We model a quarter of the domain, and impose symmetry boundary

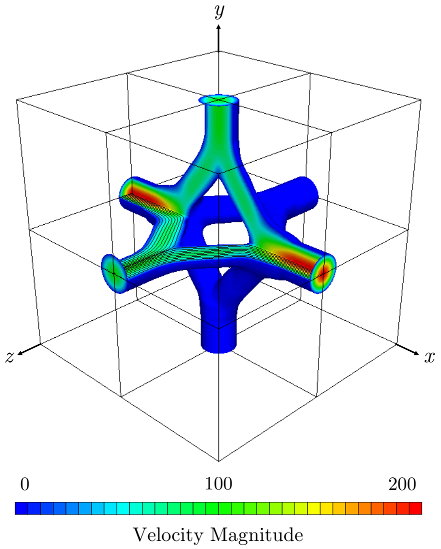
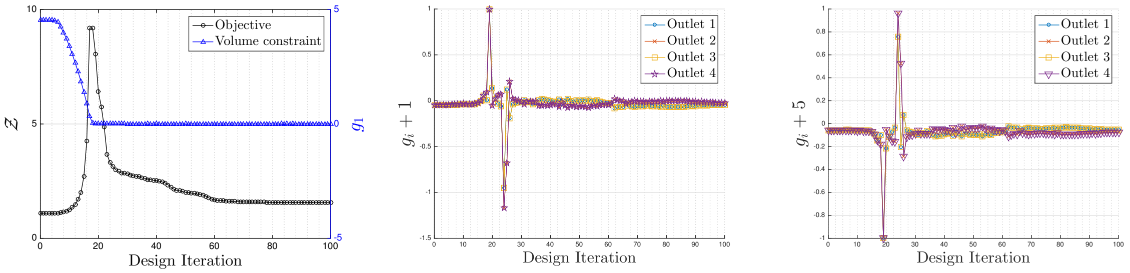


Figure 7.48: Velocity magnitude, with streamlines, of the optimized material layout for the multiple outlets example. A section of the design was removed for visualization purposes.



(a) Objective and volume constraint. (b) Lower bounds of mass inequality constraints. (c) Upper bounds of mass inequality constraints.

Figure 7.49: Convergence plots of the objective and constraints for the multiple outlets example.

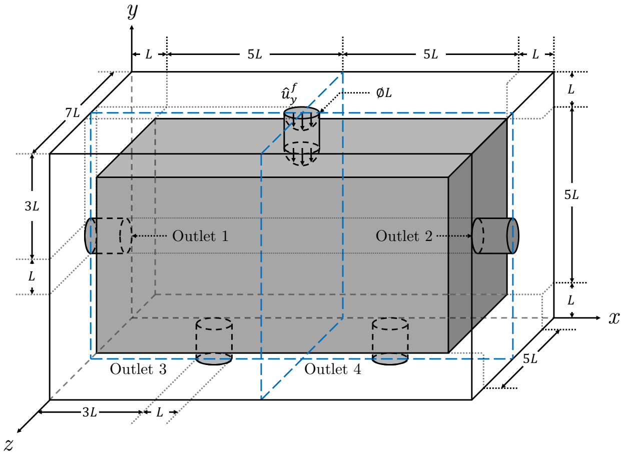


Figure 7.50: Problem setup for the variable outlets example. Blue dashed lines denote the symmetry planes.

conditons along the $x = 6L$ and the $z = 3.5L$ planes. The design domain is initialized with $3 \times 6 \times 1$ spherical solid inclusions of radii $0.625L$. The volume constraint (7.11) is set to 20%.

The optimization problem has 3 additional design variables per outlet. These design variables control the in-plane coordinates at the center of the outlets and their radii; the lower and upper bounds are given in Table 7.16. The rest of the domain is parametrized by (2.9), and the lower and upper bounds of the design variables are the same as in Table 7.15. We impose constraints on the mass flow rates such that the left and right outlets have 33.333% each, and the bottom outlets have 16.667% each, of the fluid flow entering through the inlet. Similar to (7.12), we formulate these limits as inequality constraints and initially set the lower and upper tolerances, $\dot{m}^L$ and $\dot{m}^U$, to $\mp 1.667\%$ and $\mp 0.833\%$, respectively, to not overconstrain the problem in the initial stages of the optimization process. After the optimization process converges, we use a continuation approach: we increase/decrease the tolerances every 50 iterations first to $\mp 0.833\%$ and $\mp 0.417\%$, respectively, and then to $\mp 0.333\%$ and $\mp 0.167\%$, to increase the accuracy to which the limits are satisfied. This new formulation is defined as:

$$\begin{aligned}
g_2 &= 1 - \frac{\dot{m}_{\text{out},1}}{(33.333\% - \dot{m}^L)\,\dot{m}_{\text{in}}} \ , & g_6 &= \frac{\dot{m}_{\text{out},1}}{(33.333\% + \dot{m}^U)\,\dot{m}_{\text{in}}} - 1 \ , \\
g_3 &= 1 - \frac{\dot{m}_{\text{out},2}}{(33.333\% - \dot{m}^L)\,\dot{m}_{\text{in}}} \ , & g_7 &= \frac{\dot{m}_{\text{out},2}}{(33.333\% + \dot{m}^U)\,\dot{m}_{\text{in}}} - 1 \ , \\
g_4 &= 1 - \frac{\dot{m}_{\text{out},3}}{(16.667\% - \dot{m}^L)\,\dot{m}_{\text{in}}} \ , & g_8 &= \frac{\dot{m}_{\text{out},3}}{(16.667\% + \dot{m}^U)\,\dot{m}_{\text{in}}} - 1 \ , \\
g_5 &= 1 - \frac{\dot{m}_{\text{out},4}}{(16.667\% - \dot{m}^L)\,\dot{m}_{\text{in}}} \ , & g_9 &= \frac{\dot{m}_{\text{out},4}}{(16.667\% + \dot{m}^U)\,\dot{m}_{\text{in}}} - 1 \ .
\end{aligned} \tag{7.14}$$

The converged optimized material layout after 350 iterations is shown in Figure 7.51. The mass flow through each outlet, and the radii and coordinates of the outlets are shown in Table 7.17. In the optimized material layout, the left and right outlets increase their radii and move up in the y-direction. The bottom outlets merge and form a single outlet; however, the numbers in Table 7.17 reveal that they do not fully overlap one another, and that they have moved away from the $x = 6L$ and $z = 3.5L$ planes of symmetry. The relative difference between their final positions and the axes of symmetry is small nonetheless, in the order of 0.1%. This value got increasingly closer to 0 as we varied the mass flow rate tolerances of (7.14). The mass flow exiting through each of

	x_c^L	x_c^U	y_c^L	y_c^U	z_c^L	z_c^U	r_c^L	r_c^U
Outlet 1	N/A	N/A	L	$6L$	L	$6L$	0	$2.5L$
Outlet 2	N/A	N/A	L	$6L$	L	$6L$	0	$2.5L$
Outlet 3	L	$11L$	N/A	N/A	L	$6L$	0	$2.5L$
Outlet 4	L	$11L$	N/A	N/A	L	$6L$	0	$2.5L$

Table 7.16: Lower and upper bounds of the design variables that control the position and size of the ports in the example with variables outlets.

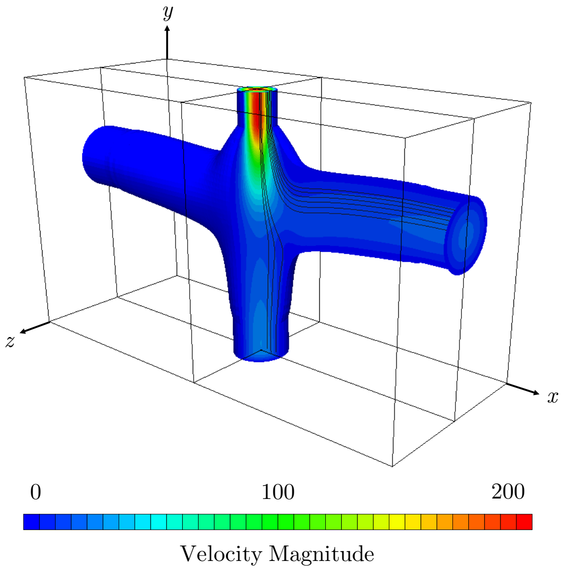


Figure 7.51: Velocity magnitude, with streamlines, of the optimized material layout for the variable outlets example. A section of the design was removed for visualization purposes.

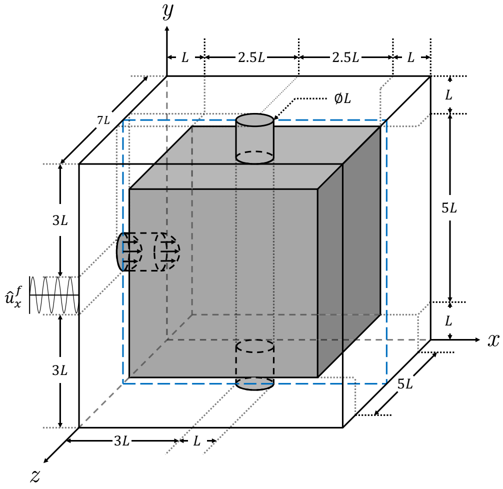


Figure 7.52: Problem setup for the transient pump example. Blue dashed lines denote the symmetry plane.

the left and right outlets is 33.2% of the mass flow entering through the inlet, and the mass flow exiting through each of the bottom outlets is 16.8%; these value are within the lower and upper bounds of the mass flow limits.

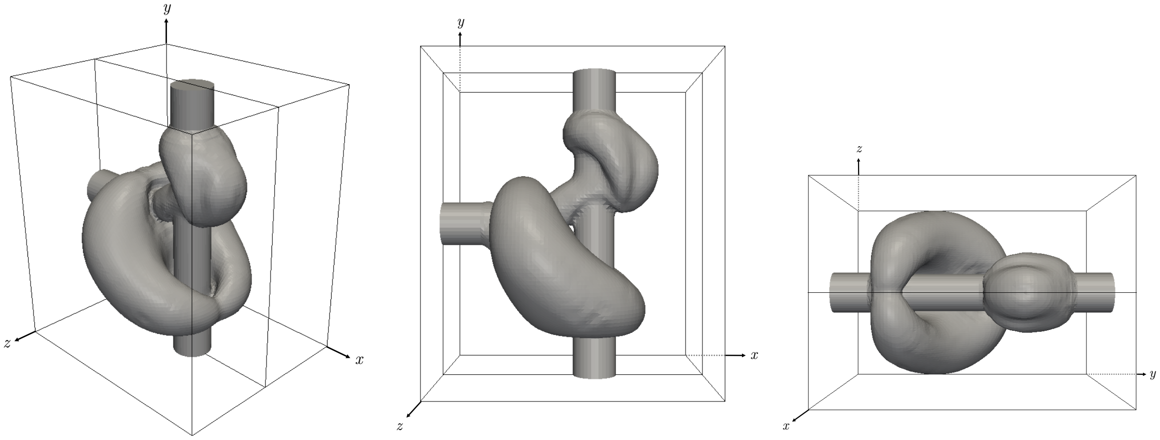
7.6.3 Design of a Transient Pump

In this example, we study the applicability of the LSM-XFEM framework to problems with transient behavior. We consider the problem of optimizing a simplified fluid pump. The example is the 3D analog to the 2D problem found in Nørgaard et al [2016]. The problem setup is shown in Figure 7.52. The design domain has an inlet (on the left), and 2 ports (on the top and bottom) at which traction-free boundary conditions are applied. The basic idea is to prescribe a harmonically oscillating inflow velocity, and optimize the design domain to maximize the amount of fluid that is transported through the port at the top. The port at the bottom effectively represents a reservoir from which additional fluid can enter the domain. The inflow condition is defined as:

$$\hat{u}_x^f\left(0,y,z,t\right) = u_c^f \cdot \sin\left(\frac{\pi t}{v}\right) \cdot \left(\left(-\frac{4}{L_c^2}\right) \cdot \left(\left(y-y_c\right)^2 + \left(z-z_c\right)^2\right) + 1\right) \ , \qquad \hat{u}_y^f = \hat{u}_z^f = 0 \ , \tag{7.15}$$

	$\dot{m}$	Fraction	Radius	x	y	z
Inlet	3.135	100%	$0.500L$	$6L$	$7L$	$3.5L$
Outlet 1	1.040	33.2%	$0.855L$	$12L$	$4.485L$	$3.557L$
Outlet 2	1.040	33.2%	$0.855L$	$0L$	$4.485L$	$3.557L$
Outlet 3	0.528	16.8%	$0.741L$	$6.024L$	$0L$	$3.546L$
Outlet 4	0.528	16.8%	$0.741L$	$5.976L$	$0L$	$3.546L$

Table 7.17: Mass flow rate, fraction of the mass flow entering through the inlet, radii, and coordinates of the outlets in the variable outlets example.



(a) Optimized layout. (b) xy plane view. (c) yz plane view.

Figure 7.53: Optimized material layout for the transient pump example.

where v is some frequency. No-slip boundary conditions are imposed on the fluid-solid interface. We only model a half of the domain, and symmetry boundary conditions are imposed on the $z = 3.5L$ plane.

We model a single pumping cycle with 8 time iterations. Numerical experiments showed that this number is sufficient to model the harmonic behavior of the flow. The objective is to maximize the mass flow through the top outlet, and to minimize the surface area of the fluid-solid interface:

$$\mathcal{Z} = -\frac{\frac{1}{N_t}\sum_{n=0}^{N_t} \dot{m}_{\text{out},1}\left(t_n\right)}{\left\|\frac{1}{N_t}\sum_{n=0}^{N_t} \dot{m}^0_{\text{out},1}\left(t_n\right)\right\|} + w_{\mathcal{S}}\frac{\mathcal{S}}{\|\mathcal{S}^0\|} \ , \tag{7.16}$$

where N_t is the total number of time iterations. The design domain is subject to a volume constraint (7.11) of 15% to suppress trivial solutions and to promote the formation of smooth fluid channels. The remaining problem parameters are given in Table 7.18. The design is initialized with the same layout as in Example 7.6.1.

The converged optimized layout after 225 iterations is shown in Figure 7.53. Similar to the results from Nørgaard et al [2016], the design exhibits a narrowing of the inflow channel, and the formation of a central reservoir from which fluid flows towards the pumping outlet during the outflow cycle. During the inflow cycle, fluid flows from the left towards the pumping outlet,

	Value
Mesh size	$72 \times 84 \times 30$ (half domain)
Element size	$h = 0.08333L$
Characteristic velocity	$u_c^f = 200$
Characteristic length	$L_c = 1$
Dynamic viscosity	$\mu^f = 1$
Density	$\rho^f = 1$
Inlet velocity frequency	$v = 4$
Time step	$\Delta t = 1$
Number of time iterations	$N_t = 8$
Nitsche velocity penalty	$\alpha_{\mathrm{N},\boldsymbol{u}} = 100$
Viscous ghost-penalty	$\alpha_{\mathrm{GP},\mu} = 0.05$
Pressure ghost-penalty	$\alpha_{\mathrm{GP},p} = 0.005$
Convective ghost-penalty	$\alpha_{\mathrm{GP},\boldsymbol{u}} = 0.05$
Pressure constraint parameter	$k_p^f = 1$
Surface area objective weight	$w_{\mathcal{S}} = 0.1$
Volume constraint	15%
Number of design variables	$226,981$ (half domain)
Design variables bounds	$s_i^L = -0.041667L$, $s_i^U = +0.041667L$
Smoothing filter radius	$2.4h$

Table 7.18: Problem parameters for the transient pump example.

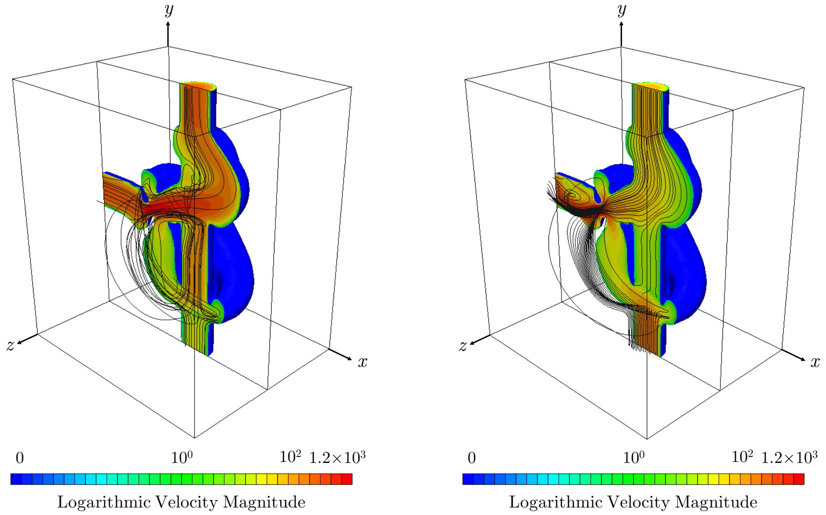
while additional fluid is pulled from the bottom reservoir (bottom open boundary) towards the central reservoir. An area of recirculation then forms around the central reservoir. During the outflow phase, fluid flows from the central reservoir towards the pumping outlet, while fluid from the bottom reservoir flows in a vortex like path around the left inlet; this fluid is then transported towards the pumping outlet during the next inflow phase. The process by which this is achieved is shown by streamlines in Figure 7.54. The flow does not exit through the bottom outlet at any of the cycles; the average mass flow rate at the final design is $+0.328$, with the positive sign indicating that the flow is entering the domain. Contrarily, the average mass flow rate at the pumping outlet is -0.328. In contrast to the results shown by Nørgaard et al [2016], our pump design does not suffer from numerical artifacts in the optimized material layout, such as isolated regions of fluid flow.

7.7 Species Transport

In this section, we study the characteristics of our framework in the context of species transport problems. The species field is modeled using the energy and species transport equations in (4.29).

7.7.1 Design of a Micromixer

In this example, we apply the framework to the modeling and optimizing of a micromixer at a low Reynolds number for steady-state conditions. The example is the 3D analog to the 2D problem found in Makhija and Maute [2014a], and is similar to the micromixer studies with flow topology optimization found in Łaniewski-Wołłk and Rokicki [2015] and Liu et al [2013]. The problem setup is shown in Figure 7.55. A "red" fluid and a "blue" fluid enter the design domain through the left inlet and exit it through the lower right side. We assume that the fluids are ideally miscible and have identical flow properties. The "red" fluid is represented by a species concentration value of $\hat{T}^f = 1$, and the "blue" fluid by $\hat{T}^f = 0$. We do not consider diffusion of the species field through the solid phase. The inflow condition is formulated using the same approach as in Example 7.5.



(a) Inflow cycle.

(b) Outflow cycle.

Figure 7.54: Velocity streamlines for half the domain of the transient pump example.

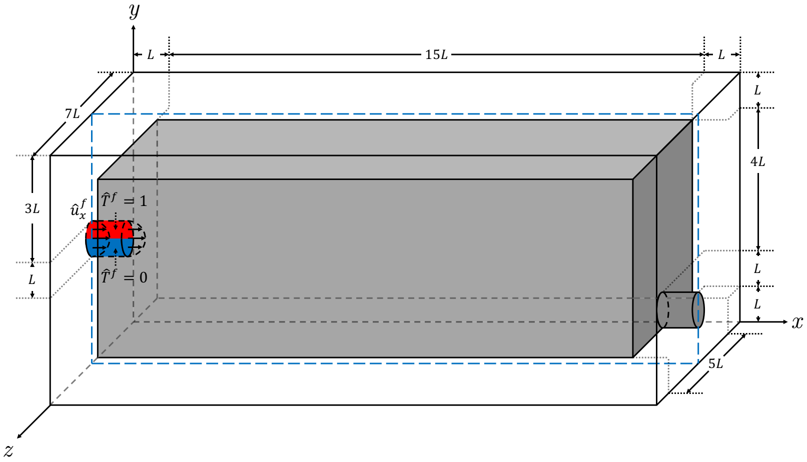


Figure 7.55: Problem setup for the micromixer example. Blue dashed lines denote the symmetry plane.

No-slip boundary conditions are applied at the fluid-solid interface, and a traction-free boundary condition is imposed on the outlet. We only model a half of the domain, and symmetry boundary conditions are imposed on the $z = 3.5L$ plane. An adiabatic condition is imposed on the fluid-solid interface for the species field.

The objective uses the target scalar value formulation from (4.74) at steady-state, and is defined as:

$$\mathcal{Z} = \frac{\mathcal{K}_{\text{out}}}{\|\mathcal{K}_{\text{out}}^0\|} + w_{\mathcal{S}} \frac{\mathcal{S}}{\|\mathcal{S}^0\|} \ , \tag{7.17}$$

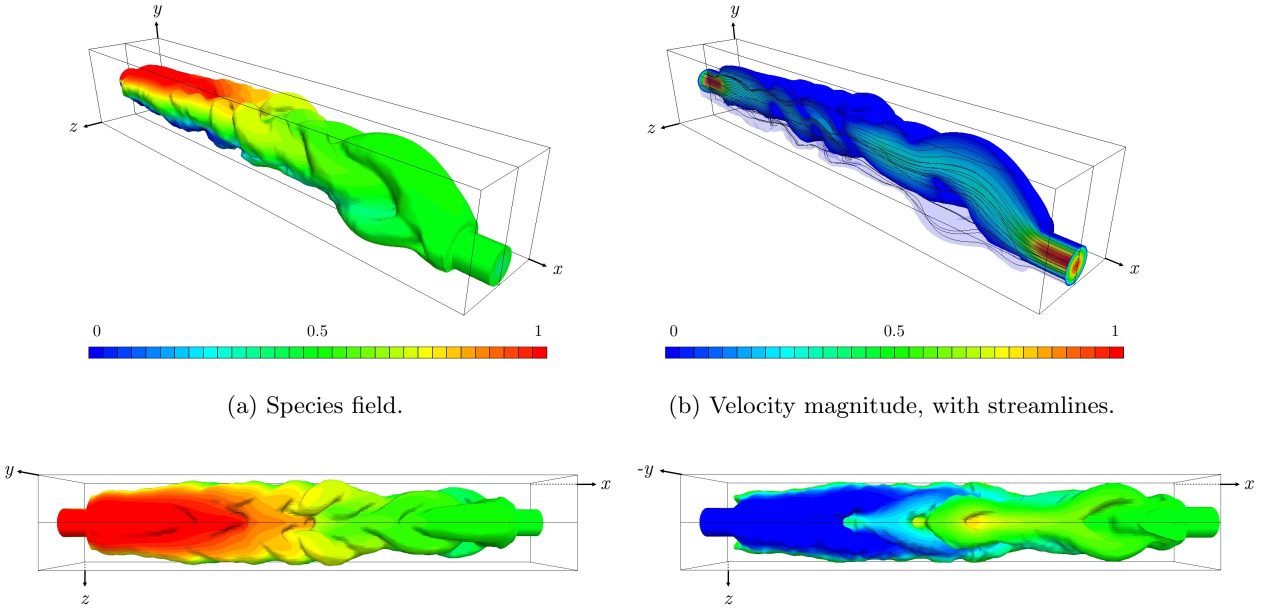
with $\beta_K = 400$, and $T_{\text{ref}}^f = 0.5$. A small surface area penalty is applied to regularize the optimization problem. The design is subject to a volume constraint (7.11) of 35% to supress trivial solutions, and to promote the formation of distinct fluid channels. Similar to Makhija and Maute [2014a], a constraint is imposed on the maximum pressure drop to prevent the formation of small geometric features:

$$g_2 = \frac{\mathcal{T}_{\text{in}} - \mathcal{T}_{\text{out}}}{\Delta p_{\text{ref}}^f} - 1 \ . \tag{7.18}$$

The problem is initialized with $15 \times 5 \times 5$ spherical solid inclusions of radii $0.5L$, similar to the previous examples. The remaining parameters are given in Table 7.19.

	Value
Mesh size	$226 \times 40 \times 20$ (half domain)
Element size	$h = 0.125L$
Characteristic velocity	$u_c^f = 1$
Characteristic length	$L_c = 1$
Dynamic viscosity	$\mu^f = 1$
Density	$\rho^f = 1$
Specific heat capacity	$c_p^f = 1$
Thermal conductivity	$k^f = 0.001$
Nitsche velocity penalty	$\alpha_{\mathrm{N},\boldsymbol{u}} = 100$
Nitsche species penalty	$\alpha_{\mathrm{N},c} = 1$
Viscous ghost-penalty	$\alpha_{\mathrm{GP},\mu} = 0.05$
Pressure ghost-penalty	$\alpha_{\mathrm{GP},p} = 0.005$
Convective ghost-penalty	$\alpha_{\mathrm{GP},\boldsymbol{u}} = 0.05$
Species ghost-penalty	$\alpha_{\mathrm{GP},c} = 0.05$
Pressure constraint parameter	$k_p^f = 1$
Pressure constraint reference	$\Delta p_{\mathrm{ref}}^f = 30$
Surface area objective weight	$w_{\mathcal{S}} = 0.001$
Volume constraint	35%
Number of design variables	$160,000$ (half domain)
Design variables bounds	$s_i^L = -0.0625L$, $s_i^U = +0.0625L$
Smoothing filter radius	$2.4h$

Table 7.19: Problem parameters for the micromixer example.



(a) Species field.

(b) Velocity magnitude, with streamlines.

(c) Species field, top view.

(d) Species field, bottom view.

Figure 7.56: Optimized material layout for the micromixer example.

The optimized material layout, along with the species field and the velocity magnitude, is shown in Figure 7.56. The average species concentration at the outlet is 0.46. Analogous to the 2D results from Makhija and Maute [2014a], the length of the channel layout increases by creating an intricate wavy design, thereby increasing the path traveled by the fluids. This lengthening mechanism is the key to enhance the mixing of the fluids in 3D laminar flows. The number of iterations is rather large: $1,500$ iterations are required to form the channel layout and fine-tune its shape. This behavior has also been seen by Makhija and Maute [2014a], who attributes this to the interplay of localized sensitivities along the fluid-solid interface and the volume constraint on the fluid phase. In contrast to the density approach of Makhija et al [2012], we do not obtain numerical artifacts in the optimized design.

7.7.2 Design of a Species Separator

In this example, we apply the framework to the modeling and optimizing of a species separator at a low Reynolds number for steady-state conditions. The problem setup is shown in Figure 7.57. A "red" fluid and a "blue" fluid enter the design domain through the left inlets and exit it through the right outlets. The objective is find a channel layout that prevents the mixing of the species fields, such that the "red" fluid exits through the bottom outlet, and the "blue" fluid exits through the top one. We assume that the fluids are ideally miscible and have identical flow properties. The "red" fluid is represented by a species concentration value of $\hat{T}^f = 1$, and the "blue" fluid by $\hat{T}^f = 0$. We do not consider diffusion of the species field through the solid phase. The inflow conditions are formulated using the same approach as in Example 7.5. No-slip boundary conditions are applied at the fluid-solid interface, and a traction-free boundary condition is imposed on the outlets. We model the full domain, and no symmetry boundary conditions are imposed. An adiabatic condition is imposed on the fluid-solid interface for the species field.

The objective formulation requires that we minimize the largest temperature difference between each pair of inlet and outlet for both the "red" and "blue" fluids. To achieve this, we utilize

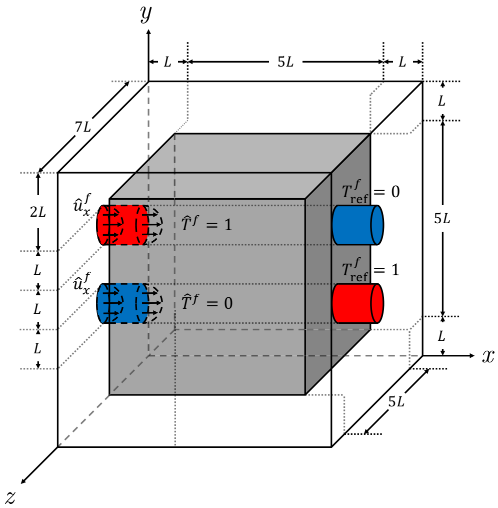


Figure 7.57: Problem setup for the species separator example.

the min-max objective formulation proposed by Svanberg [2007], and the objective is defined as:

$$\mathcal{Z} = \beta_{\mathcal{D}} \ , \tag{7.19}$$

where $\beta_{\mathcal{D}}$ is a design variable in addition to the parameters of the LSF. The constraints are formulated as:

$$g_1 = \alpha_{\mathcal{D}} \mathcal{D}_1 - \beta_{\mathcal{D}} \ , \tag{7.20}$$

$$g_2 = \alpha_{\mathcal{D}} \mathcal{D}_2 - \beta_{\mathcal{D}} \ , \tag{7.21}$$

where $\alpha_{\mathcal{D}}$ is a constant parameter. In our experience, this parameter must be chosen large enough such that the constraints remain violated for the duration of the optimization process, but not too large in order to have a well-posed optimization problem. Here, we use $\alpha_{\mathcal{D}} = 100$. No other constraints are applied in the problem. The term $\mathcal{D}_i$ is defined as:

$$\mathcal{D}_i = \int_{\Gamma_i} \left(T^f - T^f_{\text{ref}} \right)^2 \mathrm{d}\Gamma \ , \tag{7.22}$$

with the subscript i representing the i-th outlet. The problem is initialized with $5 \times 5 \times 5$ spherical solid inclusions of radii $0.5L$, similar to Example 7.6.1. The remaining parameters are given in Table 7.20.

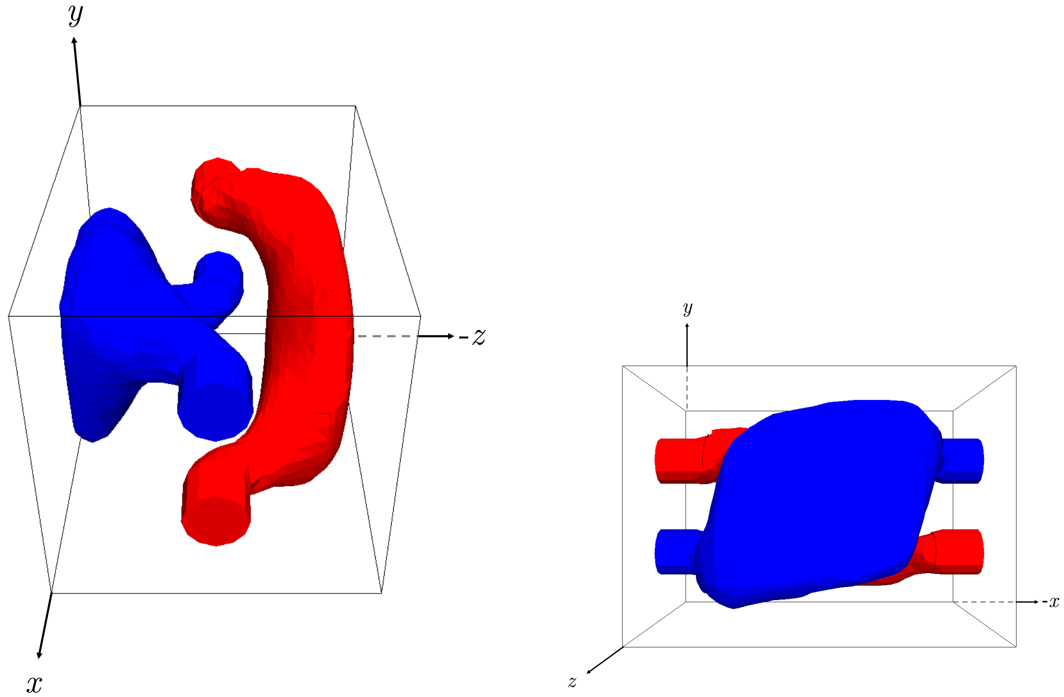
The converged optimized material layout after 100 iterations is shown in Figure 7.58. The geometry reaches an intuitive design, and the optimizer forms individual channels the separate and route the species concentrations to their respective outlets. Numerical experiments showed that utilizing an objective formulation similar to the one used in previous examples did not succeed in separating the species fields.

7.8 Natural Convection

In this section, we study the characteristics of our framework in the context of heat transfer problems dominated by natural convection.

	Value
Mesh size	$28 \times 28 \times 28$
Element size	$h = 0.25L$
Characteristic velocity	$u_c^f = 1$
Characteristic length	$L_c = 1$
Dynamic viscosity	$\mu^f = 1$
Density	$\rho^f = 1$
Specific heat capacity	$c_p^f = 1$
Thermal conductivity	$k^f = 0.001$
Nitsche velocity penalty	$\alpha_{\mathrm{N},\boldsymbol{u}} = 100$
Nitsche species penalty	$\alpha_{\mathrm{N},T} = 1$
Viscous ghost-penalty	$\alpha_{\mathrm{GP},\mu} = 0.05$
Pressure ghost-penalty	$\alpha_{\mathrm{GP},p} = 0.005$
Convective ghost-penalty	$\alpha_{\mathrm{GP},\boldsymbol{u}} = 0.05$
Species ghost-penalty	$\alpha_{\mathrm{GP},T} = 0.05$
Pressure constraint parameter	$k_p^f = 1$
Number of design variables	$9,261$
Design variables bounds	$s_i^L = -0.125L$, $s_i^U = +0.125L$
Smoothing filter radius	$2.4h$

Table 7.20: Problem parameters for the species separator example.



(a) Species field.

(b) Species field, xy plane view.

Figure 7.58: Optimized material layout for the species separator example.

7.8.1 Design of a Passive Cooler

This example studies the design of passive coolers where the energy transport in the fluid is dominated by natural convection. The objective of this example is to find the geometry of a cooler such that the temperature of a given solid material, which is subject to an internal heat source, is minimized. The problem setup is the level set-based equivalent to Alexandersen [2015] where the design problem is solved by a density approach.

The problem setup is shown in Figure 7.59. The heat source is modeled as a solid slab of aluminium with a uniform volumetric heat flux of $\hat{q}^s_\Omega = 1\,\mathrm{W}$. The design domain is defined by a cubic box of length 100mm placed on top of the heat source. The source and the cooler are modeled as being suspended in free space, and traction-free boundary conditions are imposed on all boundaries with the exception of the bottom-most boundary, where a no-slip boundary condition is applied. A quarter domain symmetry is imposed along the $x = 2.5L$ and $z = 2.5L$ planes. The temperature field is imposed to be equal to the reference room temperature at all boundaries, except at the top-most boundary, which is adiabatic. The reference temperature is set to 25 °C, and properties of air and aluminium at this reference temperature are used. The heat flux is modeled in both the fluid and the solid domains. We utilize a hierarchical mesh refinement, and set the resolution of the design domain to an element size of 0.002777m, and of the rest of the domain to 0.025m. The problem parameters are shown in Table 7.21. For an explanation on the choice of the problem parameters and the domain setup, refer to Alexandersen [2015].

The objective of the optimization problem is to maximize the thermal compliance (4.75) at steady-state of the solid phase, $\mathcal{Q}^s$, and is defined as:

$$\mathcal{Z} = -\mathcal{Q}^s \ . \tag{7.23}$$

Constraints are imposed on the maximum volume of the solid and surface values, to half the total box volume and thrice the total surface area of the design domain, respectively, and are defined as:

$$g_1 = \frac{\mathcal{V}^s}{1.0308 \times 10^{-04}} - 1 \ , \qquad g_2 = \frac{\mathcal{S}}{3.6717 \times 10^{-02}} - 1 \ . \tag{7.24}$$

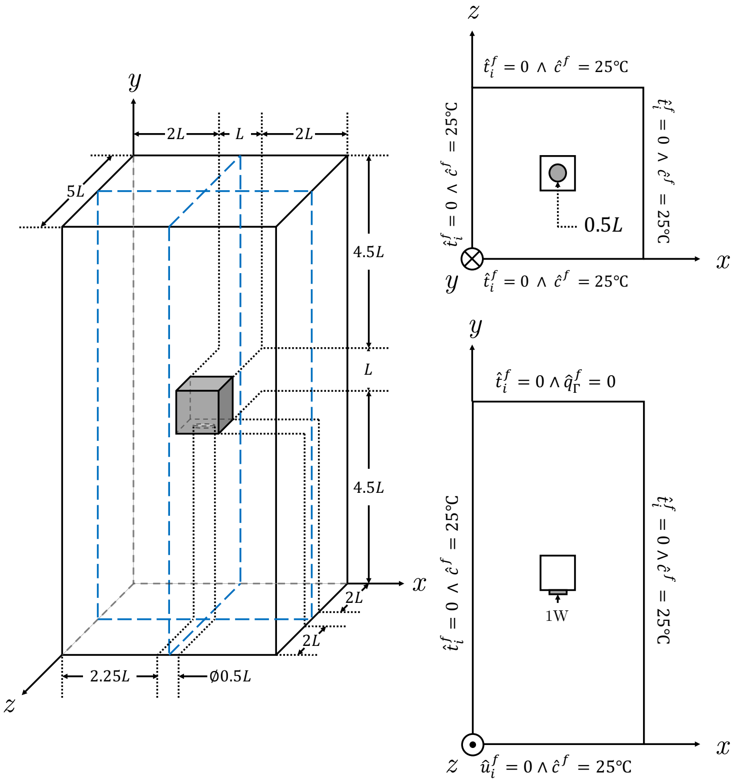

y
2L
L
2L
5L
4.5L
L
4.5L
x
2L
2L
z
2.25L
∅0.5L
z
$\hat{t}_i^f = 0 \wedge \hat{c}^f = 25$°C
$\hat{t}_i^f = 0 \wedge \hat{c}^f = 25$°C
$\hat{t}_i^f = 0 \wedge \hat{c}^f = 25$°C
0.5L
x
y
$\hat{t}_i^f = 0 \wedge \hat{c}^f = 25$°C
y
$\hat{t}_i^f = 0 \wedge \hat{q}_\Gamma^f = 0$
$\hat{t}_i^f = 0 \wedge \hat{c}^f = 25$°C
$\hat{t}_i^f = 0 \wedge \hat{c}^f = 25$°C
1W
x
z
$\hat{u}_i^f = 0 \wedge \hat{c}^f = 25$°C

Figure 7.59: Problem setup for the passive cooler example. Blue dashed lines denote the symmetry plane. The grey cubic box denotes the design domain. $L = 100\,\text{mm}$.

	Value
Element size	$h = 0.002777$m (in design domain)
Characteristic length	$L_c = 100$mm
Dynamic viscosity	$\mu^f = 1.511 \times 10^{-05} \frac{\text{kg}}{\text{m}\cdot\text{s}}$
Density	$\rho^f = 1.205 \frac{\text{kg}}{\text{m}^3}$ $\rho^s = 2.700 \times 10^3 \frac{\text{kg}}{\text{m}^3}$
Specific heat capacity	$c_p^f = 1.005 \times 10^3 \frac{\text{J}}{\text{kg} \cdot \text{K}}$ $c_p^s = 0.910 \times 10^3 \frac{\text{J}}{\text{kg} \cdot \text{K}}$
Thermal conductivity	$k^f = 257 \times 10^{-04} \frac{\text{W}}{\text{m} \cdot \text{K}}$ $k^s = 237 \frac{\text{W}}{\text{m} \cdot \text{K}}$
Gravitational vector	$\boldsymbol{g} = \{0, -9.81, 0\} \frac{\text{m}}{\text{s}^2}$
Thermal expansion coefficient	$h_T^f = 3.43 \times 10^{-03} \frac{1}{\text{K}}$
Reference temperature field	$T_\infty^f = 25\,°\text{C}$
Nitsche velocity penalty	$\alpha_{\text{N},\boldsymbol{u}} = 100$
Nitsche temperature penalty	$\alpha_{\text{N},T} = 10$
Viscous ghost-penalty	$\alpha_{\text{GP},\mu} = 0.005$
Pressure ghost-penalty	$\alpha_{\text{GP},p} = 0.0005$
Convective ghost-penalty	$\alpha_{\text{GP},\boldsymbol{u}} = 0.005$
Temperature ghost-penalty	$\alpha_{\text{GP},T} = 0.005$
Pressure constraint parameter	$k_p^f = 1$
Maximum volume constraint	$1.0308 \times 10^{-04}\text{m}^3$
Maximum surface area constraint	$3.6717 \times 10^{-02}\text{m}^2$
Number of design variables	$11,340$ (quarter domain)
Design variables bounds	$s_i^L = -h/2$, $s_i^U = +h/2$
Smoothing filter radius	$2.4h$

Table 7.21: Problem parameters for the passive cooler example.

The initial design consists of a semisphere of radius 25mm placed on top of the solid slab.

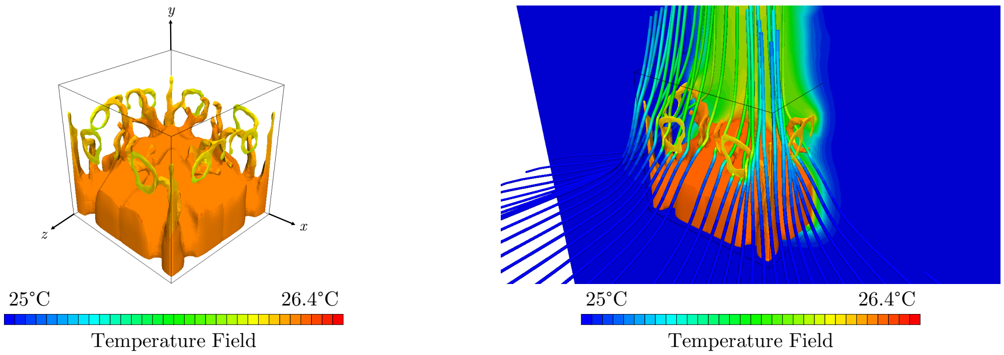
Numerical experiments studied in Coffin and Maute [2015b] showed that this class of conjugate heat transfer problems lead to designs that form very thin features, which affects the convergence of the optimization problem. To prevent the formation of subelement-size features, we utilize the gradient measure in (6.24) to regularize the optimization problem. However, imposing strict limits on the gradient measure may prevent the occurrence of changes in the topology of the design during the optimization process [Coffin and Maute, 2015b]. To mitigate this issue, we adopt a continuation approach, where we start with a large constraint limit, and successively lower the limit in the course of the optimization process. Our gradient measure constraint is then formulated as follows:

$$g_3 = \frac{\mathcal{G}}{c_{\mathcal{G}}} - 1 \ . \tag{7.25}$$

Initially, no constraint is imposed and the value of $c_{\mathcal{G}}$ is set to ∞. As the design evolves during the optimization process, and the minimum feature size drops below the length of an element (after 100 design iterations), the value of $c_{\mathcal{G}}$ is modified to 1.5×10^{-04}.

The converged material layout of the cooler after 275 iterations is shown in Figure 7.60. The thermal compliance, $\mathcal{Q}^s$, increases from 0.56 to 2.89. The Grashof number of the fluid measured at the top edge of the design domain is 3.75×10^4. The optimized design exhibits tree-like branches extending out from the center of the design domain, which conduct the heat away from the heat source and transfer it to the flowing air by allowing the flow to move between the branches. Figure 7.60b shows the fluid flowing through the branches in a zig-zag fashion in the vertical direction. Figure 7.61 shows the velocity and temperature fields for the optimized material layout. The highest velocity is found some distance above the cooler, which agrees well with the results from Alexandersen [2015]. The air moves slowly away from the lamp, but accelerates above it, when it has been drawn in from the surroundings. The temperature field shows the ambient temperature in the entire computational domain, except near the lamp, on which a funnel-like shape is formed.

The converged design is conceptually similar to the solution from Alexandersen [2015]. The



(a) Final design of the passive cooler.

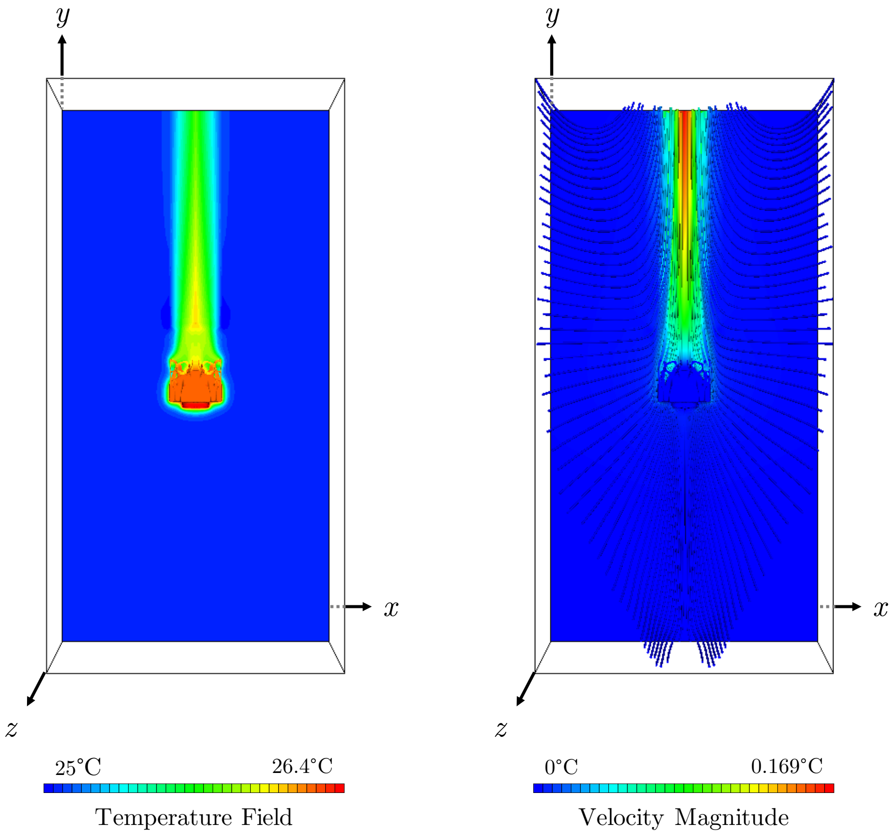
(b) Sectioning of the fluid temperature field, with streamlines.

Figure 7.60: Optimized material layout for the passive cooler example.

difference in designs can be attributed to the lack of constraining the minimum feature size to values above an element length in our level-set based optimization framework. In constrast, density-based topology optimization problems utilize smoothing filters and projection schemes to discourage the formation of features smaller than a prescribed size, which can be set to any value larger than the size of an element. More flexible methods for constraining feature sizes in level set-based optimization have been proposed by Chen et al [2008] and Guo et al [2014], but not yet integrated into our optimization framework.

7.9 Minimum Feature Size

In this section, we examine the application of this measure to different design problems. It is important to note that to strictly enforce the feature size measure in both phases prevents changes in the topology of the designs. In our LSM-XFEM framework, the creation, merging and/or removal of holes requires the formation of small features. If these small features are not allowed, then changes in topology cannot happen, and the variation of the designs is severely limited. This consideration will be discussed in this section. We will use a continuation approach, and gradually impose the feature size constraint in a more strict manner, but in a way that does not prevent



(a) Temperature field.

(b) Velocity magnitude, with streamlines.

Figure 7.61: Slice of the optimized material layout, with the velocity magnitude and the temperature field.

topological changes.

First, we compare our approach against the quadratic energy function of Chen et al [2008]. Then, we validate our feature size measure using shape optimization on a 3D sphere. Subsequently, we apply the feature size measure to the well-known MBB beam design problem. This solid mechanics problem leads to the development of many thin, truss-like structures [Michell, 1904]. The MBB problem will be studied to illustrate the differences between applying the measure to one or both phases. We also study the enforcement of the measure considering different minimum feature sizes.

Then, we apply the feature size measure to the force inverter design problem of Bendsøe and Sigmund [2003]. In the context of our LSM-XFEM framework, this problem is particularly difficult to solve with a solid-void configuration because the optimizer drives the connection towards a thin hinge. This hinge can cause some features to disconnect, leading towards a discontinuous response of the performance with respect to the design variables.

Subsequently, a convective design problem is studied. In this example, the objective is to minimize the temperature at a point where heat is applied by arranging a limited amount of diffusive material. This class of problems also leads to thin geometric features, in an effort to maximize the surface area of the interface. The complex, wavy surface that is produced by this problem provides a clear demonstration of the influence of varying the geodesic parameter r_{tx} in (6.29).

Finally, we study the analog versions of the MBB beam and the convective design problems in 3D.

7.9.1 Comparison with Chen et al [2008]

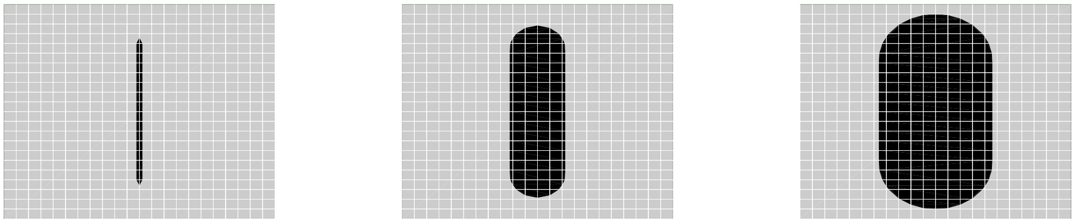
Chen et al [2008] utilized a so-called quadratic energy function to control the minimum feature size. This function is a double integral over the phase interface and provided inspiration for the measure defined here. The quadratic energy function of Chen et al [2008] is computed as:

$$E_q = -\int_{\Gamma_2^0} \int_{\Gamma_1^0} \boldsymbol{T}(\boldsymbol{x}_1^\Gamma) \cdot \boldsymbol{T}(\boldsymbol{x}_2^\Gamma) \; \psi_x(|\boldsymbol{x}_1^\Gamma - \boldsymbol{x}_2^\Gamma|) \, \mathrm{d}\Gamma \mathrm{d}\Gamma \;, \tag{7.26}$$

where $\boldsymbol{T}\left(\boldsymbol{x}_i^\Gamma\right)$ is the tangent vector on the interface, $\boldsymbol{x}_i^\Gamma$ is the position vector of intersection point i, and $\psi_x\left(a\right)$ is computed as:

$$\psi_x\left(a\right) = \begin{cases} a < r_x - w_x & 1.0 \ , \\ r_x - w_x \geq a \geq r_x + w_x & \frac{1}{2}\left[1 - \frac{a-r_x}{w_x} - \frac{1}{\pi} sin\left(\pi \frac{a-r_x}{w_x}\right)\right] \ , \\ a > r_x + w_x & 0.0 \ . \end{cases} \tag{7.27}$$

Here, $\psi_x\left(a\right)$ has the same purpose as the second Heaviside in (6.27) and (6.33), and identifies points that are closer than the minimum feature size. In (7.26), the dot product of the interface tangent vectors, $\boldsymbol{T}(\boldsymbol{x}_1^\Gamma) \cdot \boldsymbol{T}(\boldsymbol{x}_2^\Gamma)$, is used to identify points that fall on different portions of the interface. This construction leads to complexity when considering the discretized form of curved surfaces. We will demonstrate the differences between the measures using a series of different shapes in the following.



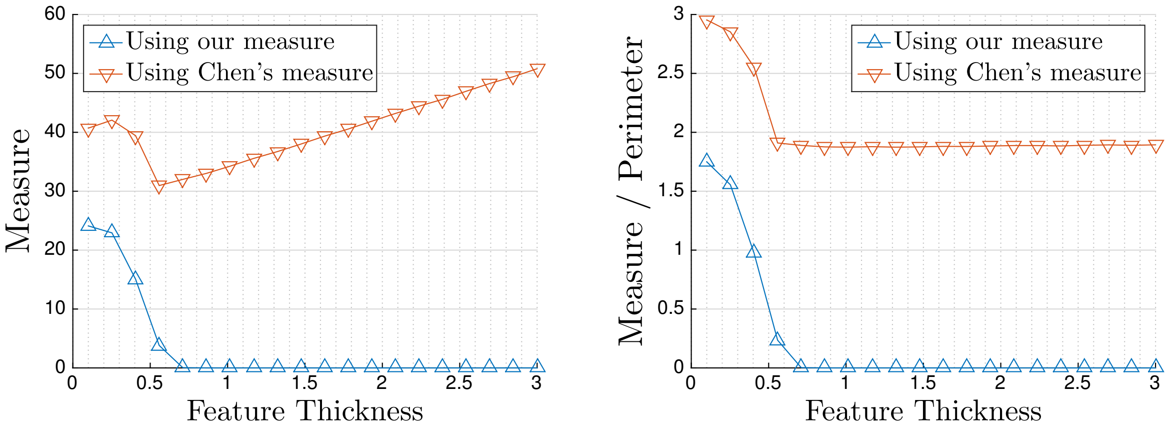
(a) Feature Thickness: 0.1

(b) Feature Thickness: 1.0

(c) Feature Thickness: 2.0

Figure 7.62: Test geometries for comparing the feature size measures in closed surfaces.

First, we study the feature size measures with closed surfaces. The surfaces are modeled as ellipsoids, and their thickness is varied from 0.1 to 3.0, as shown in Figure 7.62. We compare our measure in (6.27) against the quadratic energy function in (7.26). The value of r_x in (6.28) is chosen as 0.5. Figure 7.63a depicts the response of both measures with respect to an increasing feature thickness. We can observe that both measures increase in value as the thickness gets smaller, which implies that both formulations correctly identify when the minimum feature size is being violated. Notice, however, than the effective value of the minimum feature size in our measure is not the desired value of 0.5, but rather approximately 0.75, which is equivalent to $1.5 \times r_x$. This behavior is attributed to the w_x parameter of the second Heaviside function in (6.27), which is set to $r_x/2$. Furthermore, as the feature size becomes larger than $1.5 \times r_x$, the measures display a significantly different behavior. Our measure displays a more intuitive response, and simply turns off with the value of $\mathcal{M}^0$ dropping to 0. On the other hand, Chen's formulation becomes a linear function of the perimeter, that is, as the perimeter of the shape increases, so does the value of the quadratic energy function. The dependence of Chen's formulation on the perimeter is further demonstrated in Figure 7.63b, which displays the measures normalized by the perimeter. Again, while our formulation turns off for a feature size larger than $1.5 \times r_x$, Chen's formulation remains constant with a value greater than 0.

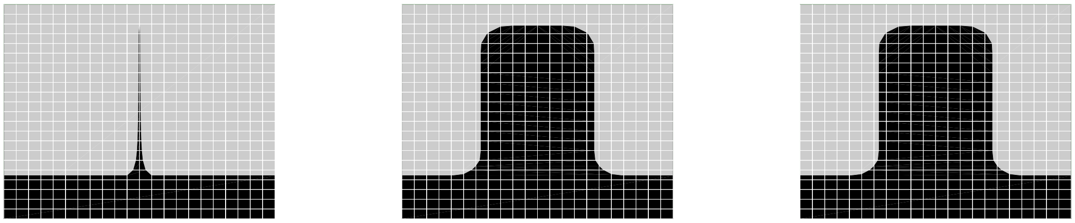


(a) Feature size measure.

(b) Feature size measure normalized by the perimeter.

Figure 7.63: Comparison of the feature size measures for the closed surfaces depicted in Figure 7.62.

Next, we study the feature size measures using shapes with an open surface, as shown in Figure 7.64. We vary the feature thickness of a protrusion from 0.1 to 3.0. Notice that due to the open surface geometry and the shape of the protrusion, the perimeter remains roughly constant as the feature thickness varies. The response of the measures is shown in Figure 7.65. Here, the response of the quadratic energy function is more complicated, with the minimum measure



(a) Feature Thickness: 0.1

(b) Feature Thickness: 1.0

(c) Feature Thickness: 2.0

Figure 7.64: Test geometries for comparing the feature size measures in open surfaces.

occurring at some intermediate feature thickness value. Accounting for changes in perimeter, as shown in Figure 7.63b, does not alleviate this issue. Our measure, on the other hand, presents again a more intuitive response that easily identifies the violation of the feature size.

7.9.2 Verification using a Spherical Shape

To validate the minimum feature size framework, we formulate a shape optimization problem in which we minimize the volume of the negative level set domain subject to a minimum feature size measure constraint. We model a two-material linear diffusion problem; however, note that the overall setup of the physics does not matter in this scenario because the optimization problem is formulated with respect to geometric properties.

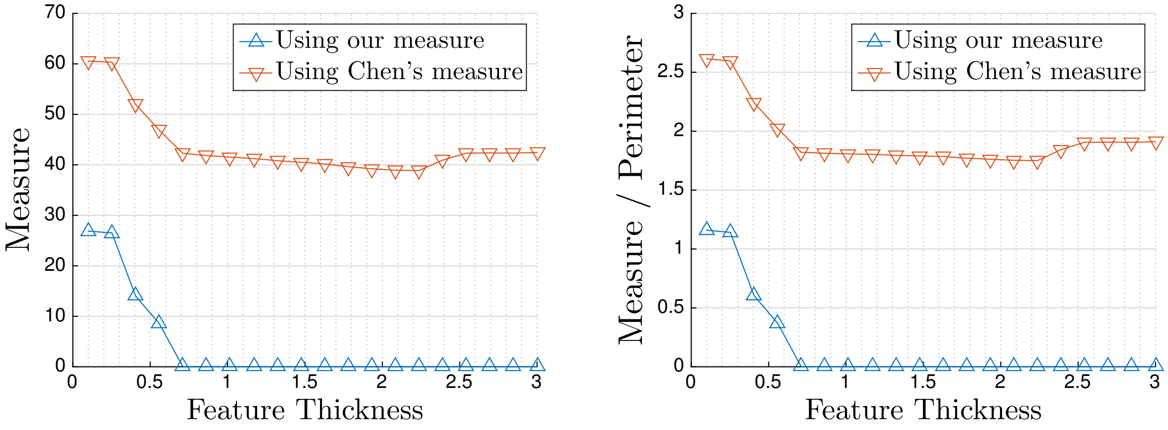
The initial material layout is given by a sphere with a radius of 0.25. The objective, $\mathcal{Z}$, is given as:

$$\mathcal{Z} = \mathcal{V}^- \ , \tag{7.28}$$

where $\mathcal{V}^-$ is defined in (4.71), and the only constraint, g_1, is given by:

$$g_1 = \frac{\mathcal{M}^0}{\mathcal{S}^2} - w_{\mathcal{M}} \ , \tag{7.29}$$

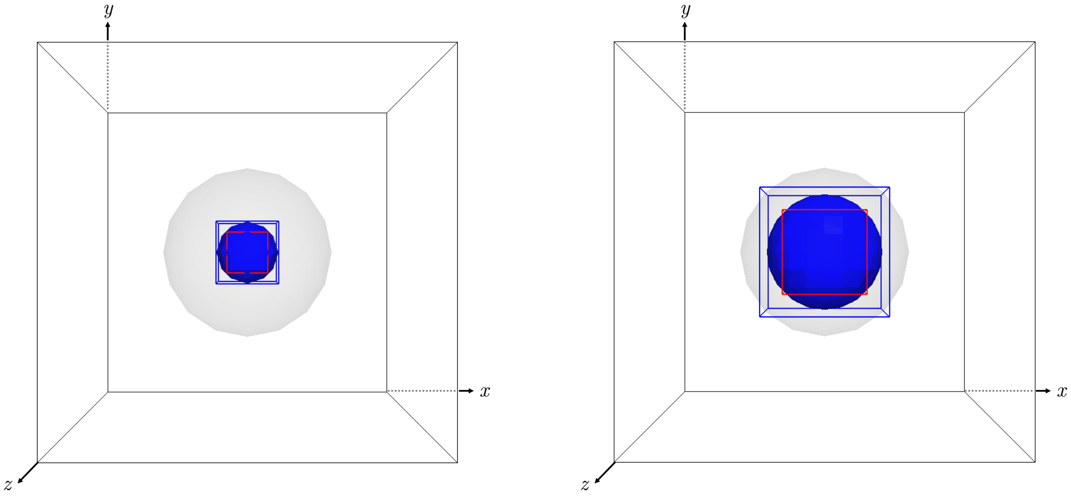
where $\mathcal{M}^0$ is defined in (6.27), $\mathcal{S}$ is defined in (4.73), and the term $w_{\mathcal{M}}$ is a small bound set to 10^{-04}.



(a) Feature size measure.

(b) Feature size measure normalized by the perimeter.

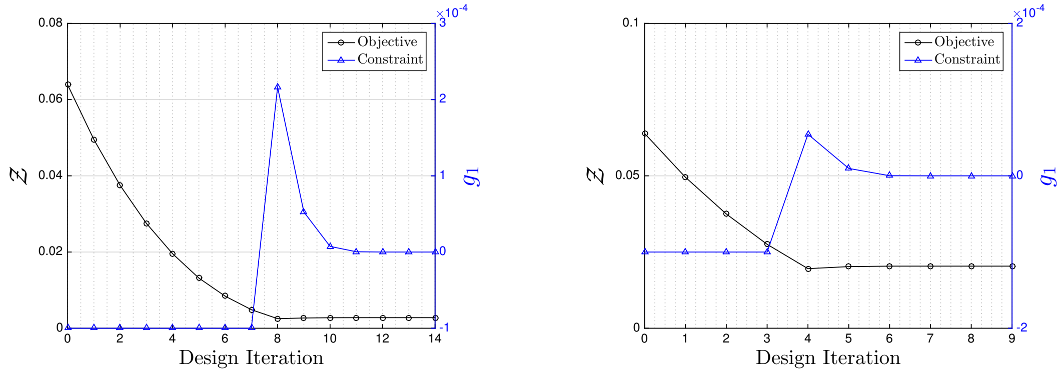
Figure 7.65: Comparison of the feature size measures for the open surfaces depicted in Figure 7.64.



(a) $r_x = 2.4h$.

(b) $r_x = 4.8h$.

Figure 7.66: Minimum feature size validation examples. The grey surfaces denote the initial spheres. The blue spheres denote the optimized material layout. The red solid lines denote a cubic box with dimension r_x. The blue solid lines denote a cubic box with dimension $1.5 \times r_x$.



(a) $r_x = 2.4h$.

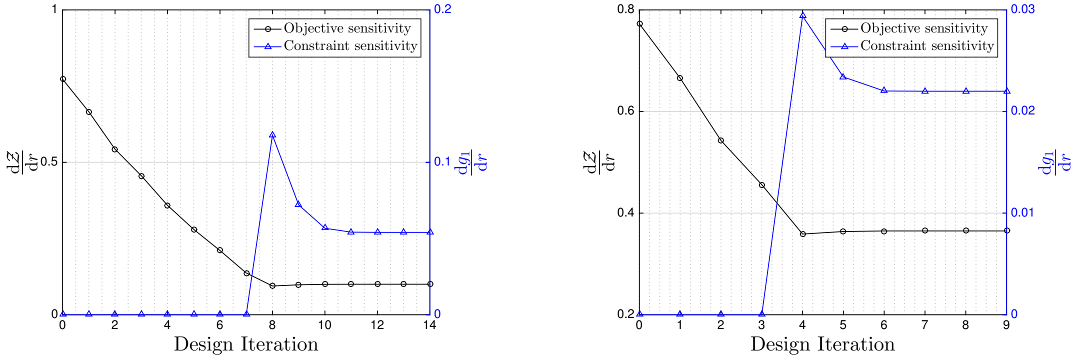
(b) $r_x = 4.8h$.

Figure 7.67: Objective and constraint plots for the minimum feature size validation examples.

A single design variable is used to control the radius of the sphere. The expected behavior would be that the radius is reduced in order to minimize the volume until the minimum feature size constraint is activated. We test two different feature size values, $r_x = 2.4h$ and $r_x = 4.8h$. The optimized material layouts are shown in Figure 7.66, the objective and constraint plots are shown in Figure 7.67, and the sensitivities are shown in Figure 7.68.

Note that the results reflect a minimum feature size closer to $1.5 \times r_x$; this behavior is attributed to the smoothed Heaviside added to the prescribed feature size, $w_x = r_x/2$.

If it is benefitial to the optimization problem to further minimize the objective, at the cost of violating the constraint, the problem might get stuck and prevent topological changes from occurring. Furthermore, the objective will display wiggliness in the convergence plot of the objective function. For example, an additional problem was run where r_x was set to $12h$. The objective function is shown in Figure 7.69, and it displays this wiggliness behavior. The optimization problem reduces the volume of the sphere, which violates the constraint. The constraint in turn causes the design to increase in size. Then, the optimization problem sees this change and reduces the volume again, violating the constraint, and the cycle repeats. Reducing the optimization step size or increasing the resolution of the mesh might help ameliorate this issue.



(a) $r_x = 2.4h$.

(b) $r_x = 4.8h$.

Figure 7.68: Objective and constraint sensitivity plots for the minimum feature size validation examples.

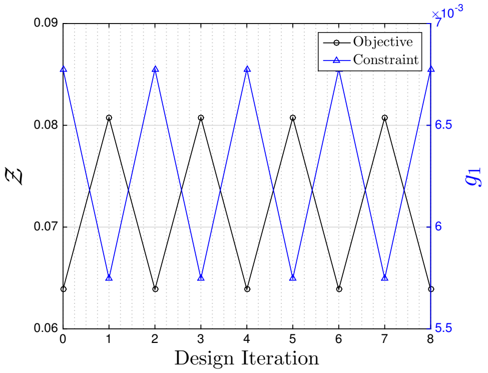


Figure 7.69: Objective and constraint sensitivity plots for the minimum feature size validation example with $r_x = 12h$.

7.9.3 Verification with 2 Inclusions

In this example, we study the influence of the r_x parameter in (6.28) by maximizing the surface area of two 2D circular inclusions of radii $0.16L$, subject to a design constraint on the minimum feature size. The problem setup is shown in Figure 7.70. We utilize the $\mathcal{M}^-$ measure from (6.33), and set $r_\mathcal{M} = 3 \times r_x$. The objective is defined as:

$$\mathcal{Z} = \mathcal{V}^- \ , \tag{7.30}$$

$$g_0 = \mathcal{M}^-/c_\mathcal{M} - 1.0 \ , \tag{7.31}$$

where $c_\mathcal{M} = 0.01$. The problem parameters are shown in Table 7.22.

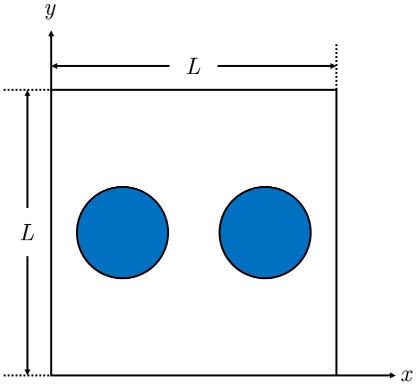


Figure 7.70: Problem setup for the minimum feature size example with 2 circular inclusions. The blue domains represent the positive material phase. The exact location of the inclusions is not relevant for this problem.

The optimized designs are shown in Figure 7.71. We can observe that a small value of r_{tx} prevents the formation of any features, and does not allow the circular inclusions to change in size or in shape. The designs for $r_{tx} = \sqrt{2} + \sqrt{6}$ display a more wavy pattern and a less tick strip than the geometries for $r_{tx} = \sqrt{2}\pi/4$ and $r_{tx} = \sqrt{2}$. This behavior is attributed to the fact that, in this example, r_{tx} is larger than the factor 3 used in $r_\mathcal{M}$. The results for $r_{tx} = \sqrt{2}\pi/4$ and $r_{tx} = \sqrt{2}$ display a single vertical strip, which matches the analytical optimal design for this

	Value
Element edge length	$h = 6.667 \times 10^{-03} L$
Minimum Feature Size	$r_x = \{6h, 12h, 24h\}$
Geodesic to Euclidean distance ratio	$r_{tx} = \left\{1, \sqrt{2}\pi/4, \sqrt{2}, \sqrt{2} + \sqrt{6}\right\}$
Maximum search radius	$r_{\mathcal{M}} = 3 \times r_x$
Minimum feat. size smooth Heaviside width	$w_x = 0.5$
Geodesic smooth Heaviside width	$w_{tx} = 0.01$
Phase domain smooth Heaviside width	$w_n = 0.01$
Constraint weight	$c_{\mathcal{M}} = 0.01$

Table 7.22: Problem parameters for the minimum feature size validation examples of Section 7.9.3.

problem. Additionally, both results do not display a wavy pattern on the material interface. It is not clear from these results if a larger value of r_{tx} is the cause of the wavy design, because the value of $r_{\mathcal{M}}$ was chosen to be too small. We can observe the effects of the w_x parameter used on ϕ_x in (6.28), as the width of the strip for $r_{tx} = \sqrt{2}\pi/4$ and $r_{tx} = \sqrt{2}$ is closer to $1.5 \times r_x$.

7.9.4 Verification with 4 Inclusions

This example uses the same problem setup as in Example 7.9.3. The initial design has 4 circular inclusions of radii $0.16L$, as shown in Figure 7.72. We study the influence of the r_{tx} parameter in (6.29) on penalizing sharp corners. We utilize the $\mathcal{M}_x^-$ measure from (6.37), and set $r_{\mathcal{M}} = r_x \times r_{tx}$. The value of $c_{\mathcal{M}}$ is set to 10^{-06} to account for the smaller value of the feature size measure utilized here.

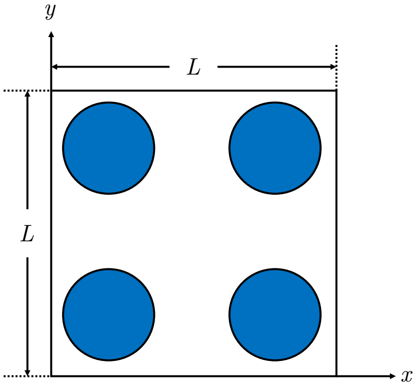
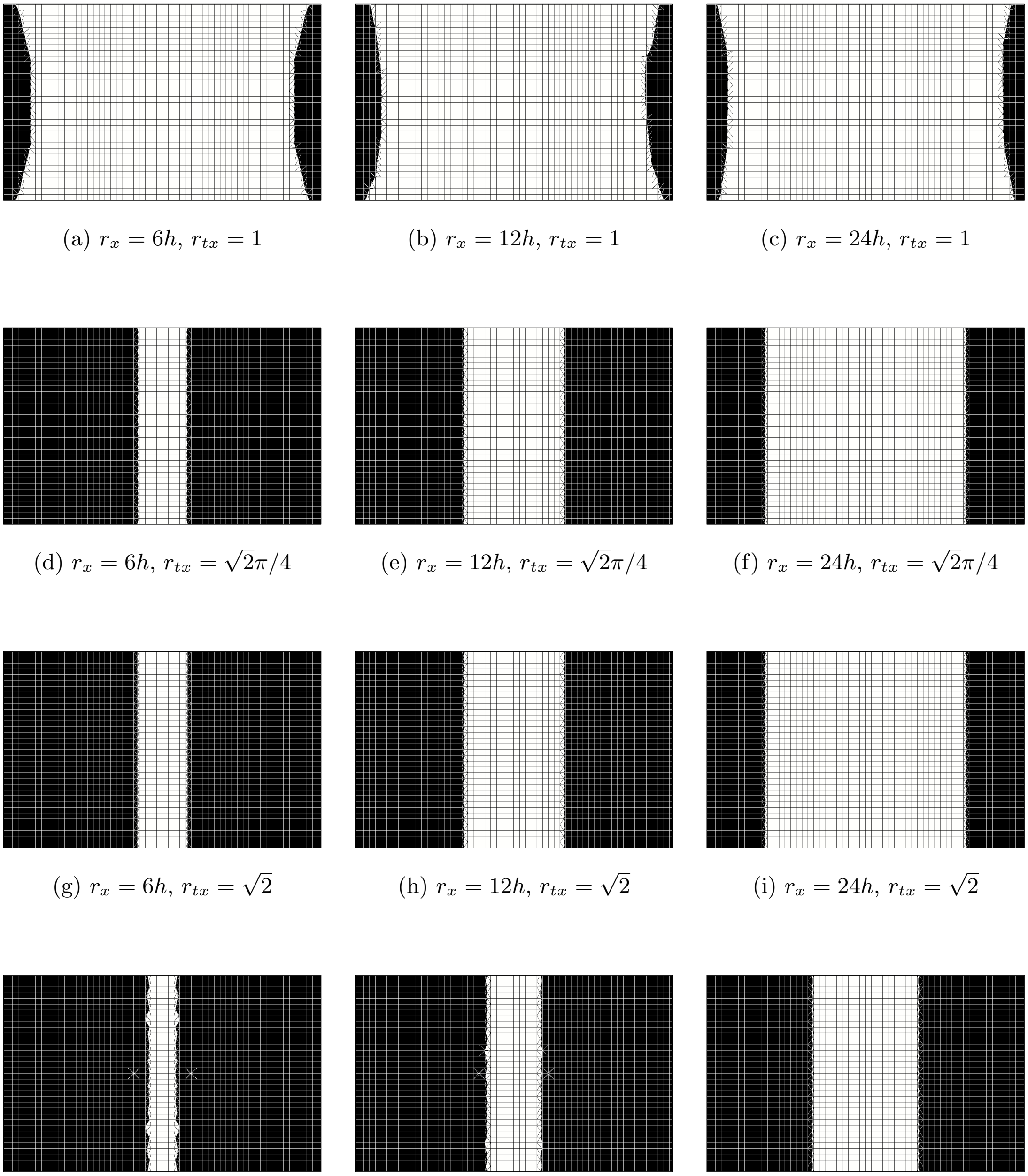


Figure 7.72: Problem setup for the minimum feature size example with 4 circular inclusions. The blue domains represent the positive material phase. The exact location of the inclusions is not relevant for this problem.

The optimized designs are shown in Figure 7.73. We observe that for all examples, the minimum feature size is enforced well. We can also observe that as the geodesic parameter, r_{tx}, is increased, the interface of the design becomes more oscillatory. It is not clear from these results whether this behavior is a consequence of the geodesic parameter or of the feature size measure.

To study the influence of the feature size measure, we take the setup from Figure 7.73i, and use



(a) $r_x = 6h$, $r_{tx} = 1$

(b) $r_x = 12h$, $r_{tx} = 1$

(c) $r_x = 24h$, $r_{tx} = 1$

(d) $r_x = 6h$, $r_{tx} = \sqrt{2}\pi/4$

(e) $r_x = 12h$, $r_{tx} = \sqrt{2}\pi/4$

(f) $r_x = 24h$, $r_{tx} = \sqrt{2}\pi/4$

(g) $r_x = 6h$, $r_{tx} = \sqrt{2}$

(h) $r_x = 12h$, $r_{tx} = \sqrt{2}$

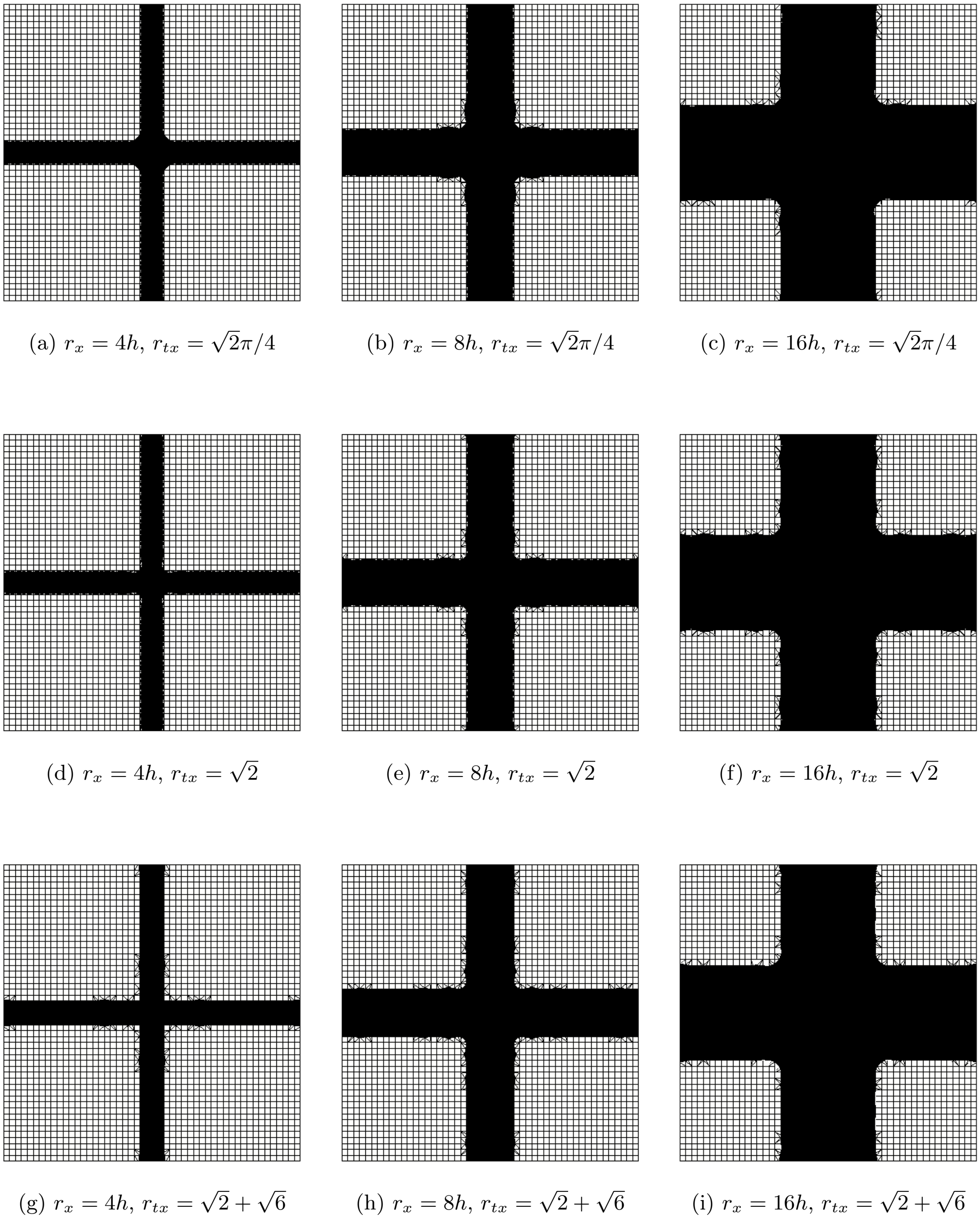
(i) $r_x = 24h$, $r_{tx} = \sqrt{2}$

(j) $r_x = 6h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

(k) $r_x = 12h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

(l) $r_x = 24h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

Figure 7.71: Zoomed display of the optimized material layouts for the minimum feature size validation examples of Section 7.9.3. The white domains represent the negative material phase.



(a) $r_x = 4h$, $r_{tx} = \sqrt{2}\pi/4$

(b) $r_x = 8h$, $r_{tx} = \sqrt{2}\pi/4$

(c) $r_x = 16h$, $r_{tx} = \sqrt{2}\pi/4$

(d) $r_x = 4h$, $r_{tx} = \sqrt{2}$

(e) $r_x = 8h$, $r_{tx} = \sqrt{2}$

(f) $r_x = 16h$, $r_{tx} = \sqrt{2}$

(g) $r_x = 4h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

(h) $r_x = 8h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

(i) $r_x = 16h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

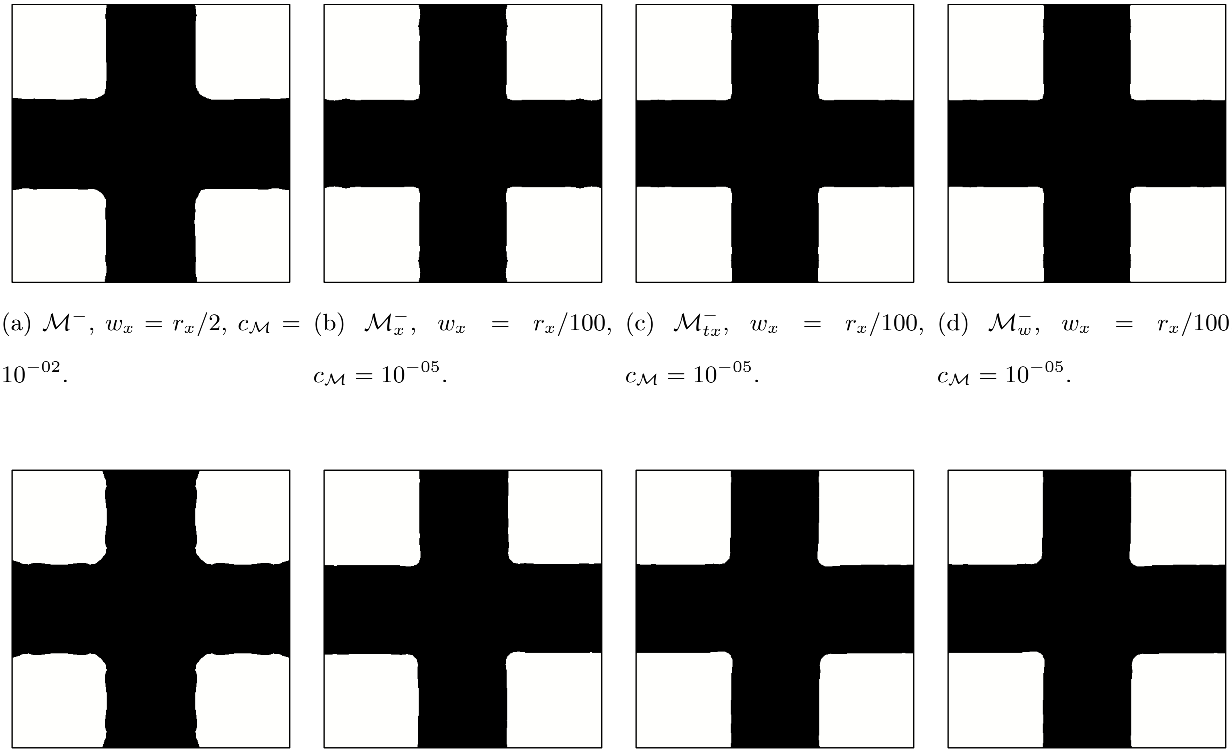
Figure 7.73: Optimized material layouts for the minimum feature size validation examples of Section 7.9.4. The black regions denote the negative material phase.

it to compare it against the different feature size formulations introduced in Section 6.3.2 and against the constraint parameter, $c_{\mathcal{M}}$. We utilize the minimum feature size measures in (6.33), (6.37), (6.38), and (6.40) for a feature size value of $r_x = 16h$, and a geodesic parameter of $r_{tx} = \sqrt{2} + \sqrt{6}$. Unless otherwise stated, the values for all w_ζ parameters are set to 0.001, effectively turning off the smoothing of the Heaviside functions. The parameters w_x and $c_{\mathcal{M}}$ are set on a per-problem basis.

The optimized designs are shown in Figure 7.74. We can observe that the minimum feature size is well imposed in all of the examples. The measure $\mathcal{M}^-$ in (6.33) yields more wavy designs, when compared to the other formulations and used in combination with a smaller $c_{\mathcal{M}}$ parameter. The remaining formulations do not display any significant difference among them, although the designs that utilize the smaller $c_{\mathcal{M}}$ parameter shown slightly straighter lines. It is clear from these examples that the formulations in (6.37), (6.38), and (6.40) are equivalent for this class of problems, and that they yield smoother designs that the $\mathcal{M}^-$ formulation.

7.9.5 Verification with 5 Inclusions

This example uses the same problem setup as in Example 7.9.3. The initial design has 5 circular inclusions of radii $0.16L$, as shown in Figure 7.75. We study the influence of the r_x and the r_{tx} parameters, in (6.29), in the $\mathcal{M}_x^-$ measure from (6.37), and set $r_{\mathcal{M}} = r_x \times r_{tx}$. The value of $c_{\mathcal{M}}$ is set to 10^{-06}.



(a) $\mathcal{M}^-$, $w_x = r_x/2$, $c_{\mathcal{M}} = 10^{-02}$.

(b) $\mathcal{M}_x^-$, $w_x = r_x/100$, $c_{\mathcal{M}} = 10^{-05}$.

(c) $\mathcal{M}_{tx}^-$, $w_x = r_x/100$, $c_{\mathcal{M}} = 10^{-05}$.

(d) $\mathcal{M}_w^-$, $w_x = r_x/100$, $c_{\mathcal{M}} = 10^{-05}$.

(e) $\mathcal{M}^-$, $w_x = r_x/2$, $c_{\mathcal{M}} = 10^{-06}$.

(f) $\mathcal{M}_x^-$, $w_x = r_x/100$, $c_{\mathcal{M}} = 10^{-06}$.

(g) $\mathcal{M}_{tx}^-$, $w_x = r_x/100$, $c_{\mathcal{M}} = 10^{-06}$.

(h) $\mathcal{M}_w^-$, $w_x = r_x/100$, $c_{\mathcal{M}} = 10^{-06}$.

Figure 7.74: Optimized material layouts for the minimum feature size validation examples of Section 7.9.4, with $r_x = 16h$, and $r_{tx} = \sqrt{2} + \sqrt{6}$. The black regions denote the negative material phase.

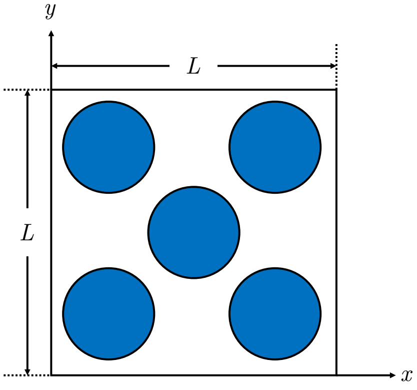


Figure 7.75: Problem setup for the minimum feature size example with 5 circular inclusions. The blue domains represent the positive material phase. The exact location of the inclusions is not relevant for this problem.

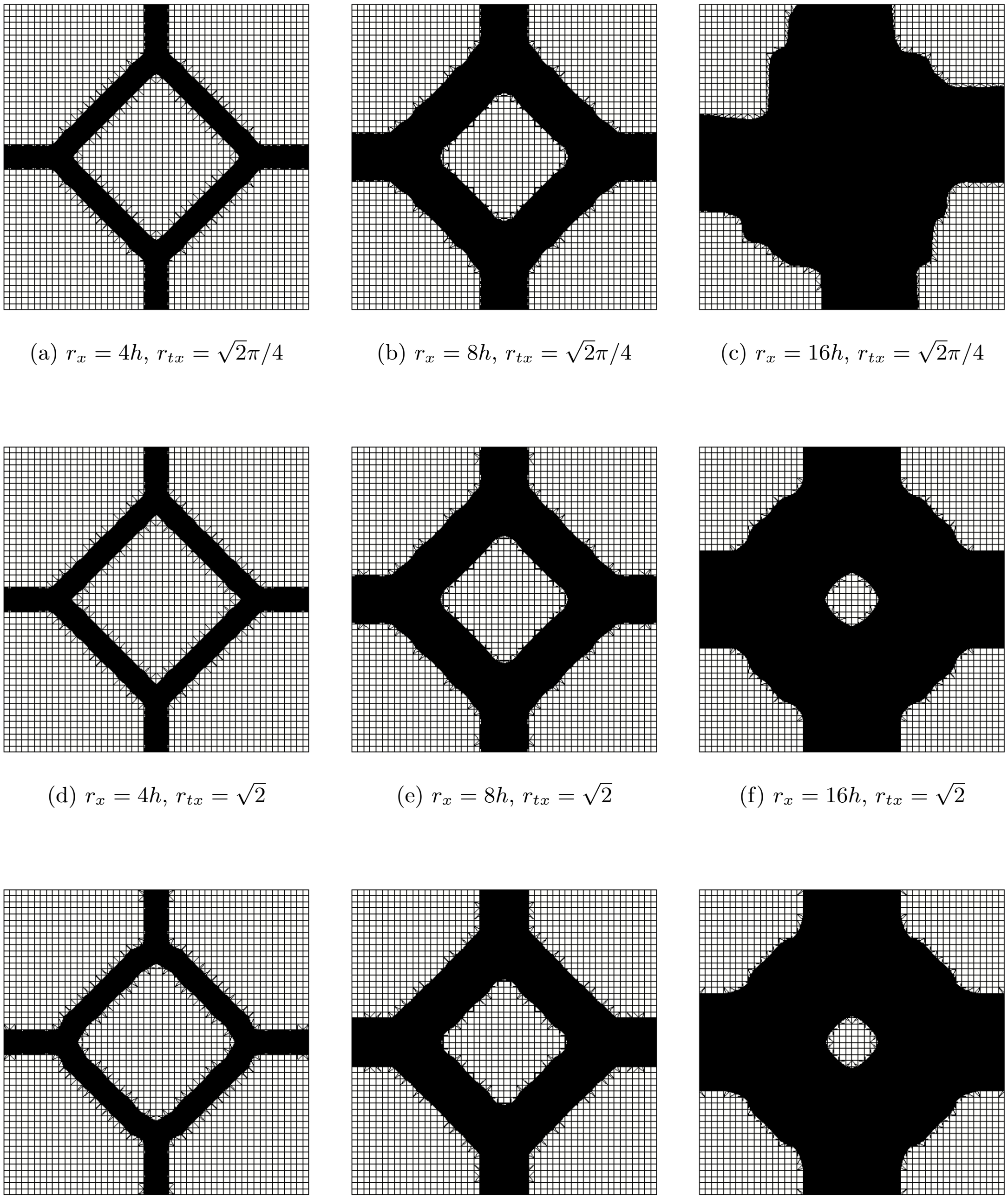
The optimized designs are shown in Figure 7.76. We observe that for all examples, the minimum feature size is enforced well. The problems that utilize a feature size value of $r_x = 4h$ and $r_x = 8h$ display similar designs. The designs for the largest feature size display significant differences, which seem to be related to the influence of r_{tx}. A value of $r_{tx} = \sqrt{2}\pi/4$ yields a more wavy design, while the other two showcase a more doughnut-like pattern. It is not clear from these results why the geodesic parameter causes this behavior, although we can infer that it is related to the smootheness of the designs.

7.9.6 Analysis Sweeps

In this section, we will study the minimum feature size measures in (6.33) and (6.35) with respect to different shapes and parameters.

7.9.6.1 Sweep over the Minimum Feature Size Parameter

Our problem setup for this example is shown in Figure 7.77. The analysis domain is a $3L \times 2L$ mesh with 150×100 quadrilateral bilinear elements, for an h value of $0.02L$. The domain contains an inclusion in the form of a strip, with a thickness value of h_{gap}. The inclusion is represented by



(a) $r_x = 4h$, $r_{tx} = \sqrt{2}\pi/4$

(b) $r_x = 8h$, $r_{tx} = \sqrt{2}\pi/4$

(c) $r_x = 16h$, $r_{tx} = \sqrt{2}\pi/4$

(d) $r_x = 4h$, $r_{tx} = \sqrt{2}$

(e) $r_x = 8h$, $r_{tx} = \sqrt{2}$

(f) $r_x = 16h$, $r_{tx} = \sqrt{2}$

(g) $r_x = 4h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

(h) $r_x = 8h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

(i) $r_x = 16h$, $r_{tx} = \sqrt{2} + \sqrt{6}$

Figure 7.76: Optimized material layouts for the minimum feature size validation examples of Section 7.9.5. The black regions denote the negative material phase.

the negative phase in (2.2). We modify the value of r_x in (6.33) from 0.1 to 0.3, and set h_{gap} to a constant value of 0.2. We measure the feature size formulation in (6.33). The problem parameters are shown in Table 7.23. The values for all w_ζ parameters are set to 0.001, effectively turning off the smoothing of the Heaviside functions.

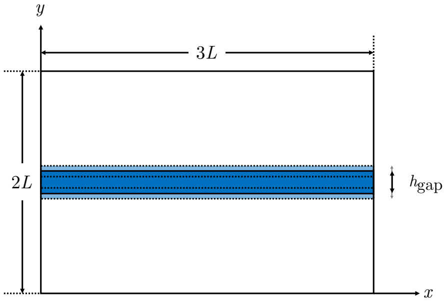
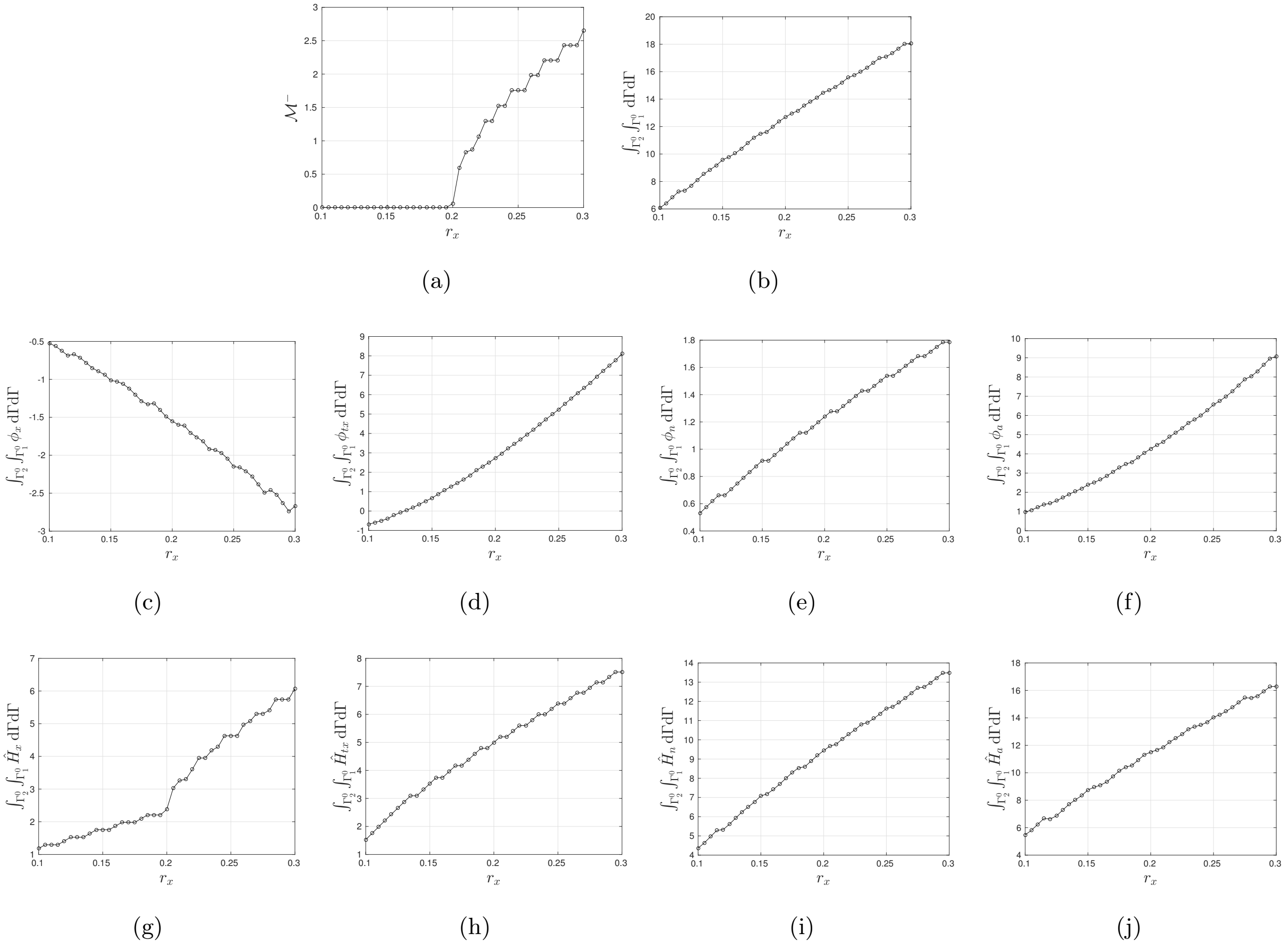


Figure 7.77: Problem setup for the sweep over the strip with thickness h_{gap}.

The results for the minimum feature size measure in (6.33) over the negative phase, the formulations in (6.28), (6.29), (6.34), and (6.36), and their corresponding smoothed Heaviside functions are shown in Figure 7.78. We can observe that the integral for ϕ_{tx} in Figure 7.78d increases as r_x is varied. This is counterintuitive since ϕ_{tx} in (6.29) does not depend on r_x. However, this behavior is attributed to the behavior of the double integral over the perimeter in Figure 7.78b. Our minimum feature size implementation integrates over all connected elements to the intersection point of interest, $\boldsymbol{x}_i^\Gamma$, within a search radius of $r_\mathcal{M}$. In this example, $r_\mathcal{M}$ is a function of r_x, and therefore, as r_x increases, so do the integration area and the integral for ϕ_{tx}. The jagged behavior seen in the measures of $\mathcal{M}^-$ and ϕ_x can also be attributed to the value of $r_\mathcal{M}$, as a small value can lead to discretization errors in the measure. We will show the influence of $r_\mathcal{M}$ in subsequent sweeps.

	Value
Thickness gap	$h_{\text{gap}} = 0.2$
Minimum Feature Size	$r_x = \{0.1 : 0.005 : 0.3\}$
Geodesic to Euclidean distance ratio	$r_{tx} = 1.2$
Maximum search radius	$r_{\mathcal{M}} = 3 \times r_x$
Minimum feat. size smooth Heaviside width	$w_x = 0.001$
Geodesic smooth Heaviside width	$w_{tx} = 0.001$
Phase domain smooth Heaviside width	$w_n = 0.001$

Table 7.23: Problem parameters for the sweep over the strip with thickness h_{gap}.



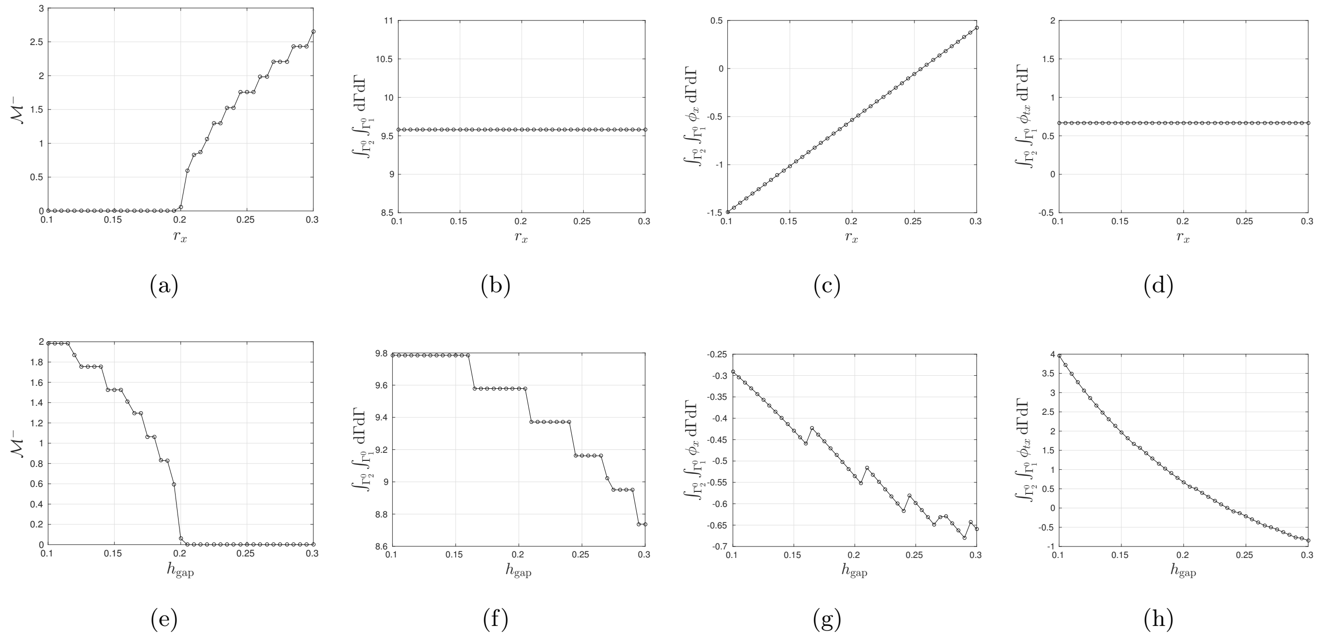
(a) (b)

(c) (d) (e) (f)

(g) (h) (i) (j)

Figure 7.78: Minimum feature size results for the strip inclusion sweep of Section 7.9.6.1.

In the following sweep, we set $r_{\mathcal{M}}$ to a constant value of $5h$. We also study the effects of keeping r_x constant and instead modifying h_{gap} in the same range as r_x was modified previously. The results are shown in Figure 7.79. We observe that with a constant value for $r_{\mathcal{M}}$, the integral of ϕ_{tx} remains constant for the sweep over r_x. The value does change for the sweep over h_{gap}; however, this is expected as ϕ_{tx} depends on the Euclidean distance $\boldsymbol{x}_{12}^{\Gamma}$, which is modified as the gap varies. The measures of $\mathcal{M}^-$ and ϕ_x still display a jagged behavior. This again can be attributed to discretization errors produced by a small $r_{\mathcal{M}}$. In a subsequent sweep, we will set $r_{\mathcal{M}} = \infty$.

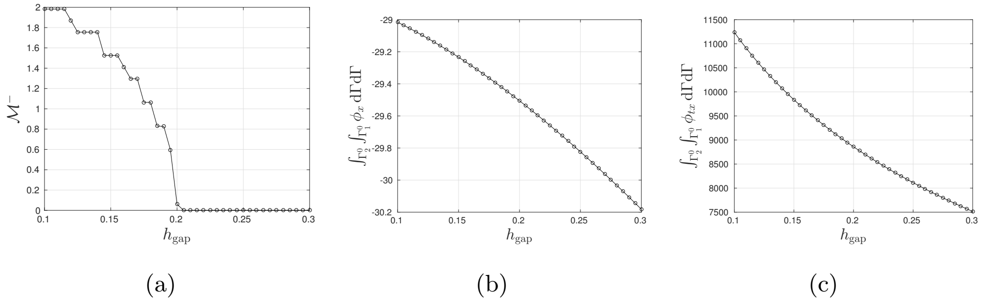


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 7.79: Minimum feature size results for the strip inclusion sweep of Section 7.9.6.1, with constant $r_{\mathcal{M}} = 5h$.

In the following sweep, we set the value of $r_{\mathcal{M}}$ to ∞, and sweep over h_{gap}. The results are shown in Figure 7.80. While ϕ_x and ϕ_{tx} display a smooth behavior, the minimum feature size measure $\mathcal{M}^-$ is still not smooth. This is attributed to the Heaviside projection used by the formulation in (6.33). We will study the characteristics of (6.35) in a subsequent sweep to allow for a smoother response of the feature size measure.

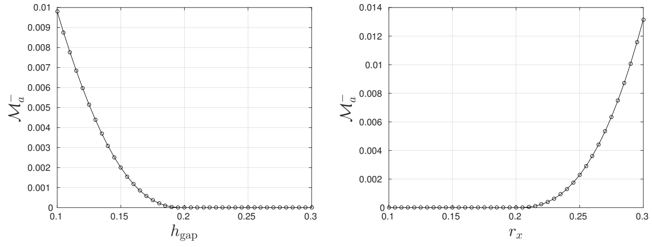


(a) (b) (c)

Figure 7.80: Minimum feature size results for the strip inclusion sweep of Section 7.9.6.1, with $r_{\mathcal{M}} = \infty$.

Remark 1 Note that the integral for ϕ_{tx} increases considerably in value in Figure 7.80c. This is because of the value $\left|\boldsymbol{t}_{12}^{\Gamma}\left(\boldsymbol{s}\right)\right|$ is assigned if two points, $\boldsymbol{x}_1^{\Gamma}$ and $\boldsymbol{x}_2^{\Gamma}$ are within an Euclidean distance $\left|\boldsymbol{x}_{12}^{\Gamma}\left(\boldsymbol{s}\right)\right| < r_x$, but on topologically disconnected surfaces. In such a scenario, the geodesic distance is set to $r_{\mathcal{M}}$.

In this sweep, we utilize the $\mathcal{M}_a^-$ formulation, and sweep over r_x and h_{gap}. The results are shown in Figure 7.81. We can observe than modifying the formulation results in a smoother response for the minimum feature size measure.



(a)

(b)

Figure 7.81: Minimum feature size results for the strip inclusion sweep of Section 7.9.6.1, using the $\mathcal{M}_a^-$ formulation.

We can conclude from these sweeps that the search radius, $r_{\mathcal{M}}$, must be set to a large value in order to avoid discretization errors, and that utilizing the formulations that include the square power of (6.28) results in a smoother response of the measure, which is preferrable when computing the sensitivities.

7.9.6.2 Sweep over the p-Norm of an Inclusion

In this study, we sweep over the p-norm of a circular inclusion. We modify the norm from 1, which gives us a diamond shape, to 10, which gives us a cuboid-shape with rounded corners. We seek to study the effects of the r_x, r_{tx} and r_a parameters in (6.35).

First, we study the influence of the r_x and the r_{tx} parameters. We choose two values to analyze for r_{tx}, 1.2 and $\sqrt{2}$, and three values for r_x, 0.25, 0.50, and 0.75. r_a is set to a constant

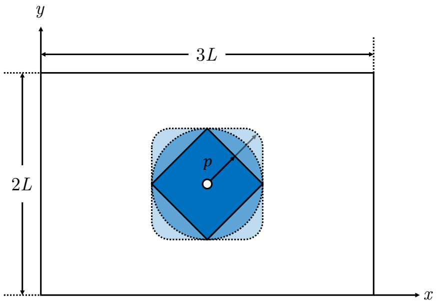
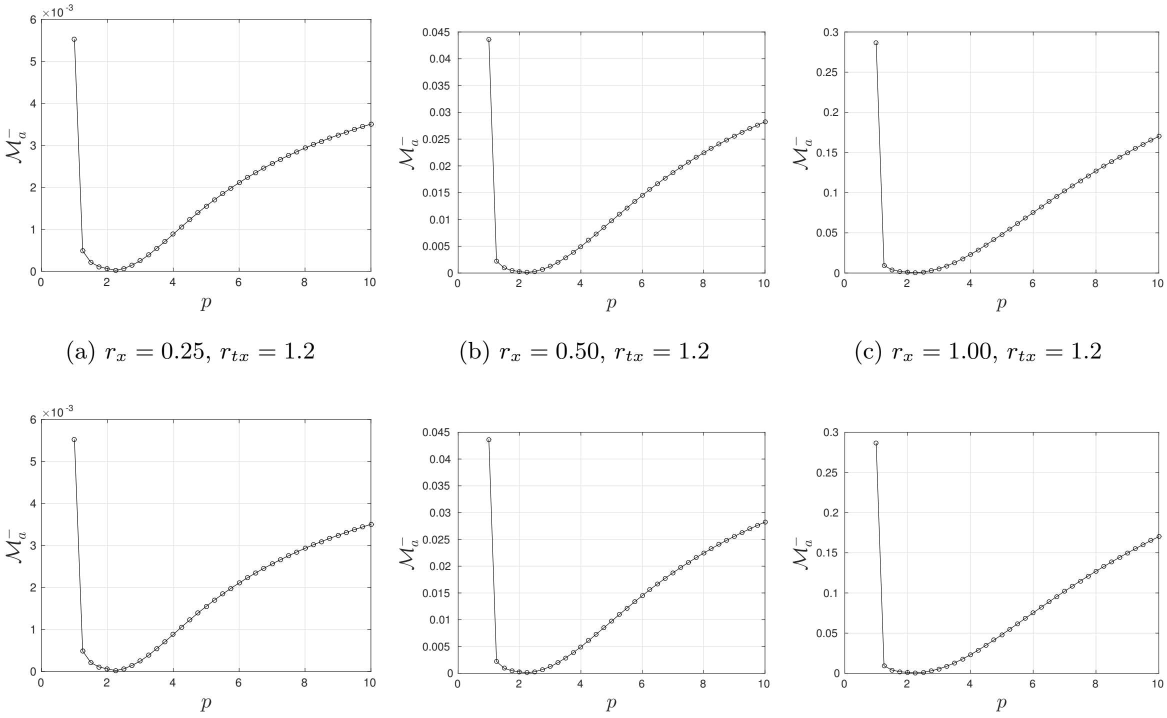


Figure 7.82: Problem setup for the sweep over the p-norm of an inclusion.

value of 0.5. The results are shown in Figure 7.83. We can observe that the parameter r_x scales the value of $\mathcal{M}_a^-$, but that the parameter r_{tx} has no influence in this formulation. We will see later that this issue is related to the fact that $r_a < r_{tx}$.



(a) $r_x = 0.25$, $r_{tx} = 1.2$

(b) $r_x = 0.50$, $r_{tx} = 1.2$

(c) $r_x = 1.00$, $r_{tx} = 1.2$

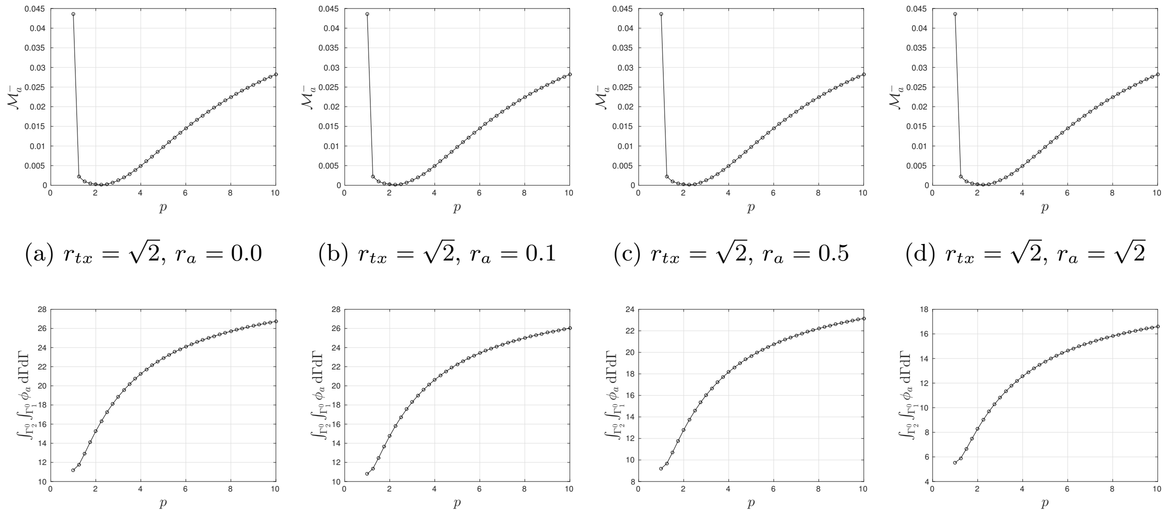
(d) $r_x = 0.25$, $r_{tx} = \sqrt{2}$

(e) $r_x = 0.50$, $r_{tx} = \sqrt{2}$

(f) $r_x = 1.00$, $r_{tx} = \sqrt{2}$

Figure 7.83: Minimum feature size results for the p-norm sweep of Section 7.9.6.2, with $r_{\mathcal{M}} = \infty$ and $r_a = 0.5$.

For this sweep, we focus on the characteristics of the r_a parameter. We keep r_x constant to 0.5, and r_{tx} constant to $\sqrt{2}$. We observe that the parameter r_a has virtually no influence on the feature size measure. However, the function ϕ_a does change with a varying r_a.



(a) $r_{tx} = \sqrt{2}$, $r_a = 0.0$ (b) $r_{tx} = \sqrt{2}$, $r_a = 0.1$ (c) $r_{tx} = \sqrt{2}$, $r_a = 0.5$ (d) $r_{tx} = \sqrt{2}$, $r_a = \sqrt{2}$

(e) $r_{tx} = \sqrt{2}$, $r_a = 0.0$ (f) $r_{tx} = \sqrt{2}$, $r_a = 0.1$ (g) $r_{tx} = \sqrt{2}$, $r_a = 0.5$ (h) $r_{tx} = \sqrt{2}$, $r_a = \sqrt{2}$

Figure 7.84: Minimum feature size results for the p-norm sweep of Section 7.9.6.2, with $r_{\mathcal{M}} = \infty$ and $r_x = 0.50$.

This behavior can be attributed to the relationship between r_{tx} and r_a. For (6.29) to be positive, we must have:

$$\left|\boldsymbol{t}_{12}^{\Gamma}\right| > r_{tx} \left|\boldsymbol{t}_{12}^{\Gamma}\right| \ . \tag{7.32}$$

For (6.28), we must have:

$$r_x > \left|\boldsymbol{x}_{12}^{\Gamma}\right| \ . \tag{7.33}$$

And for (6.36), we must have:

$$\left|\boldsymbol{t}_{12}^{\Gamma}\right| > r_a \cdot r_{tx} \ . \tag{7.34}$$

Inserting (7.33) into (7.34) gives us:

$$\left|\boldsymbol{t}_{12}^{\Gamma}\right| > r_a \cdot \left|\boldsymbol{x}_{12}^{\Gamma}\right| \ . \tag{7.35}$$

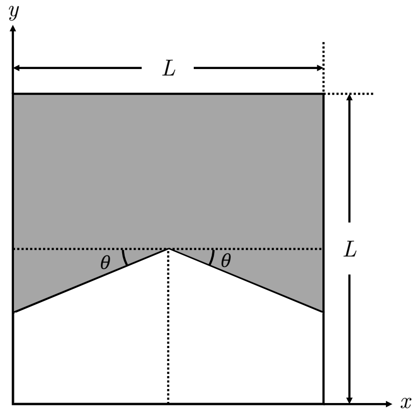


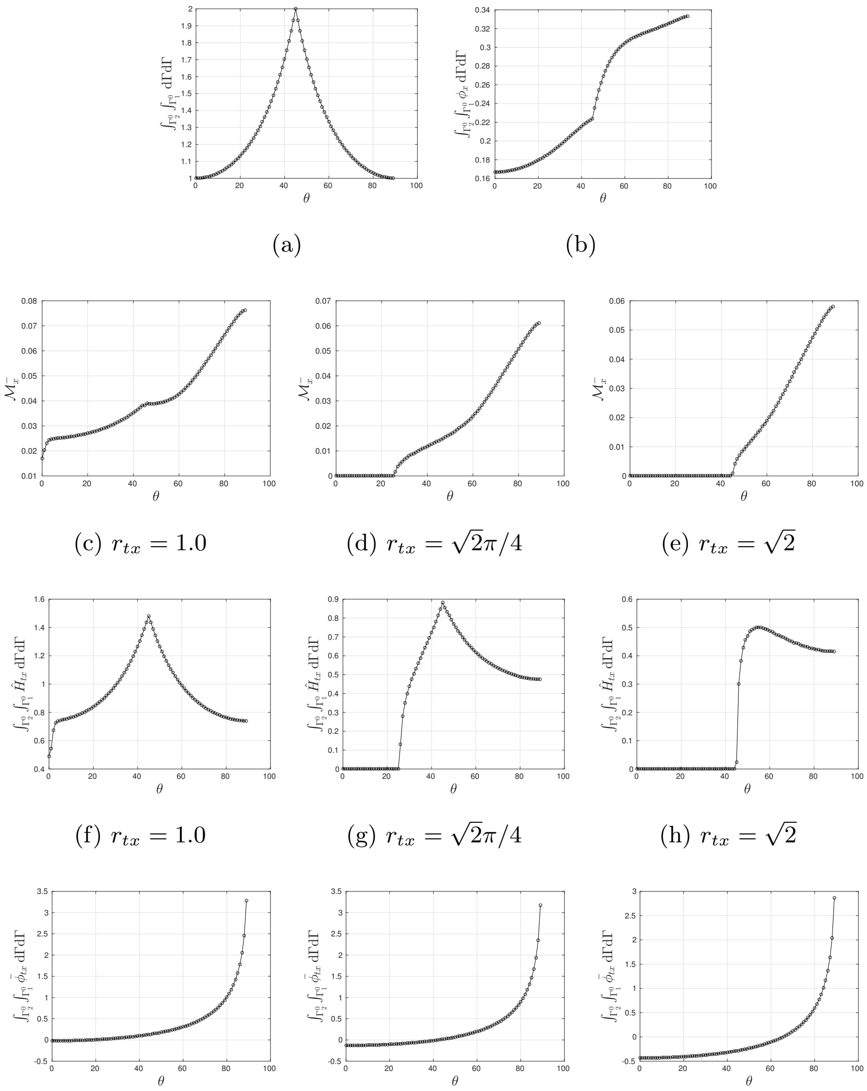
Figure 7.85: Problem setup for the sweep over different angles example.

This indicates that the factor r_a has the same effect as r_{tx}, and that if $r_a \leq r_{tx}$, the first Heaviside in (6.35) has no effect. Also, we can conclude that as long as $r_{\mathcal{M}} \leq r_{tx} \cdot r_x$, the first Heaviside will not vanish.

7.9.6.3 Sweep over the Angle of an Inclusion

In this example, we seek to study the relationship between the geodesic parameter, r_{tx}, and the sharp corners (i.e. the smoothness) of the design. Our problem setup consists of a sharp corner inclusion, as shown in Figure 7.85, where the angle θ is varied from 0° to 89°. We utilize the feature size measure of (6.37) over the negative phase. The value of r_x is set to 0.5, and the search radius $r_{\mathcal{M}}$ is set to ∞.

The plots for the feature size measure and its components are shown in Figure 7.86. As the angle θ increases, so does the sharpness of the corners, which increases the curvature of the problem. Analogous to this behavior, the feature size measure also increases with an increasing value of θ; however, the angle at which it activates is dependent on the geodesic parameter r_{tx}. We can conclude that the parameter r_{tx} is a function of the secant of θ, where θ, through the Heaviside function, controls which sharp corners are accounted for in the measure. Therefore, utilizing a small value for r_{tx} will result in all curvatures being penalized in order to obtain a smoother geometries, whereas a large value will discourage smooth designs.



(a)

(b)

(c) $r_{tx} = 1.0$

(d) $r_{tx} = \sqrt{2}\pi/4$

(e) $r_{tx} = \sqrt{2}$

(f) $r_{tx} = 1.0$

(g) $r_{tx} = \sqrt{2}\pi/4$

(h) $r_{tx} = \sqrt{2}$

(i) $r_{tx} = 1.0$

(j) $r_{tx} = \sqrt{2}\pi/4$

(k) $r_{tx} = \sqrt{2}$

Figure 7.86: Minimum feature size results for the angle sweep of Section 7.9.6.3, with $r_{\mathcal{M}} = \infty$ and $r_x = 0.50$.

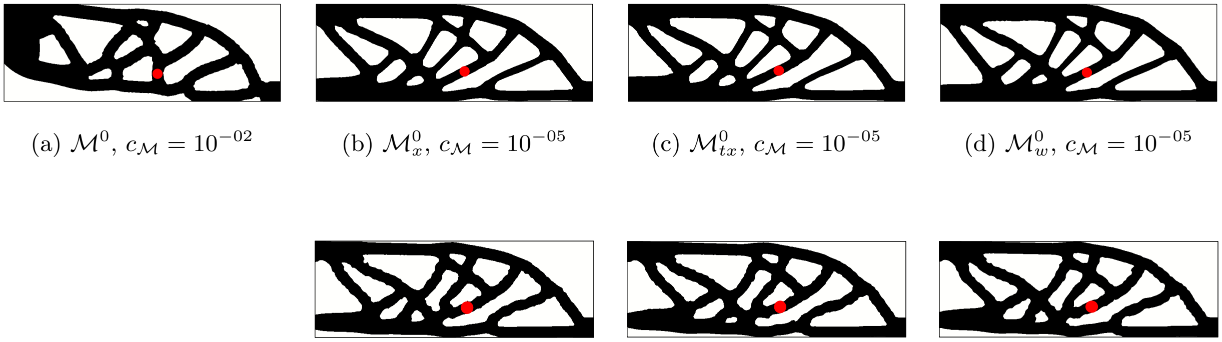
7.9.7 Design of a 2D MBB Beam

In this example, we study the characteristics of the minimum feature size measures in (6.33), (6.37), (6.38), and (6.40) on the optimized MBB beam design of Example 7.3.2. An additional constraint is imposed on the problem to control the minimum feature size over both phases as:

$$g_1 = \mathcal{M}_\zeta^0 / c_{\mathcal{M}} - 1.0 \; , \tag{7.36}$$

where ζ determines the specific measure used and $c_{\mathcal{M}}$ is the constraint weight. The problem parameters are shown in Table 7.24.

The optimized material layouts are shown in Figure 7.87. We can observe that the thin features of the MBB beam design in Example 7.3.2 grow in size; however, the material interface loses smoothness, and small wavy patterns emerge as seen in previous examples. We can attribute this to the fact that once this patterns emerge, the minimum feature size measure will not do anything to smooth them out, and that the perimeter penalty is not strong enough to eliminate them. Additional surface smoothers measures are necessary to get rid of these oscillatory patterns on the interface. Furthermore, we see that the oscillatory behavior increases with a decreasing constraint parameter $c_{\mathcal{M}}$. This can be attributed to the interplay of localized sensitivities along the material interface and the volume constraint on the solid phase.



(a) $\mathcal{M}^0$, $c_{\mathcal{M}} = 10^{-02}$

(b) $\mathcal{M}^0_x$, $c_{\mathcal{M}} = 10^{-05}$

(c) $\mathcal{M}^0_{tx}$, $c_{\mathcal{M}} = 10^{-05}$

(d) $\mathcal{M}^0_w$, $c_{\mathcal{M}} = 10^{-05}$

(e) $\mathcal{M}^0_x$, $c_{\mathcal{M}} = 10^{-06}$

(f) $\mathcal{M}^0_{tx}$, $c_{\mathcal{M}} = 10^{-06}$

(g) $\mathcal{M}^0_w$, $c_{\mathcal{M}} = 10^{-06}$

Figure 7.87: Optimized material layouts for the 2D MBB beams with a minimum feature size constraint of of $r_x = 6h$.

We utilize the design from Figure 7.87b to restart the optimization process with a larger feature size value, $r_x = 9h$, and measure the feature size over the negative phase, $\mathcal{M}^-_x$. The full design is shown in Figure 7.88. We can observe that the feature size is well-imposed, and that the design does not display large wavy patterns on the interface.

	Value
Element edge length	$h = 6.667 \times 10^{-03} L$
Minimum Feature Size	$r_x = 6h$
Geodesic to Euclidean distance ratio	$r_{tx} = \sqrt{2} + \sqrt{6}$
Maximum search radius	$r_{\mathcal{M}} = r_{tx} \times r_x$
Minimum feat. size smooth Heaviside width	$w_x = 0.5$ for (6.33), 0.01 for all others
Geodesic smooth Heaviside width	$w_{tx} = 0.01$
Phase domain smooth Heaviside width	$w_n = 0.01$
Perimeter penalty	$w_{\mathcal{S}} = 0.1$
Smoothing filter radius	$r_\phi = 2.4h$

Table 7.24: Problem parameters for the minimum feature size validation examples of Section 7.9.3.

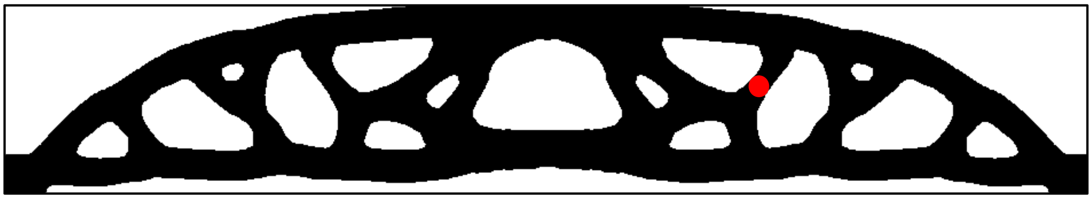


Figure 7.88: Optimized material layout for the 2D MBB beams with a minimum feature size constraint of $r_x = 9h$.

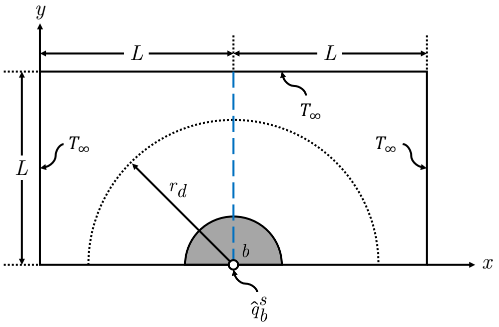


Figure 7.89: Convective design problem setup.

7.9.8 Design of a 2D Convective Design

In this example, we study the application of the feature size measure on a different physical model. We take the heat conduction problem with edge convection of Coffin and Maute [2015b]. The objective of this problem is to minimize the temperature at a point where heat is applied by arranging a limited amount of diffusive material. The in-plane boundaries of the diffusive material are subject to a simplified convection boundary condition. This class of problems leads to very thin solid and void features that can hamper the convergence of the design.

In a classical Newton's Law of Cooling formulation, a single diffusive material would be used, and the temperature in the fluid (or void) domain would be prescribed to be the far-field temperature. To deter the formation of disconnected fluid inclusions, we utilize two diffusive materials, one for the solid and one for a fictitious fluid. The solid domain is represented by the negative phase, and denoted as Ω^s, and the fluid domain is represented by the positive phase, and denoted as Ω^f. The problem setup is shown in Figure 7.89. A heat flux, $\hat{q}_b^s$, is applied at point b, and the temperature is fixed as $T^s = T_\infty$ at the top, left, and right walls. The conductivity in the fictitious fluid is set to be large relative to the solid, i.e. $\kappa^f >> \kappa^s$, such that on the fluid-solid interface the temperature is nearly the far-field value $T^f \approx T_\infty$. The objective $\mathcal{Z}$ is written as:

$$\mathcal{Z}\left(\boldsymbol{s}, \boldsymbol{u}\left(\boldsymbol{s}\right)\right) = T_b^s \ , \tag{7.37}$$

where T_b^s is the temperature of the solid at point b.

Table 7.25: 2D convective design problem parameters.

Property	Value
Element length scale	$h = 0.02L$
Applied heat	$\hat{q}_b^s = 1.0$
Solid diffusivity	$\kappa^s = 1.0$
Fluid diffusivity	$\kappa^f = 5.0$
Convection coefficient	$h = 0.1$
Characteristic length	$L_c = 1.0$
Far field temperature	$T_\infty = 0.0$
Reference temperature	$T_{\text{ref}} = 0.0$
Filter Radius	$r_\phi = 0.048$
Outer design radius	$r_d = 0.8$
Maximum search radius	$r_{\mathcal{M}} = r_{tx} \times r_x$
Minimum feat. size smooth Heaviside width	$w_x = 0.01$ for (6.33), 0.01 for all others
Geodesic smooth Heaviside width	$w_{tx} = 0.01$
Phase domain smooth Heaviside width	$w_n = 0.01$
Smoothing filter radius	$r_\phi = 2.4h$

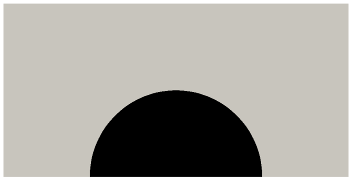


Figure 7.90: Initial design for the convective design problem.

In this problem, the design is prescribed to be symmetric, that is, the design variables are only defined on the nodes in the left half of the mesh. Level set values at nodes on the right half of the mesh are set equal to those on the left. The lower and upper limits, s_i^L and s_i^U, of the design variables are set to $\mp h/2$, respectively. An outer design radius is also specified, r_d, beyond which the design variables are set equal to the upper bound. The problem parameters are described in Table 7.25.

To simplify the initial design of the convective problem, we choose a semi-circle with radius $r = 0.5$, as shown in Figure 7.90. As observed in previous studies [Coffin and Maute, 2015b], the design is expected to grow branches out from the initial circle. While the geometry changes are radical, there is little change in topology. This particular evolution of geometry in the design process ensures that for all values of minimum feature size less than the initial circle radius, there will be no initial violation of the feature size constraint. With no continuation approach, the constraint can be enforced strongly, with $c_{\mathcal{M}} = 10^{-06}$, and studying the different formulations of Section 6.3.2.

Figure 7.91 shows the result of not enforcing any minimum feature size. This optimization does not yield a converged design. Branches are formed from the circular base, but their connection to the base grows thin, and eventually they disconnect and are removed from the design. After the branches disconnect from the base, new branches form. Without additional regularization, this behavior leads to a continuous morphing of the geometry. More details on this issue and the need for regularization are provided in Coffin and Maute [2015b].

Controlling the mininum feature size for $r_x = 4h$ and $r_{tx} = \sqrt{2}$ using the formulations in (6.37), (6.38), and (6.40) yields a series of complicated geometrical structures, as shown in Figure

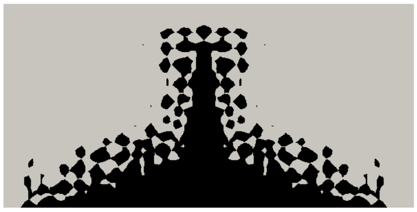
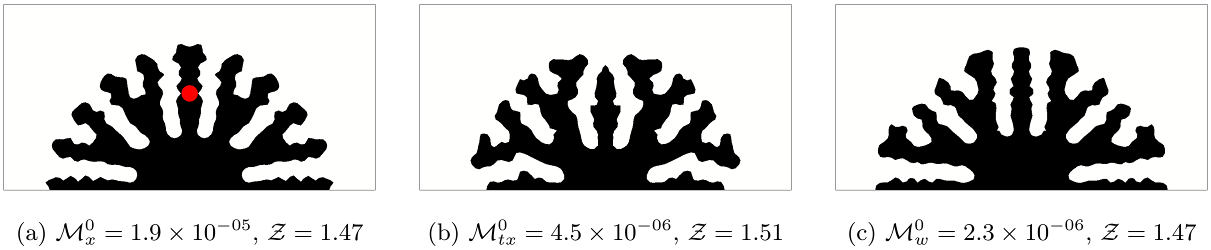


Figure 7.91: Convective design resulting from no feature size enforcement ($c_{\mathcal{M}} = \infty$).

7.92. The nature of the design problem, which is strongly non-convex, leads to these complicated shapes. Conceptually, all three formulations produce a similar design, with branches growing out from the initial circle. All formulations display a wavy material interface, which is attributed to the lack of smoothing applied on the designs. The formulations in Figures 7.92a and 7.92c display a smaller objective than the remaining formulation. It is not clear from these examples if any of the formulations is significantly better than the other ones in terms of minimizing the objective and satisfying the feature size.

Utilizing the feature size measure in (6.37), we optimize for different minimum feature sizes, as shown in Figure 7.93. All designs yield conceptually similar geometries, where branches grow out of the initial circle. We can observe that for all designs, the minimum feature size is well imposed. As the feature size parameter increases, the number of branches decreases, which yields a higher objective value.

The optimization problem drives the design to increase the surface area, and to form wavy

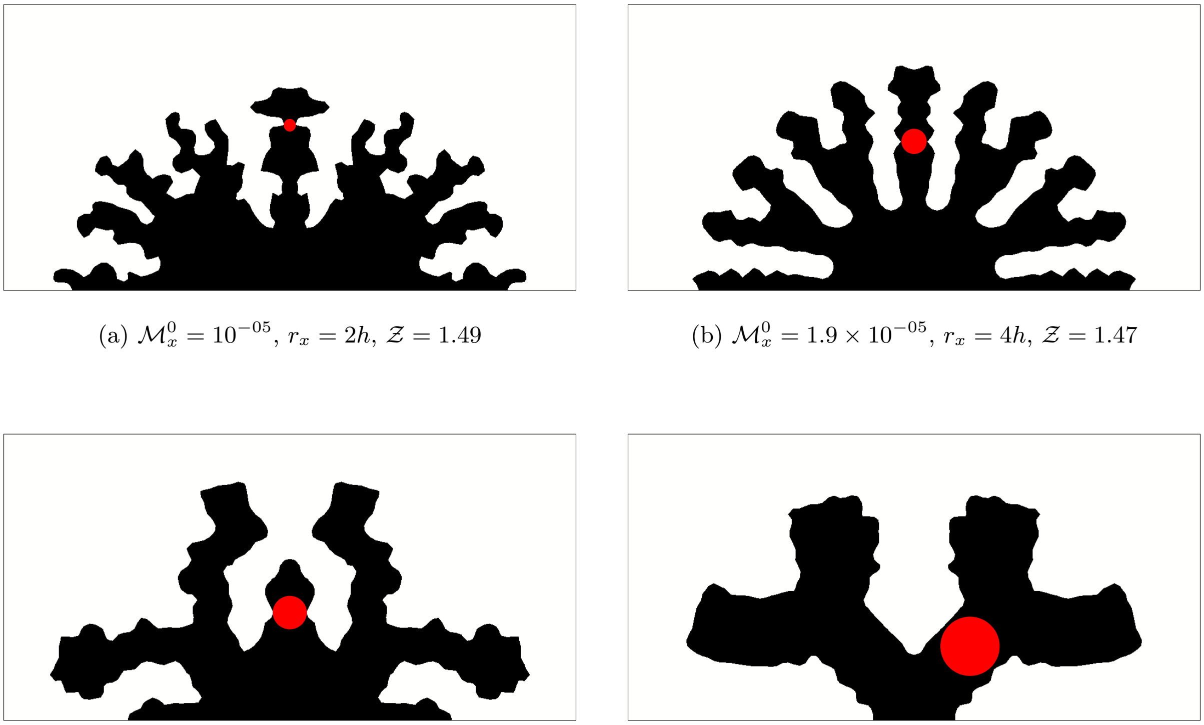


(a) $\mathcal{M}_x^0 = 1.9 \times 10^{-05}$, $\mathcal{Z} = 1.47$

(b) $\mathcal{M}_{tx}^0 = 4.5 \times 10^{-06}$, $\mathcal{Z} = 1.51$

(c) $\mathcal{M}_w^0 = 2.3 \times 10^{-06}$, $\mathcal{Z} = 1.47$

Figure 7.92: Optimized convective designs for different minimum feature size formulations.



(a) $\mathcal{M}_x^0 = 10^{-05}$, $r_x = 2h$, $\mathcal{Z} = 1.49$

(b) $\mathcal{M}_x^0 = 1.9 \times 10^{-05}$, $r_x = 4h$, $\mathcal{Z} = 1.47$

(c) $\mathcal{M}_x^0 = 3.6 \times 10^{-04}$, $r_x = 6h$, $\mathcal{Z} = 1.69$

(d) $\mathcal{M}_x^0 = 5.3 \times 10^{-04}$, $r_x = 12h$, $\mathcal{Z} = 1.95$

Figure 7.93: Optimized convective designs for different minimum feature size values after 1,000 iterations. Red circle indicates the desired minimum feature size.

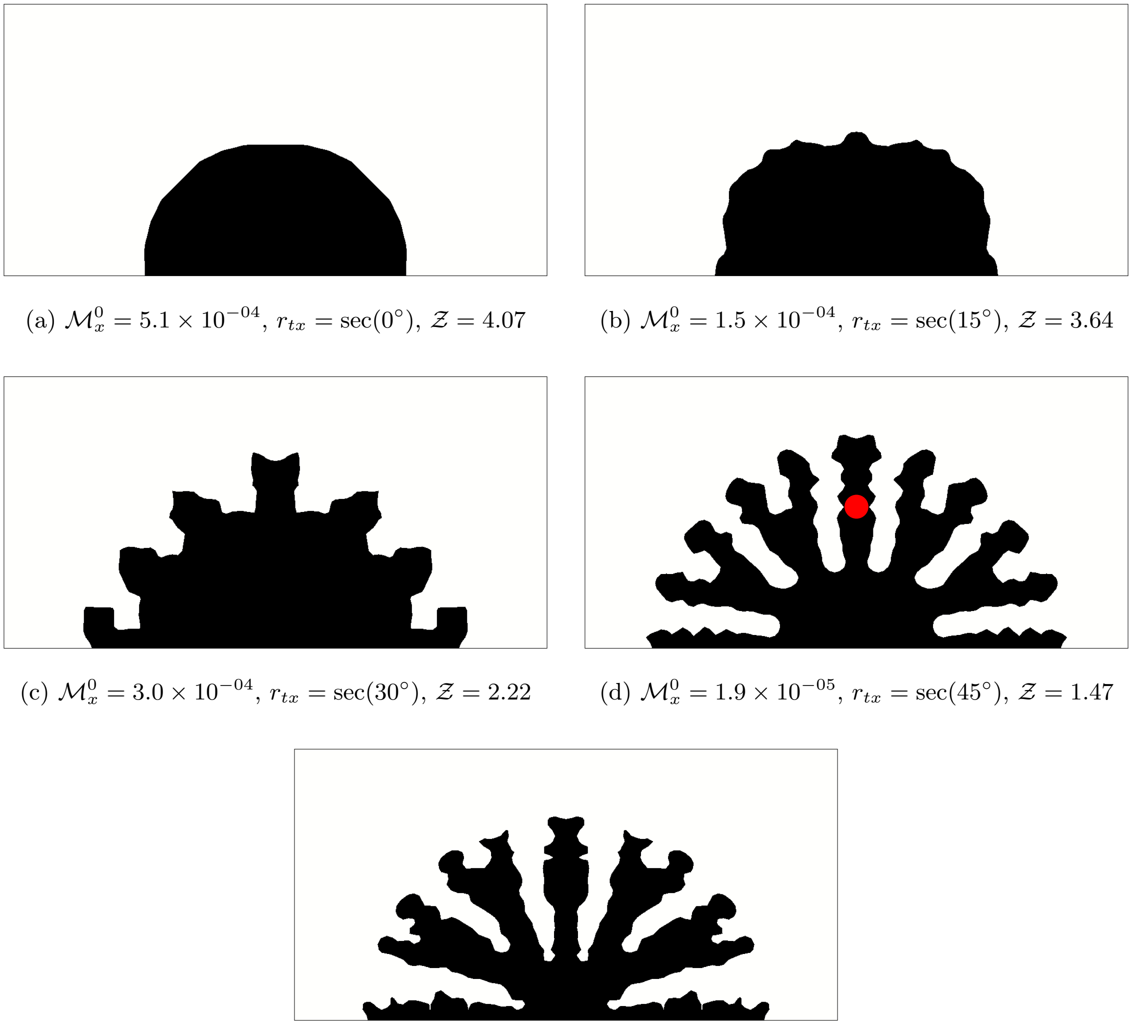
boundaries (Figure 7.93). This behavior makes the problem a good candidate for demonstrating the influence of the r_{tx} parameter. The parameter r_{tx} is an important scaling factor in (6.37). It identifies regions of nearby points that are excluded as their geodesic distance is small relative to their Euclidean distance. In a previous example in Section 7.9.6.3, the parameter r_{tx} was related to the secant function of an angle θ, where θ determines the smoothness and/or sharpness of a corner. Selecting the configuration of Figure 7.93b, the scaling r_{tx} is varied and the resulting designs are shown in Figure 7.94. Note that Figure 7.93b and 7.94d are the same problem. For small values of scaling, $r_{tx} \leq \sec(15°) = \sqrt{6} - \sqrt{2}$, little feature development occurs because the measure identifies nearly all curvatures as violations. As the value of the scaling increases, more features and wavy patterns are allowed. The choice of $r_{tx} = \sec(45°) = \sqrt{2}$ (Figure 7.94d) displays a visually appealing degree of feature development, while $r_{tx} = \sec(30°) = 2\sqrt{3}/3$ (Figure 7.94c) may be a good choice to promote smooth designs.

7.9.9 Design of a 3D MBB Beam

Here we study the capabilities of the feature size measure on 3D problems that have an initial design with multiple void inclusions. The example is the 3D analog the problem studied in Section 7.9.7. We study two configurations: one where no feature size constraint is imposed, and another where the feature size (6.37) is imposed as an objective penalty and computed over the negative phase. The penalty is imposed from the initial design, and no continuation approach is used. The feature size measure is enforced as a penalty on the objective, rather than as a constraint, to prevent the optimization algorithm from not merging holes and preventing changes in the topology of the design. The objective is then defined as:

$$\mathcal{Z} = \frac{\mathcal{U}}{\|\mathcal{U}^{(0)}\|} + w_{\mathcal{S}} \frac{\mathcal{S}}{\|\mathcal{S}^{(0)}\|} + w_{\mathcal{M}} \mathcal{M}_x^- \ . \tag{7.38}$$

The weights of the penalties, $w_{\mathcal{S}}$ and $w_{\mathcal{M}}$, are set to 10^{-01} and 10^{+04}, respectively. The feature size measure is not normalized in the objective function because its value at the initial design is 0 due to the minimum size being satisfied. The problem setup for both configurations is shown in Figure



(a) $\mathcal{M}_x^0 = 5.1 \times 10^{-04}$, $r_{tx} = \sec(0°)$, $\mathcal{Z} = 4.07$

(b) $\mathcal{M}_x^0 = 1.5 \times 10^{-04}$, $r_{tx} = \sec(15°)$, $\mathcal{Z} = 3.64$

(c) $\mathcal{M}_x^0 = 3.0 \times 10^{-04}$, $r_{tx} = \sec(30°)$, $\mathcal{Z} = 2.22$

(d) $\mathcal{M}_x^0 = 1.9 \times 10^{-05}$, $r_{tx} = \sec(45°)$, $\mathcal{Z} = 1.47$

(e) $\mathcal{M}_x^0 = 2.7 \times 10^{-05}$, $r_{tx} = \sec(75°)$, $\mathcal{Z} = 1.45$

Figure 7.94: Optimized convective designs with varying r_{tx} parameter after 1,000 iterations.

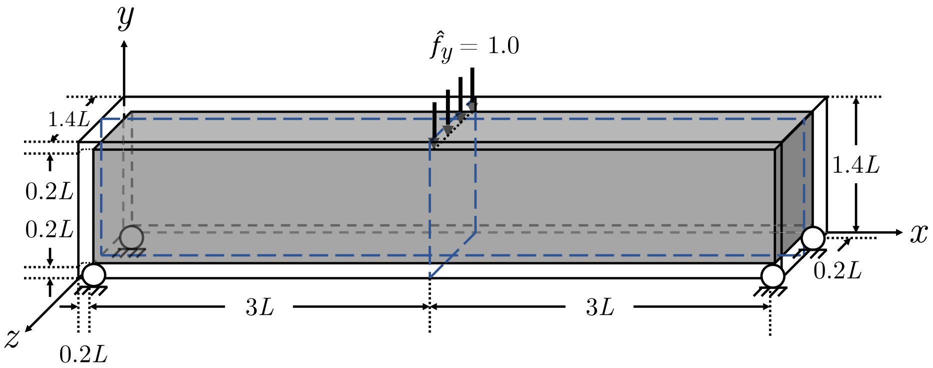


Figure 7.95: Problem setup for the 3D MBB beam example. Blue dashed lines denote the symmetry planes of the analysis domain.

7.95. The problem parameters are shown in Table 7.26. The initial design is shown in Figure 7.96. The minimum feature size is set to six times the element length scale $6h$.

The optimized material layouts after 1,000 iterations are shown in Figure 7.97, and the objective and volume constraint plots are shown in Figure 7.98. Similar to the problem studied in Example 7.3.4, the optimized material layout for the design without a feature size constraint (i.e. $c_M = \infty$) displays thin-wall features. The cross-sectional area of the design with the feature size measure shows that the feature size measure is well-imposed. The lack of smoothness on certain surface areas of the design and the wiggliness in the objective plot are attributed to the gradients of the geodesic Heaviside (6.29) not being computed in the sensitivity analysis. Imposing the feature size measure as a constraint may prevent changes in the topology of the design. Enforcing the measure as a penalty on the objective encourages design changes while satisfying the minimum feature size. The development of robust continuation approaches is necessary to prevent reaching local minima when using the measure as a constraint.

7.9.10 Design of a 3D Convective Design

This example is the 3D analog to the problem studied in Section 7.9.8. The problem setup is shown in Figure 7.99. We will a convective problem with feature size constraints (6.33) of $2h$ and

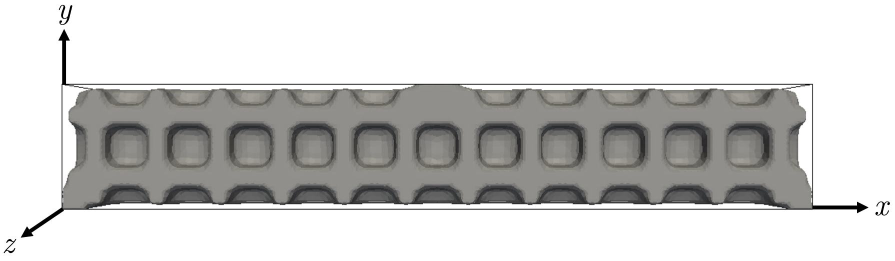
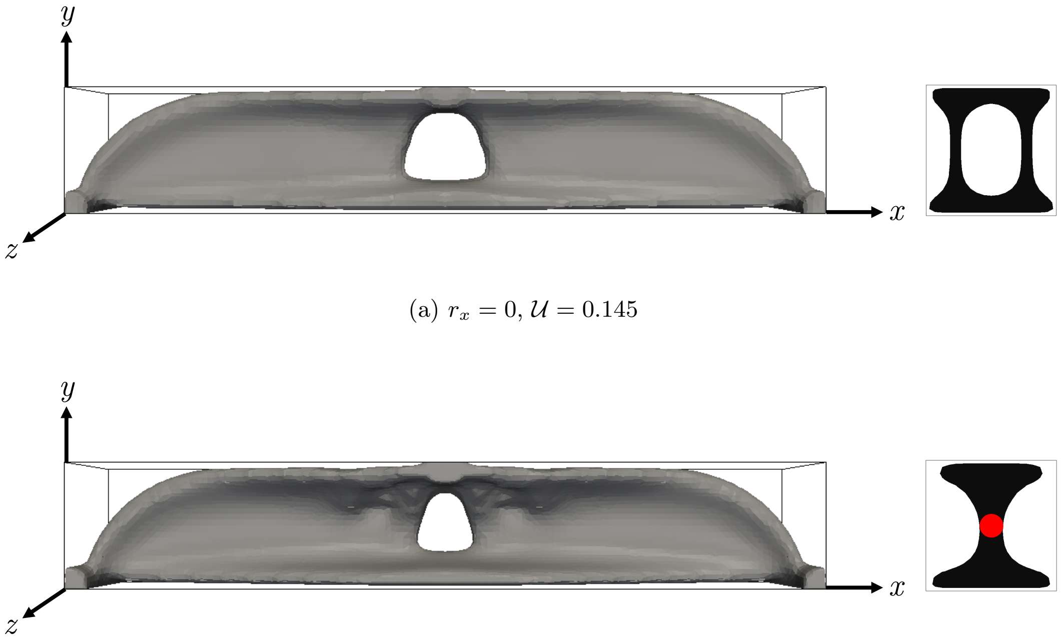


Figure 7.96: Initial design for the 3D MBB beam example. The domain is initialized with $13\times3\times3$ void inclusions of radii $0.16L$.

	Value
Element length scale	$h = 8.0 \times 10^{-02}L$
Density	$\rho = 1.0$
Young's modulus	$E = 1.0$
Poisson's ratio	$\nu = 3.0 \times 10^{-01}$
Volume ratio constraint	$c_V = 3.0 \times 10^{-01}$
Minimum feature size	$r_x = 6h$
Geodesic to Euclidean distance ratio	$r_{tx} = \sqrt{2} + \sqrt{6}$
Maximum search radius	$r_{\mathcal{M}} = r_{tx} \times r_x$
Minimum feat. size smooth Heaviside width	$w_x = 0.01$
Geodesic smooth Heaviside width	$w_{tx} = 0.01$
Phase domain smooth Heaviside width	$w_n = 0.01$
Smoothing filter radius	$r_\phi = 1.6h$

Table 7.26: Problem parameters for the 3D MBB beam example.



(a) $r_x = 0$, $\mathcal{U} = 0.145$

(b) $r_x = 6h$, $\mathcal{U} = 0.154$

Figure 7.97: Optimized material layout for the 3D MBB beam examples, with and without a feature size constraint.

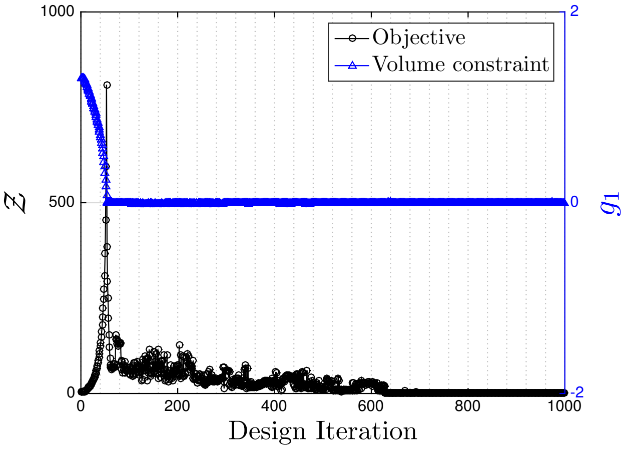


Figure 7.98: Objective and volume constraint plots for the 3D MBB beam example.

$6h$, applied over both phases from the beginning of the optimization process, without a continuation approach. The constraint parameter c_M is set to 10^{-04}. Both problems are computed for a total of $2,000$ designs iterations.

The optimized material layouts are shown in Figure 7.100. The cross-sectional areas show that the feature size constraint is well-imposed. Imposing the minimum feature size on this class of problems is simpler due to the initial design, and they do not require a continuation approach. As shown in previous examples, there is a trade-off between performance and enforcement of the minimum feature size; this is shown in Table 7.27.

7.10 Shape Smoothers

Manufacturing constraints considered in shape and topology optimization are often expressed in terms of the curvature value on the surface of the optimized material layout. In this section, we study the application of shape smoothers to increase the smoothness, and therefore, decrease the curvature of discretized surfaces. The discretized surface is represented by the level set interface (2.2) and modeled as a polygon mesh using the XFEM.

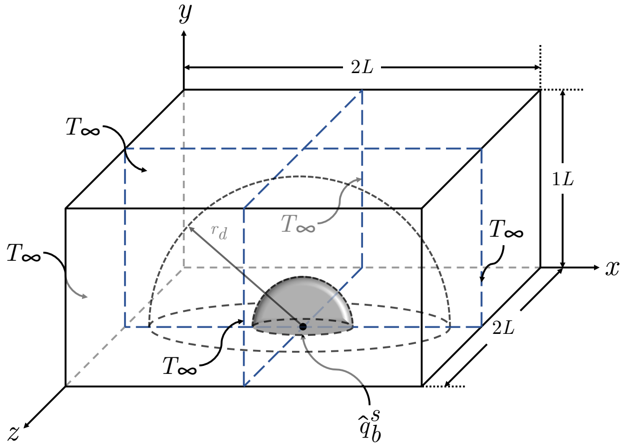
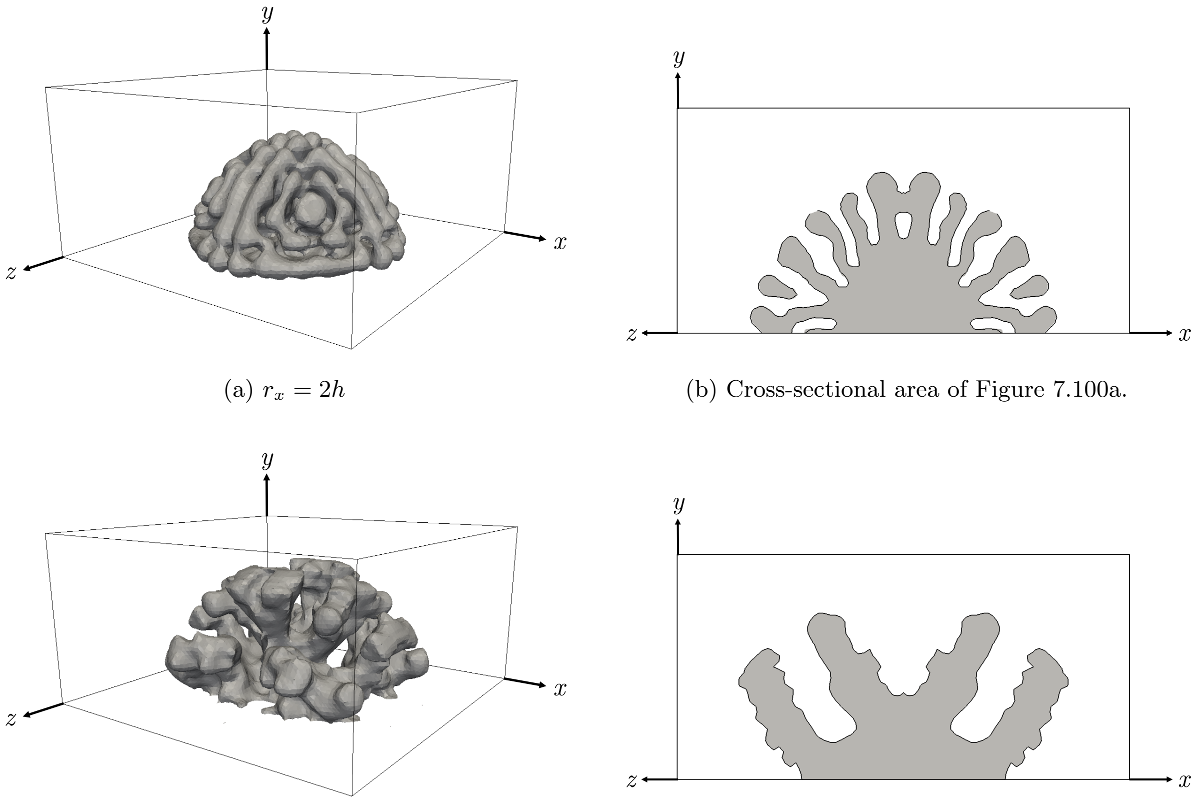


Figure 7.99: Problem setup for the 3D simplified convection example. Blue dashed lines denote the symmetry planes of the analysis domain.

	Temperature at point B
Figure 7.100a	3.559e+01
Figure 7.100c	3.558e+01

Table 7.27: Temperature values for the 3D convective design examples.



(a) $r_x = 2h$

(b) Cross-sectional area of Figure 7.100a.

(c) $r_x = 6h$

(d) Cross-sectional area of Figure 7.100c.

Figure 7.100: Optimized material layouts for the 3D convective design examples.

7.10.1 Curvature Sweeps

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 6.1.2. We will use the results of this example to select the most promising approaches and later minimize the curvature of the material boundaries.

7.10.1.1 Sweep Setup

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

$$\phi_i = r_c - \sqrt{x_i^2 + y_i^2} \ , \tag{7.39}$$

where ϕ_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 7.101. From (6.9) and (6.16), the analytical curvature is equal to:

$$\int_{\Gamma^0} \kappa_{\boldsymbol{n}}^2 \, \mathrm{d}\Gamma = \frac{2\pi}{r_c} \ . \tag{7.40}$$

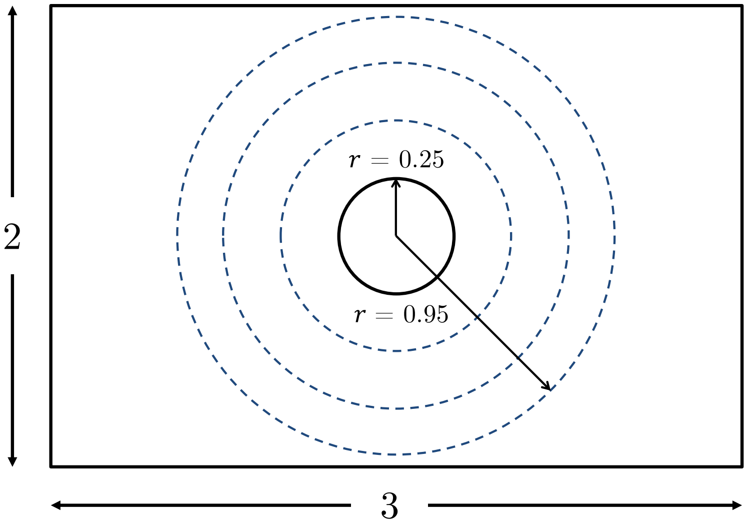


Figure 7.101: Setup for sweeping over the radius of a circle from 0.25 to 0.95.

The sinusoidal wave will be modeled by the following equation:

$$\phi_i = y_i - A_s \sin \pi x_i \ , \tag{7.41}$$

where we will sweep over the amplitude of the sinusoidal wave, A_s, from 0.25 to 0.75, as shown in Figure 7.102.

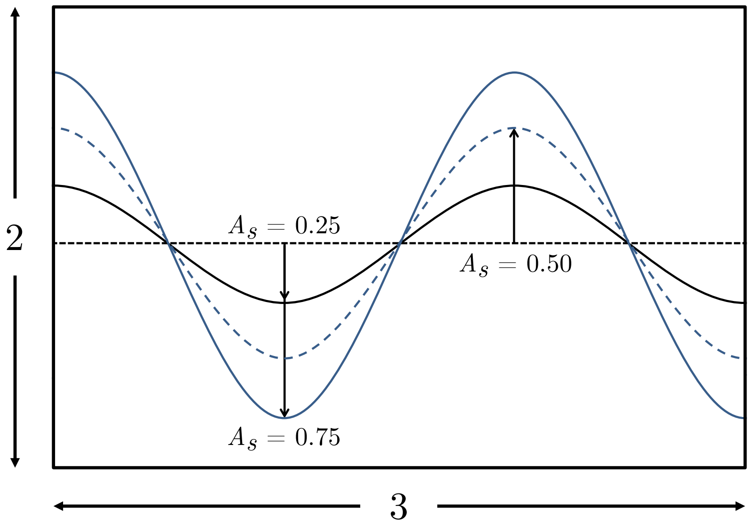


Figure 7.102: Setup for sweeping over the amplitude of a sinusoidal wave ranging from 0.25 to 0.75.

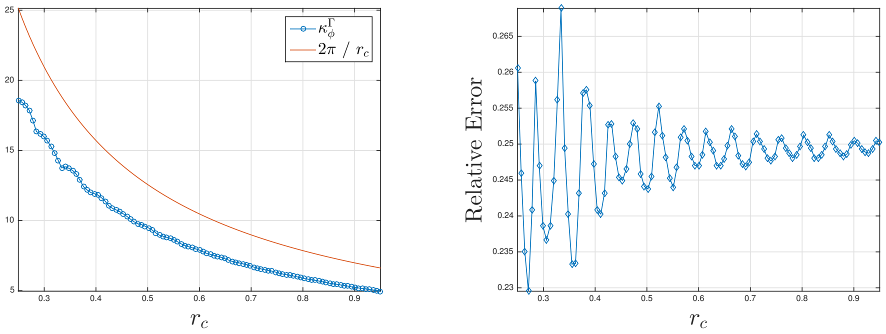
The mesh has spatial dimensions of $3L \times 2L$, and will be divided into 45×30 elements. The finer version of the mesh will have 180×120 elements.

7.10.1.2 κ_ϕ^Γ Results

The results for the circular sweep with the κ_ϕ^Γ formulation and mesh 45×30 are shown in Figure 7.103. The finer mesh results are shown in Figure 7.104.

7.10.1.3 $\kappa_{\boldsymbol{n}}^\Gamma$ Results

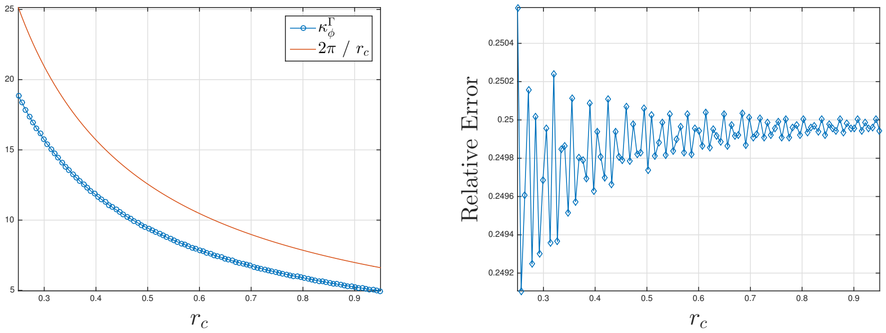
Using the $\kappa_{\boldsymbol{n}}^\Gamma$ formulation, we get the results displayed in Figures 7.105 and 7.106 for a circle and sinusoidal wave, respectively. The mesh size is 45×30. The scaling factor, γ_κ, is 10^{+02}.



(a)

(b)

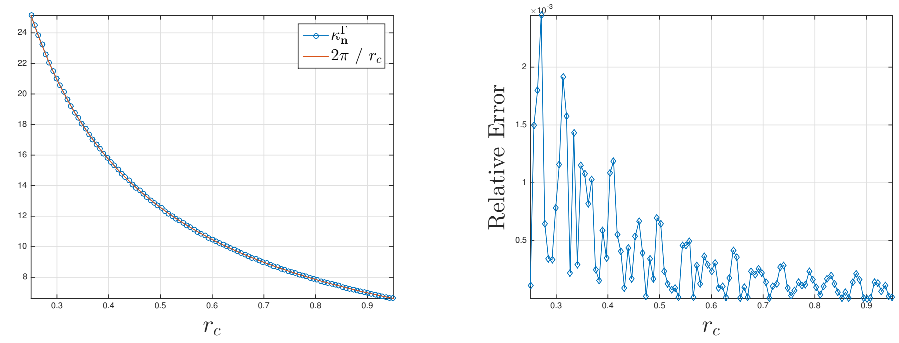
Figure 7.103: Squared curvature and absolute error for the circular sweep using the κ_ϕ^Γ formulation, with mesh 45 × 30.



(a)

(b)

Figure 7.104: Squared curvature and absolute error for the circular sweep using the κ_ϕ^Γ formulation, with mesh 180 × 120.



(a) (b)

Figure 7.105: Squared curvature and absolute error for the circular sweep using the $\boldsymbol{n}_u$ formulation, with mesh 45 × 30.

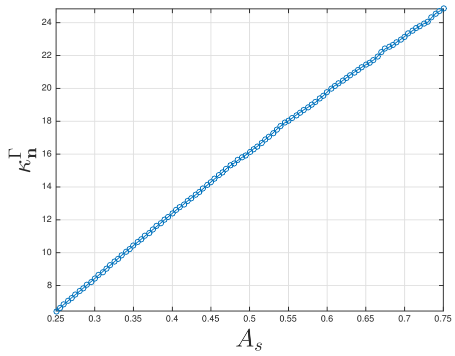


Figure 7.106: Squared curvature for the sinusoidal wave sweep using the $\boldsymbol{n}_u$ formulation, with mesh 45 × 30. This formulation does not have an analytical solution to compare against.

7.10.1.4 Discussion

Figures 7.103 and 7.104 show that using κ_{ϕ}^{Γ} causes oscillations in the measurement if the mesh is not fine enough. Using a projection scheme in the $\kappa_{\boldsymbol{n}}^{\Gamma}$ formulations yields a smoother curvature measure and decreases the error.

7.10.2 Projected Normal Curvature as an Objective Penalty

In this section, we impose our $\kappa_{\boldsymbol{n}}^{\Gamma}$ curvature formulation as a penalty on the objective. We utilize the problem setup from Section 7.3.1, without the additional padding domain. The initial design is shown in Figure 7.107. Our goal is to increase the smoothness of the design, and to reduce the sharpness of the internal corners. The objective formulation is given by:

$$\mathcal{Z} = \frac{\mathcal{U}^{s}}{\left\|\mathcal{U}^{s(0)}\right\|} + w_{\kappa} \frac{\kappa_{\boldsymbol{n}}^{\Gamma}}{\left\|\kappa_{\boldsymbol{n}}^{\Gamma(0)}\right\|} \ , \tag{7.42}$$

where the superscript (0) denotes the value at the beginning of the optimization process.

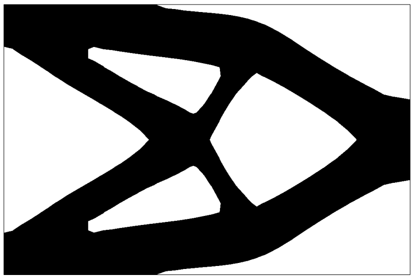
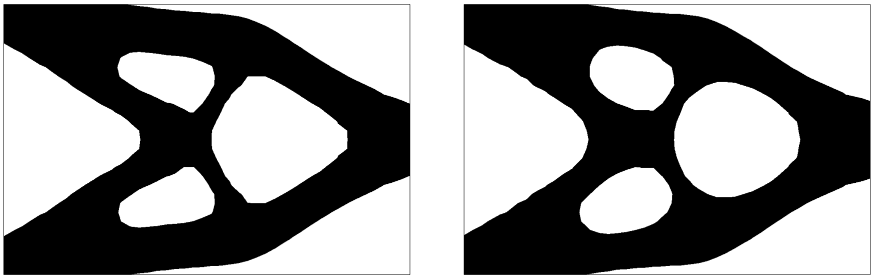


Figure 7.107: Initial design for the $\kappa_{\boldsymbol{n}}^{\Gamma}$ curvature optimization problems.

The optimized material layouts are shown in Figures 7.108a and 7.108b. We can observe that while the formulation smooths certain sections, it also generates sharp corners on other areas. This is attributed to the fact that the measure is dependent on the level set function, and does not operate directly on the discretized surface, Γ^{0}.



(a) $w_\kappa = 0.1$

(b) $w_\kappa = 1.0$

Figure 7.108: Optimized geometry for computing curvature using the $\kappa_{\boldsymbol{n}}^{\Gamma}$ formulation.

7.10.3 Spline Curvature as an Objective Penalty

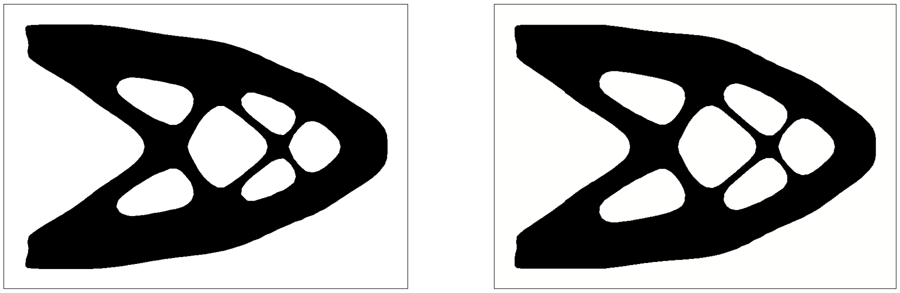
In this section, we impose our spline curvature formulation as a penalty on the objective. This formulation operates directly on the discretized material interface. We utilize the problem setup from Section 7.3.1. The initial design is given by the optimized material layout of Figure 7.20b. Our goal is to increase the smoothness of the design, and to reduce the sharpness of the internal corners. The objective formulation is given by:

$$\mathcal{Z} = \frac{\mathcal{U}^s}{\left\|\mathcal{U}^{s(0)}\right\|} + w_\kappa \frac{\kappa_{\boldsymbol{u}}^{\Gamma}}{\left\|\kappa_{\boldsymbol{u}}^{\Gamma(0)}\right\|} \ , \tag{7.43}$$

where $w_\kappa = 10^{-01}$, and the superscript (0) denotes the value at the beginning of the optimization process. Additionally, we study the effects of lower and upper bounds on the perimeter of the level set interface. We utilize the perimeter of Figure 7.20b and impose a ±5% variation on this value as:

$$\begin{aligned} g_2 &= \mathcal{S} / \left(1.05 \cdot \hat{\mathcal{S}}\right) - 1.0 \ , \\ g_3 &= 1.0 - \mathcal{S} / \left(0.95 \cdot \hat{\mathcal{S}}\right) \ , \end{aligned} \tag{7.44}$$

where $\hat{\mathcal{S}}$ is the prescribed perimeter with a value of $2.108 \times 10^{+01}$. The optimized material layouts are shown in Figure 7.109. The computed results are shown in Table 7.28. Without the perimeter constraints (Figure 7.109a), the curvature formulation smooths the interface and reduces the size of the holes at the cost of increasing the strain energy. Given a large amount of optimization



(a) No perimeter constraints.

(b) Lower and upper bounds are imposed on the perimeter.

Figure 7.109: Optimized material layout for the 2D short cantilever beam example, with a curvature penalty weight of $w_\kappa = 10^{-01}$.

iterations, the curvature will form circular shapes out of all the inclusions. On the other hand, imposing constraints in the optimization problem (Figure 7.109b) smooths the design but preserves the original shape of the design.

7.10.4 Spline Curvature Density

In this section, we compute the spline curvature density for a shape with a sharp corner in the middle, as shown in Figure 7.110. The spline curvature density is defined as the ratio of the spline curvature value (6.20) and the length of the zero level set interface (4.73). The sharpness of the corner is modified by increasing the angle at the edges that form the corner. We set the curvature search radius as $r_\kappa = \infty$, and the curvature springs stifness as $k_\kappa = 10^{-02}$.

The curvature density results are shown in Figure 7.111. We can observe that the curvature density is directly proportional to the angle at the corner. We will use this information to penalize intermediate curvature values using a smooth Heaviside projection.

	Strain Energy	Perimeter	Curvature
Original	1.363×10^{-01}	$2.108 \times 10^{+01}$	N/A
Figure 7.109a	1.402×10^{-01}	$1.886 \times 10^{+01}$	3.693×10^{-05}
Figure 7.109b	1.384×10^{-01}	$1.986 \times 10^{+01}$	1.004×10^{-04}

Table 7.28: Strain energy, perimeter, and curvature for the 2D short cantilever beam example of Section 7.10.3.

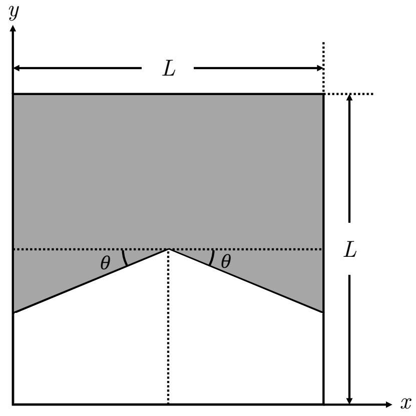


Figure 7.110: Problem setup for the sweep over different angles example.

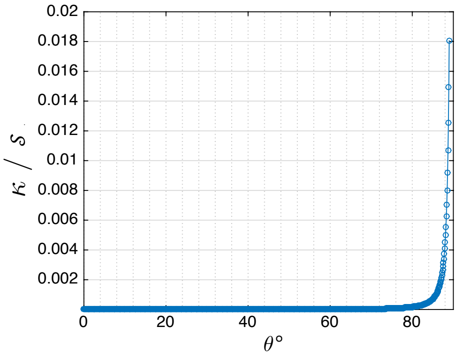
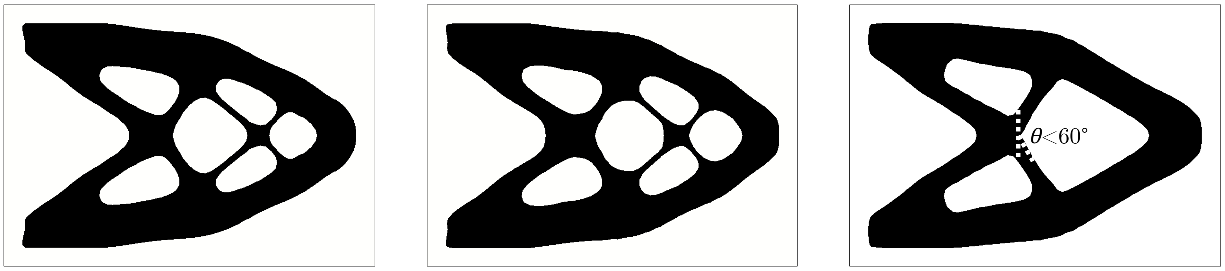


Figure 7.111: Curvature density distribution for different angles.



(a) 30 degrees.

(b) 45 degrees.

(c) 60 degrees.

Figure 7.112: Optimized material layout for the 2D short cantilever beam example, with a penalty weight of $w_\kappa = 10^{-01}$ applied over intermediate curvature values.

7.10.5 Penalizing Intermediate Spline Curvature Values

In this example, we utilize the curvature density information from Section 7.10.4 to only penalize curvature values above a certain threshold. We penalize curvature for density values above 30, 45, and 60 degrees. We utilize the same problem setup as in Section 7.10.3, but we impose a ±10% variation on the perimeter (7.44).

The optimized material layouts are shown in Figure 7.112. The strain energy, perimeter, and curvature are shown in Table 7.29. We can observe that using a curvature density threshold to penalize the curvature of the interface leads to different geometries. The layout in Figure 7.112c displays a kink in one of the inclusions; however, this is expected, as the formulation only penalizes curvature with a degree larger than 60° on the outside of the shape.

	Strain Energy	Perimeter	Projected curvature
Original	1.363×10^{-01}	$2.108 \times 10^{+01}$	N/A
Figure 7.112a	1.385×10^{-01}	$1.984 \times 10^{+01}$	9.820×10^{-05}
Figure 7.112b	1.397×10^{-01}	$1.984 \times 10^{+01}$	6.566×10^{-05}
Figure 7.112c	1.415×10^{-01}	$1.879 \times 10^{+01}$	9.978×10^{-05}

Table 7.29: Strain energy, perimeter, and curvature for the 2D short cantilever beam example of Section 7.10.5.

Chapter 8

Conclusions

In this thesis, we presented a topology optimization framework combining an explicit Level Set Method (LSM) for describing the geometry and an eXtended Finite Element Method (XFEM) for predicting the physical response. The framework builds upon previous optimization studies that utilized the LSM and the XFEM. The framework presents state-of-the-art stabilization methods for immersed boundary techniques, such as the face-oriented ghost-penalty methods, and state-of-the-art methods for the weak enforcement of boundary conditions.

The framework can be divided into 4 areas: the optimization model, the design model, the analysis model, and the fabrication model. The conclusions with respect to each area of study are summarized subsequently.

8.1 Analysis model

Departing from simplified and potentially inaccurate enrichment strategies frequently used in topology optimization, we have presented a generalized Heaviside-step enrichment strategy that consistently interpolates the state variables for complex geometries without the need to adaptively refine the mesh. The XFEM formulation presented in this paper is robust and efficient in analyzing configurations with complex geometries, which often emerge in topology optimization. Heaviside-step enrichment functions provide great flexibility in handling a broad class of physical problems, but may require enforcing additional continuity conditions across the phase boundaries. The proposed optimization scheme was applied to two-phase material-void and material-material problems in 2D and 3D, with steady-state and transient behavior, in linear elastic structures and laminar incompressible Navier-Stokes flow problems, coupled with energy and species transport models. For material-void problems, elements entirely in the void phase were omitted in the element assembly process and the degrees-of-freedom that interpolate the void phase were eliminated from the system of equations; both techniques reduced the computational cost of solving the XFEM problem. The boundary conditions for the material interface and on the external surfaces were enforced weakly using either a stabilized Lagrange multiplier method or Nitsche's method.

We studied the geometric preconditioner for Heaviside-enriched XFEM problems of Lang et al [2014] and face-oriented ghost-penalty methods to stabilize the system of equations for cases where the degrees-of-freedom interpolate the solution in small intersection areas. The geometric preconditioner is constructed from both the nodal basis functions and the interface configuration, and can be computed prior to constructing the system matrices, making it well-suited for nonlinear problems. The scheme was extended to 3D, and studied in the context of linear elasticity. The face-oriented ghost-penalty formulation was extended to work with a generalized Heaviside-step enrichment strategy, and studied in the context of laminar incompressible flow problems. By implementing either of the proposed stabilization schemes, the ill-conditioning due to small element intersections is eliminated. Furthermore, the condition number of the XFEM problem is comparable to that of a body-fitted mesh using the traditional FEM.

A novel auxiliary indicator field was modeled in order to identify isolated fluid regions surrounded by solid in the flow problems, and free-floating solid particles surrounded by void in the linear elasticity problems. For the flow problems, we weakly enforced a pressure constraint on these regions to prevent a singular analysis problem. For the linear elasticity problems, we added soft springs to the computational model to bypass the ill-conditioning issues due to the free-floating material.

The analysis capability of the framework was validated quantitatively through comparison against previous benchmark studies, and qualitatively through its application to topology optimization problems. The design optimization problems converged to intuitive designs and resembled well the results from previous 2D studies. As demonstrated by the numerical studies presented in this thesis, our LSM-XFEM approach is accurate, robust, and applicable to a broad range of design problems for different physical phenomena. However, the computational costs can be significant, in particular if accurate flow solutions are required. This is in parts due to the enlarged bandwidth of the linear systems caused by the face-oriented ghost-penalty formulation, but more importantly due to the lack of an adaptive mesh refinement strategy. Future research may focus on integrating mesh refinement strategies into the LSM-XFEM framework such that the boundary layer phenomena are captured accurately and efficiently.

8.2 Design model

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. We applied a perimeter constraint to suppress small geometric features in the optimized design. The SIMP approach may require a strong projection to achieve clear binary “0-1” results with comparable physical response. 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. These observations are in agreement with the findings of van Dijk et al [2013] and Sigmund and Maute [2013] for level set methods using Ersatz material and emphasize the need for regularization techniques with local shape control in level set topology optimization.

8.3 Optimization model

The optimization problems used a gradient-based algorithm, and were solved using a nonlinear programming method; the flexibility of this scheme allowed us to define additional design variables in addition to the parameters of the LSF. For example, additional design variables were used to control the position and size of the outlets in the flow problems.

Shape sensitivities in the context of Heaviside-enriched XFEM were studied. Using the adjoint method, shape sensitivities were shown to be a function of the sensitivities of the residual of the governing equations, $\partial \boldsymbol{R}/\partial s_i$. Computing the semi-analytical shape sensitivities involved finite differencing the term $\partial \boldsymbol{R}_e/\partial \boldsymbol{x}_k^{\Gamma}$ in (4.66); this was shown to be an effective and easy to implement alternative to analytical approaches. The corresponding shape sensitivities were observed to be almost insensitive to the perturbation sizes.

In accordance with the adopted generalized enrichment strategy for the XFEM, the interface is not allowed to intersect a node. Consequently, the material interface is shifted if it gets too close to a node. It was observed that performing this shift in the interface position during the finite difference perturbations resulted in an inaccurate computation of shape sensitivities.

Several material-material examples were studied. The accuracy of the semi-analytical shape sensitivities was established via comparisons against finite differenced shape sensitivities. A numerical comparison with shape sensitivities obtained using a body-fitted mesh was performed. Although, the latter framework resulted in smoother sensitivities, an average relative difference of only 0.69% was observed when using the XFEM. This difference further reduced by 36% after two levels of mesh refinement. One disadvantage of the Heaviside-enriched XFEM is the need for interface conditions to enforce continuity in the solution across the material interface. These interface conditions have influence over the behavior of the sensitivities. Using a circular inclusion, we showed that the smoothness of the shape sensitivities is heavily affected by the discretization of the design geometry. As a result, it is possible to obtain smoother sensitivities through mesh refinement. A two-inclusion problem was studied to demonstrate the effect of merging topologies on shape sensitivities. The merging of topologies, not being a continous phenomenon, resulted in a distinct spike in the shape sensitivities. Application to an incompressible fluid flow example demonstrated the suitability of the Heaviside-enriched XFEM framework for the computation of shape sensitivities in 3D nonlinear problems.

8.4 Fabrication model

A level set gradient measure was applied to prevent the formation of features smaller than an element size. Although the gradient penalty measure prevented the formation of very thin branches in the conjugate heat transfer problem, the optimized design was different from the original study (c.f. Alexandersen [2015]) due to the lack of a feature size control for arbitrary sizes.

We presented a novel measure to identify and control the minimum feature size in LSM-XFEM topology optimization. The scalar measure identified features using the discretized surface geometry of the XFEM model. This differentiates it from existing methods that directly utilize the LSF. The measure differs from the quadratic energy approach of Chen et al [2008] in that it does not use surface tangent information. This allows it to permit sharp corners found on a discretized surface.

The measure was demonstrated via numerical experiments using two design problems: the MBB beam and a heat transfer device with edge convection. The measure can be incorporated into the optimization problem as a constraint or as a penalty on the objective. In cases where the feature size could be enforced strictly the constraint was used. In cases where a strict enforcement overly restricted design changes the measure was enforced with the penalty.

The measure was shown to be effective in identifying and enforcing a minimum feature size, and it was enforced in either a single material phase or both. The construction makes deterring the formation of small features convenient. Removing existing small features requires more care. Incorporating the measure as an inequality constraint allows for an strict enforcement, but is a difficult approach when initial designs violate the constraint. Furthermore, when used as a constraint, it can be problematic to enforce a minimum feature size much larger than the current geometry, and it is necessary to use a continuation approach. The application of the measure to the convective design problem was shown to be simpler due to the initial design. The use of a simple initial design also ensured that the constraint was satisfied from the beginning. Future work is necessary to develop robust continuation schemes or formulations to balance feature development and minimum feature size constraint.

The feature size measure allows some control over the surface roughness or curvature via its geodesic parameter r_{tx}. The parameter was shown to be related to the secant function of an angle θ, and to be able to prevent the formation of curvatures associated with this angle.

Manufacturing constraints in milling processes can be expressed in terms of the smoothness of the optimized design. We have developed several smoothing techniques to control and maximize the smoothness of our geometries. Numerical studies showed that our methods were effective in maximizing the smoothness of the designs.

Chapter 9

Contributions

This section builds upon the issues, challenges, research topics, and results introduced in previous sections, and summarizes the contributions of this thesis. These are outlined as follows:

(1) **Development of an efficient algorithm to identify disconnected topologies.** The Heaviside-enriched XFEM described in this work augments the interpolation space of the solution field within a fixed background mesh. In the context of topology optimization, this approach meant that for some design iteration, the geometry would be described by disconnected pieces of material, as shown in Figure 3.5. In order to avoid the artificial coupling of these pieces, the XFEM needed to properly identify such regions as disconnected and apply the correct enrichment function (3.1). This problem is in its core a computer vision issue. We developed a topology-flipping algorithm, which was used in combination with a flood-fill algorithm, to correctly identify both the connected and disconnected topologies in the analysis domain. This algorithm constitutes the foundation of the XFEM approach utilized in this document. Numerical experiments performed during the course of this dissertation showed that the algorithm could accurately identify the topology of an analysis domain with millions of finite elements in milliseconds. The algorithmic implementation of the method is described in the internal document in Appendix A, and in the publication in Appendix C.

(2) **Study of the correctness and accuracy of the shape sensitivities of the opti-**

mization criteria. In this dissertation, we used a gradient-based optimization algorithm as part of our optimization framework. The algorithm required that we provide the sensitivities of the objective function and the inequality constraints of the optimization problem. These sensitivities were computed with respect to the design variables (1.3), which in turn were used as the parameters of a discretized Level Set Function (2.9). During the computation of the shape sensitivities, it was necessary to compute the gradient of the residual function of the governing equations with respect to the coordinates of the points at which the zero level set of the Level Set Function intersected the finite elements (4.66). The gradient of the residual could be computed analytically or by finite difference. The analytical route was complex and computationally expensive because the nodal basis functions were defined as functions of the design variables. The finite difference approach was easier to implement, but could present issues in terms of the accuracy of the computation. Furthermore, the Heaviside function was discontinuous by construction. This quality of the enrichment function presented advantages as well as challenges in the computation of the shape sensitivities. We studied the shape sensitivity analysis using backward, forward, and central finite differences approaches. We developed a methodology to select an epsilon value for the perturbations such that the sensitivities are well-posed. We compared the results against analytical shape sensitivities, and showed that our method is accurate, and that finite differencing the residual sensitivities is an effective and easy to implement alternative.

(3) **Application of a geometric preconditioner for the Heaviside-enriched XFEM in 3D problems.** As the geometry of the design evolves during the optimization process, the interface of the embedded body may lead to intersection configurations where certain degrees-of-freedom interpolate in small intersection areas, as shown in Figure 5.1. This produces an ill-conditioning of the nonlinear problem, which manifests itself through an increase in the condition number of the linearized system. This phenomenon is more pronounced in 3D problems than in 2D ones. To guarantee stability, as well as to improve

the conditioning of the system, we expanded the geometric preconditioning scheme of Lang et al [2014], which was introduced and studied for 2D heat conduction and flow problems, onto 3D problems in structural mechanics. The preconditioner scheme builds a scaling matrix to balance the influence of all degrees-of-freedom in the system. Numerical experiments showed that the preconditioner successfully decreased the condition number of the system. The method is generic in that it was built based on the design geometry, which in turn depends on the Level Set Function. This meant that the approach can be applied to multiple physical phenomena without any modifications to the scaling matrix. The study of the preconditioner is shown in Appendix C. This work constitutes the first use of the preconditioner in topology optimization problems.

(4) **Development of an auxiliary indicator field to efficiently recognize isolated regions arising during the optimization process.** A novel auxiliary indicator field is developed in order to identify isolated regions of a specific phase surrounded by the domain of the opposite phase. For example, fluid flow “puddles” surrounded by the solid domain, or free-floating structural particles surrounded by the void domain. These isolated regions lead to an ill-conditioned system of equations because the absolute value of the pressure is not governed in the flow problems, or because they produce a rigid body mode in the linear elasticity problems. In order to stabilize the system, we added a penalty formulation to weakly enforce a pressure constraint in the flow problems, and modeled an artificial system of soft springs to the linear elasticity equations in the structural problems. This penalty is only applied to the domains identified by the auxiliary indicator field as isolated regions. The auxiliary field is modeled as a linear diffusion problem in the phase region of interest. The diffusion model is not applied in the other phase. The auxiliary field serves as a binary indicator that is active inside the isolated regions. Numerical experiments showed that applying the penalty only over the isolated regions does not affect the optimization process nor the accuracy of the physical response of the system. Similar approaches in

the literature apply the penalty over the entire domain; in our study, we showed that this approach can cause the solution to violate the incompressibility condition in incompressible flow problems.

(5) **Comparison of the LSM-XFEM framework against traditional density-based topology optimization methods.** The first application of both the Level Set Method and the Heaviside-enriched eXtended Finite Element Method as a topology optimization framework for problems in 3D was performed as part of this dissertation. The application was studied in the context of 3D linear elastic structures. The study compared the framework against traditional density-based topology optimization methods, such as SIMP. The study revealed several differences between the two approaches; for example, it showed that the XFEM may provide a faster convergence of the optimization algorithm, especially for multimaterial problems. The LSM-XFEM framework also provides a better description of the geometry and material distribution than density methods, and therefore requires coarser meshes which leads to faster computations. However, the study proved that, in contrast to density methods and its smoothing filter (1.12), the LSM-XFEM framework is not capable of controlling the minimum feature size of the design without additional regularization techniques. The minimum feature size is relevant in order to accurately represent the physics and to meet manufacturing constraints; for example, it is not desirable for the design to have a small feature that the manufacturing process cannot represent. Furthermore, we showed that the LSM-XFEM framework provides a more accurate response of the system because of the crisp definition of the boundaries. The method can accurately describe the physics at the material interface without extensive mesh refinement with the use of stabilized Lagrange multipliers or Nitsche's method. The framework does not require any postprocessing to extract the optimized material layout. The latter proved significantly advantageous for the use of the framework in rapid prototyping. The study is provided in Appendix C.

(6) **Extension of face-oriented ghost-penalty methods for use in Heaviside-enriched XFEM and application in topology optimization.** Small intersection areas in incompressible flow problems may cause an ill-conditioning of the system, which manifests itself through oscillations in the stresses. Convective flows require special treatment and additional stability terms. To properly model incompressible Navier-Stokes flow, and energy and species transport in the context of topology optimization, we extended the face-oriented ghost-penalty formulations of Burman and Hansbo [2012], Burman and Hansbo [2014], and Schott and Wall [2014] onto Heaviside-enriched XFEM. The formulation adds additional terms to the incompressible Navier-Stokes and advection-diffusion equations to penalize discontinuties in the spatial gradients of the velocity, pressure, and temperature fields across the common facets of intersected elements. These terms accounted for **viscous**-dominated and **convection**-dominated flow regimes. The accuracy of the method was validated quantitatively through comparison against benchmark examples, and qualitatively through their application to 3D laminar flow topology optimization problems. The topology optimization problems were modeled after density-based examples found in the literature. This work constitutes the first validation and application of face-oriented ghost-penalty methods in topology optimization.

(7) **Study the accurate and robust application of boundary conditions in the context of topology optimization.** Boundary conditions on the level set interface are enforced weakly. Enforcing the boundary conditions in an accurate and robust manner is important to properly model the physical response of the system, and to compute the sensitivities needed for the optimization problem. Two methods were studied to weakly enforce boundary conditions: stabilized Lagrange multipliers and Nitsche's method. We applied stabilized Lagrange multipliers to 3D linear elasticity problems, and utilized Nitsche's method in 3D incompressible laminar flow. We show that to ensure robustness, we needed additional terms in the formulations to account for the nonlinearity of the problems, as shown in

Schott and Wall [2014]. We validated the enforcement of boundary conditions quantitatively through benchmark examples in the literature.

(8) **Application of the framework to a real-life design problem.** In Section 7.8.1, we studied the design of passive coolers. The engineering design problem was modeled after the setup provided by Alexandersen [2015] for a light-emitting diode (LED) lamp. LED lamps have a lifespan and electrical efficiency that is several times better than incandescent and fluorescent lamps. However, unless adequately cooled, their lifespan is reduced because 70% of the energy supplied to them is converted to heat. The objective of the example was to minimize the temperature of the LED package, which is subject to heat transfer dominated by natural convection. To the authors knowledge, this work is the first to treat a real-life application, using the LSM and the XFEM to model a correct and coupled physical model. The optimized design showed interesting features that are currently being incorporated into industrial designs of passive coolers. The design exhibited tree-like branches extending out from the center of the design domain, which conduct the heat away from the LED package and transfer it to the flowing air by allowing the flow to move between the branches.

(9) **Application of a minimum feature size control measure in 3D problems.** Controlling the minimum feature size of an optimized material layout is paramount to ensure that the governing equations can accurately model the physical response of the feature, and that the design can be manufactured. In this thesis, we introduced a novel measure to control the minimum feature size of a design geometry in 3D topology optimization problems. Our feature size control method relied on the explicit Level Set Method (2.9) and on the eXtended Finite Element Method (3.1). Several feature size control techniques for LSM-based topology optimization rely on preserving a signed distance field in the LSF; however, this requires a refined mesh to accurately represent the geometry. An example of a feature size control technique which does not depend on a signed distance field is the quadratic energy method of Chen et al [2008], which compares the tangent vectors on the

interface at different points. Nevertheless, this approach is not effective when computing the measure on discretized surfaces. The feature size control measure developed here identifies violations of a minimum feature size, and does not require a signed distance field because it operates on the discretized polygon mesh of the level set interface, as provided by the XFEM. A comparison of our formulation against the one from Chen et al [2008] is also provided. The measure was demonstrated on structural and convective heat transfer topology optimization problems.

(10) **Development of shape smoothers to control the shape of the design geometry and meet manufacturing constraints.** The minimum feature size measure may not be sufficient to meet manufacturing constraints. Sharp edges also present issues in 3D printing and milling processes, due to the minimum resolution of the machines. In this thesis, we studied several novel measures to control and improve the smoothness of the material interface in the context of topology optimization.

Chapter 10

Future Work

This thesis has introduced a topology optimization framework that utilizes the Level Set Method to describe the design geometry and the eXtended Finite Element Method to solve the governing equations and measure the performance of the design. There are a number of short term (shown first) and long term questions and avenues for further research, summarized here:

(1) **Extend shape smoothers to 3D.** Natural convection problems in 2D and 3D lead to designs that form several thin branches, and may require additional smoothing to meet manufacturing constraints. The shape smoothers introduced here need to be implemented and studied for 3D problems.

(2) **Study the feature size measure with topological derivatives to reduce influence of the initial design.** In our framework, the initial design is usually set as an agglomeration of multiple inclusions that later merge or change shapes during the optimization process. The minimum feature size measure studied here may prevent the merging of these inclusions, and therefore, prevent changes in the design. Alternate methods that do not require us to feed a "swiss cheese"-like initial design need to be studied, such as topological derivatives [Sá et al, 2016].

(3) **Study adaptive mesh refinement to perform topology optimization with boundary layer meshes.** We currently do not have the capability to adaptively refine the mesh nor to generate boundary layer meshes at the material interface for design changes during

the optimization process. This is important for optimization problems that require the computation of the spatial gradients of the solution at the material interface with high accuracy.

Bibliography

Alexandersen J (2015) Topology optimisation of passive coolers for light-emitting diode lamps. In: 11th World Congress of Structural and Multidisciplinary Optimisation

Allaire G, Jouve F, Toader A (2002) A level-set method for shape optimization. Comptes Rendus Mathematique 334(12):1125–1130

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

Allaire G, Jouve F, Michailidis G (2014) Thickness control in structural optimization via a level set method, URL `https://hal.archives-ouvertes.fr/hal-00985000`

Annavarapu C, Hautefeuille M, Dolbow JE (2012) A robust nitsche's formulation for interface problems. Computer Methods in Applied Mechanics and Engineering 225:44–54

Babuška I, Melenk JM (1997) The partition of unity method. International Journal for Numerical Methods in Engineering 40(4):727–758

Bazilevs Y, Hughes TJ (2007) Weak imposition of dirichlet boundary conditions in fluid mechanics. Computers & Fluids 36(1):12–26

Bazilevs Y, Michler C, Calo V, Hughes T (2010) Isogeometric variational multiscale modeling of wall-bounded turbulent flows with weakly enforced boundary conditions on unstretched meshes. Computer Methods in Applied Mechanics and Engineering 199(13):780–790

Béchet É, Minnebo H, Moës N, Burgardt B (2005) Improved implementation and robustness study of the x-fem for stress analysis around cracks. International Journal for Numerical Methods in Engineering 64(8):1033–1056

Belytschko T, Black T (1999) Elastic crack growth in finite elements with minimal remeshing. International Journal for Numerical Methods in Engineering 45

Bendsøe M (1989) Optimal shape design as a material distribution problem. Structural and Multidisciplinary Optimization 1(4):193–202

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

Bendsøe M, Sigmund O (1999) Material interpolation schemes in topology optimization. Archive of Applied Mechanics 69(9-10):635–654

Bendsøe M, Lund E, Olhoff N, Sigmund O (2005) Topology optimization-broadening the areas of application. Control and Cybernetics 34(1):7

Bendsøe MP, Sigmund O (2003) Topology Optimization: Theory, Methods and Applications. Springer

Borrvall T, Petersson J (2003) Topology optimization of fluids in Stokes flow. International Journal for Numerical Methods in Fluids 41(1):77–107

Burman E (2012) A penalty-free nonsymmetric nitsche-type method for the weak imposition of boundary conditions. SIAM Journal on Numerical Analysis 50(4):1959–1981

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

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

Burman E, Hansbo P (2014) Fictitious domain methods using cut elements: Iii. a stabilized nitsche method for stokes' problem. ESAIM: Mathematical Modelling and Numerical Analysis 48(03):859–874

Burman E, Fernández MA, Hansbo P (2006) Continuous interior penalty finite element method for oseen's equations. SIAM journal on numerical analysis 44(3):1248–1274

Burman E, Claus S, Hansbo P, Larson MG, Massing A (2014) Cutfem: discretizing geometry and partial differential equations. International Journal for Numerical Methods in Engineering

CGAL (2009) Computational Geometry Algorithms Library. http://www.cgal.org

Chen S, Wang M, Liu A (2008) Shape feature control in structural topology optimization. Computer-Aided Design 40(9):951–962

Chern IL, Shu YC (2007) A coupling interface method for elliptic interface problems. Journal of Computational Physics 225(2):2138–2174

Choi YJ, Hulsen MA, Meijer HE (2012) Simulation of the flow of a viscoelastic fluid around a stationary cylinder using an extended finite element method. Computers & Fluids 57:183–194

Coffin P, Maute K (2015a) A level-set method for steady-state and transient natural convection problems. Structural and Multidisciplinary Optimization pp 1–21

Coffin P, Maute K (2015b) Level set topology optimization of cooling and heating devices using a simplified convection model. Structural and Multidisciplinary Optimization

Coolidge J (1952) The unsatisfactory story of curvature. American Mathematical Monthly pp 375–379

van 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

Dolbow J, Harari I (2009) An efficient finite element method for embedded interface problems. Int J Numer Meth Engng 78:229–252

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

Erentok A, Sigmund O (2011) Topology optimization of sub-wavelength antennas. Antennas and Propagation, IEEE Transactions on 59(1):58–69

Eschenauer H, Olhoff N (2001) Topology optimization of continuum structures: a review. Applied Mechanics Reviews 54:331

Franca LP, Frey SL, Hughes TJ (1992) Stabilized finite element methods: I. application to the advective-diffusive model. Computer Methods in Applied Mechanics and Engineering 95(2):253–276

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

Fuchs M, Jiny S, Peleg N (2005) The srv constraint for 0/1 topological design. Structural and Multidisciplinary Optimization 30(4):320–326

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

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

Gill PE, Murray W, Saunders MA (2002) Snopt: An sqp algorithm for large-scale constrained optimization. SIAM Journal on Optimization 12(4):979–1006

Golmon S, Maute K, Dunn ML (2012) Multiscale design optimization of lithium ion batteries using adjoint sensitivity analysis. International Journal for Numerical Methods in Engineering 92(5):475–494

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

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

Guo X, Zhang W, Zhong W (2014) Explicit feature control in structural topology optimization via level set method. Computer Methods in Applied Mechanics and Engineering 272:354–378

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

Hashin Z, Shtrikman S (1962) On some variational principles in anisotropic and nonhomogeneous elasticity. Journal of the Mechanics and Physics of Solids 10(4):335 – 342

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

Jenkins N, Maute K (2015) Level set topology optimization of stationary fluid-structure interaction problems. Structural and Multidisciplinary Optimization 52(1):179–195

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

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

Kreissl S, Maute K (2011) Topology optimization for unsteady flow. International Journal for Numerical Methods in Engineering 87:1229–1253

Kreissl S, Maute K (2012) Levelset based fluid topology optimization using the extended finite element method. Structural and Multidisciplinary Optimization 46(3):311–326

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

Labbé T, Glineur F, Dehez B (2009) Topology optimization method applied to the design of electromagnetic devices: focus on convexity issues. In: 8th International Symposium on Advanced Electromechanical Motion Systems and Electric Drives Joint Symposium, ELECTROMOTION 2009

Lang C, Makhija D, Doostan A, Maute K (2014) A simple and efficient preconditioning scheme for heaviside enriched xfem. Computational Mechanics 54(5):1357–1374

Łaniewski-Wołłk Ł, Rokicki J (2015) Adjoint lattice boltzmann for topology optimization on multi-gpu architecture. arXiv preprint arXiv:150104741

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

Legay A, Chessa J, Belytschko T (2006) An eulerian–lagrangian method for fluid–structure interaction based on level sets. Computer Methods in Applied Mechanics and Engineering 195(17):2070–2087

Li L, Wang M, Wei P (2012) Xfem schemes for level set based structural optimization. Frontiers of Mechanical Engineering 7(4):335–356

Liu Y, Deng Y, Zhang P, Liu Z, Wu Y (2013) Experimental investigation of passive micromixers conceptual design using the layout optimization method. Journal of Micromechanics and Microengineering 23(7):075,002

Luo J, Luo Z, Chen L, Tong L, Wang M (2008a) A semi-implicit level set method for structural shape and topology optimization. Journal of Computational Physics 227(11):5561–5581

Luo J, Luo Z, Chen S, Tong L, Wang M (2008b) A new level set method for systematic design of hinge-free compliant mechanisms. Computer Methods in Applied Mechanics and Engineering 198(2):318–331

Makhija D, Maute K (2014a) Level set topology optimization of scalar transport problems. Structural and Multidisciplinary Optimization

Makhija D, Maute K (2014b) Numerical instabilities in level set topology optimization with the extended finite element method. Structural and Multidisciplinary Optimization 49(2):185–197

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

Maute K (2014) Topology optimization of diffusive transport problems. In: Rozvany G, Lewiński T (eds) Topology Optimization in Structural and Continuum Mechanics, CISM International Centre for Mechanical Sciences, vol 549, Springer Vienna, pp 389–407

Maute K, Ramm E (1995) Adaptive topology optimization. Structural and Multidisciplinary Optimization 10(2):100–112

Maute K, Ramm E (1997) Adaptive topology optimization of shell structures. AIAA Journal 35(11):1767–1773

Maute K, Schwarz S, Ramm E (1998) Adaptive topology optimization of elastoplastic structures. Structural and Multidisciplinary Optimization 15(2):81–91

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 ‚Äì An Additive Manufacturing Conference, August 12-14, 2013, Austin, TX

Menk A, Bordas S (2011) A robust preconditioning technique for the extended finite element method. International Journal for Numerical Methods in Engineering 85(13):1609–1632

Michell AGM (1904) The limits of economy of material in frame-structures. Philos Magazine S6 8(47):589–597

Nemhauser GL, Wolsey LA (1988) Integer and combinatorial optimization, vol 18. Wiley New York

Newton I, Colson J (1736) The Method of Fluxions and Infinite Series; with Its Application to the Geometry of Curve-lines... Translated from the Author's Latin Original Not Yet Made Publick. To which is Subjoin'd a Perpetual Comment Upon the Whole Work... by J. Colson

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

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

Nitsche J (1975) Uber ein Variationsprinzip zur Loesung von Dirichlet-Problemen bei Verwendung von Teilraeumen, die keinen Randbedingungen unterworfen sind. Abhandlungen aus dem Mathematischen Seminar der Universitaet Hamburg 36:9–15

Nørgaard S, Sigmund O, Lazarov B (2016) Topology optimization of unsteady flow problems using the lattice boltzmann method. Journal of Computational Physics 307:291–307

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

Osher S, Paragios N (2003) Geometric level set methods in imaging, vision, and graphics. Springer-Verlag New York Inc

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

Osher SJ, Sethian JA (1988) Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations. Journal of Computational Physics 79:12—49

Park J, Sutradhar A (2015) A multi-resolution method for 3d multi-material topology optimization. Computer Methods in Applied Mechanics and Engineering 285:571–586

Parvizian J, Düster A, Rank E (2012) Topology optimization using the finite cell method. Optimization and Engineering 13(1):57–78

Pingen G, Waidmann M, Evgrafov A, Maute K (2010) A parametric level-set approach for topology optimization of flow domains. Structural and Multidisciplinary Optimization 41(1):117–131

Poulsen TA (2003) A new scheme for imposing a minimum length scale in topology optimization. International Journal for Numerical Methods in Engineering 57(6):741–760

Rozvany G (2009) A critical review of established methods of structural topology optimization. Structural and Multidisciplinary Optimization 37(3):217–237

Rozvany G, Zhou M, Sigmund O (1994) Optimization of topology. Advances in design optimization Chapman & Hall, London pp 340–399

Sá L, Amigo R, Novotny A, Silva E (2016) Topological derivatives applied to fluid flow channel design optimization problems. Structural and Multidisciplinary Optimization pp 1–16

Saad Y (1994) Ilut: A dual threshold incomplete lu factorization. Numerical Linear Algebra with Applications 1(4):387–402

Saad Y, Schultz M (1986) GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM Journal on Scientific and Statistical Computing 7(3):856–869

Sauerland H, Fries TP (2013) The stable xfem for two-phase flows. Computers & Fluids 87:41–49

Schäfer M, Turek S, Durst F, Krause E, Rannacher R (1996) Benchmark computations of laminar flow around a cylinder. Springer

Schevenels M, Lazarov BS, Sigmund O (2011) Robust topology optimization accounting for spatially varying manufacturing errors. Computer Methods in Applied Mechanics and Engineering 200(49):3613–3627

Schleupen A, Maute K, Ramm E (2000) Adaptive fe-procedures in shape optimization. Structural and Multidisciplinary Optimization 19:282–302

Schmitt O, Steinmann P (2015) On curvature control in node–based shape optimization. PAMM 15(1):579–580

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

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 URL http://dx.doi.org/10.1002/nme.4789

Sethian J, Wiegmann A (2000) Structural boundary design via level set and immersed interface methods. Journal of Computational Physics 163(2):489–528

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

Sigmund O (2001) A 99 line topology optimization code written in matlab. Structural and Multidisciplinary Optimization 21(2):120–127

Sigmund O (2007) Morphology-based black and white filters for topology optimization. Structural and Multidisciplinary Optimization 33(4-5):401–424

Sigmund O (2009) Manufacturing tolerant topology optimization. Acta Mechanica Sinica/Lixue Xuebao 25(2):227–239

Sigmund O (2011) On the usefulness of non-gradient approaches in topology optimization. Structural and Multidisciplinary Optimization 43(5):589–596

Sigmund O, Jensen JS (2003) Systematic design of phononic band-gap materials and structures by topology optimization. Philosophical Transactions of the Royal Society A 361(1806):1001–1019

Sigmund O, Maute K (2013) Topology optimization approaches: A comparative review. Structural and Multidisciplinary Optimization 48(6):1031–1055

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

Squillacote AH, Ahrens J (2007) The paraview guide, vol 366. Kitware

Stolpe M, Svanberg K (2001) An alternative interpolation scheme for minimum compliance topology optimization. Structural and Multidisciplinary Optimization 22(2):116–124

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

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

Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J on Optimization 12(2):555–573

Svanberg K (2007) Mma and gcmma-two methods for nonlinear optimization. Tech. rep., Technical report

Tcherniak D, Sigmund O (2001) A web-based topology optimization program. Structural and Multidisciplinary Optimization 22(3):179–187

Terada K, Asai M, Yamagishi M (2003) Finite cover method for linear and non-linear analyses of heterogeneous solids. International Journal for Numerical Methods in Engineering 58(9):1321–1346

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

Tran TQN, Lee HP, Lim SP (2011) Modelling porous structures by penalty approach in the extended finite element method. Computer Methods in Biomechanics and Biomedical Engineering ahead-of-p:1–11

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

Villanueva CH, Maute K (2014) Density and level set-xfem schemes for topology optimization of 3-d structures. arXiv preprint arXiv:14016475

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

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

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

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

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

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

Xia Q, Shi T (2015) Constraints of distance from boundary to skeleton: For the control of length scale in level set based structural topology optimization. Computer Methods in Applied Mechanics and Engineering 295:525–542

Xu S, Cai Y, Cheng G (2010) Volume preserving nonlinear density filter based on heaviside functions. Structural and Multidisciplinary Optimization 41(4):495–505

Yaji K, Yamada T, Kubo S, Izui K, Nishiwaki S (2015) A topology optimization method for a coupled thermal–fluid problem using level set boundary expressions. International Journal of Heat and Mass Transfer 81:878–888

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):2876–2891

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

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

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

Yoon G, Jensen JS, Sigmund O (2007) Topology optimization of acoustic-structure interaction problems using a mixed finite element formulation. International Journal for Numerical Methods in Engineering 70(9):1049–1075

Zhang C, Zhang P, Cheng FF (2001) Fairing spline curves and surfaces by minimizing energy. Computer-Aided Design 33(13):913–923

Zhang W, Zhong W, Guo X (2014) An explicit length scale control approach in simp-based topology optimization. Computer Methods in Applied Mechanics and Engineering 282:71 – 86

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

Zhou M, Lazarov BS, Sigmund O (2014) Topology optimization for optical projection lithography with manufacturing uncertainties. Appl Opt 53(12):2720–2729

Zhou M, Lazarov BS, Wang F, Sigmund O (2015) Minimum length scale in topology optimization by geometric constraints. Computer Methods in Applied Mechanics and Engineering 293:266–282

Appendix A

Internal I1: A Complete Methodology for the Implementation of XFEM Inclusive Models

A Complete Methodology for the Implementation of XFEM Inclusive Models

Hernan Villanueva

Fall 2012

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.

1 Introduction

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 (Abdelaziz and Hamouine, 2008). The process is laborious and difficult (Zienkiewicz et al, 2005). 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.

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 (Abdelaziz and Hamouine, 2008). 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 (Hansbo and Hansbo, 2004).

1.3 eXtended Finite Element Method

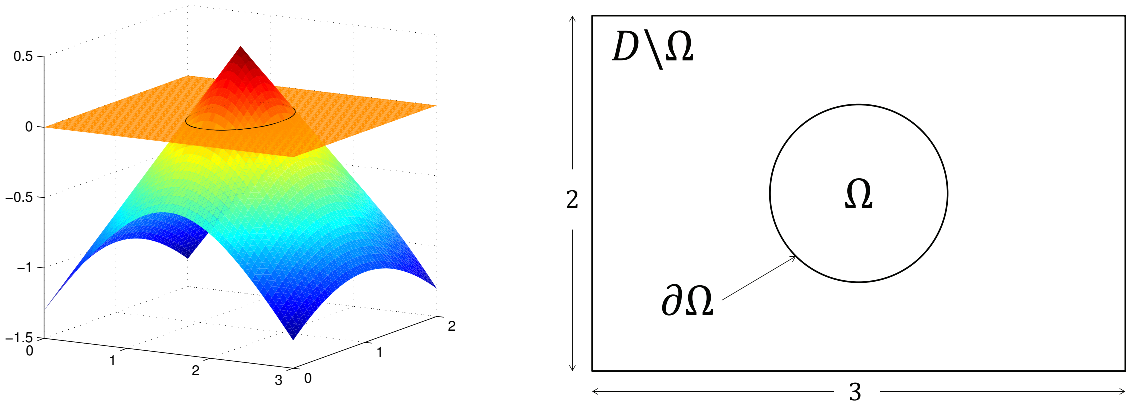
The XFEM arose as a numerical technique capable of providing local enrichment functions at the position of the material interface, avoiding the need to re-mesh the entire structure while finding solutions for the discontinuous functions (Fries and Belytschko, 2010). By doing this, XFEM appears to solve the shortcomings of the FEM by providing accurate solutions for complex problems in engineering that would be impossible to solve otherwise (Abdelaziz and Hamouine, 2008).

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 1).

$$
\begin{aligned}
&\phi\left(\mathbf{x}\right) > 0 \quad \forall\mathbf{x} \in \Omega\backslash\partial\Omega \text{ (inside the region)} \\
&\phi\left(\mathbf{x}\right) = 0 \quad \forall\mathbf{x} \in \partial\Omega \text{ (on the boundary)} \\
&\phi\left(\mathbf{x}\right) < 0 \quad \forall\mathbf{x} \in D\backslash\Omega \text{ (outside the region)}
\end{aligned}
\tag{1}
$$

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.



(a)

(b)

Figure 1: The zero level set isolevel of the level set function, $\partial\Omega = \Gamma_{\phi=0}$, in 1a divides the fixed mesh grid into different phase regions in 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 (Stolarska et al, 2001).

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 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) (Lee and Schachter, 1980). 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

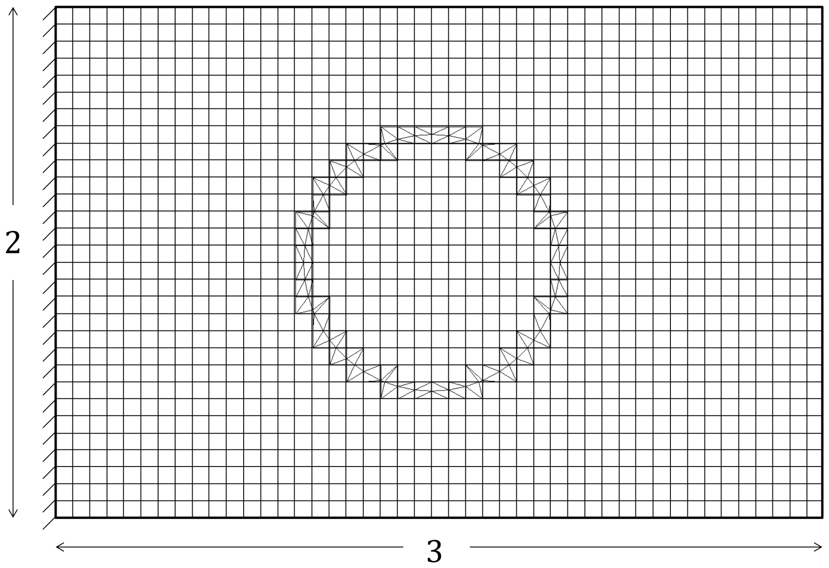


Figure 2: The Delaunay formulation triangulates QUAD4 finite elements cut by the level set zero isolevel from Figure 1b into TRI3 elements.

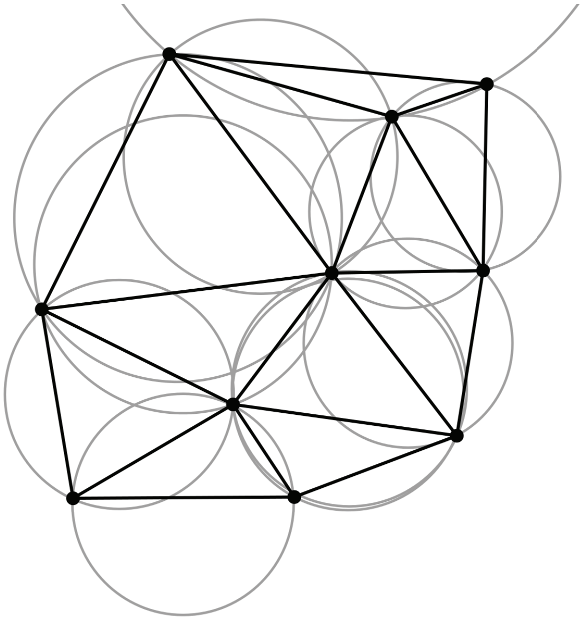


Figure 3: Delaunay circumcircles - A set of points can be uniquely triangulated in a way that the points form circumcircles.

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 4 shows the different cases for 2D, while Figure 5 shows the triangulation of a 3D element with four different discontinuities.

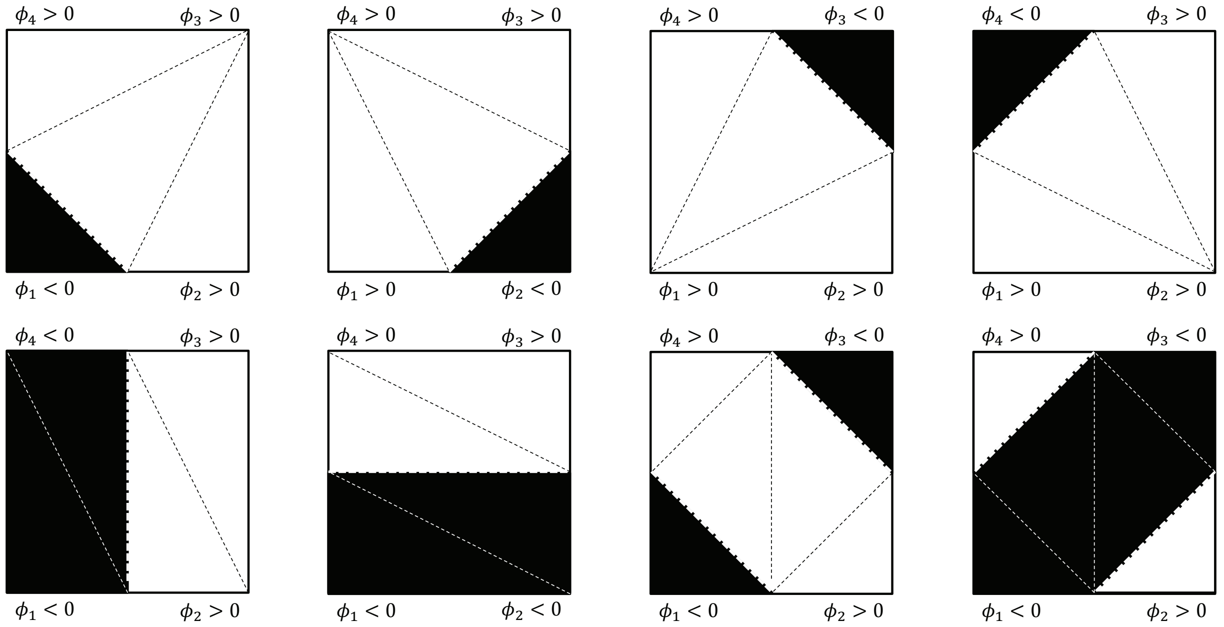


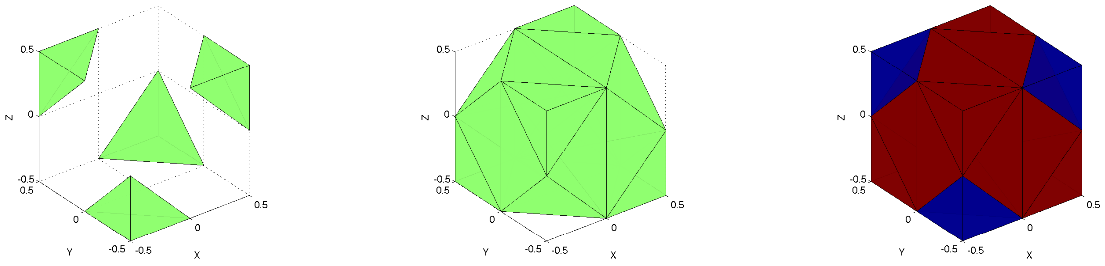
Figure 4: There are only 8 different triangulation configurations for an inclusion-based XFEM model.

2 Implementation

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.



(a) Phase “A”. (b) Phase “B”. (c) Both phase regions.

Figure 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.

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 intersection 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.

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.

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

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 6 leads to the intersection pattern shown in Figure 7. The mesh in Figure 8 shows the indices of the nodes and the cells.

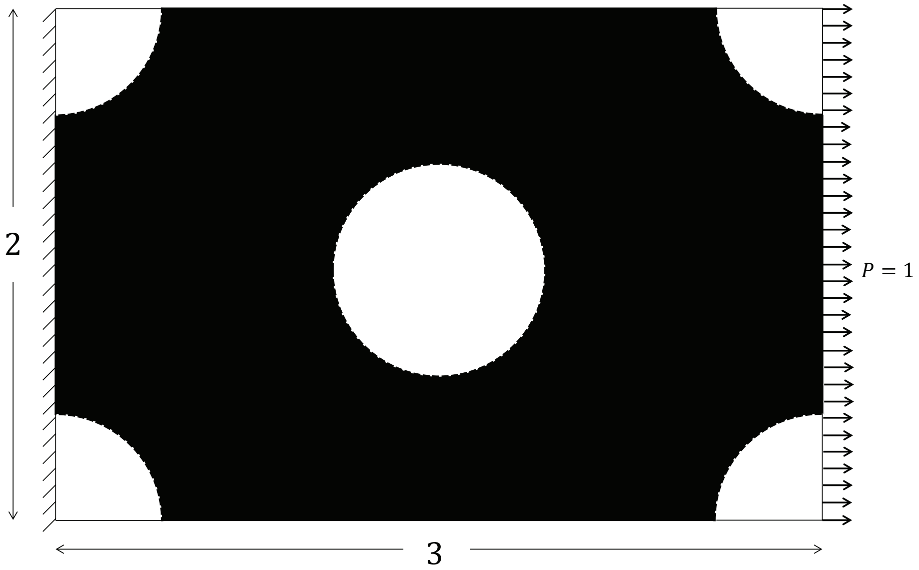


Figure 6: Structural problem setup. The domain contains multiple level set inclusions.

2.4.1 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 1):

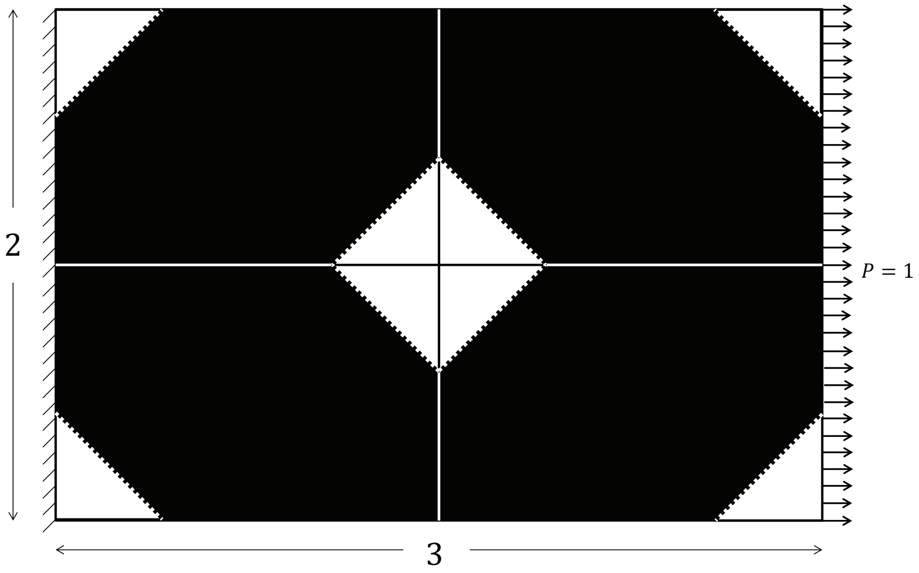


Figure 7: 4-element 2D mesh. Black areas: material phase 1, negative level-set value at the nodes; white areas: material phase 2, positive level-set value.

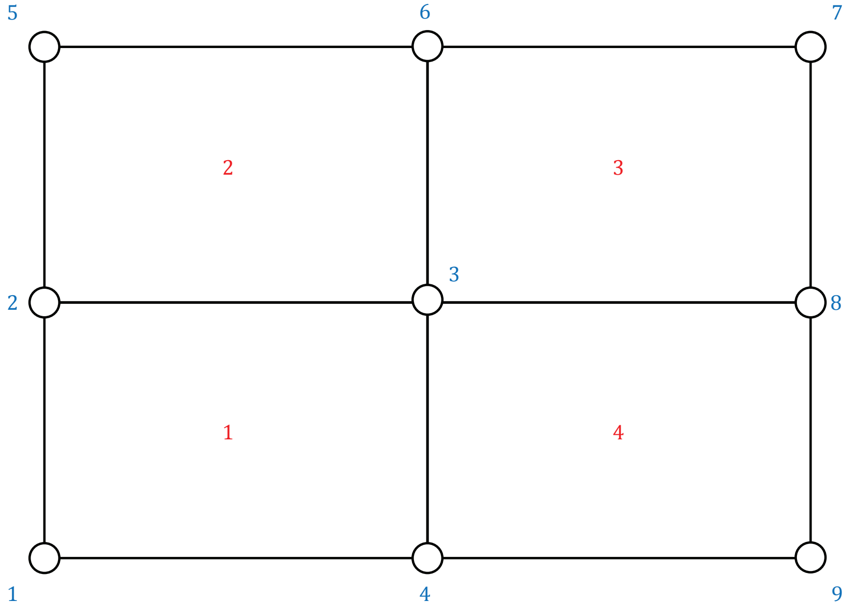


Figure 8: 4-element 2D mesh. Red numbers represent the global element identifiers, blue numbers represent the global node identifiers.

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 1: Point ID to cell IDs connectivity table

2.4.2 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 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 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 9).

2.4.3 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

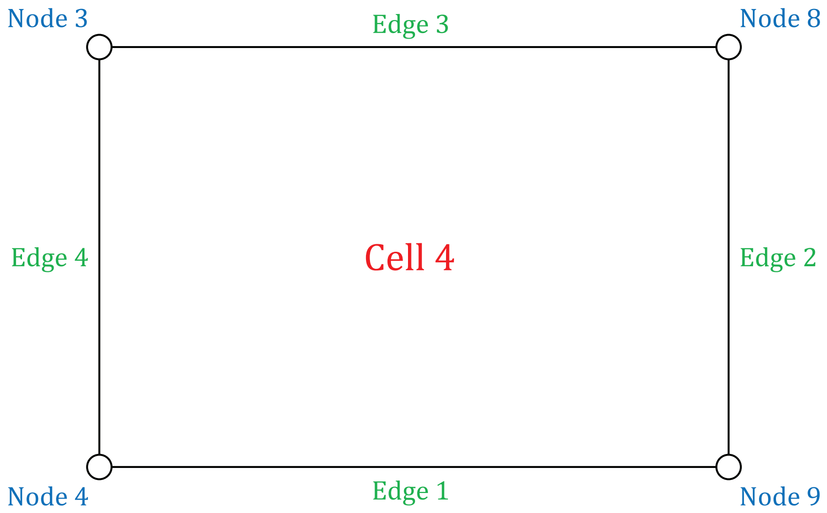


Figure 9: Edge representation in a QUAD4 element. Green edges represent the local edge index at the element.

be obtained by looping over the connected elements of a node, and obtaining the node list for each element.

2.4.4 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 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.

2.4.5 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

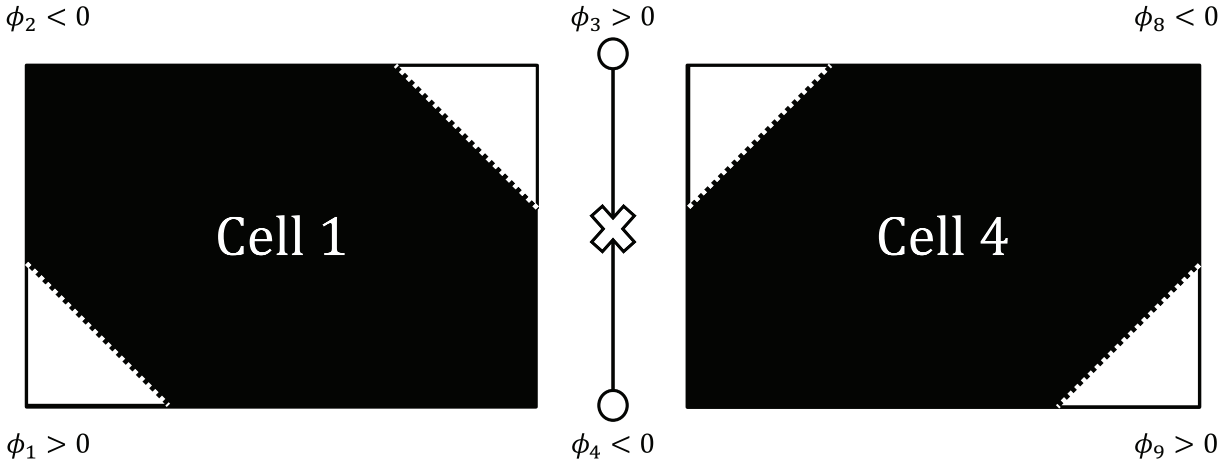


Figure 10: Mapping of the intersection points to the elements. An edge contains two nodes that have level set values of opposite sign. The location of the intersection point is labeled with an "X" in the figure. The information of the intersection point is sent to all the neighboring elements of the edge.

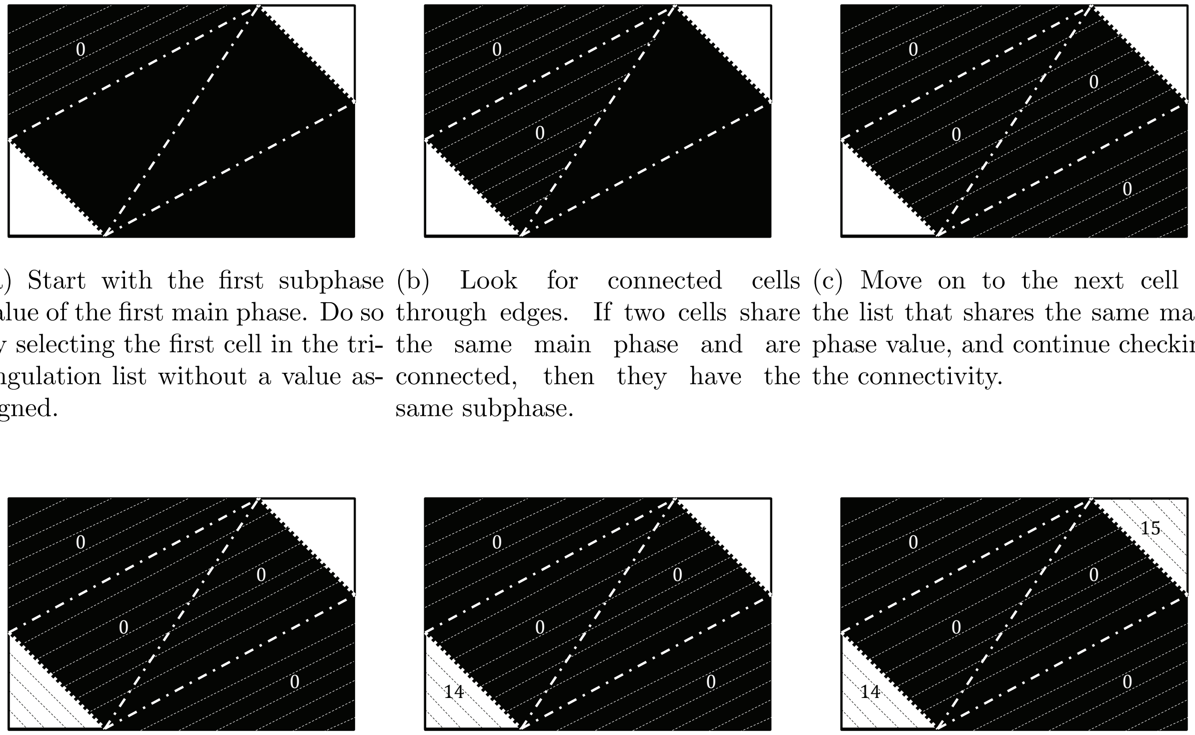
Step 3. The Delaunay triangulation requires only the coordinates of the element corner and edge intersection points. The Delaunay triangulation will return a list of triangles for 2D or tetrahedrons for 3D problems.

2.4.6 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 11) of the flood-fill algorithm used are:

- Build edge table for pseudo-elements of current XFEM element, analogue to Step 2.4.2.
- 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 value of the first main phase. Do so by selecting the first cell in the triangulation list without a value assigned.

(b) Look for connected cells through edges. If two cells share the same main phase and are connected, then they have the same subphase.

(c) Move on to the next cell in the list that shares the same main phase value, and continue checking the connectivity.

(d) Once the connectivity for the specific subphase of the main phase is achieved, check if additional cells have the same main phase but different subphase. If there are not any cells with the same main phase, move on the next main phase.

(e) Select the first subphase of the second main phase. Repeat steps 1 through 3.

(f) Once the connectivity for a subphase is computed, increase the subphase value within the main phase. Look for cells in the triangulation list that have not been processed yet, and check their connectivity.

Figure 11: Subphase computation algorithm. Refer to section 2.4.6 for a more detailed description.

For Figure 11, we would get the list in table 3:

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); the assignment is stored, using a one-to-one map by the XFEM element.

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

2.4.7 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.

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 4: Initial node-element table.

The initial node-element table (see Table 4) shows that the consistency condition is not 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 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 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 7. The rows correspond to the elements, the columns list the original sub-phases, and the entries are the enrichment levels used by

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

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

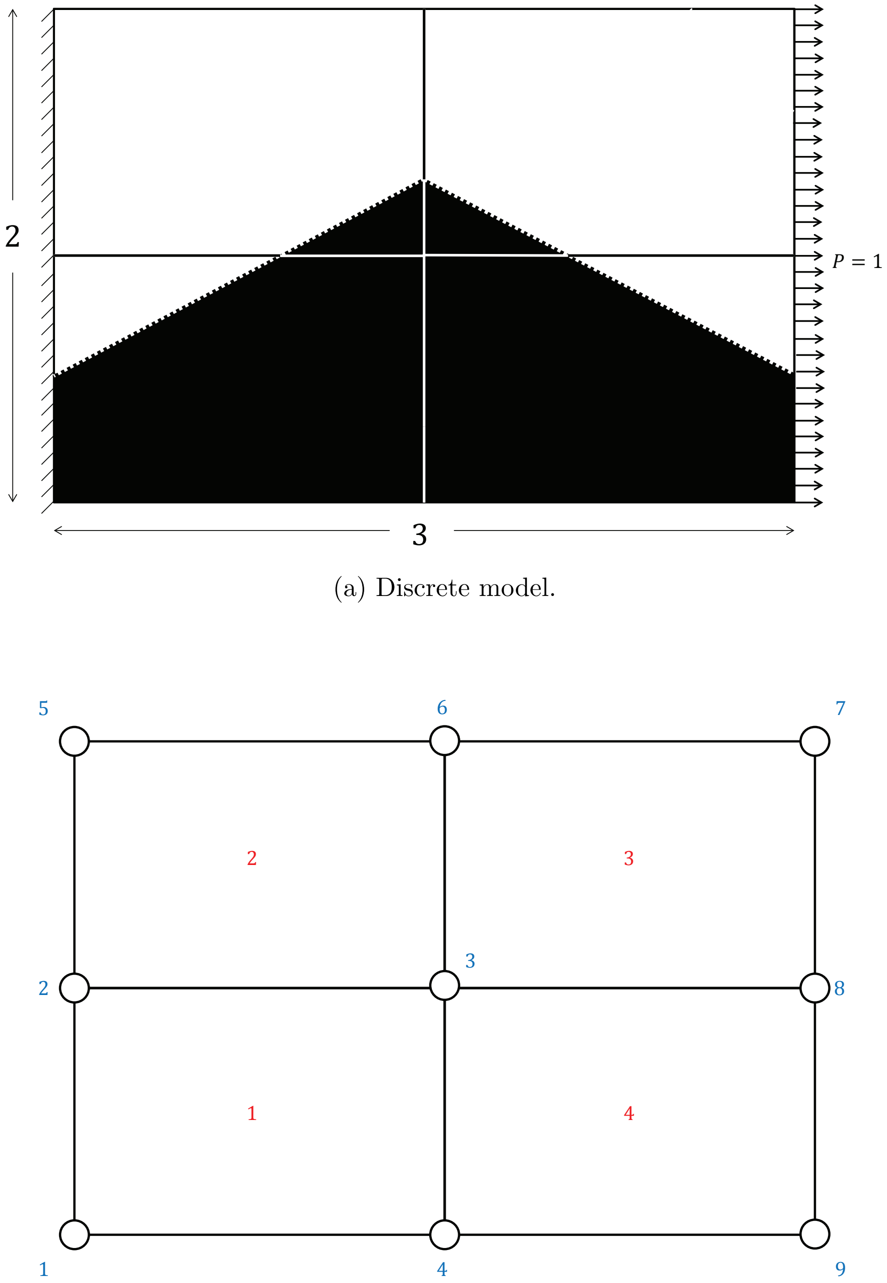
Table 7: Element to enrichment table for node ID 3. This table shows the initial enrichments node 3 received during the sub-phase algorithm and the enrichment levels after the enrichment algorithm.

5. 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.

2.4.8 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.

2.4.9 Implementation example 2



(a) Discrete model.

(b) Mesh information.

Figure 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 8: Initial node-element table for configuration 2.

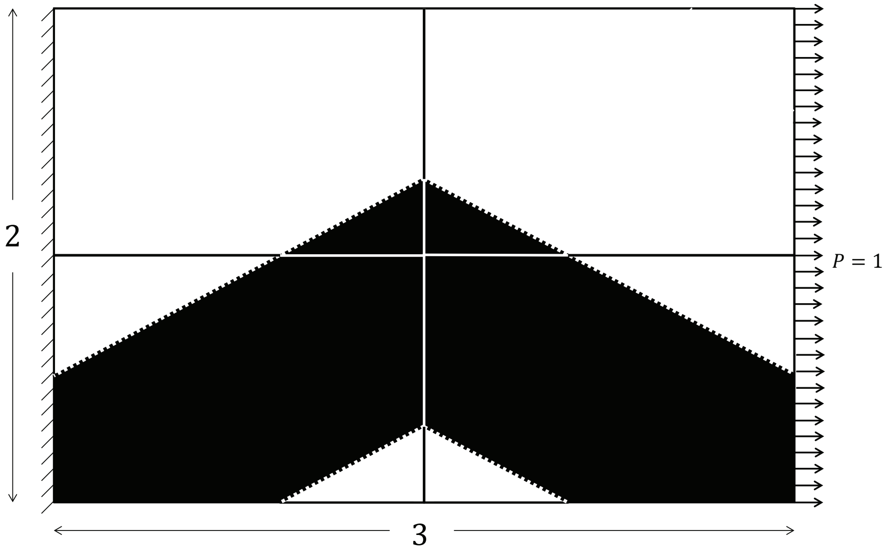
Nodes/Elements	1	2	3	4
1	0			
2 “C”	14	14		
3 “C”	0	0	0	0
4 “C”	0			0
5		14		
6 “C”		14	14	
7			14	
8 “C”			14	14
9				0

Table 9: Final node-element table for configuration 2.

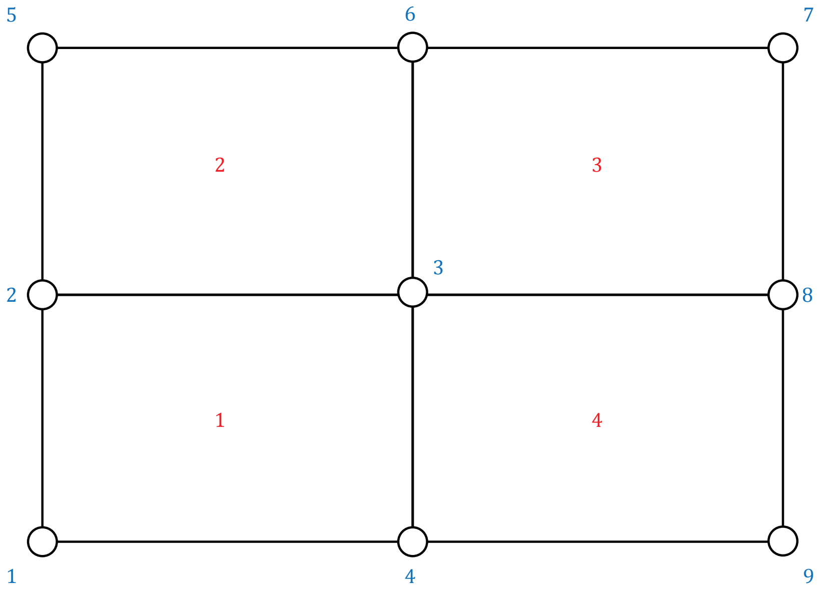
For Node 3: Element / Enrichment level	0	14
1	0	14
2	0	14
3	0	14
4	0	14

Table 10: Enrichment level map for configuration 2.

2.4.10 Implementation example 3



(a) Discrete model.



(b) Mesh information.

Figure 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 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 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 13: Enrichment level map for configuration 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 (2014) which resolves consistently the temperature fields in the presence of small features and does not suffer from artificially coupling disconnected phases.

$$u(\mathbf{x}) = \sum_{m=1}^{M} \left(H(-\phi) \sum_{i=1}^{n} \mathbf{N}_i \; u_{i,m}^{A} + H(\phi) \sum_{i=1}^{n} \mathbf{N}_i \; u_{i,m}^{B} \right) \tag{2}$$

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 $l = [A, B]$, ϕ 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:

$$H(z) = \begin{cases} 1 & z > 0 \\ 0 & z \leq 0 \end{cases} \tag{3}$$

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.

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 $\boldsymbol{T}$ 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 $\boldsymbol{T}$ 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:

$$\boldsymbol{T}^l_{i,m} = \left(\max_{e \in E_i} \frac{\int_{\mathcal{D}^e_l} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x}) \, d\mathbf{x}}{\int_{\mathcal{D}^e} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x}) \, d\mathbf{x}} \right)^{-1/2} \tag{4}$$

where $\boldsymbol{T}^l_{i,m}$ corresponds to the degree-of-freedom $\mathbf{u}^l_{i,m}$, 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 $\mathcal{D}^e_l$ 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 $\boldsymbol{T}^l_{i,m}$ 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 $\boldsymbol{T}$, the degrees of freedom associated with the diagonal entry $\boldsymbol{T}^l_{i,m}$ are constrained to zero if the following condition is satisfied:

$$\boldsymbol{T}^l_{i,m} \geq T_{tol} \tag{5}$$

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 Corroboration and results

3.1 Methodology

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

Equation 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.

$$\sqrt{\frac{\sum_{\text{element}}\sum_{\text{interface}}\int u^{+}-u^{-}\mathrm{d}\Gamma_{i}}{\sum_{\text{element}}\sum_{\text{interface}}\int \mathrm{d}\Gamma_{i}}} \tag{6}$$

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 7 compares the relative difference between the XFEM solution and the FEM solution.

$$\sqrt{\frac{\int u_{\text{XFEM}}-u_{\text{FEM}}\mathrm{d}\Omega}{\int u_{\text{FEM}}\mathrm{d}\Omega}} \tag{7}$$

u_{XFEM} represents the XFEM solution, while u_{FEM} represents the FEM solution.

XFEM was used to solve a thermal problem with the configuration of Figure 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

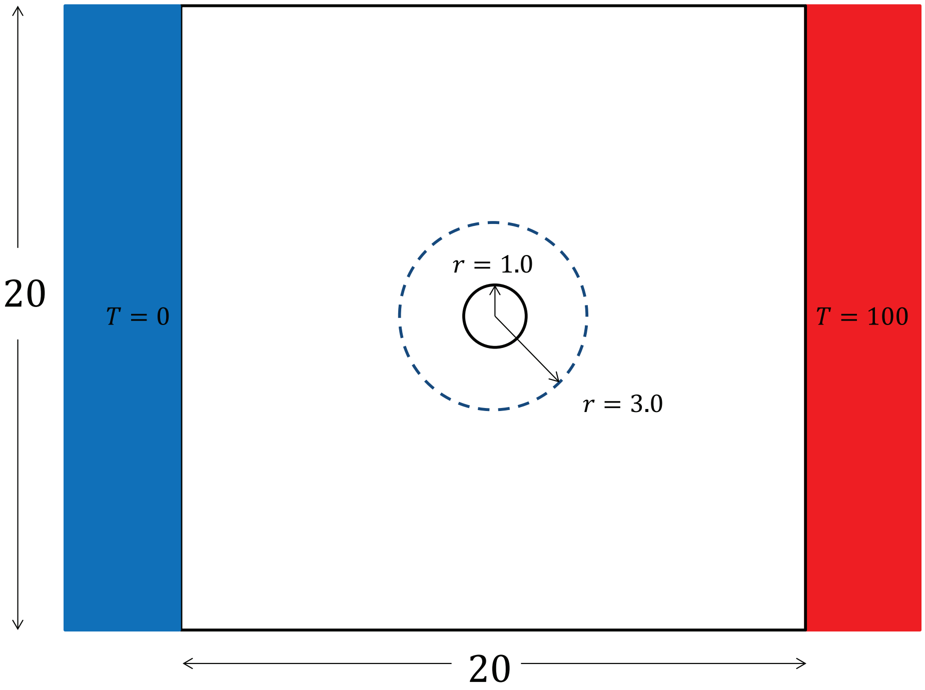


Figure 14: Diffusion problem setup.

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.

3.2 Tests

3.2.1 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 × 20, 30 × 30, 40 × 40, and 50 × 50.

Figure 15 shows that as the mesh is refined, the interface error converges to zero.

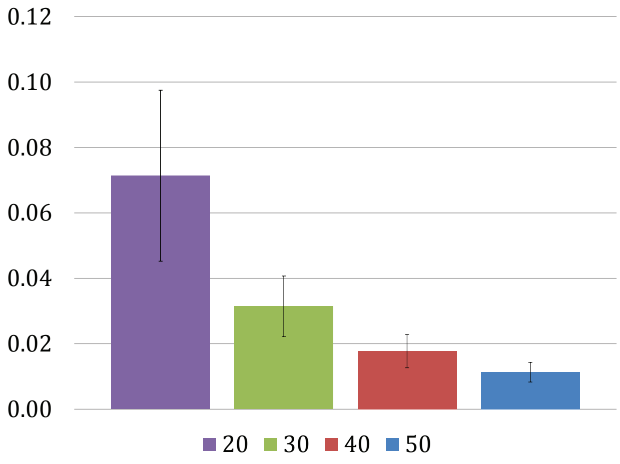


Figure 15: Mesh refinement sweep interface error.

Figure 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 × 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.

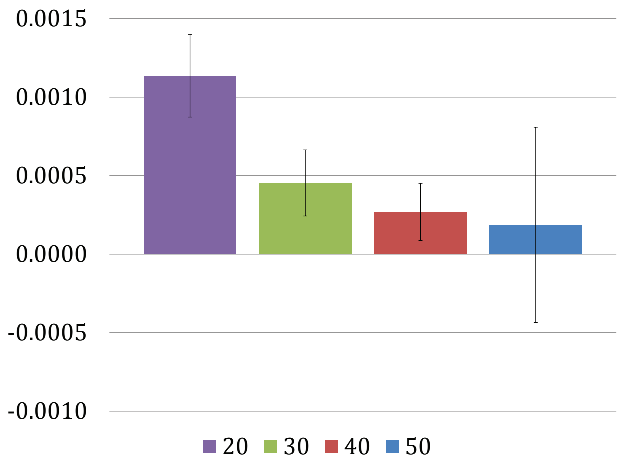


Figure 16: Mesh refinement sweep L2 error.

3.2.2 Conductivity ratio sweep

The conductivity ratio between the different materials was the variable in this test. The mesh size was 30 × 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 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.

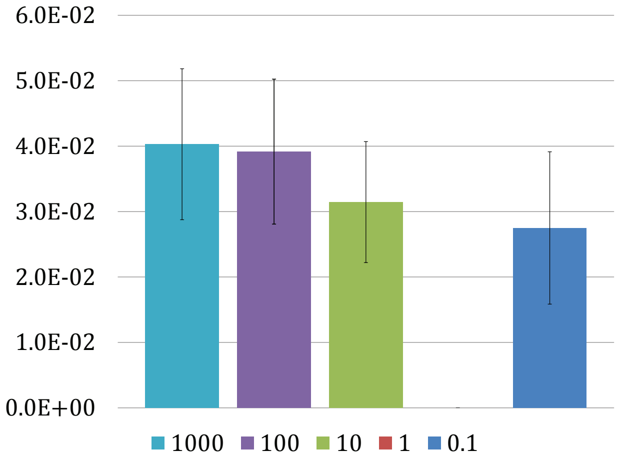


Figure 17: Conductivity refinement sweep interface error.

For the L2 computation, only FEM solutions with conductivity ratios of 10, 100, and 1000 were computed. Figure 18 shows that the difference in solutions is very small $O(10^{-4})$.

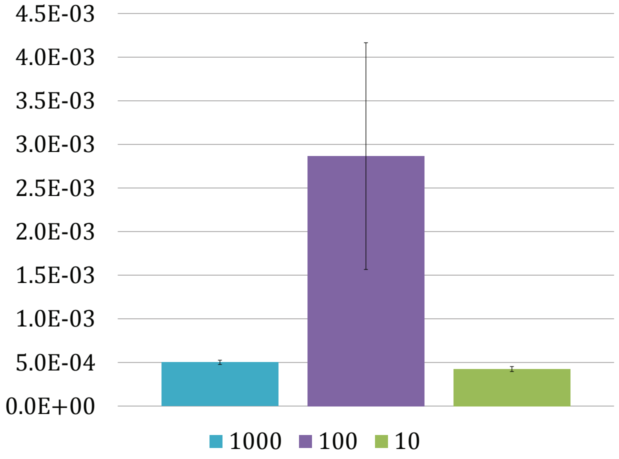


Figure 18: Conductivity refinement sweep L2 error.

3.2.3 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 × 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 19 shows that the Jacobian matrix has a condition number in the order of 10^{15} when no scaling is applied [1], while Figure 20 shows that the condition number decreases to the order of 10^4 when scaling is applied.

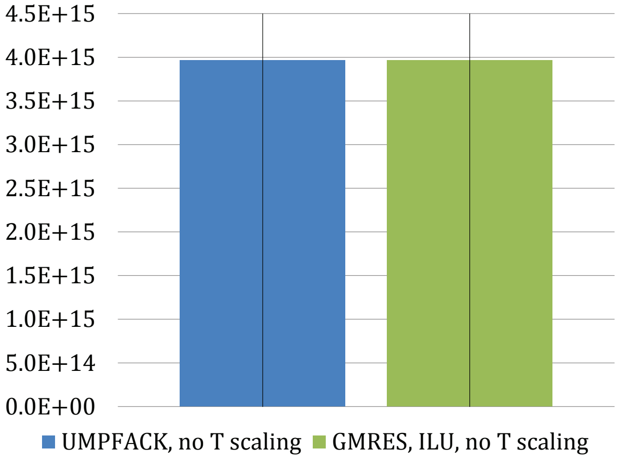


Figure 19: Condition number comparison - no pre-conditioner.

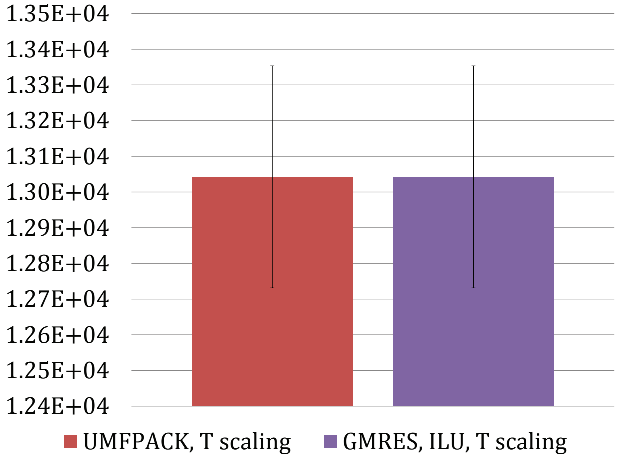


Figure 20: Condition number comparison - with pre-conditioner.

[1] 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.

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 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.

The 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.

Appendix

5 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.

5.A main.m

MATLAB code

```
function[] = main()
% Main program
% Modify ex, ey, and ez to change shape of element
% Change levs to change level-set configuration

% Global x, y, z coordinates of element.
% This will form a 3D cubic element cube.
ex = [-0.5 -0.5 -0.5 -0.5 +0.5 +0.5 +0.5 +0.5];
ey = [-0.5 -0.5 +0.5 +0.5 -0.5 -0.5 +0.5 +0.5];
ez = [+0.5 -0.5 -0.5 +0.5 +0.5 -0.5 -0.5 +0.5];

%% Extract a particular case (set k manually).

% Initial test for a particular levelset function.
% Randn function will return a m x n matrix of random positive and negative
% numbers.
hexsect = cell(127, 8);
k = round(1 + (127-1).*rand);
% levs = randn(1, 8);
% levs = [-1 -1 -1 -1 -1 -1 -1 1];
levs = [+1 -1 +1 -1 -1 +1 -1 +1];
[nsct, isct, xsct, ysct, zsct, levs] = xfem8isct(ex, ey, ez, levs);
[xtet, ytet, ztet, ctet, ptet, pnd, plist, Tetp1, Tetp2, tet1G, tet1L, tet2G, tet2L, size_phase1, size_phase2, volume1,
     volume2, total_volume1, total_volume2] = xfem8tet(isct, xsct, ysct, zsct, ex, ey, ez, levs);
hexsect{k, 2} = xtet;
hexsect{k, 3} = ytet;
hexsect{k, 4} = ztet;
hexsect{k, 5} = ctet;
hexsect{k, 6} = ptet;
hexsect{k, 7} = pnd;
hexsect{k, 8} = plist;
[max(plist) min(plist)];

figure(1)
tetramesh(Tetp1, [xtet(1, :)’, ytet(1, :)’, ztet(1, :)’], -ones(size(Tetp1, 1), 1));
xlabel(’X’)
ylabel(’Y’)
zlabel(’Z’)
grid
axis equal

figure(2)
tetramesh(Tetp2, [xtet(1, :)’, ytet(1, :)’, ztet(1, :)’], ones(size(Tetp2, 1), 1));
xlabel(’X’)
ylabel(’Y’)
zlabel(’Z’)
grid
axis equal

figure(3)
tetramesh(ctet, [xtet(1, :)’, ytet(1, :)’, ztet(1, :)’], ptet);
xlabel(’X’)
ylabel(’Y’)
zlabel(’Z’)
axis equal

figure(4)
plot3(xtet(1, :), ytet(1, :), ztet(1, :), ’X’)
xlabel(’X’)
ylabel(’Y’)
zlabel(’Z’)
axis equal
```

```

end
```

5.B xfem8isct.m

MATLAB code

```
function [nsct, isct, xsct, ysct, zsct, levs] = xfem8isct(ex, ey, ez, levs)

% Intersection of hex8 based on level-set values.
%
% Input:   ex    : global x-coordinates of nodes of 3D element
%          ey    : global y-coordinates of nodes of 3D element
%          ez    : global z-coordinates of nodes of 3D element
%          levs  : level set values randomly generated
%
% Output:  nsct  : number of intersected edges
%          isct : vector flags of intersected edges
%          xsct : x-coordinates of intersections in global and local coordinates
%          ysct : y-coordinates of intersections in global and local coordinates
%          zsct : z-coordinates of intersections in global and local coordinates

% Map of node connections across edges in a 3D element.
% Edgmap is a 12 x 2 matrix.
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];

% Values of nodes in master element (local coordinates).
xp = [-1 -1 -1 -1 +1 +1  +1 +1];
yp = [-1 -1 +1 +1 -1 -1  +1 +1];
zp = [+1 -1 -1 +1 +1 -1  -1 +1];

% Set initial value of intersected edges to zero.
% Create a zero 12 x 1 matrix to flag edges with intersected edges.
% Create a zero 12 x 2 matrix to record x, y, z coordinates in global and
% local coordinates.
nsct = 0;
isct = zeros(12, 1);
xsct = zeros(12, 2);
ysct = zeros(12, 2);
zsct = zeros(12, 2);

for i = 1:12
    % ic1 and ic2 return the values of the first and second columns of
    % edgmap, respectively, for a determined row. ic1 and ic2 represent two
    % nodes connected by a cube edge.
    ic1 = edgmap(i, 1);
    ic2 = edgmap(i, 2);
    % The level set values at those nodes are multiplied to determined
    % intersection.
    if levs(ic1)*levs(ic2) < 0      % If so, then there is an intersection.
        nsct = nsct+1;              % Increase number of intersected edges by one.
        isct(i) = 1;                % Flag edge as having an intersection in isct matrix.
        sctr = -levs(ic1)/(levs(ic2)-levs(ic1));    % Dimensionless location (ratio) of intersection.
        % Insection in glb coordinates
        xsct(i, 1) = ex(ic1)+sctr*(ex(ic2)-ex(ic1));
        ysct(i, 1) = ey(ic1)+sctr*(ey(ic2)-ey(ic1));
        zsct(i, 1) = ez(ic1)+sctr*(ez(ic2)-ez(ic1));
        % Insection in local coordinates
        xsct(i, 2) = xp(ic1)+sctr*(xp(ic2)-xp(ic1));
        ysct(i, 2) = yp(ic1)+sctr*(yp(ic2)-yp(ic1));
        zsct(i, 2) = zp(ic1)+sctr*(zp(ic2)-zp(ic1));
    end
end

display(’Number␣of␣intersected␣edges:␣’);
disp(nsct);
display(’Edges␣with␣intersections:␣’);
disp(isct’);
display(’x-coordinates␣of␣intersections␣in␣global␣coordinates:␣’);
disp(xsct(:, 1));
display(’x-coordinates␣of␣intersections␣in␣local␣coordinates:␣’);
disp(xsct(:, 2));
display(’y-coordinates␣of␣intersections␣in␣global␣coordinates:␣’);
disp(ysct(:, 1));
display(’y-coordinates␣of␣intersections␣in␣local␣coordinates:␣’);
disp(ysct(:, 2));
display(’z-coordinates␣of␣intersections␣in␣global␣coordinates:␣’);
disp(zsct(:, 1));
display(’z-coordinates␣of␣intersections␣in␣local␣coordinates:␣’);
disp(zsct(:, 2));

end
```

5.C xfem8tet.m

MATLAB code

```
function [xtet, ytet, ztet, ctet, ptet, pnd, plist, Tetp1, Tetp2, tet1G, tet1L, tet2G, tet2L, size_phase1, size_phase2,
     volume1, volume2, total_volume1, total_volume2] = xfem8tet(isct, xsct, ysct, zsct, ex, ey, ez, levs)

% Intersection of hex8 based on level-set values.
%
% Input:  nsct   :  number of insected edges
%         isect  :  vector flags intersected edges
%         xsect  :  x-coordinates of insections in global and local coordinates
%         ysect  :  y-coordinates of insections in global and local coordinates
%         zsect  :  z-coordinates of insections in global and local coordinates
%         ex     :  global x-coordinates of nodes
%         ey     :  global y-coordinates of nodes
%         ez     :  global z-coordinates of nodes
%         levs   :  levs
%
% Output: xtet   :  x-coordinates of nodes of triangulated element
%         ytet   :  y-coordinates of nodes of triangulated element
%          ztet   :  z-coordinates of nodes of triangulated element
%          ctet   :  connectivity of tets in triangulated element
%          ptet   :  phase of each tetrahedron
%          pnd    :  nodal levelset values (actual values), # of nodes
%          plist  :  phase of each node (in terms of -1, 1 and 0)
%          Tetp1  :  nodes of tetrahedrons for main phase 1
%          Tetp2  :  nodes of tetrahedrons for main phase 2
%          tet1G  :  global coordinates of tetrahedron for phase 1
%          tet1L  :  local coordinates of tetrahedron for phase 1
%          tet2G  :  global coordinates of tetrahedron for phase 1
%          tet2L  :  local coordinates of tetrahedron for phase 2
%          size_phase1 : number of tetrahedrons in phase 1
%          size_phase2 : number of tetrahedrons in phase 2
%          volume1     : volume of a tetrahedron in phase 1
%          volume2     : volume of a tetrahedron in phase 2
%          total_volume1 : total volume of all tetrahedrons in phase 1
%          total_volume2 : total volume of all tetrahedrons in phase 2

%% Identify coordinates and phases to triangulate

% Nodes in local coordinates
xp = [-1 -1 -1 -1 +1 +1  +1 +1];
yp = [-1 -1 +1 +1 -1 -1  +1 +1];
zp = [+1 -1 -1 +1 +1 -1  -1 +1];

% Find intersected edges. Order of edges depends on configuration of edgmap
% isct flagged intersected edges with a 1
itr = find(isct>0);

% Sort nodes by main phase. Determine which nodes have positive or negative
% level-set values.
ip1 = find(levs<0);
ip2 = find(levs>0);

% Id-numbers of node at edge intersections, i.e. create pseudonodes
ips = 9:8+length(itr);

% Create phase vector for triangulated element. Vector contains level-set
% values at the 8 original nodes plus zero values for the new pseudonodes
pnd = [levs zeros(1, length(itr))];

% Create nodes for main phase 1
% Coordinates where nodes are negative plus coordinates of intersections in
% global coordinates
exp1 = [ex(ip1) xsct(itr, 1)'];
eyp1 = [ey(ip1) ysct(itr, 1)'];
ezp1 = [ez(ip1) zsct(itr, 1)'];

% Nodes and pseudonodes numbers of main phase 1
ipx1 = [ip1 ips];

% Create nodes for main phase 2
% Coordinates where nodes are positive plus coordinates of intersections in
% global coordinates
exp2 = [ex(ip2) xsct(itr, 1)'];
eyp2 = [ey(ip2) ysct(itr, 1)'];
```

```
ezp2 = [ez(ip2) zsct(itr, 1)’];

% Nodes and pseudonodes numbers of main phase 2
ipx2 = [ip2 ips];

% Combine triangulation points of main phase 1 and 2 in local and global
% coordinates
xtet = [ex xsct(itr, 1)’;xp xsct(itr, 2)’];
ytet = [ey ysct(itr, 1)’;yp ysct(itr, 2)’];
ztet = [ez zsct(itr, 1)’;zp zsct(itr, 2)’];

% Same as xtet, ytet, ztet, but only with global coordinates. Useful for
% triangulation below.
exp = [ex xsct(itr, 1)’];
eyp = [ey ysct(itr, 1)’];
ezp = [ez zsct(itr, 1)’];

%% Triangulate main phase 1 and 2 together

% If we triangulate main phase 1 and 2 separately, tetrahedron will
% superpose for some level-set combinations. We need to triangulate the
% entire element as a whole.
Tp = DelaunayTri(exp’, eyp’, ezp’);
Tetp = Tp.Triangulation(:, :);

%% Separate triangulation into main phases 1 and 2

Tetp1 = zeros(1, 4);
Tetp2 = zeros(1, 4);
index1 = 1;
index2 = 1;

% If tetrahedron contains a negative node, it is phase 1. Phase 2,
% otherwise.
for i = 1:size(Tetp, 1)
    if pnd(Tetp(i, 1)) < 0||pnd(Tetp(i, 2)) < 0||pnd(Tetp(i, 3)) < 0||pnd(Tetp(i, 4)) < 0
        Tetp1(index1, :) = Tetp(i, :);
        index1 = index1 + 1;
    else
        Tetp2(index2, :) = Tetp(i, :);
        index2 = index2 +1;
    end
end

%% Display coordinates of phase 1 tetrahedrons

% Display coordinates of phase 1 tetrahedrons in global coordinates
% Display volume of each phase 1 tetrahedron
total_volume1 = 0;
display(’The␣global␣coordinates␣for␣the␣phase␣1␣tetrahedrons␣are:␣’)
for i = 1:size(Tetp1, 1)
    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)), 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)), ytet(1, Tetp1(i, 4)), ztet(1, Tetp1(i, 4))];
    disp(tet1G)
    % Calculate volume. Algorithm: For 4 points in tetrahedron P, Q, R, S,
    % volume = abs(det([Q-P;R-Q;S-R]))/6
    a = tet1G(2, :)-tet1G(1, :);
    b = tet1G(3, :)-tet1G(2, :);
    c = tet1G(4, :)-tet1G(3, :);
    volume1 = abs(det([a;b;c]))/6;
    total_volume1 = total_volume1 + volume1;
    display('The␣volume␣of␣phase␣1␣tetrahedron␣is:␣')
    disp(volume1)
end

% Display coordinates of phase 1 tetrahedrons in local coordinates
display('The␣local␣coordinates␣for␣the␣phase␣1␣tetrahedrons␣are:␣')
for i = 1:size(Tetp1, 1)
    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))];
    disp(tet1L)
end

% Display total volume of phase 1 tetrahedrons
display('The␣total␣volume␣of␣phase␣1␣tetrahedrons␣is:␣')
disp(total_volume1)

%% Display coordinates of phase 2 tetrahedrons

% Display coordinates of phase 2 tetrahedrons in global coordinates
% Display volume of each phase 2 tetrahedron
total_volume2 = 0;
display('The␣global␣coordinates␣for␣the␣phase␣2␣tetrahedrons␣are:␣')
for i = 1:size(Tetp2, 1)
    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))];
    disp(tet2G)
    % Calculate volume.
    a = tet2G(2, :)-tet2G(1, :);
    b = tet2G(3, :)-tet2G(2, :);
    c = tet2G(4, :)-tet2G(3, :);
    volume2 = abs(det([a;b;c]))/6;
    total_volume2 = total_volume2 + volume2;
    display('The␣volume␣of␣phase␣2␣tetrahedron␣is:␣')
    disp(volume2)
end

% Display coordinates of phase 2 tetrahedrons in local coordinates
display('The␣local␣coordinates␣for␣the␣phase␣2␣tetrahedrons␣are:␣')
for i = 1:size(Tetp2, 1)
    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))];
    disp(tet2L)
end

% Display total volume of phase 2 tetrahedrons
display('The␣total␣volume␣of␣phase␣2␣tetrahedrons␣is:␣')
disp(total_volume2)
```

```

%% Display number of tetrahedrons in each phase

size_phase1 = size(Tetp1(:, 1));
size_phase1 = size_phase1(1);
size_phase2 = size(Tetp2(:, 1));
size_phase2 = size_phase2(1);
display('The␣number␣of␣tetrahedrons␣in␣phase␣1␣is:␣')
disp(size_phase1);
display('The␣number␣of␣tetrahedrons␣in␣phase␣2␣is:␣')
disp(size_phase2);

%% Locate triangle interfaces - new algorithm

% Use function ismember to compare array vectors
% Display only result if three nodes repeat
display('The␣interfaces␣can␣be␣found␣on␣the␣triangles␣with␣nodes:␣')
for i = 1:size(Tetp1, 1)
    for j = 1:size(Tetp2, 1)
        r = ismember(Tetp1(i, :), Tetp2(j, :));
        Tetp1_tri = Tetp1(i, :);
        Tetp1_tri = Tetp1_tri(r);
        if size(Tetp1_tri, 2) == 3
            disp(Tetp1_tri)
        end
    end
end

%% Combine triangulation of nodes in ctet

% Size of ptet depends on number of tetrahedrons
% Phase 1 tetrahedrons have value of -1, phase 2 value of 1 in ptet

% Original:
% ctet=[Tetp1;Tetp2];
% ptet=[-ones(size(Tetp1, 1), 1);ones(size(Tetp2, 1), 1)];

% New method:
% Provide two options for triangulation
% Option 1: based on volume
% If one phase is significantly larger than the other, switch tetramesh
if total_volume2 <= (0.2*(total_volume1 + total_volume2))
    ctet=[Tetp2;Tetp1];
    ptet=[-ones(size(Tetp2, 1), 1);ones(size(Tetp1, 1), 1)];
else
    ctet=[Tetp1;Tetp2];
    ptet=[-ones(size(Tetp1, 1), 1);ones(size(Tetp2, 1), 1)];
end

% Option 2: based on user's choice
% option = input('Choose triangulation option 1 or 2: ');
% if option == 1
%     ctet=[Tetp2;Tetp1];
%     ptet=[-ones(size(Tetp2, 1), 1);ones(size(Tetp1, 1), 1)];
% elseif
%     ctet=[Tetp1;Tetp2];
%     ptet=[-ones(size(Tetp1, 1), 1);ones(size(Tetp2, 1), 1)];
% end

%% Check connectivity between phases. PART of ORIGINAL CODE

% Obsolete? plist produces the same value as pnd
% Check connectivity for main phase 1 and creat sub-phase information

% Matrix plisp1 formed of (nodes + pseudonodes) x 1 elements with value of
% -2. ipx1 represents location of phase 1 nodes + pseudonodes. At these
% locations, values are replaced by -1
plisp1 = -2*ones(size(xtet, 2), 1);
plisp1(ipx1) = -1;
domp1 = 0;

% While there are phase 1 nodes
while ismember(-1, plisp1) > 0
    domp1 = domp1 + 1;
    % Find where the negative nodes are in the nodes + pseudonodes vector
    % Value are progressively changed by 0, one at a time
    ppp = find(plisp1 == -1);
    plisp1(ppp(1)) = 0;
    % Locate where values were changed to zero and create new variable ppp
    % pid represents the node where value has transformed into zero
    while ismember(0, plisp1) > 0
        ppp = find(plisp1 == 0);
        pid = ppp(1);
        for it = 1:size(ctet, 1)
            % If the node at pid belongs to Tetp1 and is negative
            if ismember(pid, ctet(it, :)) > 0 && ptet(it) < 0
                plisp1(ctet(it, :)) = max(0, plisp1(ctet(it, :)));
            end
        end
        % At the location of phase 1 nodes + pseudonodes, the value will be
        % replaced to 1.
        plisp1(pid) = domp1;
    end
end

% Check connectivity for main phase 2 and creat sub-phase information
% Functions the same as previous routine, but for phase 2
plisp2 = -2*ones(size(xtet, 2), 1);
plisp2(ipx2) = -1;

domp2 = 0;
while ismember(-1, plisp2) > 0
    domp2 = domp2+1;
    ppp = find(plisp2 == -1);
    plisp2(ppp(1)) = 0;
    while ismember(0, plisp2) > 0
        ppp = find(plisp2 == 0);
        pid = ppp(1);
        for it = 1:size(ctet, 1)
            if ismember(pid, ctet(it, :)) > 0 && ptet(it) > 0
                plisp2(ctet(it, :)) = max(0, plisp2(ctet(it, :)));
            end
        end
        plisp2(pid) = domp2;
    end
end

% Build phase list including subphase information
% Replace all pseudonodes by 0
plisp1(ips) = 0;
plisp2(ips) = 0;

% Find location of the pseudonodes
idp1 = find(plisp1>0);
idp2 = find(plisp2>0);

% Create a zero (nodes + pseudonodes) x 1 matrix plist
plist = zeros(size(xtet, 2), 1);
% Create list that shows which original nodes belong to phase I and II
plist(idp1) = -plisp1(idp1);
plist(idp2) = plisp2(idp2);

end
```

5.D number_configurations.m

MATLAB code

```
function[] = number_configurations()
% This function computes all possible combinations of level-set function
% values at the corner nodes to obtain the number of 3D triangulation
% combinations possible.

% x, y, z coordinates of element.
% This will form a 3D cubic element cube.
ex = [0 1 1 0 0 1 1 0];
ey = [0 0 1 1 0 0 1 1];
ez = [0 0 0 0 1 1 1 1];

%% Sweep over all possible level-set configurations.

% Twelve edges. Maximum number of configurations:
maxc = 2^12;
icase = zeros(maxc, 1);
hexsect = cell(127, 8);

% Set initial configuration, k
% This loop calculates the total number of possible configurations of
% edge intersections
% Routine uses simple negative/positive level set values
k = 0;
for i1 = -1:2:1
    for i2 = -1:2:1
        for i3 = -1:2:1
```

```
            for i4 = -1:2:1
                for i5 = -1:2:1
                    for i6 = -1:2:1
                        for i7 = -1:2:1
                            for i8 = -1:2:2
                                levs = [i1 i2 i3 i4 i5 i6 i7 i8];
                                [nsct, isct, xsct, ysct, zsct, levs] = xfem8isct(ex, ey, ez, levs);
                                % isct is a 12 x 1 matrix representing each edge of the 3D cube element
                                % Since each edge can have one or zero intersections, the isct vector becomes a binary display
                                cbin = sprintf('%d%d%d%d%d%d%d%d%d%d%d%d', isct(1), isct(2), isct(3), isct(4), isct(5), isct(6), isct(7), isct(8), isct(9), isct(10), isct(11), isct(12));
                                % cbin is a binary number displayed as a string, and converted into the decimal cdec.
                                cdec = bin2dec(cbin);
                                % If icase is equal to zero, it means it is a new unique configuration and therefore,
                                % number of total configurations k should
                                % be increased by one
                                if icase(cdec+1) == 0 && nsct > 0
                                    k = k+1;
                                    [xtet, ytet, ztet, ctet, ptet, pnd, plist, Tetp1, Tetp2, tet1G, tet1L, tet2G, tet2L] = xfem8tet(isct, xsct, ysct, zsct, ex, ey, ez, levs);
                                    hexsect{k, 1} = cdec;
                                    hexsect{k, 2} = xtet;
                                    hexsect{k, 3} = ytet;
                                    hexsect{k, 4} = ztet;
                                    hexsect{k, 5} = ctet;
                                    hexsect{k, 6} = ptet;
                                    hexsect{k, 7} = pnd;
                                    hexsect{k, 8} = plist;
                                    [max(plist) min(plist)];
                                    icase(cdec+1)=1;
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end

fprintf('Number␣of␣configurations␣=␣%d\n', k);

end
```

References

Abdelaziz Y, Hamouine A (2008) A survey of the extended finite element. Computers & Structures 86(11):1141–1151

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

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

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

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

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

Zienkiewicz O, Taylor RL, Zhou S (2005) Finite Element Method - Its Basis and Fundamentals, 4th edn, Elsevier, chap 9, pp 206–256

Appendix B

Internal I2: Minimum Feature Size Measure with CGAL

The Computational Geometry Algorithms Library (CGAL) is a software library that aims to provide easy access to efficient and reliable algorithms in computational geometry. Similar to the Trilinos library, it is a collection of packages ranging from Delaunay triangulation to geodesic capabilities in graph theory.

In this document, we use the Surface Mesh Shortest Path module to compute geodesic paths on XFEM surface meshes. We provide a brief tutorial on how to represent surface meshes (with a femdoc-to-CGAL parser) and on how to find the shortest geodesic distance between two points.

The module has a dependency on the Boost Graph and GMP libraries; therefore, the program must be compiled with `libCGAL`, `libgmp`, and `libboost_system`.

The following class was adapted from the tutorials provided by the CGAL package. In this example, we model a 3D triangular surface mesh. The class is constructed by taking two matrices. The first matrix is a n-by-3 matrix, where n is the number of points in the surface mesh, and the number of columns correspond to the x, y, and z coordinates. The second matrix is a m-by-3, where m is the number of triangles in the surface mesh, and each column entry maps to a coordinate location.

C++ code

```cpp
// in file Parser_CGAL.hpp

#include <cstdlib>
```

```
#include <iostream>
#include <fstream>
#include <iterator>

#include <CGAL/Exact_predicates_inexact_constructions_kernel.h>
#include <CGAL/Random.h>
#include <CGAL/Polyhedron_3.h>
#include <CGAL/Polyhedron_items_with_id_3.h>
#include <CGAL/IO/Polyhedron_iostream.h>
#include <CGAL/Surface_mesh_shortest_path.h>
#include <CGAL/boost/graph/graph_traits_Polyhedron_3.h>
#include <CGAL/boost/graph/iterator.h>

typedef CGAL::Exact_predicates_inexact_constructions_kernel           Kernel;
typedef CGAL::Polyhedron_3<Kernel, CGAL::Polyhedron_items_with_id_3>
    Polyhedron_3;
typedef Kernel::Point_3                                               Point_3;
typedef Polyhedron_3::HalfedgeDS                                      HalfedgeDS;
typedef Polyhedron_3::Vertex_iterator
    Vertex_iterator;
typedef Polyhedron_3::Edge_iterator
    Edge_iterator;
typedef Polyhedron_3::Facet_iterator
    Facet_iterator;
typedef CGAL::Surface_mesh_shortest_path_traits<Kernel, Polyhedron_3> Traits;
typedef CGAL::Surface_mesh_shortest_path<Traits>
    Surface_mesh_shortest_path;
typedef boost::graph_traits<Polyhedron_3>
    Graph_traits;
typedef Graph_traits::vertex_iterator
    vertex_iterator;
typedef Graph_traits::face_iterator
    face_iterator;
```

```
typedef Polyhedron_3::Halfedge_around_facet_circulator
    Halfedge_facet_circulator;

template <class HDS>
class polyhedron_builder : public CGAL::Modifier_base<HDS>
{
public:
  std::vector<std::vector<double> > & coords;
  std::vector<std::vector<unsigned int> > & tris;

  polyhedron_builder(
      std::vector<std::vector<double> > & aCoords,
      std::vector<std::vector<unsigned int> > & aTris)
    : coords(aCoords)
    , tris(aTris)
  {
  }

  void operator()(HDS & hds)
  {
    // Postcondition: hds is a valid polyhedral surface.
    typedef typename HDS::Vertex   Vertex;
    typedef typename Vertex::Point Point;

    // Create a cgal incremental builder.
    CGAL::Polyhedron_incremental_builder_3<HDS> B(hds, true);
    B.begin_surface(coords.size(), tris.size());

    // Add the polyhedron vertices.
    for(unsigned int i = 0; i < (unsigned int) coords.size(); ++i)
    {
      B.add_vertex(Point(coords.at(i).at(0), coords.at(i).at(1),
    coords.at(i).at(2)));
  }

  // Add the polyhedron triangles.
  for(unsigned int i = 0; i < (unsigned int) tris.size(); ++i)
  {
   B.begin_facet();
   B.add_vertex_to_facet(tris.at(i).at(0));
   B.add_vertex_to_facet(tris.at(i).at(1));
   B.add_vertex_to_facet(tris.at(i).at(2));
   B.end_facet();
  }

  // Finish up the surface.
  B.end_surface();
 }
};
```

The matrices are modeled as Standard Template Library (STL) vectors of vectors. The following example describes the triangular mesh of Figure B.1.

C++ code

```
// in main.cpp

#include "ParserCGAL.h"

std::vector<std::vector<double> > coords;
std::vector<std::vector<unsigned int> > tris;

// ... Allocate memory space for vectors.

// x, y, z coordinates of all points in surface mesh.
coords[0][0] = 0.0; coords[0][1] = 0.0; coords[0][1] = 0.0;
coords[1][0] = 1.0; coords[1][1] = 0.0; coords[1][1] = 0.0;
```

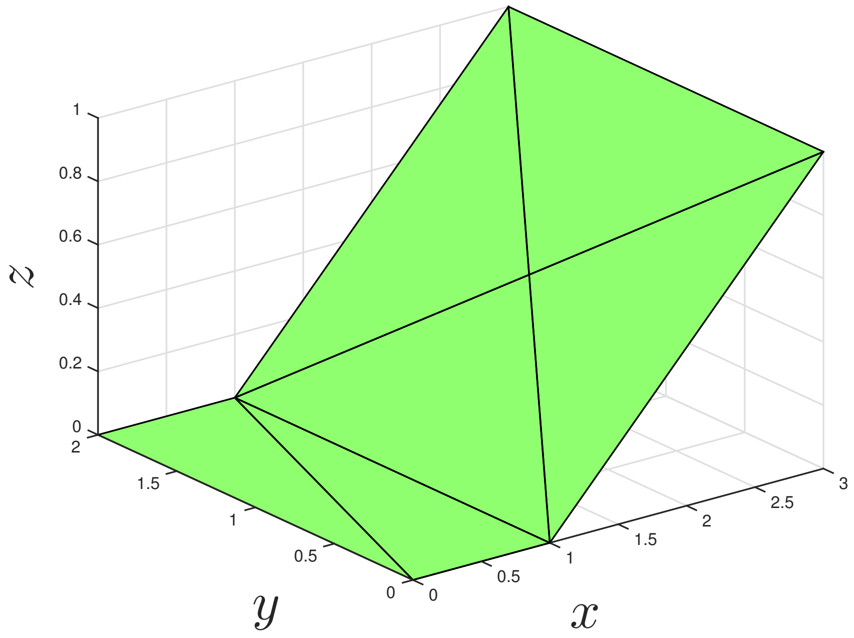


Figure B.1: Surface mesh example.

```
coords[2][0] = 3.0; coords[2][1] = 0.0; coords[2][1] = 1.0;
coords[3][0] = 3.0; coords[3][1] = 2.0; coords[3][1] = 1.0;
coords[4][0] = 1.0; coords[4][1] = 2.0; coords[4][1] = 0.0;
coords[5][0] = 2.0; coords[5][1] = 1.0; coords[5][1] = 0.5;
coords[6][0] = 0.0; coords[6][1] = 2.0; coords[6][1] = 0.0;

// Surface mesh topology.
tris[0][0] = 0; tris[0][1] = 6; tris[0][2] = 4;
tris[1][0] = 0; tris[1][1] = 4; tris[1][2] = 1;
tris[2][0] = 1; tris[2][1] = 4; tris[2][2] = 5;
tris[3][0] = 4; tris[3][1] = 3; tris[3][2] = 5;
tris[4][0] = 5; tris[4][1] = 3; tris[4][2] = 2;
tris[5][0] = 1; tris[5][1] = 5; tris[5][2] = 2;
```

The surface meshes in femdoc are described in the same manner as above, by coordinates and topology. The surface mesh can be mapped to the CGAL `Polyhedron_3` format using the parser class formerly described.

C++ code

```
// in main.cpp

// Build the surface mesh.
Polyhedron_3 P;
polyhedron_builder<HalfedgeDS> triangle(coords, tris);
P.delegate(triangle);

// Initialize indices of vertices, halfedges and facets.
CGAL::set_halfedgeds_items_id(P);

// Assert it was built correctly.
CGAL_assertion(P.is_pure_triangle());
```

Now, in order to find the shortest distance between two points, we need to instantiate the

`Surface_mesh_shortest_path` CGAL class.

C++ code

```cpp
// in main.cpp

// Construct a shortest path query object.
Surface_mesh_shortest_path shortest_paths(P);
```

Then, we need to add a source point from which we will compute the distance.

C++ code

```cpp
// in main.cpp

// Select a vertex index.
const int target_vertex_index = 6;

// Build a vertex iterator.
vertex_iterator vertex_it = vertices(P).first;
std::advance(vertex_it, target_vertex_index);

// Add the source point.
shortest_paths.add_source_point(*vertex_it);
```

Alternatively, we can add some point on a triangle facet as the source point. This option is particularly attractive because we can use the integration points as the baryocentric coordinates.

C++ code

```cpp
// in main.cpp

// Select a triangle index.
const int target_triangle_index = 0;

// Build a face iterator.
face_iterator face_it = faces(P).first;
```

```
std::advance(face_it, target_triangle_index);

// Integration point in local coordinates.
double xi = 0.0;
double eta = 0.0;
double zeta = 1.0;

// Define a barycentric coordinate inside the face.
Traits::Barycentric_coordinate face_location = {{xi, eta, zeta}};

// Add the source point.
shortest_paths.add_source_point(*face_it, face_location);
```

The following example shows how to get the shortest path to every vertex on the surface from the source point defined above.

C++ code

```
vertex_iterator vit, vit_end;
for (boost::tie(vit, vit_end) = vertices(P); vit != vit_end; ++vit)
{
  std::vector<Traits::Point_3> points;
  shortest_paths.shortest_path_points_to_source_points(*vit,
    std::back_inserter(points));

  // Print the points.
  std::cout << points.size() << " ";
  for (std::size_t i = 0; i < points.size(); ++i)
    std::cout << " " << points[i];
  std::cout << std::endl;
}
```

Alternatively, you can define another facet with baryocentric coordinates and find its distance to the source point.

C++ code

```cpp
double t12_cgal = shortest_paths.shortest_distance_to_source_points (*face_it2,
    face_location2).first;
```

A CGAL mesh can be converted to an ASCII file. These ASCII files can be written in the `.off` format, and visualized with the Geomview library. Alternatively, the code below could be easily modified to output `.stl` files.

C++ code

```cpp
// in main.cpp
CGAL::set_ascii_mode(std::cout);
std::cout << "OFF" << std::endl << P.size_of_vertices() << ' '
      << P.size_of_facets() << " 0" << std::endl;

std::cout << std::endl;

std::copy( P.points_begin(), P.points_end(),
    std::ostream_iterator<Point_3>(std::cout, "\n"));

std::cout << std::endl;

for (Facet_iterator i = P.facets_begin(); i != P.facets_end(); ++i)
{
  Halfedge_facet_circulator j = i->facet_begin();

  // Facets in polyhedral surfaces are at least triangles.
  CGAL_assertion(CGAL::circulator_size(j) >= 3);
  std::cout << CGAL::circulator_size(j) << ' ';
  do {
    std::cout << ' ' << std::distance(P.vertices_begin(), j->vertex());
  } while (++j != i->facet_begin());
  std::cout << std::endl;
}
```

```

std::cout << std::endl;
```

For more details on the library and additional API calls, the reader is referred to the CGAL user manual [CGAL, 2009].

Appendix C

Publication P1: Density and Level-Set XFEM Schemes for Topology Optimization of 3D Structures

Computational Mechanics manuscript No.
(will be inserted by the editor)

Density and Level-Set XFEM Schemes for Topology Optimization of 3D Structures

Topology Optimization in 3D

Carlos H. Villanueva · Kurt Maute

Received: date / Accepted: date

Abstract 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.

Keywords eXtended Finite Element Method · Topology Optimization · Solid Isotropic Material with Penalization · Level Set Methods · Additive Manufacturing · 3D Printing

C. H. Villanueva
Department of Mechanical Engineering,
University of Colorado at Boulder,
Boulder, CO 427 UCB, USA
e-mail: carlos.villanueva@colorado.edu

K. Maute
Department of Aerospace Engineering,
University of Colorado at Boulder,
Boulder, CO 427 UCB, USA
e-mail: maute@colorado.edu

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 (Ning and Pellegrino, 2012; Meisel et al, 2013). 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 (Bendsøe and Sigmund, 2003; Rozvany, 2009). 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 (Sigmund, 2001a,b). 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 (1989) and Zhou and Rozvany (1991). It features great versatility, robustness, efficiency, and ease of implementation for a broad range of applications (Sigmund and Maute, 2013; Deaton and Grandhi, 2013).

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 (Maute et al, 1998). Often this issue can be mitigated by mesh refinement or adaptive re-meshing (Maute and Ramm, 1995, 1997). 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 (Sigmund and Maute, 2013).

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 (Allaire et al, 2004; Sethian and Wiegmann, 2000; Wang et al, 2003).

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 (Dijk et al, 2013). 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 (2013) and Gain and Paulino (2013).

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) (Wang and Wang, 2006), the eXtended finite element method (XFEM) (van Miegroet and Duysinx, 2007; Wei et al, 2010; Kreissl and Maute, 2012), and local re-meshing schemes (Yamasaki et al, 2011).

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 (Wang and Wang, 2006) 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 (1997). The XFEM was originally proposed by Belytschko and Black (1999) to model crack propagation. The reader is referred to Yazid et al (2009) 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 (Gerstenberger and Wall, 2008b,a), multi-phase flows (Fries, 2009), and nano-scale heat transfer (Lee et al, 2011). A general overview of the method is presented by Fries and Belytschko (2010).

Duysinx et al (2006) 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 (Makhija and Maute, 2014). 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 (2006), van Miegroet et al (2005), and van Miegroet and Duysinx (2007), and to topology optimization of two and three dimensional structures by Herrero et al (2013) and Li et al (2012), 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 (2010) to solve "solid-void" structural topology optimization problems, modeling the "void" phase as a soft material. Maute et al (2011) 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 (2014), who presented a generalized enrichment strategy based on the step enrichment of Hansbo and Hansbo (2004) 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 (2012), we will adopt the generalized enrichment strategy of Makhija and Maute (2014) 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 (2013) 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 2. The mechanical model and the XFEM formulation are summarized in Section 3. Details of the LSM-XFEM and SIMP optimization approaches are presented in Section 4. Section 5 highlights specific computational challenges of the XFEM approach for three dimensional problems. Section 6 presents structural topology optimization examples, comparing the LSM-XFEM and SIMP approaches. Section 7 summarizes the main conclusions drawn from this study.

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 (2010). The optimization variables define analytically or by means of auxiliary partial differential equations the nodal or element density parameters (Sigmund and Maute, 2013). 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 (Guest et al, 2004; Sigmund, 2007).

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 $\mathit{\Gamma}_{A,B}$ is defined as follows:

$$\begin{aligned} &\phi(\mathbf{x}) < 0, \quad \forall\ \mathbf{x} \in \mathit{\Omega}_A, \\ &\phi(\mathbf{x}) > 0, \quad \forall\ \mathbf{x} \in \mathit{\Omega}_B, \\ &\phi(\mathbf{x}) = 0, \quad \forall\ \mathbf{x} \in \mathit{\Gamma}_{A,B}, \end{aligned} \tag{1}$$

where the vector $\mathbf{x}$ collects the spatial coordinates, $\mathit{\Omega}_A$ is the domain of material phase "A", $\mathit{\Omega}_B$ is the domain of material phase "B", and $\mathit{\Gamma}_{A,B}$ 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:

$$\phi_i = (x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2 - r^2, \tag{2}$$

where r represents the radius of the sphere, and the sign of ϕ_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 (Dijk et al, 2013). 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 4.1). 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 (Eschenauer et al, 1994; Sokolowski and Zochowski, 1999; Burger et al, 2004; Norato et al, 2007). An example of an initial design with a regular pattern of spherical inclusions is shown in Fig. 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. 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.

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.

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. 2, where $\mathit{\Gamma}_N$ denotes the surface where traction forces are applied, $\mathit{\Gamma}_D$ 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 can be decomposed into the following terms:

$$W = W_S + W_L + W_k = 0, \tag{3}$$

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".

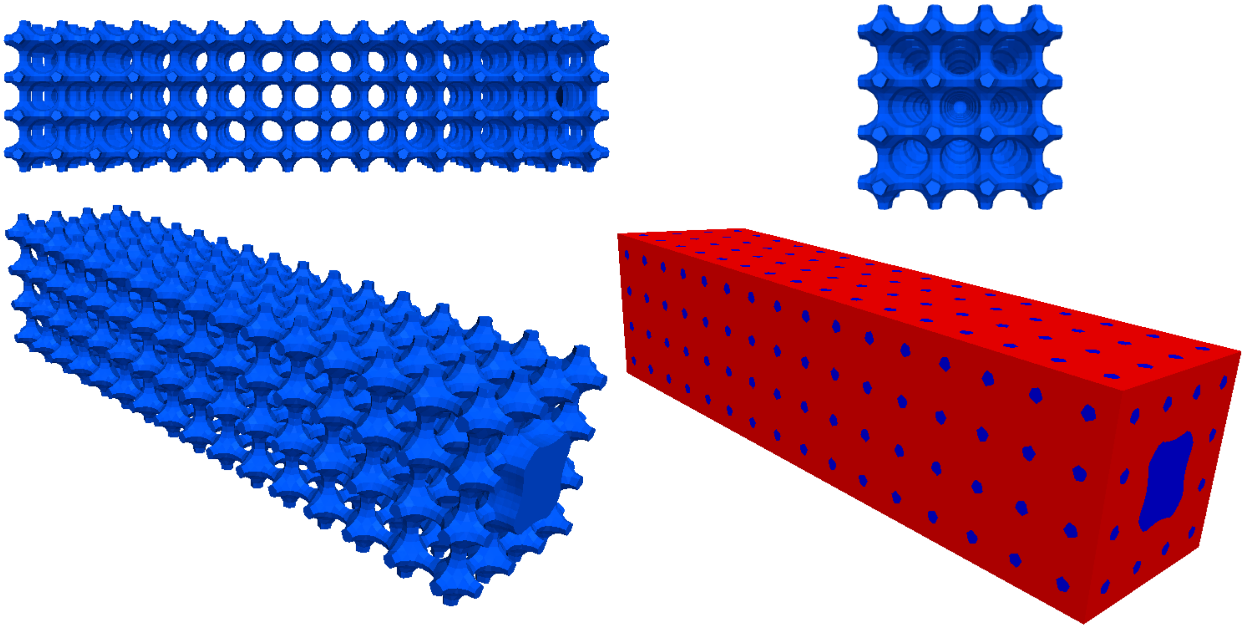


Fig. 1: Initial design with array of spherical inclusions for the cantilever beam example of Section 6.3; blue represents phase “A” material and red represents phase“B”.

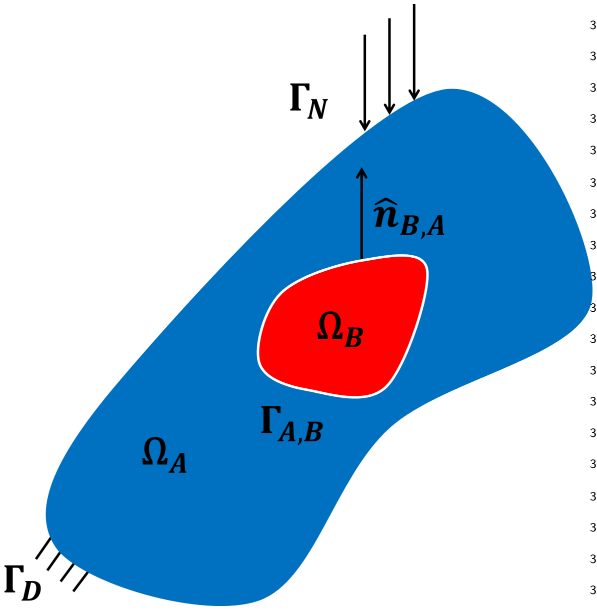


Fig. 2: Two-phase problem.

The weak form of the structural equilibrium equations is:

$$W_S = \int_{\Omega_A} \boldsymbol{\eta} \colon \boldsymbol{\sigma}(\mathbf{u}) \; d\Omega + \int_{\Omega_B} \boldsymbol{\eta} \colon \boldsymbol{\sigma}(\mathbf{u}) \; d\Omega - \int_{\Omega_A} \mathbf{v} \cdot \mathbf{b} \; d\Omega - \int_{\Omega_B} \mathbf{v} \cdot \mathbf{b} \; d\Omega - \int_{\Gamma_N} \mathbf{v} \cdot \mathbf{f} \; d\Gamma_N, \quad (4)$$

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 Γ_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 (1995), Juntunen and Stenberg (2009), and Dolbow and Harari (2009) for more details.

In this paper, we enforce displacement continuity along phase boundaries for “solid-solid” problems via the following stabilized Lagrange multiplier method (Makhija and Maute, 2014):

$$W_L = -\int_{\Gamma_{A,B}} [\mathbf{v}] \cdot \boldsymbol{\lambda} \; d\Gamma_{A,B} + \gamma \int_{\Gamma_{A,B}} \boldsymbol{\mu} \cdot [\mathbf{u}] \, d\Gamma_{A,B} + \int_{\Gamma_{A,B}} \mu \cdot (\boldsymbol{\lambda} - \bar{\boldsymbol{\sigma}} \cdot \mathbf{n}_{A,B}) \, d\Gamma_{A,B}, \quad (5)$$

$$[\mathbf{u}] = \mathbf{u}^{(A)} - \mathbf{u}^{(B)} \; , [\mathbf{v}] = \mathbf{v}^{(A)} - \mathbf{v}^{(B)} \; , \quad (6)$$

$$\bar{\boldsymbol{\sigma}} = \frac{1}{2} \left(\boldsymbol{\sigma}^{(A)} + \boldsymbol{\sigma}^{(B)} \right), \quad (7)$$

where $\boldsymbol{\lambda}$ is the Lagrange multiplier, and $\boldsymbol{\mu}$ is the associated test function. The higher the weight γ 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 (Wei et al, 2010). However, this approach requires accounting for the interface contributions (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 (2014) 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:

$$W_k = \int_{\Omega_A} k \; \mathbf{v} \cdot \mathbf{u} \; d\Omega, \tag{8}$$

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 (2014).

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 (2014) 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:

$$\mathbf{u}(\mathbf{x}) = \sum_{m=1}^{M} \left(H(-\phi) \sum_{i=1}^{n} \mathbf{N}_i \; \mathbf{u}_{i,m}^{A} + H(\phi) \sum_{i=1}^{n} \mathbf{N}_i \; \mathbf{u}_{i,m}^{B} \right) \tag{9}$$

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 $l = [A, B]$, ϕ 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 5.

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

$$H(z) = \begin{cases} 1 & z > 0, \\ 0 & z \leq 0, \end{cases} \tag{10}$$

The Heaviside functions "turns on/off" the standard finite element interpolations in the particular phases. The approximation (9) allows for discontinuities of the displacements along the phase boundaries. Therefore the continuity is enforced weakly via the stabilized Lagrange multiplier method (5).

Following a Bubnov-Galerkin scheme, we test the governing equations with the same subspace as we use for the trail functions; see Eq. 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.

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 (2013), 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:

$$\tilde{\mathbf{u}} = \boldsymbol{T}\mathbf{u}, \tag{11}$$

where $\mathbf{u}$ is the vector of displacement degrees of freedom according to Eq. 9, $\boldsymbol{T}$ is a transformation matrix, and $\tilde{\mathbf{u}}$ is the solution vector in the transformed space. The residual, $\tilde{\boldsymbol{R}}$, and stiffness matrix, $\tilde{\boldsymbol{K}}$, in the transformed space are defined as:

$$\tilde{\boldsymbol{R}} = \boldsymbol{T}^T \boldsymbol{R} \qquad \tilde{\boldsymbol{K}} = \boldsymbol{T}^T \boldsymbol{K} \boldsymbol{T}, \tag{12}$$

where the residual, $\boldsymbol{R}$, and the stiffness matrix, $\boldsymbol{K}$, result from integrating the weak form of the governing equations using the XFEM approximation (9).

The preconditioner $\boldsymbol{T}$ 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:

$$\boldsymbol{T}^l_{i,m} = \left(\max_{e \in E_i} \frac{\int_{\mathcal{D}^e_l} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x})\, d\mathbf{x}}{\int_{\mathcal{D}^e} \nabla \mathbf{N}_i(\mathbf{x}) \cdot \nabla \mathbf{N}_i(\mathbf{x})\, d\mathbf{x}} \right)^{-1/2}, \tag{13}$$

where $\boldsymbol{T}^l_{i,m}$ corresponds to the degree of freedom $\mathbf{u}^l_{i,m}$, 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 $\mathcal{D}^e_l$ 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 $T^l_{i,m}$ 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 $\boldsymbol{T}$, the degrees of freedom associated with the diagonal entry $\boldsymbol{T}^l_{i,m}$ are constrained to zero if the following condition is satisfied:

$$\boldsymbol{T}^l_{i,m} \geq T_{tol}, \tag{14}$$

where T_{tol} is a specified tolerance. As studies by Lang et al (2013) 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 (2013).

4 Optimization Model

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

$$\begin{aligned} &\min_{\mathbf{s}} \mathcal{F}(\mathbf{s}, \mathbf{u}(\mathbf{s})), \\ &\text{s.t.} \begin{cases} \mathbf{s}, & \text{subject to design constraints } \mathcal{G}_j \leq 0, \\ \mathbf{u}, & \text{solves } W = 0 \text{ for a given } \mathbf{s}, \end{cases} \end{aligned} \tag{15}$$

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 (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.

4.1 XFEM

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

$$\phi^n(\mathbf{s}) = \frac{\sum\limits_{i=1}^{P} \mathbf{w}^n_i \mathbf{s}_i}{\sum\limits_{i=1}^{P} \mathbf{w}^n_i}, \tag{16}$$

with

$$\mathbf{w}^n_i = max\left(0, r_\phi - \parallel \mathbf{x}_i - \mathbf{x}_n \parallel\right), \tag{17}$$

where ϕ^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, $\mathbf{w}^n_i$ is the weight of node n with respect to design variable i, r_ϕ 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 (2012) and Makhija and Maute (2014), 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 6.

4.2 SIMP

Here, the material distribution is parameterized by nodal density values, ρ_i, which are treated as optimization variables, i.e. $\rho_i = \mathbf{s}_i$. Following the work of Guest et al (2004), we compute the elemental density by combining a linear density filter and a projection scheme as follows:

$$\rho^e(\mathbf{s}) = \frac{\sum\limits_{i=1}^{E} \mathbf{w}^e_i \mathbf{s}_i}{\sum\limits_{i=1}^{E} \mathbf{w}^e_i}, \tag{18}$$

with

$$\mathbf{w}_i^e = max\left(0, r_\rho - \parallel \mathbf{x}_i - \mathbf{x}_e \parallel\right), \tag{19}$$

where ρ^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_ρ is the filter radius, and E is the number of elements in the computational mesh.

Guest et al (2004) 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:

$$\hat{\rho}^e(\mathbf{s}) = 1 - e^{-\beta\rho^e(\mathbf{s})} + \rho^e(\mathbf{s})e^{-\beta} \tag{20}$$

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:

$$E(\mathbf{x}) = E_B + |E_A - E_B|\hat{\rho}^e(\mathbf{s})^p \tag{21}$$

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 (18) prevents the formation of features smaller than r_ρ, typically at the cost of generating intermediate density values along the phase boundaries. This effect is mitigated by the projection (20) which, in the limit for $\beta \to \infty$, maps non-zero ρ^e values into "1". The reader is referred to the papers by Guest et al (2004) and Guest et al (2011) for further details of the scheme presented above, and to Sigmund and Maute (2013) for a comprehensive discussion of projection schemes.

5 Computational Considerations

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. 3. In three dimensions, there are 127 intersection configurations. To handle this complexity, we compute 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.

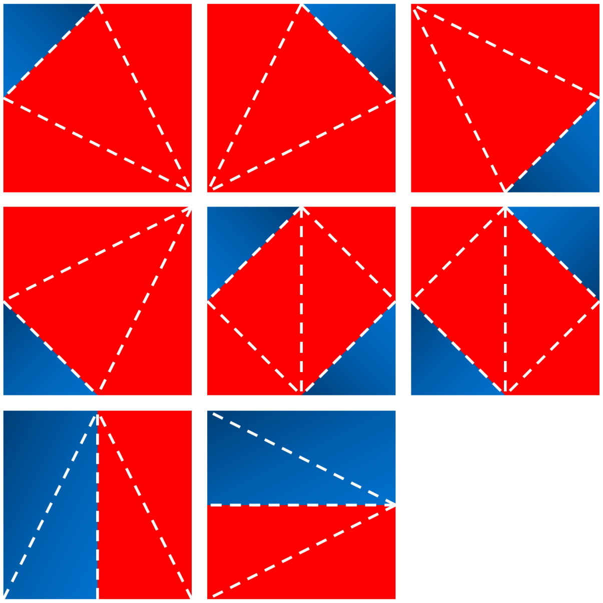


Fig. 3: Intersection patterns for a two-dimensional QUAD4 element.

Previous studies on topology optimization for three dimensional structures with the XFEM (Li et al, 2012) 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 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.

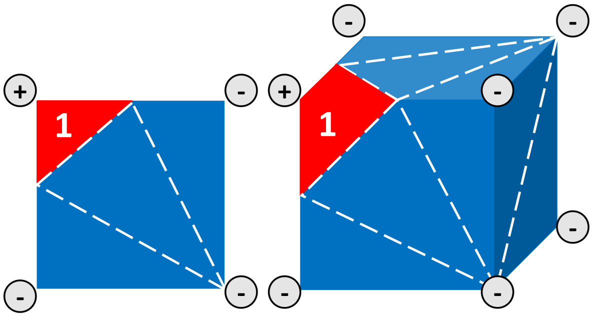


Fig. 4: Initial assignment of enrichment levels.

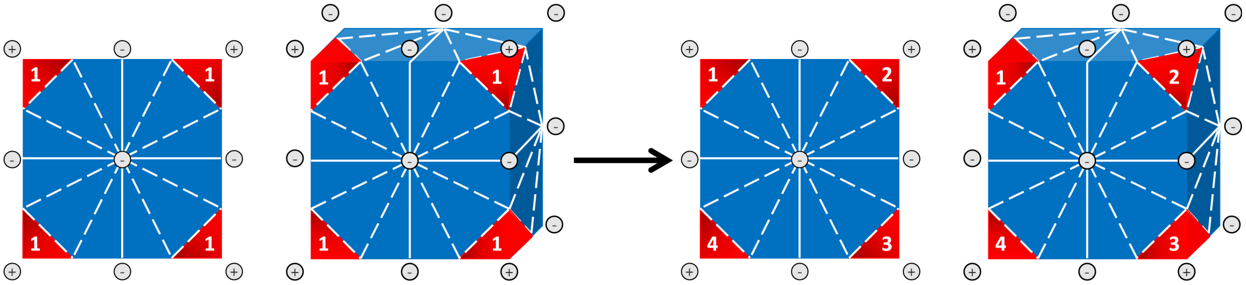


Fig. 5: Initial and final assignment of enrichment levels considering a cluster of elements.

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 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. 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 (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 (12).

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 4.2. 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 (16) and (18), and perimeter constraints.

In all examples, the optimization problems are solved by the Globally Convergent Method of Moving Asymptotes (GCMMA) of Svanberg (2002). 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 (Heroux et al, 2003). 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 β 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 3D 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, ρ_T, from the nodal density values, ρ_i; see Section 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 ρ_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:

$$\bar{\rho} = \frac{1}{\int_{\Omega_D} d\Omega_D} \int_{\Omega_D} \hat{\rho}^e (1 - \hat{\rho}^e)\, d\Omega_D \tag{22}$$

where Ω_D denotes the design domain.

6.1 Cube with center load

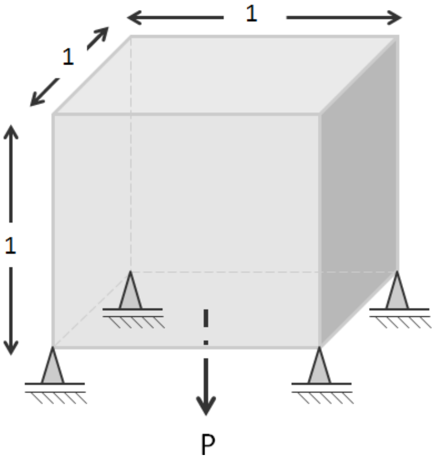


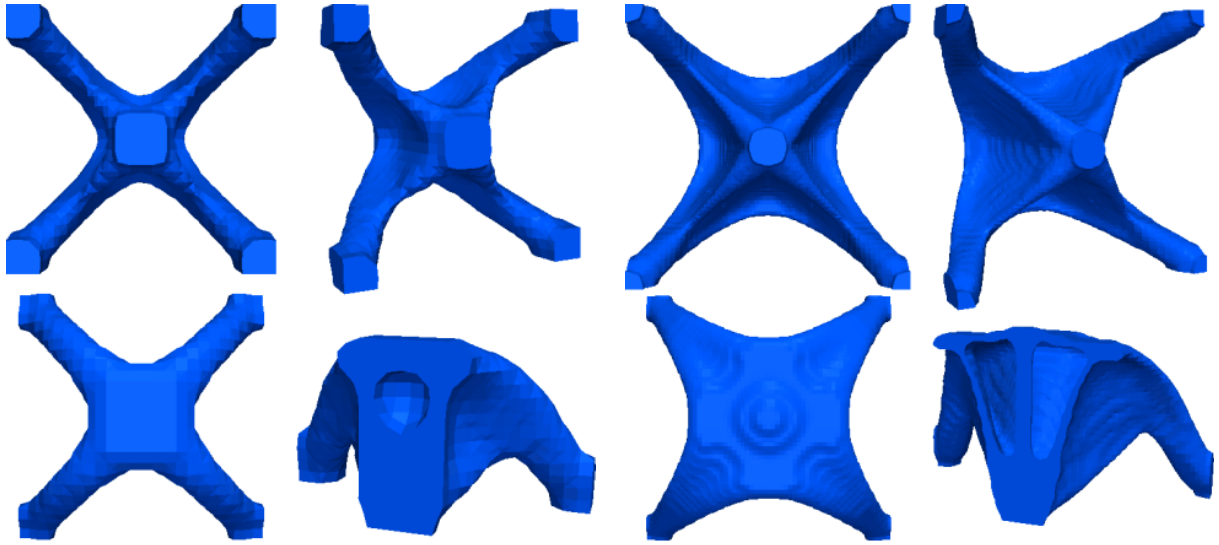
Fig. 6: Cube with center load.

We consider the “solid-void” optimization problem depicted in Fig. 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: $24 \times 24 \times 24$ 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× element edge length). While this approach does not ensure mesh-independent optimization results, it still prevents the formation of checker-board 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 7 where material with a density lower than $\rho_i < 0.75$ is considered void. The strain energies are reported in Tab. 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 approach described above. The strain energies for varying threshold values, ρ_T, are plotted in Fig. 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 ρ_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 (22) are 0.285 and 0.0189 for the coarse and fine mesh, re-



(a) 24x24x24 mesh

(b) 65x65x65 mesh

Fig. 7: SIMP results for cube with center load problem; clockwise: bottom, side, top, and clipped views.

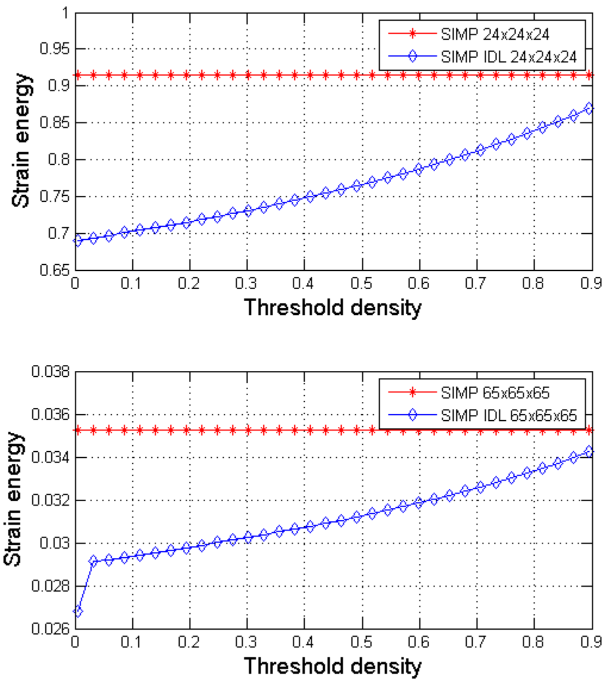


Fig. 8: IDL post-processing of SIMP results for cube with center load problem.

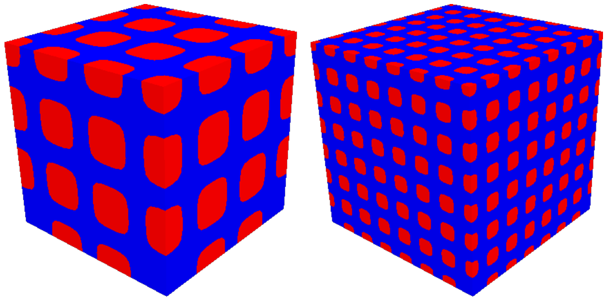
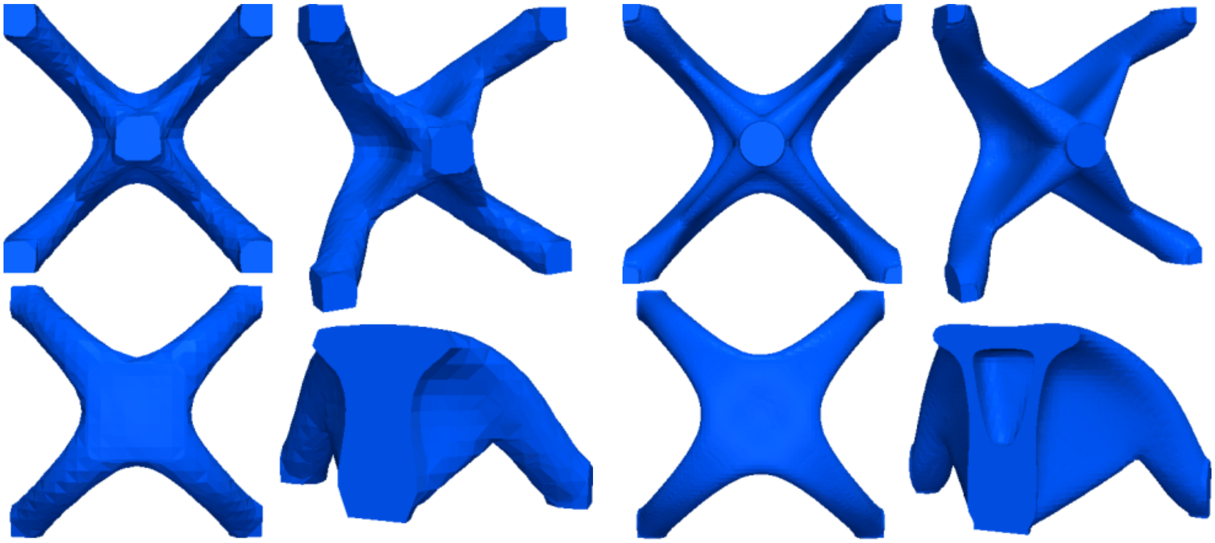


Fig. 9: Initial level set configurations for cube with center load problem.

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

Table 1: Comparison of strain energies of SIMP and LSM-XFEM results for cube with center load problem; the corresponding designs are shown in Figs. 7 and 10.

spectively. The post-processed designs have lower strain energies because the post-processing counteracts the effect of the density filter (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 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



(a) 24x24x24 mesh

(b) 65x65x65 mesh

Fig. 10: LSM-XFEM results for cube with center load problem; clockwise: bottom, side, top, and clipped views.

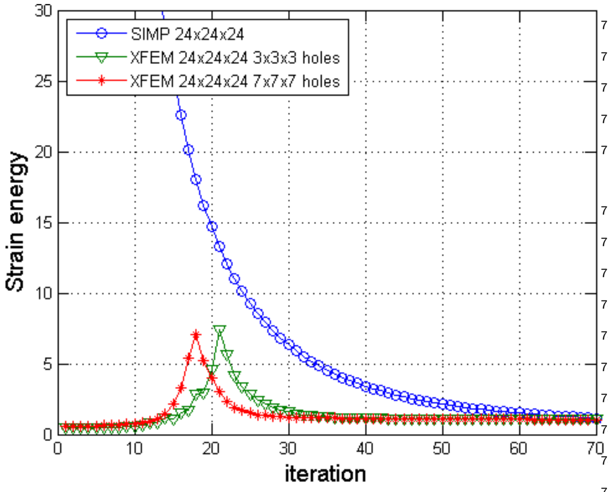


Fig. 11: Evolution of strain energies in the optimization process for SIMP and LSM-XFEM approaches.

rounded corners, by modifying Eq. 2 into the following:

$$\phi_i = (x_i - x_c)^{10} + (y_i - y_c)^{10} + (z_i - z_c)^{10} - r^{10}. \quad (23)$$

One configuration has $3\times3\times3$ equally spaced holes with radius 5.50, the other $7\times7\times7$ holes with radius 2.0, as shown in Fig. 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. 10. The strain energies of the optimized designs are given in Tab. 1. The convergence history for the coarse meshes in SIMP and LSM-XFEM is shown in Fig. 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.

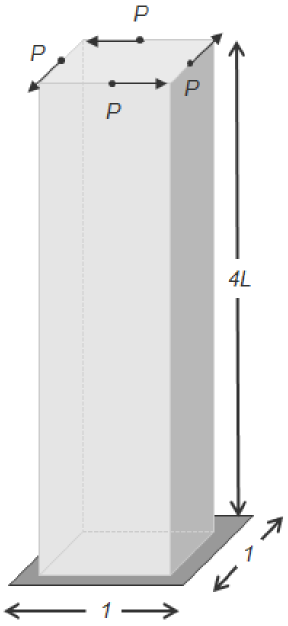


Fig. 12: Cuboid under torsion model.

6.2 Cuboid under torsion

The second “solid-void” example is taken from Nguyen et al (2012) 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. 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.

6.2.1 Mesh convergence study

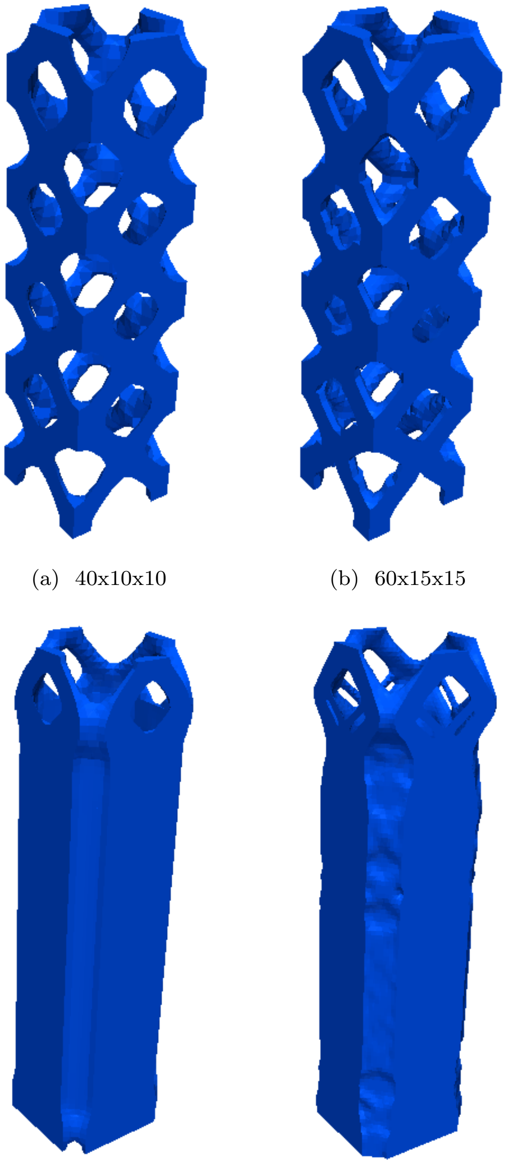
The optimization problem is solved with the SIMP approach for four different mesh sizes: $40 \times 10 \times 10$, $60 \times 15 \times 15$, $80 \times 20 \times 20$, and $120 \times 30 \times 30$. The Young’s modulus of the void phase is set to $E_B = 10^{-9}$. The design domain is initialized with 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.

	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 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. 13.

The optimized material distributions are shown in Fig. 13 where material with a density lower than $\rho_i < 0.35$ is considered void. The strain energies are reported in Tab. 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 dimensional problems. The optimized material layout switches abruptly from a grid-type structure, which conceptually agrees with the results of Nguyen et al (2012), 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. 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. 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. Al-



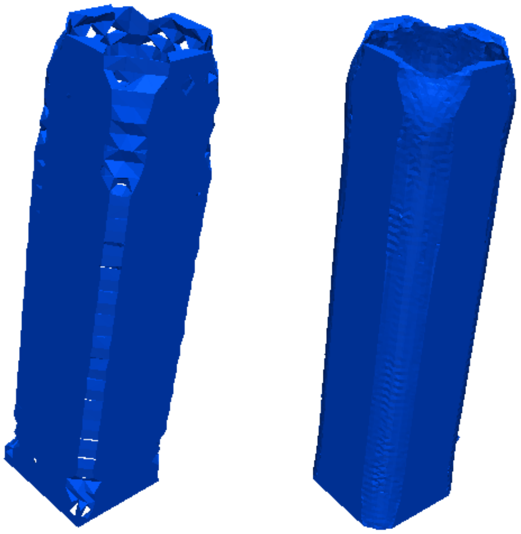
(a) 40x10x10 (b) 60x15x15

(c) 80x20x20 (d) 120x30x30

Fig. 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	$40 \times 10 \times 10$	$8.7551e + 02$
	$120 \times 30 \times 30$	$9.8262e + 02$

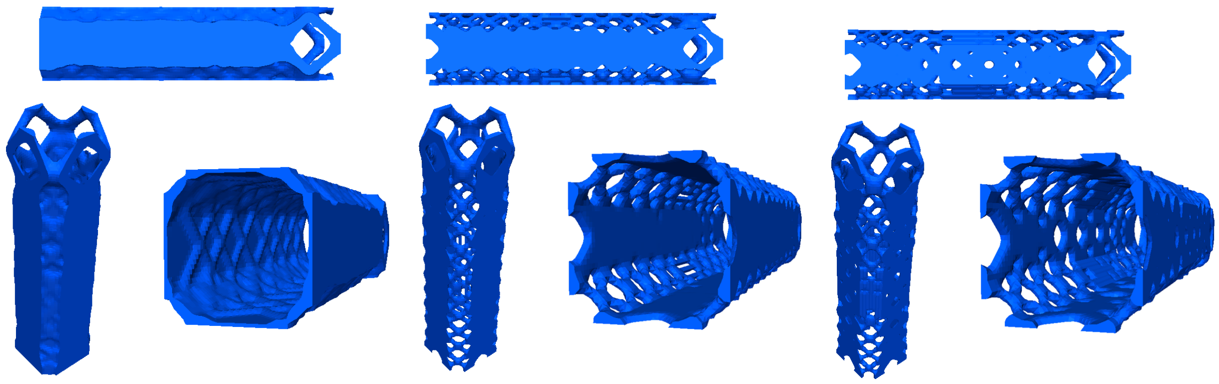
Table 3: Strain energies of LSM-XFEM results for cuboid under torsion problem; LSM-XFEM parameters: $r_\phi = 1.6$ of the element edge length, no perimeter constraint; the corresponding designs are shown in Fig. 14.



(a) 40x10x10 (b) 120x30x30

Fig. 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.

though 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 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 (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. 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 reports on the strain energies and the volume fractions of intermediate densities, $\bar{\rho}$, as the projection parameter, β, is increased. The higher β, the lower $\bar{\rho}$ and the lower the strain energy, approaching the one of the LSM-XFEM result. Note that as β 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 β, we post-process the SIMP results for $\beta = 0$ by the IDL post-processing method. Figure 16 shows the strain energy of the post-processed design over the threshold density, ρ_T, for the coarsest and the finest mesh. The volume constraint is satis-



(a) $\beta = 0$ (b) $\beta = 4$ (c) $\beta = 8$

Fig. 15: SIMP results for different projection parameters $\beta = [0.0, 4.0, 8.0]$; mesh size: $120 \times 30 \times 30$.

	β 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: Strain energies of SIMP results for different projection parameters $\beta = [0.0, 4.0, 8.0]$; the corresponding designs are shown in Fig. 15.

	Mesh	ρ_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 5: Strain energies of the SIMP-IDL designs of cuboid under torsion problem; the corresponding designs are shown in Fig. 13.

fied 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. 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 and the LSM-XFEM results in Tab. 3. For the coarse mesh, the strain energy is well below the raw SIMP results from Tab. 2 but still above the results for the LSM-XFEM approach in Tab. 3. As we will see below, this is because of the larger smoothing radius which prevents the formation of smaller features and thinner walls.

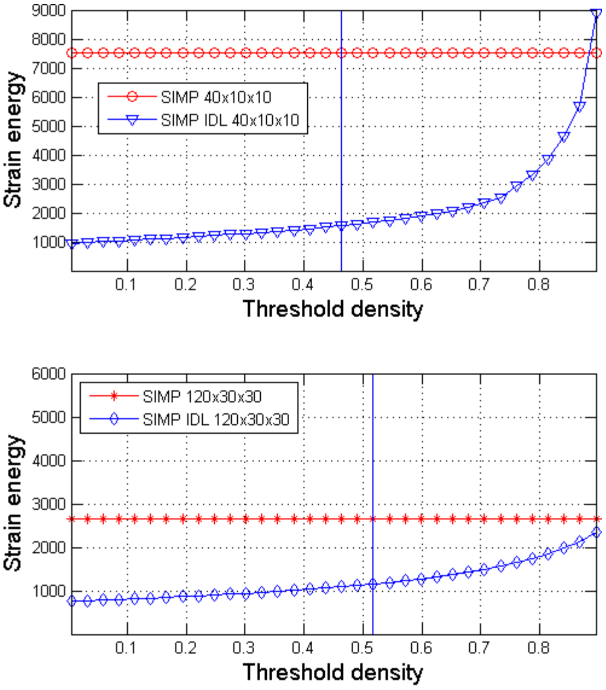
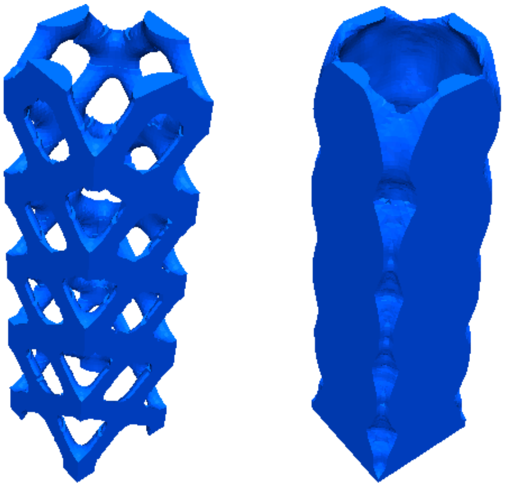


Fig. 16: IDL post-processing of SIMP results for cuboid under torsion problem; the vertical lines mark the threshold values at which the volume constraint is satisfied.

6.2.2 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



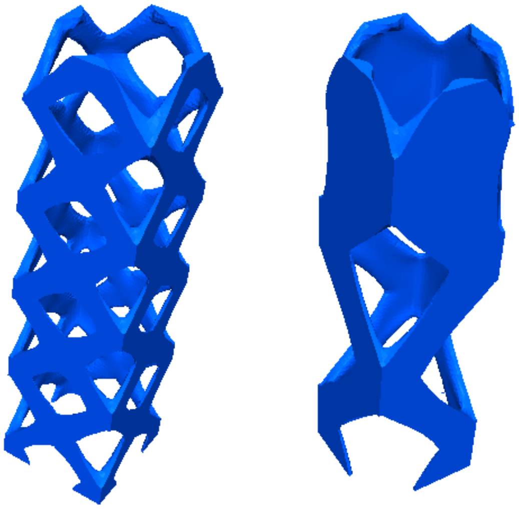
(a) SIMP (b) LSM-XFEM

Fig. 17: SIMP and LSM-XFEM results for larger smoothing radius; mesh size: 120 × 30 × 30.

meshes, independent of their size. This observation is in agreement with studies for two-dimensional problems, see for example Kreissl and Maute (2012), 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 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. 13(a). Comparing the SIMP results in Fig. 17(a) and Fig. 13(a) confirms the finding of numerous studies (Bendsøe and Sigmund, 2003) 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. 17(a) is given in Tab. 6. A similar effect is not observed in the LSM-XFEM approach when we apply the same filter radius, r_ϕ, used earlier for the coarse mesh to the fine mesh. Figure 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. 6. To control the overall structural complexity in the LSM, the formulation of the optimization problem (15) is often augmented by a perimeter constraint (Dijk et al, 2013). 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. 17(a) and impose this value as an upper bound on the perimeter. One problem uses the SIMP result in Fig. 17(a) as the initial design, and the other one uses the LSM-XFEM result in Fig. 14(b). The results are shown in Fig. 18 and the strain energies are given in Tab. 7.

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

Table 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. 17.



(a) (b)

Fig. 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 × 30 × 30.

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

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

Table 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. 18.

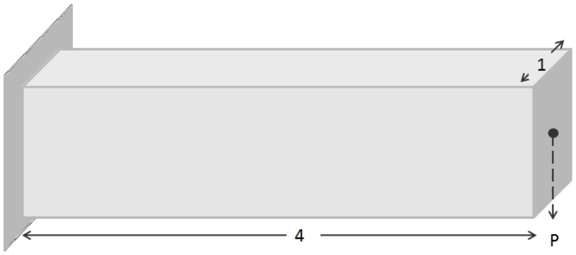


Fig. 19: Initial setup for two-phase cantilever beam problem.

sign in Fig. 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. 18(a) resembles closely the SIMP result from which it was restarted. However, considering the strain energy in Tab. 7, this design has a larger strain energy than the one in Fig. 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. 18(b) shows. The design has more structural features than the design without perimeter constraint in Fig. 14(b).

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 discussed in Section 1 is not applicable to such problems. The generalized enrichment strategy of Section 3.2 is required and the interface conditions of Eq. 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. 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× 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. 20 where material with a density lower than $\rho_i < 0.25$ is transparent. The strain energies are reported in Tab. 8.

The LSM-XFEM results for a smoothing radius of $r_\phi = 0.05333$ (1.6× the element edge) are shown in Fig. 20 and the strain energies are given in Tab. 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. 1 and satisfies the volume constraint for the stiff phase. Note that the interface condition is enforced via the stabilized Lagrange multiplier method (5) with an element wise constant Lagrange multiplier, λ, 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 $E_B = 0.1 E_A$ in Fig. 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.

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,

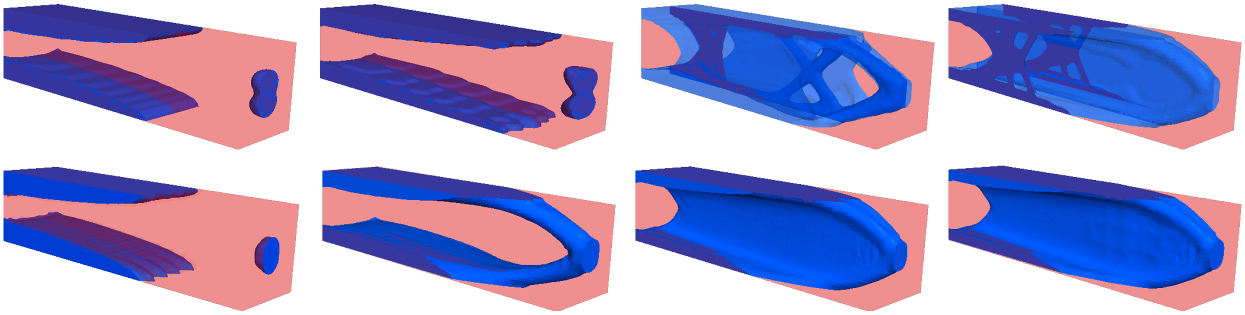


Fig. 20: SIMP and LSM-XFEM results for different stiffness ratios: (from left to right) $E_B = 0.5E_A$, $E_B = 0.1E_A$, $E_B = 0.01E_A$, E_B is void; SIMP results (top row); LSM-XFEM results (bottom row).

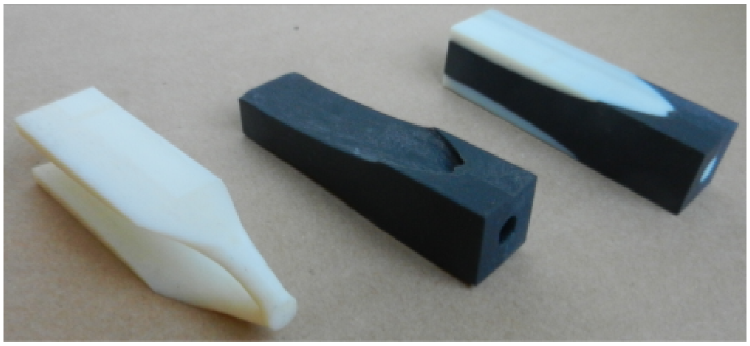


Fig. 21: LSM-XFEM optimized two-phase design for $E_B = 0.1E_A$ realized by 3D printing.

	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 8: Strain energies of SIMP and LSM-XFEM results for different stiffness ratios; the corresponding designs are shown in Fig. 20.

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 (2012) 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.

Acknowledgement

The authors acknowledge the support of the National Science Foundation under grant EFRI-ODISSEI 1240374 and CBET 1246854. The opinions and conclusions presented in this paper are those of the authors and do not necessarily reflect the views of the sponsoring organization.

References

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

Babuška I, Melenk JM (1997) The partition of unity method. International Journal for Numerical Methods in Engineering 40(4):727–758

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

Bendsøe M (1989) Optimal shape design as a material distribution problem. Structural and Multidisciplinary Optimization 1(4):193–202

Bendsøe MP, Sigmund O (2003) Topology Optimization: Theory, Methods and Applications. Springer

Burger M, Hackl B, Ring W (2004) Incorporating topological derivatives into level set methods. Journal of Computational Physics 194(1):344–362

Deaton J, Grandhi R (2013) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Structural and Multidisciplinary Optimization pp 1–38, URL http://dx.doi.org/10.1007/s00158-013-0956-z

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

Dolbow J, Harari I (2009) An efficient finite element method for embedded interface problems. Int J Numer Meth Engng 78:229–252

Duysinx P, Miegroet 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

Eschenauer H, Kobelev V, Schumacher A (1994) Bubble method for topology and shape optimization of structures. Structural and Multidisciplinary Optimization 8(1):42–51

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

Fries TP (2009) The intrinsic xfem for two-fluid flows. Int J Numer Meth Fluids 60(4):437–471

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

Gerstenberger A, Wall WA (2008a) Enhancement of fixed-grid methods towards complex fluid-structure interaction applications. Int J Numer Meth Fluids 57(9):1227–1248

Gerstenberger A, Wall WA (2008b) An extended finite element method/Lagrange multiplier based approach for fluid-structure interaction. Computer Methods in Applied Mechanics and Engineering 197:1699–1714

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

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

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

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

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

Juntunen M, Stenberg R (2009) Nitsches method for general boundary conditions. Mathematics of Computation 78:1353–1374

Kreissl S, Maute K (2012) Levelset based fluid topology optimization using the extended finite element method. Structural and Multidisciplinary Optimization 46(3):311–326

Lang C, Makhija D, Doostan A, Maute K (2013) A simple and efficient preconditioning scheme for xfem with heaviside enrichments. Computational Mechancis

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

Li L, Wang M, Wei P (2012) Xfem schemes for level set based structural optimization. Frontiers of Mechanical Engineering 7(4):335–356

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

Maute K, Ramm E (1995) Adaptive topology optimization. Structural and Multidisciplinary Optimization 10(2):100–112

Maute K, Ramm E (1997) Adaptive topology optimization of shell structures. AIAA Journal 35(11):1767–1773

Maute K, Schwarz S, Ramm E (1998) Adaptive topology optimization of elastoplastic structures. Structural and Multidisciplinary Optimization 15(2):81–91

Maute K, Kreissl S, Makhija D, Yang R (2011) Topology optimization of heat conduction in nanocomposites. In: 9th World Congress on Structural and Multidisciplinary Optimization, Shizuoka, Japan

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

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

van Miegroet L, Moës N, Fleury C, Duysinx P (2005) Generalized shape optimization based on the level set method. In: 6th World Congress of Structural and Multidisciplinary Optimization

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

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

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

Norato J, Bendsøe M, Haber R, Tortorelli D (2007) A topological derivative method for topology optimization. Structural and Multidisciplinary Optimization 33(4):375–386

Rozvany G (2009) A critical review of established methods of structural topology optimization. Structural and Multidisciplinary Optimization 37(3):217–237

Sethian J, Wiegmann A (2000) Structural boundary design via level set and immersed interface methods. Journal of Computational Physics 163(2):489–528

Sigmund O (2001a) Design of multiphysics actuators using topology optimization – Part I: One–material structures. Computer Methods in Applied Mechanics and Engineering 190(49–50):6577–6604

Sigmund O (2001b) Design of multiphysics actuators using topology optimization - part II: Two-material structures. Computer Methods in Applied Mechanics and Engineering 190(49-50):6605–6627

Sigmund O (2007) Morphology-based black and white filters for topology optimization. Structural and Multidisciplinary Optimization 33(4-5):401–424

Sigmund O, Maute K (2013) Topology optimization approaches: A comparative review. Structural and Multidisciplinary Optimization

Sokolowski J, Zochowski A (1999) Topological derivatives for elliptic problems. Inverse problems 15:123

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

Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J on Optimization 12(2):555–573

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

Wang S, Wang MY (2006) A moving superimposed finite element method for structural topology optimization. Int J Numer Meth Engng 65:1892–1922

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

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

Yazid A, Abdelkader N, Abdelmadjid H (2009) A state-of-the-art review of the x-fem for computational fracture mechanics. Applied Mathematical Modelling 33(12):4269 – 4282

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